Alma Mater Studiorum - Università di Bologna

DOTTORATO DI RICERCA IN

DATA SCIENCE AND COMPUTATION

Ciclo 37

Settore Concorsuale: 13/D1 - STATISTICA

Settore Scientifico Disciplinare: SECS-S/01 - STATISTICA

DATA ANALYSIS WITH MERGE TREES

Presentata da: Matteo Pegoraro

Coordinatore Dottorato

Daniele Bonacorsi

Supervisore

Piercesare Secchi

Co-supervisore

Andrea Cavalli

Esame finale anno 2023

Acknowledgments

I would like to thank my supervisor Piercesare, who allowed me to freely explore ideas and topics, even when links with his usual research fields where far from obvious. With constant trust and dialogues he greatly enriched my scientific perspectives and this, in turn, allowed me to get the most out of collaborations and discussions with other colleagues. A special thanks goes to Mario Beraha and Lara Cavinato: with them I made a clear experience of how enriching is the daily collaboration between researches with very diverse backgrounds, creating an environment which ended up affecting, enriching and inspiring in many unpredictable and precious ways my scientific work.

Finally, I thank my family and my friends.

Abstract

Today's data are increasingly complex and classical statistical techniques need growingly more refined mathematical tools to be able to model and investigate them. Paradigmatic situations are represented by data which need to be considered up to some kind of transformation and all those circumstances in which the analyst finds himself in the need of defining a general concept of *shape*. Topological Data Analysis (TDA) is a field which is fundamentally contributing to such challenges by extracting topological information from data with a plethora of interpretable and computationally accessible pipelines. We contribute to this field by developing a series of novel tools, techniques and applications to work with a particular topological summary called *merge tree*. To analyze sets of merge trees we introduce a novel metric structure along with an algorithm to compute it, define a framework to compare different functions defined on merge trees and investigate the metric space obtained with the aforementioned metric. Different geometric and topological properties of the space of merge trees are established, with the aim of obtaining a deeper understanding of such trees. To showcase the effectiveness of the proposed metric, we develop an application in the field of Functional Data Analysis, working with functions up to homeomorphic reparametrization, and in the field of radiomics, where each patient is represented via a clustering dendrogram.

"What could he see, that I was missing?
It had to be the glasses.
So I made myself an identical pair. (Better, though)
And, with those on, I looked around."

Gipi, La mia vita disegnata male, pg. 141

Contents

1. General Introduction

When looking at two objects, one of the questions we may ask is: "Do these objects have the same shape?". To answer that question it is likely that we may picture ourselves squinting our eyes or taking a step backwards so that the contours of the objects become slightly blurred and we may better see which macroscopic features are shared among them.

These common behaviours unveil two linked and very interesting ideas behind the concept of shape: first, two objects may be different but have the same shape, second, to capture the shape of an object we need to neglect some information - possibly related to uninteresting details - and capture the most prominent, characteristic attributes of the object.

Whenever the notion of shape enters a data analysis situation, there is a clear need of a more precise and formal mathematical formulation of the aforementioned ideas, often application-driven. It is not surprising that two of the most powerful mathematical concepts which inevitably are employed in doing so are the idea of transformation and the notion of equivalence classes. A suitable class of transformations may be used to describe which object should be regarded as equal: two objects have the same shape if they can be transformed one in the other - e.g. if they become the same upon blurring their contours. Then the set of all possible transformations, representations of an object, naturally becomes the candidate for representing the shape of the object, and thus the statistical atom of the subsequent analysis. In this case, the shape of a datum collects all the objects we do not wish to distinguish and so it really represents some information which all the possible representations of the datum share.

Modeling from scratch the space of shapes, talking about functions, distances, curves, deformations of objects, and generalizing concepts like mean and variance, are all aspects of the analysis which need to be carefully and formally addressed. Among the fields of mathematics, the one which is most suited to describe and tackle those fundational challenges is geometry. All the aforementioned ideas have been extensively explored employing a very diverse set of geometric notions: from spaces of landmark points (Kendall, 1977, 1984) and spaces of diffemorphisms (Trouvé, 1998; Younes, 1998; Dupuis et al., 1998) to more high level works trying to establish a general framework to tackle these mathematical problems (Ripley and Grenander, 1995; Dryden and Mardia, 1998; Huckemann et al., 2010), to very recent approaches relying on sheaves and Groethendieck topologies (Arya et al., 2022). On top of such foundations, many interactions between geometry and data analysis have been developed with increasing potential (Davis, 2008; Curry, 2014; Tralie, 2017; Bône, 2020; Pennec et al., 2019; Bronstein et al., 2021) to the point that terms like "Geometric Statistics", "Geometric Data Analysis" and "Geometric Machine Learning" are commonly used and attract more and more researchers.

1.1 Topology and Data analysis

Topology is a branch of mathematics which has always been interested in classifying topological spaces considered by means of very large equivalence classes. In fact, one of the most recent breakthroughs within the broad area of shape analysis is the diffusion of ap-

plied topology techniques, which can be collected under the name of Topological Data Analysis (TDA).

Consider for instance the case of homology and cohomology groups, which are basic topology tools used to summarize some topological information of a space (Hatcher, 2000; Munkres, 2018). The information they capture can be interpreted in terms of holes and obstructions and in many cases is easily accessible from the computational point of view, especially if considered with field coefficients. Moreover, these groups are invariant to large sets of deformations of the base topological space, induced by homotopy equivalence. These facts make homology an excellent starting point to build tools to extract information from data considered up to some set of transformations. Indeed, one of the most successful ideas in TDA relies on homology groups to extract information from data. Suppose we are given a finite point cloud in $\mathbb{R}^n$. The subspace topology of such object is very poor, and the only information contained in homology groups is the number of connected components, that is, the number of points. There are however many ways to build topological spaces starting from a point cloud, which try to capture the "shape" of such point cloud (Chazal and Michel, 2017). It is then quite natural to think that, instead of comparing the information associated to the point clouds, one can compare the information obtained from the topological spaces induced by the point clouds. Moreover, such topological spaces are often dependent on one real parameter and thus one can obtain a whole set of topological spaces parametrized by a subset of the real line - i.e. a *filtration* of topological spaces. Relationships between spaces obtained at different values of the parameter in many cases allow for a very interesting approach: homology groups can be tracked along the ordered family of topological spaces, capturing the most *persistent* homological features that appear in this sequence. This idea is in fact the foundation of *persistent homology* (Edelsbrunner and Harer, 2008).

This is just one well-established example of a more general pattern: topological spaces are often understood in terms of information collected by polynomials and group structures (groups of automorphisms, cohomology groups, groups of sections of vector bundles etc) and these algebraic objects can be studied in order to better understand the underlying topological spaces (Xu et al., 2019; Scoccola and Perea, 2022). We also point out that this idea of following some quantity of interest along a filtration is so flexible that it does not suit only topological approaches (Mémoli et al., 2022).

1.2 Statistics in non-Euclidean Spaces

A critical point raised by all the ideas presented in the previous section is how to represent the (topological) information collected with different techniques so that it can be used to answer data analysis questions related to classification, regression, clustering and, possibly, uncertainty quantification. The amount of questions that can be answered is a direct consequence of the mathematical structures that can be built in the spaces in which the topological information is embedded.

The problem of carrying out statistical analysis for data which do not lie, at least naively, in $\mathbb{R}^n$, has always propelled many fruitful research investigations often relying on the mathematical developments fostered by physics and engineering. Modeling points or quantities bound by certain laws often involves calculation of differential quantities like derivatives along some "direction", and many of such differential problems involve constraints which greatly increase the modeling and computational complexity. The "curved" and non-Euclidean nature of many constraints asks for non-trivial generalization of operations which are very well understood in linear spaces - and are pivotal in statistics - and go from vector sums and directional derivatives, to the behaviour of all kinds of differential operators.

There are many examples of analyses carried out on data embedded in non-Euclidean spaces, often exploiting a *smooth manifold* structure: functions up to reparametrizaion, symmetric positive definite matrices (like covariance matrices, see Arsigny et al. (2006); Moakher and Zéraï (2011); Pigoli et al. (2014)), sets of orthonormal vectors (James, 1976; Turaga et al., 2011), rotation matrices, densities of probability distributions (Pegoraro and Beraha, 2022), even patches of images (Carlsson et al., 2008) cannot be analyzed using the linear structure of the ambient space but must be approached by considering an appropriate mathematical structure. The situation becomes further challenging when one cannot build a differential or even a topological structure which falls into the realm of well-known and deeply studied geometrical objects like manifolds. In this case, ad-hoc and meaningful tools and definitions must be carefully obtained in order to be able to work in such spaces (Turner et al., 2014; Calissano et al., 2020; Garba et al., 2021).

The information represented by topological summaries presents, very often, two different natures: a continuous nature - linked to the space in which the variables parametrizing the families of topological spaces live e.g. $\mathbb{R}$ for the scenarios considered in the thesis - and a discrete nature, derived from the discrete topological invariant which is summarized. For instance, a *persistence diagram*, used to represent information related to homology groups along a nested family of topological spaces, is a point cloud in $\mathbb{R}^2$ whose cardinality represents the amount of "holes" which arise along the nested family of spaces and whose - continuous - coordinates are the "times" at which such holes arise and get filled. This combination creates highly non-Euclidean situations, with non-unique minimal paths arising for any reasonable metric structure. Due to their importance, such challenges have sparkled a great amount of strategies, approaches and developments on the representation of topological information (Bubenik, 2015; Adams et al., 2017; Chazal et al., 2015; Mileyko et al., 2011; Turner et al., 2014; Bubenik and Wagner, 2020; Che et al., 2021).

We would like to highlight another non-trivial facet of this broad topic: when employing a representation of data, it is fundamental to formally understand what kind of information is really preserved by the representation itself and how can we "measure" the interpretability of the framework. A very reasonable approach is to embed the data themselves into some kind of metric space, where the metric employed has some clear and desired behaviour, for instance coherent with the specific application at hand, and then study the *stability* of the representation i.e. the continuity of the operator mapping the data into the representation. In this way the analyst is sure that objects which he would like to regard as similar, are indeed close in the representation space. The opposite it is clearly not desirable in general: if two input data look very different, then their representation should be allowed to be closer together, otherwise our representation is (metrically) equivalent to the space of original data, and not much is gained by employing it.

1.3 Merge Trees

In this thesis we focus on a particular topological summary called merge tree and in this section we would like to briefly motivate such choice. This is clearly a very short and introductory paragraph on this topic, as these motivations are going to be constantly updated and reinforced throughout the thesis.

The first point is that merge trees are in some sense - and up to some minor technical details -, very well established tools in many areas of science where analysts are accustomed to using trees to infer information about different objects; for instance, phylogenetic trees and hierarchical clustering dendrograms are frequently used by statisticians and other scientists. The present thesis has been carried out in collaboration with a research group of statisticians, and thus it was quite natural to start a topological investigation on objects which they were very familiar with. While carrying out such investigation we could also

appreciate two other facts about merge trees: first they are very effective visual summaries. In many situations it is very simple to make a qualitative picture of the underlying datum by looking at a merge tree. Second they offer the possibility to extract many other kinds of local additional information from data in the form of functions defined on thetopological features appearing along a filtration. This is a qualitative difference w.r.t. other topological summaries like persistence diagrams, where the connections between elements of the summary and the topological features is very unstable due to a process called *the elder rule*.

1.4 Outline of the Dissertation

As already mentioned, in this thesis we contribute to the topics presented in the previous sections of the introduction: we consider merge trees and establish novel ways in which they can be fruitfully employed in data analysis situations, we define a metric framework to compute distances between different merge trees and test such framework with two applications, studying its stability properties.

The chapters collect the contributions of the manuscript dividing them between different areas and different research perspectives.

- Chapter 2 introduces spaces of functions defined on merge trees, showing how they enhance the possibilities given by such tree-shaped topological summaries, introduces a metric which can be used to compare functions defined on different merge trees and tests it with some simulations.

- Chapter 3 exploits the general results of the previous chapter to define a novel edit distance between merge trees. The metric is then compared to already established metric structures, highlighting pros and cons of our approach. The obtained metric space is then investigated from the topological and metric point of view, proving a series of results which lead to the existence of Frechét means and to some local approximations of the space via $\mathbb{R}^n$.

- Chapter 4 concentrates on the use of merge trees as topological summaries of real valued functions in one real variable, proving stability properties and tackling a benchmark case study in functional data analysis to show the effectiveness of the topological approach when non-trivial re-parametrization procedures are needed for the analysis.

- Chapter 5 stems from the need of analysing another kind of data: point clouds with different cardinalities. Employing a tree representation of the point cloud which is consistent with the meaning of the information contained in the point clouds, a data set of patients is analysed in an unsupervised fashion. A modification of the metric defined in Chapter 3 is employed, tailored on the considered application. Stability properties are proven w.r.t. an interpretable metric between finite point clouds.

- Chapter 6 contains a computational investigation on a well established metric between merge trees. Such approach leads to a novel approximation procedure which we use to evaluate a previous attempt on the same approximation task - highlighting pros and cons of both numerical schemes. One of the theoretic results is key for the comparison with the metric defined in Chapter 3.

- Chapter 7 concludes the dissertation drawing some general conclusions and suggesting some other further research directions.

1.5 Note to the reader: Structure of the Thesis

The thesis is organized in independent chapters. With this we mean that each chapter is an investigation of standalone interest and could attract potentially different readers and scientists. Accordingly, each chapter is endowed with a preliminary introduction tailored on the audience which could be more interested on the topic, avoiding unnecessary technicalities when possible. The notation is consistent through all the chapters, and so redundant preliminaries can be skipped when going through the chapters followings their numeration.

There are, however, "non-linear" relationships between the chapters. In particular:

- the stability result in Chapter 4 is used in Chapter 2, Chapter 3 and Chapter 5. It is presented in Chapter 4 because it is part of the more general investigation related to the statistical analysis of functions carried out in such chapter.
- The main theorems in Chapter 6 are used in in Chapter 3 to obtain relationships between the *interleaving distance* and the distance defined in Chapter 3.

Moreover, the content of Chapter 5 is obtained in collaboration with other authors and has marked applied nature: the case study considered is not a benchmark data set - as in Chapter 4 -, but is on the frontiers of non-invasive cancer treatments. Thus, a good portion of the chapter is devoted to the introduction and understanding of the clinical problem and of the medical methodologies involved. The methodological novelties of the chapter in terms of data analysis pipeline and merge trees are all contained in the section "Methods" and in the appendices. This is in contrast with the other chapters, where the theoretic content is always the focal point of the developments. In this chapter, instead, it is the specific application that drives the theoretic analysis: the expertise of the other collaborators on the clinical problem considered has fostered the introduction of a modified version of the metric defined in Chapter 3 which greatly enhances the interpretability of the pipeline. And such interpretability is of utmost importance to tackle an unsupervised problem, like the one we consider.

As a general rule, the proofs of the results are contained in the appendices of every chapters. The proofs which are left inline are either very short or are constructive proofs, and so they may be useful to understand the following parts of the manuscript.

1.6 Note to the reader: previous works

This thesis is part of a double-degree program established by Politecnico di Milano and Universitá di Bologna. In particular, it is meant as a one year extension of the PhD thesis Pegoraro (2021a).

For the sake of clarity and for a better understanding of the original contributions of the present work, we detail which sections do overlap with the previous thesis, which sections have undergone substantial modifications, and which are completely original.

- Chapter 2: this chapter is partially overlapping with Chapter 2 in Pegoraro (2021a). In particular:
  - Section 2.4 shares the two main examples with Section 2.2 of Pegoraro (2021a).
  - Section 2.6 shares most definitions and results with Section 2.4 of Pegoraro (2021a). There are however some novel results and subtle modifications in the definitions.

  - Section 2.8 is the same as Section 2.7 in Pegoraro (2021a) a part from some minor details.
  - Section 2.9 contains the results in Section 2.5.1 of Pegoraro (2021a) while all other topics in Section 2.5 of Pegoraro (2021a) have been expanded and detailed in Section 2.B.

  The novelties of the chapter of present thesis include a completely new theoretical background, stemming from most novel literature in TDA, a formal introduction of function spaces on merge trees, which is now the main focus of the chapter and some technical but not negligible differences in how the general edit distance is adapted to work with such functions. Many details, examples, and figures have also been added.

- Chapter 3 shares some content with Chapter 4 in Pegoraro (2021a). In particular:

  - Section 3.6.2, Section 3.6.4, Section 3.6.5 and Section 3.6.6 are taken from, respectively, Section 4.2, 4.3, 4.4 and 4.5 of Pegoraro (2021a) with only minor changes.

  Some of the most important results in this chapter are taken directly from Pegoraro (2021a). However, the definition of the edit distance between merge trees is a novelty of the present work and the aforementioned sections are used to infer properties about the space of merge trees in a novel way. The notation and the background which are used, in accordance with Chapter 2, are novel, as are the detailed comparisons with other metric structures for merge trees offering new perspective on the edit distance presented. Original contributions include also the mapping decompositions, the local approximation results and most of the figures appearing in the chapter.

- Chapter 4 is an in-depth revised version of Chapter 2 of Pegoraro (2021a):

  - Section 4.3 is taken from Section 3.3 of Pegoraro (2021a) without significative changes.
  - Section 4.4 shares some results with Section 3.4 of Pegoraro (2021a), but the general discussion is more detailed, referenced and expanded.
  - Section 4.5 is roughly Section 3.5 of Pegoraro (2021a), with only some technical details being different.
  - Section 4.6 is a revised version of Section 3.6 in Pegoraro (2021a), and the revision involves a novel definition of the pruning operator, a brand new proof of the stability result - which contained some technical errors in Pegoraro (2021a), and a series of novel results concerning the pruning process.
  - Section 4.7.2 and Section 4.7.3 had already appeared in Section 3.8 of Pegoraro (2021a) - up to some minor details.
  - Section 4.8.1, Section 4.8.2 and Section 4.8.3 have only undergone marginal modifications compared to Section 3.9.1 and 3.9.2 of Pegoraro (2021a).

  The fundamental novelties of this chapter include the new proof of the stability theorem, which was necessary due to some minor errors in the previous one. In rewriting the proof a novel approach has been taken relying on previous works on the interleaving distance between merge trees. Section 4.8.4 is also completely new and it is a first attempt to establish statistical tools to interpret populations of merge trees.

- Chapter 5 and Chapter 6 are new.

1.7 Further Comments

The content of the main chapters of the thesis is also part of the following papers:

- Chapter 2: A Locally Stable Edit Distance for Functions Defined on Merge Trees (Pegoraro, 2021c)
- Chapter 3: A Locally Stable Stable Edit Distance for Merge Trees (Pegoraro, 2021d)
- Chapter 4: Functional Data Representation with Merge Trees (with P. Secchi)(Pegoraro and Secchi, 2021)
- Chapter 5: Imaging-based representation and stratification of intra-tumor Heterogeneity via tree-edit distance (with L. Cavinato, F. Ieva and A. Ragni) (Cavinato et al., 2022)
- Chapter 6: A Graph-Matching Formulation of the Interleaving Distance between Merge Trees (Pegoraro, 2021b).

2. A Locally Stable Edit Distance for Functions Defined on Merge Trees

Abstract

In this chapter we define a novel metric structure to work with functions defined on merge trees. The metric introduced possesses some stability properties and can be computed with a dynamical integer linear programming approach. We showcase its feasibility and the effectiveness of the whole framework with simulated data sets. Using functions defined on merge trees proves to be very effective in situation where other topological data analysis tools, like persistence diagrams, can not be meaningfully employed.

2.1 Introduction

Topological Data Analysis (TDA) is the name given to an ensemble of techniques which are mainly focused on retrieving topological information from different kinds of data (Lum et al., 2013). Consider for instance the case of point clouds: the (discrete) topology of a point cloud itself is quite poor and it would be much more interesting if, using the point cloud, one could gather information about the topological space data was sampled from. Since, in practice, this is often not possible, one can still try to capture the "shape" of the point cloud. The idea of *persistent homology* (PH) (Edelsbrunner and Harer, 2008) is an attempt to do so: using the initial point cloud, a nested sequence of topological spaces is built, which are heavily dependent on the initial point cloud, and PH tracks along this sequence the persistence of the different topological features which appear and disappear. As the name *persistent homology* suggests, the topological features are understood in terms of generators of the homology groups (Hatcher, 2000) taken along the sequence of spaces. One of the fundational results in TDA is that this information can be represented by a set of points on the plane (Edelsbrunner et al., 2002; Zomorodian and on, 2005), with a point of coordinates (x, y) representing a topological feature being born at time x along the sequence, and disappearing at time y. Such representation is called *persistence diagram* (PD). Persistence diagrams can be given a metric structure through the *Bottleneck* and *Wasserstein* metrics, which, despite having good properties in terms of continuity with respect to perturbation of the original data (Cohen-Steiner et al., 2007, 2010), provide badly behaved metric spaces - with non unique geodesics arising in many situations. Various attempts to define tools to work in such spaces have been made (Mileyko et al., 2011; Turner et al., 2012; Lacombe et al., 2018; Fasy et al., 2014), but this still proves to be an hard problem. In order to obtain spaces with better properties - e.g. with unique means - and/or information which is vectorized, a number of topological summaries alternative to PDs have been proposed, such as: persistence landscapes (Bubenik, 2015), persistence images (Adams et al., 2017) and persistence silhouettes (Chazal et al., 2015).

All the aforementioned machinery has been successfully applied to a great number of problems in a very diverse set of scientific fields: complex shape analysis (MacPherson and Schweinhart, 2010), sensor network coverage (Silva and Ghrist, 2007), protein structures (Kovacev-Nikolic et al., 2016; Gameiro et al., 2014), DNA and RNA structures (Emmett

et al., 2015; Rizvi et al., 2017), robotics (Bhattacharya et al., 2015; Pokorny et al., 2015), signal analysis and dynamical systems (Perea and Harer, 2013; Perea et al., 2015; Maletić et al., 2015), materials science (Xia et al., 2015; Kramár et al., 2013), neuroscience (Giusti et al., 2016; Curto, 2016), network analysis (Sizemore et al., 2015; Pal et al., 2017), and even deep learning theory (Hofer et al., 2017; Naitzat et al., 2020).

Related Works

Close to the definition of persistent homology for 0 dimensional homology groups, lie the ideas of *merge trees* of functions, *phylogenetic trees* and *hierarchical clustering dendrograms*. Merge trees of functions (Pascucci and Cole-McLaughlin, 2003) describe the path connected components of sublevel sets of a real valued function and are a particular case of *Reeb graphs* (Shinagawa et al., 1991a; Biasotti et al., 2008), representing the evolution of the sublevel sets of a bounded Morse function (Audin et al., 2014) defined on a path connected domain. Phylogenetic trees and clustering dendrograms are very similar objects which describe the evolution of a set of labels under some similarity measure or agglomerative criterion. Both objects are widely used respectively in phylogenetic and statistics and many complete overviews can be found, for instance see Felsenstein and Felenstein (2004), Garba et al. (2021) for phylogenetic trees and Murtagh and Contreras (2017), Xu and Tian (2015) for clustering dendrograms. Informally speaking, while persistence diagrams record only that, at certain level along a family of topological spaces some path connected components merge, merge trees, phylogenetic trees and clustering dendrograms encode also the information about which components merge with which (Kanari et al., 2020; Curry et al., 2021). Usually tools like phylogenetic trees and clustering dendrograms are used to infer something about a fixed set of labels, for instance an appropriate clustering structure; however, we are more interested in looking at the information they carry as unlabeled objects obtained with different sets of labels. For this reason most of the metrics available for phylogenetic trees and clustering dendrograms are not valuable for our purposes.

In the last years a lot of research sparkled on such topics, starting from the more general case of Reeb graphs, to some more specific works on merge trees. Different but related metrics have been proposed to compare Reeb graphs (Di Fabio and Landi, 2016; De Silva et al., 2016; Bauer et al., 2020, 2014), which have been shown to posses very interesting properties in terms of Morse functions on manifolds, connecting the combinatorial nature of Reeb Graphs with deformation-invariant characterizations of manifolds which are smooth, compact, orientable and without boundary. On the specific case of merge trees, there has been some research on their computation (Pascucci and Cole-McLaughlin, 2003) and on using them as visualization tools (wu and Zhang, 2013; Bock et al., 2017), while other works (Beketayev et al., 2014; Morozov et al., 2013) started to build frameworks to analyze sets of merge trees, mainly proposing a suitable metric structure to compare them, as do some recent preprints (Gasparovic et al., 2019; Touli, 2020; Cardona et al., 2021). Some works specifically tackle the problem of finding a suitable metric structure via edit distances (Sridharamurthy et al., 2020; Wetzels et al., 2022), which however lack suitable stability properties. The main issue with most of the proposed metrics is their computational cost, causing a lack for examples and applications also when algorithms are available (Touli and Wang, 2018). When applications and analysis are carried out due to the good computational properties (Sridharamurthy et al., 2020; Wetzels et al., 2022), either the employed metric does not have suitable properties and thus the authors must resort to a "computational solution to handle instabilities" (Sridharamurthy et al. (2020), Section 1.2), or the stability properties of the metric are not studied. Recently, Curry et al. (2022) proposed an approximation scheme for the *interleaving distance* between merge trees, describing a procedure to obtain suitable set of labels to turn the original unlabeled

problem into a labeled one. While the computational advantages of this approach are outstanding, the reliability of the approximation is yet to be formally assessed - simulations in Chapter 6 show that in some scenarios it may produce big errors. In the same work the authors propose also the idea of *decorated merge trees*, which, philosophically, goes in the same direction of the novelties presented in this manuscript. See Section 2.5.6.4 for more details. Lastly, there is a recent preprint investigating structures lying in between merge trees and persistence diagrams, to avoid computational complexity while retaining some of the additional information provided by such objects (Elkin and Kurlin, 2020).

Instead of modifying the aforementioned metrics or other metrics for trees (Billera et al., 2001; Feragen et al., 2012; Wang and Marron, 2007) in order to account for functions defined on unlabelled trees, we follow the path of edit distances because of the computational properties which they often possess, making them suited for dealing with unordered and unlabelled trees (Hong et al., 2017). The computational issues raised by those kind of trees are in fact a primary obstacle to designing feasible algorithms (Hein et al., 1995). The edit distance we propose starts from usual tree edit distances (Tai, 1979; Bille, 2005) but adds fundamental modifications in order to allow the comparison of functions defined on different trees.

Main Contributions

The success of PDs highlighted before, strongly motivates the development of more refined and computable techniques to work with merge trees, phylogenetic trees and clustering dendrograms. Our contributions to such topic can be split into four points: first we propose the use of functions defined on merge trees to enrich such trees with additional information, second we propose a metric structure for the space of these enriched topological summaries, in the form of a novel edit distance between weighted (in a very broad sense), unlabeled, unordered trees; third, we prove that this metric satisfies some stability properties when considering particular functions defined on merge trees. Lastly we prove some decomposition properties which are used in the supplementary material to develop a dynamical integer programming algorithm to make this metric viable for a good range of applications.

The framework we define in the present chapter is general enough to allow for a series of developments which are carried out the other chapters of the thesis. Some of these developments, in turn, contain results which have consequences also for the content of this chapter. In particular, Chapter 3 exploits a particular case of the general framework contained in the upcoming sections to induce a metric for merge trees and Chapter 4 develops a stability result for the metric defined in Chapter 3.

Outline

The chapter is organized as follows. In Section 2.2 and Section 2.3 we introduce all the definitions needed for our dissertation, starting from most recent TDA literature, tackling the problem of representing with a discrete summary - a merge tree - the merging pattern of the path-connected components of a filtration of topological spaces.

Once merge trees are introduced, we use Section 2.4 to motivate the use of such objects over more commonly used topological data analysis techniques. In Section 2.5 we formally introduce the spaces of functions on merge trees. In high generality, with Section 2.6, we tackle the problem of finding a suitable metric structure to compare such functions. Section 2.7 details how the findings of Section 2.6 interact with the problem of comparing functions defined on merge trees. In Section 2.8 we present some simulations and examples to showcase the effectiveness of the proposed framework while in Section 2.9 we prove some properties of the metric previously defined, which lead to the algorithm presented in

Section 2.C. We end up with some conclusions in Section 2.10.

The appendix contains the proofs of results in Section 2.A, while Section 2.B and Section 2.C contain theoretical and practical details needed to compute our edit distance.

2.2 Abstract Merge Trees

In TDA the main sources of information are sequences of homology groups with field coefficients: using different pipelines a single datum is turned into a filtration of topological spaces $\{X_t\}_{t\in\mathbb{R}}$, which, in turn, induces - via some homology functor with coefficients in the field $\mathbb{K}$ - a family of vector spaces with linear maps which are usually all isomorphisms but for a finite set of points in the sequence. Such objects are called (one-dimensional) *persistence modules* (Chazal et al., 2008). Any persistence module is then turned into a topological summary, for instance a persistence diagram, which completely classifies such objects up to isomorphisms. That is, if for two persistence modules there exists a family of linear isomorphisms giving a natural transformation between the two functors, then they are represented by the same persistence diagram. The first part of this chapter studies this very same pipeline but under the lenses of merge trees.

2.2.1 Preliminary Definitions

We start off by introducing the main mathematical objects of our research starting from the scientific literature surrounding these topics. In this process we also point out where there is no clear notation to be used and producing new definitions, with motivations, to avoid being caught in the trap of using ambiguous terminology or overwriting existing and established notation.

First we need to formally define a filtration of topological spaces. We do so in a categorical fashion, following the most recent literature in TDA. Figure 2.1 illustrates some of the objects we introduce in this section.

Definition 2.1 (Curry et al. (2022))**.** *A filtration of topological spaces is a (covariant) functor $X_\bullet : \mathbb{R} \to \mathrm{Top}$ from the poset $(\mathbb{R}, \leq)$ to* Top*, the category of topological spaces with continuous functions, such that: $X_t \to X_{t'}$, for $t < t'$, are injective maps.*

Example Given a real valued function $f : X \to \mathbb{R}$ the *sublevel set* filtration is given by $X_t = f^{-1}((-\infty, t])$ and $X_{t<t'} = i : f^{-1}((-\infty, t]) \hookrightarrow f^{-1}((-\infty, t'])$.

Example Given a finite set $C \subset \mathbb{R}^n$ its the *Céch* filtration is given by $X_t = \bigcup_{c\in C} B_t(c)$. With $B_t(c) = \{x \in \mathbb{R}^n \mid \| c - x \| < t\}$. As before: $X_{t<t'} = i : \bigcup_{c\in C} B_t(c) \hookrightarrow \bigcup_{c\in C} B_{t'}(c)$.

Given a filtration $X_\bullet$ we can compose it with the functor π_0 sending each topological space into the set of its path connected components. We recall that, according to standard topological notation, $\pi_0(X)$ is the set of the path connected components of X and, given a continuous functions $q : X \to Y$, $\pi_0(q) : \pi_0(X) \to \pi_0(Y)$ is defined as:

$$U \mapsto V \text{ such that } q(U) \subset V.$$

We use filtrations and path connected components to build more general objects which are often used as starting points of theoretical investigations in TDA.

Definition 2.2 (Carlsson and Mémoli (2013); Curry (2018))**.** *A persistent set is a functor $S : \mathbb{R} \to \mathrm{Sets}$. In particular, given a filtration of topological spaces $X_\bullet$, the persistent set of components of $X_\bullet$ is $\pi_0 \circ X_\bullet$.*

Note that by endowing a persistence set with the discrete topology, every persistence set can be seen as the persistence set of components of a filtration. Thus a general persistence set S can be written as $\pi_0(X_\bullet)$ for some filtration $X_\bullet$.

Now we want to carry on going towards the definition of merge trees. The existing paths for giving such notion relying on the language of TDA split at the definition of persistence module. All such approaches however share similar notions of *constructible* persistent sets (Patel, 2018) or modules (Curry et al., 2022). We report here the definition of constructible persistent sets adapted from Patel (2018). The original definition in Patel (2018) is stated for persistent modules (as defined in Patel (2018)) and it is slightly different - see Remark 2.4.

Definition 2.3 (Patel (2018))**.** *A persistent set $S : \mathbb{R} \to \text{Sets}$ is constructible if there is a finite collection of real numbers $\{t_1 < t_2 < \ldots < t_n\}$ such that:*

- *$S(t < t') = \emptyset$ for all $t < t_1$;*
- *for $t, t' \in (t_i, t_{i+1})$ or $t, t' > t_n$, with $t < t'$, then $S(t < t')$ is bijective.*

The set $\{t_1 < t_2 < \ldots < t_n\}$ is called critical set and t_i are called critical values. If $S(t)$ is always a finite set, then S is a finite persistent set.

Remark 2.4. *In literature there doesn't seem to be an univocal way to treat the critical values: in De Silva et al. (2016), Definition 3.3, constructibility conditions are stated in terms of open intervals (due to the definition of Reeb cosheaves), in Patel (2018), Definition 2.2, all the conditions are stated in terms of half-closed intervals $[t_i, t_{i+1})$, while Curry et al. (2022) differentiates between the open interval $(t_n, +\infty)$ i.e. $t, t' > t_n$, and the half closed intervals $[t_i, t_{i+1})$. For reasons which will be detailed in Section 2.2.2, we stated all the conditions following De Silva et al. (2016), with open intervals.*

At this point we highlight two different categorical approaches to obtain merge trees. Patel (2018) requires a persistence module to be a functor $F : \mathbb{R} \to C$ with C being an essentially small symmetric monoidal category with images (see Patel (2018) and references therein). If then one wants to work with values in some category of vector spaces over some field $\mathbb{K}$, it is required that $F(t)$ is always finite dimensional. A merge tree, for Patel (2018), Example 2.1, is then a constructible persistence module with values in $FSet$, the category of finite sets.

Curry et al. (2022) instead, states that a persistence module is a functor $F : \mathbb{R} \to \text{Vec}_{\mathbb{K}}$, with $\text{Vec}_{\mathbb{K}}$ being the category of vector spaces over the field $\mathbb{K}$. This definition seems to be in line with the ones given by other works, especially in multidimensional persistence (see for instance Scolamiero et al. (2017) and references therein). On top of that, Curry et al. (2022) obtains a (generalized) merge tree as a *display poset* (see Definition 2.7) of a persistent set. The constructibility condition on the persistence set then implies the merge tree to be *tame*.

In our dissertation we find natural to work with objects which are functors, as the merge trees defined in Patel (2018), but we require some properties which are closer to the ones of constructible persistent sets, as in Definition 2.3. Thus, mixing those definitions, we give the notion of an *abstract merge tree*.

Definition 2.5. *An abstract merge tree is a persistent set $S : \mathbb{R} \to \text{Sets}$ such that there is a finite collection of real numbers $\{t_1 < t_2 < \ldots < t_n\}$ which satisfy:*

- *$S(t) = \emptyset$ for all $t < t_1$;*
- *$S(t) = \{\star\}$ for all $t > t_n$;*

- *if $t, t' \in (t_i, t_{i+1})$, with $t < t'$, then $S(t < t')$ is bijective.*

The values $\{t_1 < t_2 < \ldots < t_n\}$ are called critical values of the tree.

If $S(t)$ is always a finite set, S is a finite abstract merge tree.

Assumption 2.6. *From now on we will be always working with finite abstract merge trees and, to lighten the notation, we assume any abstract merge tree to be finite, without explicit reference to its finiteness.*

We point out that two abstract merge trees $\pi_0(X_\bullet)$ and $\pi_0(Y_\bullet)$ are isomorphic if there is a natural transformation $\alpha_t : \pi_0(X_t) \to \pi_0(Y_t)$ which is bijective for every t. This is equivalent to having the same number of path connected components for every t and having bijections which make the following square commute:

$$\begin{array}{ccc} X_t & \longrightarrow & X_{t'} \\ \downarrow{\scriptstyle \alpha_t} & & \downarrow{\scriptstyle \alpha_{t'}} \\ Y_t & \longrightarrow & Y_{t'} \end{array}$$

for all $t < t'$.

We report one last definition from Curry et al. (2022) which will be needed in later sections.

Definition 2.7 (Curry et al. (2022))**.** *Given a persistent set $S : \mathbb{R} \to \text{Sets}$ we define its display poset as:*

$$D_S := \bigcup_{t \in \mathbb{R}} S(t) \times \{t\}.$$

The set D_S can be given a partial order with $(a, t) \leq (b, t)$ if $S(t \leq t')(a) = b$.

Given a persistent set S and its display poset D_S we define $h((a, t)) = t$ and $\pi((a, t)) = a$ for every $(a, t) \in D_S$. From D_S we can clearly recover S via $S(t) = \pi(h^{-1}(t))$ and $S(t \leq t')(a) = b$ with $a \leq b$. Thus the two representations are equivalent and, at any time, we will use the one which is more convenient for our purposes. Note that this construction is functorial: any natural transformation $\eta : S \to S'$ between persistent sets, gives a map of sets $D_\eta : D_S \to D_{S'}$ with $D_\eta((a, t)) = (\eta_t(a), t)$. Clearly $D_{\eta \circ \nu} = D_\eta \circ D_\nu$.

2.2.2 Critical Values

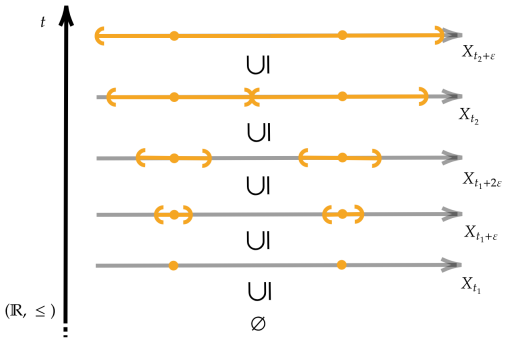
Before bridging between abstract merge trees and merge trees, we need to focus on some subtle facts about critical values.

The first fact is that neither in Definition 2.3 nor in Definition 2.5 critical values are uniquely defined. However, thanks to the functoriality of any persistence set, we can take the intersection of all the possible sets of critical values to obtain a minimal (possibly empty) one.

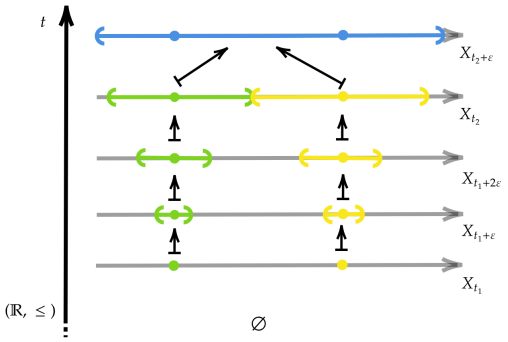
Proposition 2.8. *Let S be a constructible persistence set and let $\{C_i\}_{i \in I}$ be a family of finite critical sets of S. Then $C = \bigcap_{i \in I} C_i$ is a critical set.*

Proof. Clearly C is a finite set, possibly empty. The thesis is then a consequece of the following fact: if $t \notin C$ then there is $\varepsilon > 0$ such that $S(t - \varepsilon < t + \varepsilon)$ is bijective. So we can remove t from any critical set of S and still obtain a critical set. □

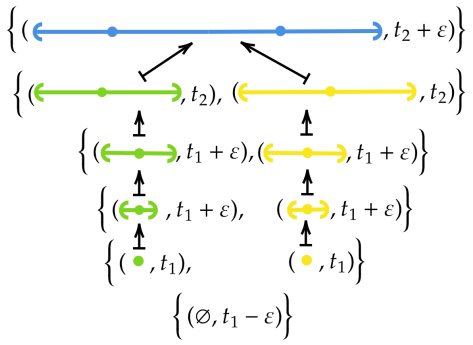
Assumption 2.9. *Leveraging on Proposition 2.8, any time we take any abstract merge tree or a constructible persistent set and consider its critical values, we mean the elements of the minimal critical set.*



(a) A filtration $X_\bullet$.



(b) The abstract merge tree $\pi_0(X_\bullet)$.



(c) The display poset $D_{\pi_0(X_\bullet)}$.

Figure 2.1: An example of a filtration along with its abstract merge tree and its display poset. The colors are used throughout the plots to highlight the relationships between the different objects.

Consider an abstract merge tree $\pi_0(X_\bullet)$ and let $t_1 < t_2 < \ldots < t_n$ be its (minimal set of) critical values. Let $i_t^{t'} := X_{t \leq t'} : X_t \to X_{t'}$. Given a critical value t_j, due to the minimality condition, we know that for $\varepsilon > 0$ small enough, at least one between $\pi_0(i_{t_j-\varepsilon}^{t_j})$ and $\pi_0(i_{t_j}^{t_j+\varepsilon})$ is not bijective.

We want to distinguish between two scenarios:

- if $\pi_0(i_{t_j}^{t_j+\varepsilon})$ is bijective, we say that topological changes in the persistence set (and in the filtration) happen *at* t_j;
- if $\pi_0(i_{t_j}^{t_j+\varepsilon})$ is not bijective, we say that topological changes in the persistence set (and in the filtration) happen *across* t_j.

Definition 2.10. *A constructible persistence set $\pi_0(X_\bullet)$ is said to be regular if all topological changes happen at its critical points.*

Consider the following filtrations of topological spaces: $X_t = (-t, t) \bigcup (1-t, 1+t)$ and $Y_t = [-t, t] \bigcup [1-t, 1+t]$ for $t > 0$ and $X_0 = Y_0 = \{0, 1\}$. For $t < 0$ the filtrations are empty. The persistent sets $\pi_0(X_t)$ and $\pi_0(Y_t)$ are two abstract merge trees and they share the same set of critical values, namely $\{0, 1/2\}$. They only differ at the critical value $1/2$: $\pi_0(X_{1/2}) = \{(-1/2, 1/2), (1/2, 1)\}$, while $\pi_0(X_{1/2}) = \{(-1/2, 1)\}$. In $X_\bullet$ changes happen across the critical values - $\pi_0(X_{1/3}) \cong \pi_0(X_{1/2})$ and $\pi_0(X_{1/2}) \ncong \pi_0(X_1)$, while in $Y_\bullet$ changes happen at the critical values - $\pi_0(Y_{1/3}) \ncong \pi_0(Y_{1/2})$ and $Y_{1/2} \cong Y_1$.

It is then clear that $X_\bullet$ and $Y_\bullet$ are not isomorphic as abstract merge trees; but at the same time they differ only by their behavior at critical points. We are not interested in distinguishing two such behaviors and for this reason we ask for a weaker notion of equivalence between abstract merge trees. Coherently, the frameworks which we will build in later sections are invariant to such weaker equivalence relationship.

Given $Z \subset \mathbb{R}$, clearly Z inherits an ordering from the one in $\mathbb{R}$ and we can consider Z as a poset category. Thus, we can take the restriction to Z of any filtration of topological spaces $X_\bullet$ (and similarly of any persistent set) via the inclusion $Z \hookrightarrow \mathbb{R}$. We indicate this restriction as $X_{\bullet\,|Z}$. Moreover, $\mathcal{L}$ is going to be the Lebesgue measure on $\mathbb{R}$. Refer to Figure 2.2a for a visual interpretation of the following definitions and propositions.

Definition 2.11. *Two persistent sets $\pi_0(X_\bullet)$ and $\pi_0(Y_\bullet)$ are almost everywhere (a.e.) isomorphic if there is a Lebesgue measurable set $Z \subset \mathbb{R}$ such that $\mathcal{L}(\mathbb{R} - Z) = 0$ and there is a natural isomorphism $\alpha : \pi_0(X_{\bullet\,|Z}) \to \pi_0(Y_{\bullet\,|Z})$. We write $\pi_0(X_\bullet) \cong_{a.e} \pi_0(Y_\bullet)$.*

Proposition 2.12. *Being a.e. isomorphic is an equivalence relationship between persistent sets.*

Proof. Reflexivity and symmetry are trivial: the first one holds with $Z = \emptyset$ and the second one holds by definition of natural isomorphism. Lastly, transitivity holds because any finite union of measure zero sets is a measure zero set. □

Now we prove that in each equivalence class of a.e. isomorphic abstract merge trees we can always pick a regular abstract merge tree, which is unique up to isomorphism.

Proposition 2.13. *For every abstract merge tree $\pi_0(X_\bullet)$ there is a unique (up to isomorphism) abstract merge tree $R(\pi_0(X_\bullet))$ such that:*

1. *$\pi_0(X_\bullet) \cong_{a.e.} R(\pi_0(X_\bullet))$;*
2. *$R(\pi_0(X_\bullet))$ is regular.*

Proof. Let $\pi_0(X_\bullet)$ be an abstract merge tree with critical values $t_1 < \ldots < t_n$. Suppose that at t_j changes happen across the critical value. Then we can fix $\varepsilon > 0$ such that $t_j + \varepsilon < t_{j+1}$ and define $X'_\bullet$ with $X'_t = X_t$ for all $t \neq t_j$ and $X'_{t_j} = X_{t_j+\varepsilon}$. Now we need to define the $X'_\bullet$ on maps:

- if $t = t_j$ and $t < t' \leq t_j + \varepsilon$, $X'_{t<t'} = (X_{t' \leq t_j+\varepsilon})^{-1}$ which is well defined as $X_{t' \leq t_j+\varepsilon}$ is an isomorphism;
- if $t' = t_j$, $X'_{t<t'} = X_{t \leq t_j+\varepsilon}$ which is well defined as $X_{t' \leq t_j+\varepsilon}$ is an isomorphism;
- otherwise $X'_{t<t'} = X_{t<t'}$.

We need to check that $X'_\bullet$ is a regular abstract merge tree. First we have:

$$X'_{t,t_j} \circ X'_{t_j,t'} = X_{t \leq t_j+\varepsilon} \circ (X_{t' \leq t_j+\varepsilon})^{-1} = X_{t,t'} = X'_{t,t'}$$

if $t' \leq t_j + \varepsilon$, otherwise

$$X'_{t,t_j} \circ X'_{t_j,t'} = X_{t \leq t_j+\varepsilon} \circ X_{t_j+\varepsilon \leq t'} = X_{t,t'} = X'_{t,t'}.$$

The filtration $X'_\bullet$ is regular at t_j by construction as $X'_{t_j} = X_{t_j+\varepsilon} \cong X'_{t'}$ for $t' \in [t_j, t_j+\varepsilon]$. Always by construction, it is a.e. isomorphic to $X_\bullet$: the natural transformation $\varphi : X_\bullet \to X'_\bullet$ is given by $\varphi_t = Id : X_t \to X'_t$ for $t \neq t_j$ and, in fact, it is defined on $\mathbb{R} - \{t_1, \ldots, t_n\}$.

If t_j is the only critical value at which changes in $X_\bullet$ happen across the value we are done, otherwise consider t_k such that changes in $X_\bullet$ happen across t_k. The same, by construction, holds also for $X'_\bullet$. Thus we can recursively apply the steps proposed up to now on $X'_\bullet$ until we obtain an abstract merge tree $R(\pi_0(X_\bullet))$ which is regular. This is reached in a finite number of steps since the critical values are a finite set.

Uniqueness (up to isomorphism) follows easily. □

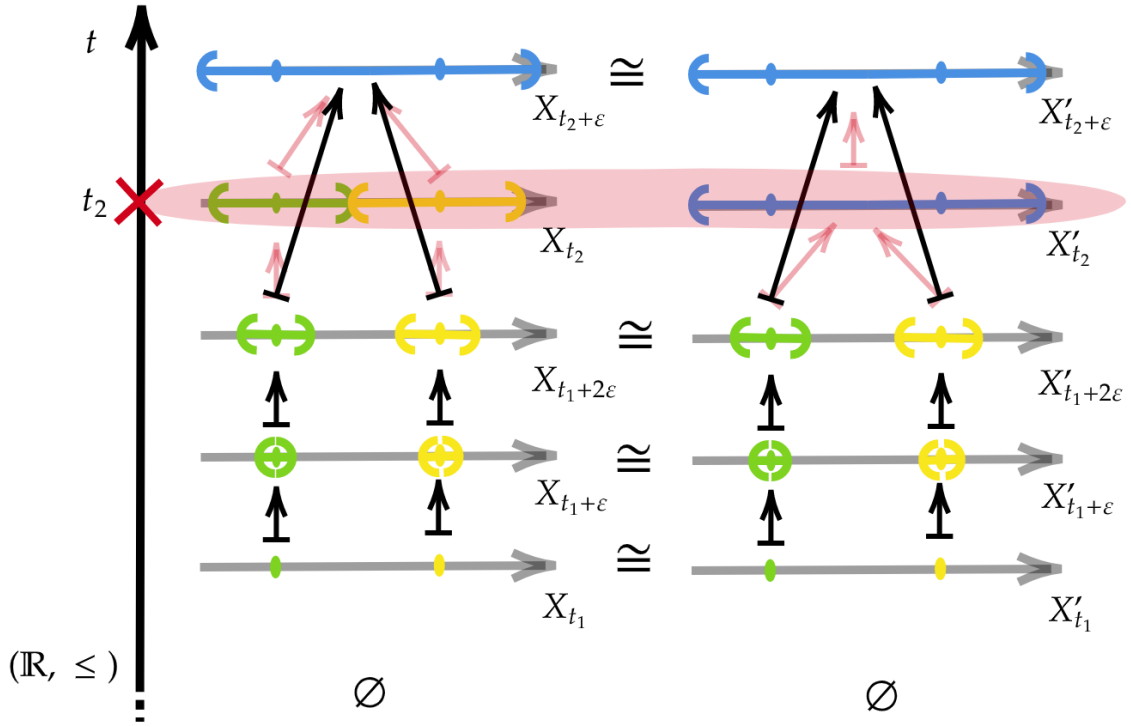
Regular abstract merge trees are the functors we want to focus on, for they make many upcoming definitions and results more natural and straightforward. With Proposition 2.13 we formally state that this choice is indeed consistent with the equivalence relationship previously established.

A more detailed discussion on the topological consequences of the regularity condition - in the particular case where $X_\bullet$ is the sublevel set filtration of a real valued function - can be found in Chapter 4.

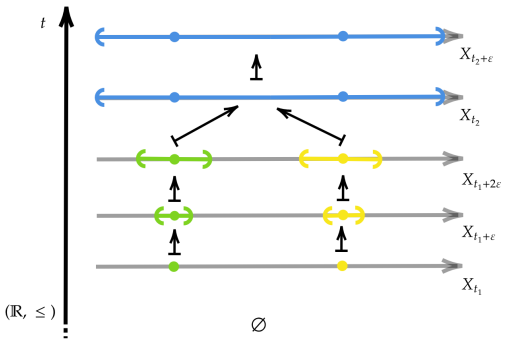
2.3 Merge Trees

We introduce now the discrete counterpart of abstract merge trees, which (up to some minor technical differences) are called *merge trees* by part of the scientific literature dealing with these topics (Gasparovic et al., 2019; Sridharamurthy et al., 2020), while Curry et al. (2022) refers to such structures as *computational* merge trees. Even thou we agree with the idea behind the latter terminology, we stick with the wording used by Gasparovic et al. (2019) and others. We do so for the sake of simplicity, as these objects will be the main focus of the theoretic investigation of the manuscript. Before proceeding, we point out that there is a third approach - on top of the categorical and the computational ones - to the definition of merge trees, followed for instance by Morozov et al. (2013), which in Curry et al. (2022) is referred to as *classical* merge trees. We avoid dealing with such objects in our dissertation and any interested reader can find in Curry et al. (2022) how that definition relates with the other ones we report.

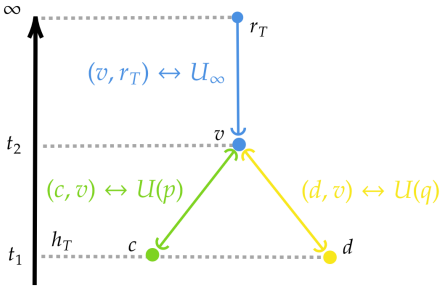
Now we need some graph-related definitions.



(a) An abstract merge tree $\pi_0(X_\bullet)$ (left) and the regular abstract merge tree $R(\pi_0(X_\bullet))$ (right).



(b) A regular abstract merge tree $R(\pi_0(X_\bullet))$.



(c) The merge tree $\mathcal{M}(R(\pi_0(X_\bullet)))$. The brackets at the end of the edges and the labels $U(p), U(q), U_\infty$ refer to the canonical a.e. covering defined in Section 2.5.2.

Figure 2.2: On the first line we can see an example of an abstract merge tree which is not regular (left) along with the regular abstract merge tree (right) obtained as in Proposition 2.13. There is also highlighted the a.e. isomorphism between them: they are isomorphic on $\mathbb{R} - \{t_2\}$. On the second line we find a regular abstract merge tree and the associated merge tree built as in Proposition 2.19. The colors are again used to highlight the relationships between the different objects.

Definition 2.14. *A tree structure T is given by a set of vertices V_T and a set of edges $E_T \subset V_T \times V_T$ which form a connected rooted acyclic graph. We indicate the root of the tree with r_T. We say that T is finite if V_T is finite. The order of a vertex $v \in V_T$ is the number of edges which have that vertex as one of the extremes, and is called $ord_T(v)$. Any vertex with an edge connecting it to the root is its child and the root is its father: this is the first step of a recursion which defines the father and children relationship for all vertices in V_T. The vertices with no children are called leaves or taxa and are collected in the set L_T. The relation $child < father$ generates a partial order on V_T. The edges in E_T are identified in the form of ordered couples (a, b) with $a < b$. A subtree of a vertex v, called $sub_T(v)$, is the tree structure whose set of vertices is $\{x \in V_T | x \leq v\}$.*

Note that, given a tree structure T, identifying an edge (v, v') with its lower vertex v, gives a bijection between $V_T - \{r_T\}$ and E_T, that is $E_T \cong V_T - \{r_T\}$ as sets. Given this bijection, we often use E_T to indicate the vertices $v \in V_T - \{r_T\}$, to simplify the notation.

We want to identify merge trees independently of their vertex set, and thus we introduce the following isomorphism classes.

Definition 2.15. *Two tree structures T and T' are isomorphic if exists a bijection $\eta : V_T \to V_{T'}$ that induces a bijection between the edges sets E_T and $E_{T'}$: $(a, b) \mapsto (\eta(a), \eta(b))$. Such η is an isomorphism of tree structures.*

Finally, we give the definition of a merge tree, slightly adapted from Gasparovic et al. (2019).

Definition 2.16. *A merge tree is a finite tree structure T with a monotone increasing height function $h_T : V_T \to \mathbb{R} \cup \{+\infty\}$ and such that 1) $ord_T(r_T) = 1$ 2) $h_T(r_T) = +\infty$ 3) $h_T(v) \in \mathbb{R}$ for every $v < r_T$.*

Two merge trees (T, h_T) and $(T', h_{T'})$ are isomorphic if T and T' are isomorphic as tree structures and the isomorphism $\eta : V_T \to V_{T'}$ is such that $h_T = h_{T'} \circ \eta$. Such η is an isomorphism of merge trees. We use the notation $(T, h_T) \cong (T', h_{T'})$.

With some slight abuse of notation we set $\max h_T = \max_{v \in V_T \,|\, v < r_T} h_T(v)$ and $\arg\max h_T = \max\{v \in V_T \,|\, v < r_T\}$. Note that, given (T, h_T) merge tree, there is only one edge of the form (v, r_T) and we have $v = \arg\max h_T$.

The relationship between abstract merge trees and merge trees is clarified in Section 2.3.1, but before going on we must introduce another equivalence relationship on merge trees.

Definition 2.17. *Given a tree structure T, we can eliminate an order two vertex, connecting the two adjacent edges which arrive and depart from it. Suppose we have two edges $e = (v_1, v_2)$ and $e' = (v_2, v_3)$, with $v_1 < v_2 < v_3$. And suppose v_2 is of order two. Then, we can remove v_2 and merge e and e' into a new edge $e'' = (v_1, v_3)$. This operation is called the* ghosting *of the vertex v_2. Its inverse transformation, which restores the original tree, is called a* splitting *of the edge e''.*

Consider a merge tree (T, h_T) and obtain T' by ghosting a vertex of T. Then $V_{T'} \subset V_T$ and thus we can define $h_{T'} := h_{T \,|\, V_{T'}}$.

Now we can state the following definition.

Definition 2.18. *Merge trees are equal up to order 2 vertices if they become isomorphic after applying a finite number of ghostings or splittings. We write $(T, h_T) \cong_2 (T', h_{T'})$.*

2.3.1 Regular Abstract Merge Trees and Merge Trees

In this section we study the relationship between abstract merge trees and merge trees. We collect all the important facts on this topic in the following proposition. Figure 2.1b and Figure 2.2c can help the reader going through the upcoming results.

Proposition 2.19. *The following hold:*

1. *we can associate a merge tree without order 2 vertices $\mathcal{M}(R(\pi_0(X_\bullet)))$ to any regular abstract merge tree $R(\pi_0(X_\bullet))$;*
2. *we can associate a regular abstract merge tree $\mathcal{F}((T,h_T))$ to any merge tree (T,h_T). Moreover, we have $\mathcal{M}(\mathcal{F}((T,h_T))) \cong_2 (T,h_T)$ and $\mathcal{F}(\mathcal{M}(R(\pi_0(X_\bullet))) \cong_{a.e.} \pi_0(X_\bullet)$;*
3. *given two abstract merge trees $X_\bullet$ and $Y_\bullet$, $\mathcal{M}(R(\pi_0(X_\bullet))) \cong \mathcal{M}(R(\pi_0(Y_\bullet)))$ if and only if $\pi_0(X_\bullet) \cong_{a.e} \pi_0(Y_\bullet)$.*
4. *given two merge trees (T,h_T) and $(T',h_{T'})$, we have $\mathcal{F}((T,h_T)) \cong \mathcal{F}((T,h_T))$ if and only if $(T,h_T) \cong_2 (T'.h_{T'})$.*

Proof. 1. WLOG suppose $\pi_0(X_\bullet) \cong R(\pi_0(X_\bullet))$; we build the merge tree $\mathcal{M}(\pi_0(X_\bullet)) = (T,h_T)$ along the following rules in a recursive fashion starting from an empty set of vertices V_T and an empty set of edges E_T. We simultaneously add points and edges to T and define h_T on the newly added vertices. Let $\{t_i\}_{i=1}^n$ be the critical set of $\pi_0(X_\bullet)$ and let $\pi_0(X_t) := a_t := \{a_1^t,\dots,a_{n_t}^t\}$. Call $\psi_t^{t'} := \pi_0(X_{t\le t'})$. Lastly, from now on, we indicate with $\#C$ the cardinality of a finite set C.

Considering in increasing order the critical values:

- for the critical value t_1 add to V_T a leaf $a_{t_1}^k$, with height t_1, for every element $a_{t_1}^k \in a_{t_1}$;
- for t_i with $i>1$, for every $a_{t_i}^k \in a_{t_i}$ such that $a_{t_i}^k \notin \mathrm{Im}(\psi_{t_{i-1}}^{t_i})$, add to V_T a leaf $a_{t_i}^k$ with height t_i;
- for t_i with $i>1$, if $a_{t_i}^k = \psi_{t_{i-1}}^{t_i}(a_{t_{i-1}}^s) = \psi_{t_{i-1}}^{t_i}(a_{t_{i-1}}^r)$, with $a_{t_{i-1}}^s$ and $a_{t_{i-1}}^r$ distinct basis elements in $a_{t_{i-1}}$, add a vertex $a_{t_i}^k$ with height t_i, and add edges so that the previously added vertices

$$v = \arg\max\{h_T(v') \,|\, v' \in V_T \text{ s.t. } \psi_{t_{v'}}^{t_i}(v') = a_{t_i}^k\}$$

and

$$w = \arg\max\{h_T(w') \,|\, w' \in V_T \text{ s.t. } \psi_{t_{w'}}^{t_i}(w') = a_{t_i}^k\}$$

connect with the newly added vertex $a_{t_i}^k$.

The last merging happens at height t_n and, by construction, at height t_n there is only one point, which is the root of the tree structure.

These rules define a tree structure with a monotone increasing height function h_T. In fact, edges are induced by maps $\psi_t^{t'}$ with $t<t'$ and thus we can have no cycles and the function h_T must be increasing. Moreover, we have $\psi_t^{t_n}(a_i^t) = a_1^{t_n}$ for every i and $t<t_n$ and thus the graph is path connected.

2. Now we start from a merge tree (T, h_T) and build an abstract merge tree $\pi_0(X_\bullet)$ such that $\mathcal{M}(\pi_0(X_\bullet)) \cong (T, h_T)$.

   To build the abstract merge tree, the idea is that we would like to "cut" (T, h_T) at every height t and take as many elements in the set of path connected components as the edges met by the cut.

   Let $\{t_1, \dots, t_n\}$ be the ordered image of h_T in $\mathbb{R}$.

   Then consider the sets $v_{t_j} = \{v_i^{t_j}\}_{i=1,\dots,n_{t_j}} = h_T^{-1}(t_j)$. We use the notation $\mathcal{F}((T, h_T))_t := a_t := \{a_1^t, \dots, a_{n_t}^t\}$. We define $a_{t_1} = v_{t_1}$. For every $\varepsilon > 0$ such that $t_1 - t_0 > \varepsilon$, we set $a_{t_1+\varepsilon} = a_{t_1}$ and consequently $\psi_t^{t'} = Id$ for every $[t, t'] \subset [t_1, t_2)$. Now we build a_{t_2} starting from a_{t_1} and using v_{t_2}. We need to consider $v_i^{t_2} \in v_{t_2}$. There are two possibilities:

   - if $v_i^{t_2}$ is a leaf, then we add $v_i^{t_2}$ to a_{t_1};
   - if $v_i^{t_2}$ is an internal vertex with $\#\text{child}(v_i^{t_2}) > 1$ - i.e. a merging point, we add $v_i^{t_2}$ to a_{t_1} and then remove $\text{child}(v_i^{t_2}) = \{v \in V_T \mid v \text{ is a children of } v_i^{t_2}\}$. Note that, by construction $\text{child}(v_i^{t_2}) \subset a_{t_1}$ and by hypothesis $\#\text{child}(v_i^{t_2}) > 1$;
   - if $v_i^{t_2}$ is an internal vertex with $\#\text{child}(v_i^{t_2}) = 1$ - i.e. an order 2 vertex, we don't do anything.

   By doing these operations for every $v_i^{t_2} \in v_1$, we obtain a_{t_2}. The map $\psi_t^{t_2}$, for $t \in [t_1, t_2)$ is then defined by setting $\psi_t^{t_2}(a_i^{t_1}) = v_i^{t_2}$ if $a_i^{t_1} \in \text{child}(v_i^{t_2})$ and $\psi_t^{t_2}(a_i^{t_1}) = a_i^{t_1}$ otherwise. To define a_t for $t > t_2$ we recursively repeat for every critical value t_i (in increasing order) the steps of defining $a_{t_i+\varepsilon}$ equal to a_{t_i} for small $\varepsilon > 0$ and then adjusting (as explained above) a_{t_i} according to the tree structure to obtain a_{i+1} and $\psi_{t_i}^{t_{i+1}}$. When reaching t_n we have $v_{t_n} = \{v_1^{t_n}\}$ and we set $a_t = v_{t_n}$ for every $t \geq t_n$.

   We call this persistent set $\mathcal{F}((T, h_T))$. Note that, by construction:

   - for every $v \in V_T$ we have $v \in a_t$ for $t \in [h_T(v), h_T(\text{father}(v))$;
   - $\mathcal{F}((T, h_T))$ is regular;
   - $\mathcal{F}((T, h_T))$ is independent from order 2 vertices of (T, h_T);
   - $\mathcal{F}((T, h_T))$ is an abstract merge tree.

   Now we need to check that $(T', h_{T'}) = \mathcal{M}(\mathcal{F}((T, h_T))) \cong_2 (T, h_T)$. WLOG we suppose (T, h_T) is without order 2 vertices and prove $(T', h_{T'}) = \mathcal{M}(\mathcal{F}((T, h_T))) \cong (T, h_T)$. Let $\pi_0(X_\bullet) = \mathcal{F}((T, h_T))$.

   As before, for notational convenience, we set $a_t := \pi_0(X_t)$ and $\psi_t^{t'} := \pi_0(X_{t \leq t'})$. By construction, $a_t \subset V_T$ for every t. Which implies $V_{T'} \subset V_T$.

   Consider now a_{t_i} with t_i critical value. To build $\pi_0(X_\bullet)$ elements $a_j^{t_{i-1}}, a_k^{t_{i-1}} \in a_{t_{i-1}}$ are replaced by v in a_{t_i} if and only if they merge with v in the merge tree (T, h_T): $(a_j^{t_{i-1}}, v), (a_k^{t_{i-1}}, v)$, with $h_T(v) = t_i$. The maps $\psi_{t_i}^{t_i} : a_{t_{i-1}} \to a_{t_{i-1}}$ are defined accordingly to represent that merging mapping $a_j^{t_{i-1}} \mapsto v$ and $a_k^{t_{i-1}} \mapsto v$. So an element v' stays in a_t until the edge $(v', father(v'))$ meets another edge in (T, h_T), and then is replaces by $father(v')$. As a consequence, we have $a_j^{t_{i-1}}, a_k^{t_{i-1}}, v \in V_{T'}$ and $(a_j^{t_{i-1}}, v), (a_k^{t_{i-1}}, v) \in E_{T'}$.

   Since (T, h_T) has no order 2 vertices then 1) $V_T = \bigcup_{i=1,\dots,n} a_{t_i}$ 2) $V_T = V_{T'}$ 3) $id : V_T \to V_{T'}$ is an isomorphism of merge trees.

Now we consider $\pi_0(X_\bullet)$ regular abstract merge tree and prove $\mathcal{F}(\mathcal{M}(R(\pi_0(X_\bullet)))) \cong \pi_0(X_\bullet)$. Consider t_i critical value, $\varepsilon > 0$ such that $t_{i-1} < t_i - \varepsilon$ and let $v_{t_i} = \{v \in \pi_0(X_{t_i}) \,|\, \#\pi_0(X_{t_i-\varepsilon<t_i})^{-1}(v) \neq 1\}$. By construction, $v_{t_i} \subset V_T$, for every t_i critical value, with $(T, h_T) = \mathcal{M}(R(\pi_0(X_\bullet)))$.

For every $v \in \pi_0(X_t)$, for any $t \in \mathbb{R}$ there is $v_j^{t_i} \in v_{t_i}$ for some t_i, such that $\pi_0(X_{t_i \leq t})(v_j^{t_i}) = v$. Moreover the following element is well defined:

$$s(v) := \max\{w \in v_{t_i}, t_i \text{ critical value } \,|\, \pi_0(X_{t_i \leq t})(w) = v\}$$

By construction we have $v = \pi_0(X_{t_i \leq t})(s(v))$.

Let $\pi_0(Y_\bullet) = \mathcal{F}(\mathcal{M}(R(\pi_0(X_\bullet))))$. Define $\alpha_t : \pi_0(X_t) \to \pi_0(Y_t)$ given by $v = \pi_0(X_{t_i \leq t})(s(v)) \mapsto s(v)$. It is an isomorphism of abstract merge trees.

3. if $\pi_0(X_\bullet) \cong_{a.e.} \pi_0(Y_\bullet)$, then $R(\pi_0(X_\bullet)) \cong R(\pi_0(Y_\bullet))$ and then the merge trees $\mathcal{M}(R(\pi_0(X_\bullet))$ and $\mathcal{M}(R(\pi_0(Y_\bullet))$ differ just by a change in the names of the vertices. If $\mathcal{M}(R(\pi_0(X_\bullet)) \cong \mathcal{M}(R(\pi_0(Y_\bullet))$ then $\mathcal{F}(\mathcal{M}(R(\pi_0(X_\bullet)))) \cong \mathcal{F}(\mathcal{M}(R(\pi_0(Y_\bullet)))) \cong R(\pi_0(X_\bullet)) \cong R(\pi_0(Y_\bullet))$.

4. the proof is analogous to the one of the previous point, with regularity condition on abstract merge trees being replaced by being without order 2 vertices for merge trees.

□

We point out an additional fact about order 2 vertices. Suppose that we were to remove a leaf in a merge tree, the father of the deleted vertex may become an order two vertex. In case that happens, such vertex carries no topological information, since the merging that the point was representing, is no more happening (was indeed removed). And in fact the abstract merge tree associated to the merge tree with the order 2 vertex and to the merge tree with the order 2 vertex ghosted are the same by Proposition 2.19. Thus working up to order two vertices is a very natural framework to work with merge trees. And this must be taken into consideration when setting up the framework to deal with functions defined on merge trees.

The proof of Proposition 2.19 carries this important corollary.

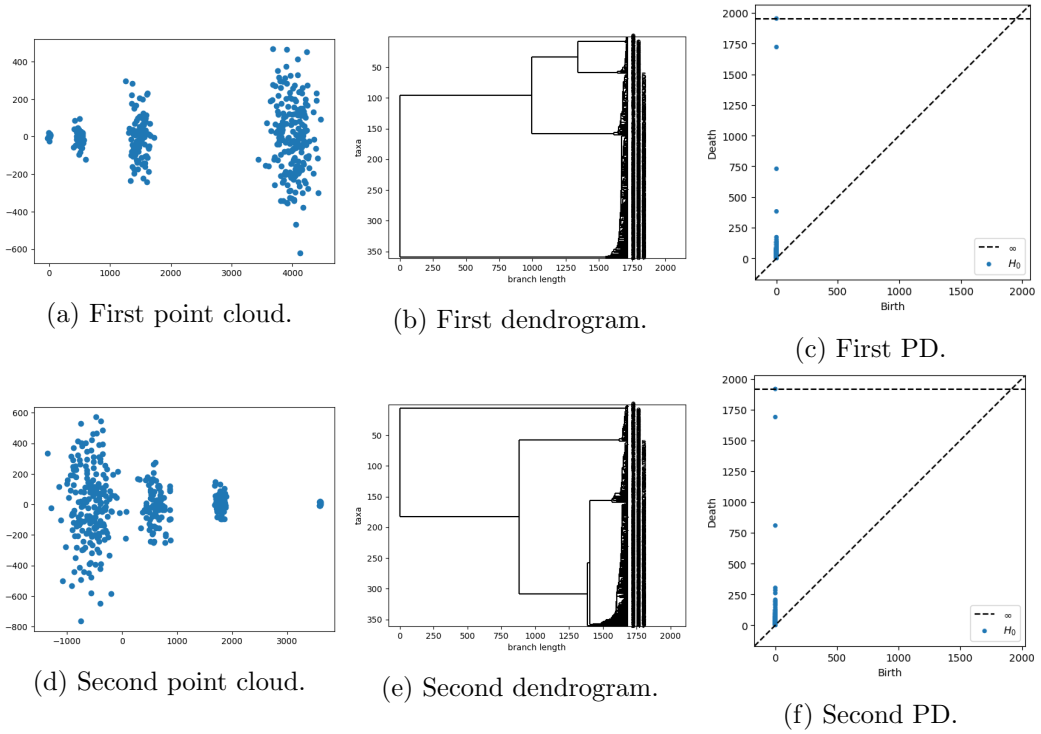
Corollary 2.20. *Given a merge tree (T, h_T) and the abstract merge tree $\pi_0(X_\bullet) = \mathcal{F}((T, h_T))$, we have $E_T \hookrightarrow D_{\pi_0(X_\bullet)}$ via $(v, v') \mapsto (v, h_T(v))$.*

2.3.2 Example

Consider the function $f = |\,|\,x\,| - 1\,|$ defined on the interval $[-2, 2]$. Consider the sublevel set filtration $X_t = f^{-1}((-\infty, t])$. The sublevel set X_t is an interval of the form $[-1 - t, -1 + t] \bigcup [1 - t, 1 + t]$, for $t \in [0, 1]$.

Consider then the abstract merge tree $\pi_0(X_\bullet)$. For any $t \in [0, 1)$, the path connected components are $a_t = \{a_1^t, a_{-1}^t\}$, with $a_1^t = [1 - t, 1 + t]$ and $a_{-1}^t = [-1 - t, -1 + t]$ and for $t \geq 1$, $a_2^t = \{[-2, 2]\}$. The critical points of the filtration are $t_1 = 0$ and $t_2 = 1$. The maps are $a_i^t \mapsto a_i^{t'}$ and with $i = -1, 1$, for $t \leq t' < 2$; $a_1^t, a_{-1}^t \mapsto a_2^{t'}$ for $t < 2 \leq t'$ and the identity for $t, t' \geq 2$.

The merge tree $\mathcal{M}(\pi_0(X_\bullet)) = (T, h_T)$ associated to $\pi_0(X_\bullet)$ has a tree structure given by a root, an internal vertex and two leaves - as in Figure 2.2c: if we call $v_1 := a_1^0$, $v_{-1} := a_{-1}^0$ and $v_2 := a_2^2$, the merge tree $\mathcal{M}(\pi_0(X_\bullet))$ is given by the vertex set $\{v_1, v_{-1}, v_2, r_T\}$ and edges $e_1 = (v_1, v_2)$, $e_2 = (v_{-1}, v_2)$ and $e_3 = (v_2, r_T)$. The height function has values $h_T(v_1) = h_T(v_{-1}) = t^- = 0$, $h_T(v_2) = 2$ and $h_T(r_T) = +\infty$.



(a) First point cloud. (b) First dendrogram. (c) First PD.

(d) Second point cloud. (e) Second dendrogram. (f) Second PD.

Figure 2.3: Data clouds, hierarchical clustering dendrograms and PDs involved in the first example.

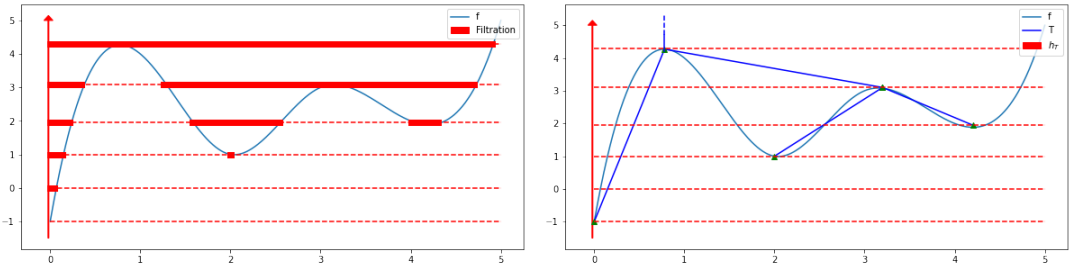
2.4 Why Using Trees

After having introduced abstract merge trees and merge trees to represent the evolution of the path connected components of a filtration, we want to give some motivation to propel the use of such topological summaries over persistence diagrams, in certain situations. We give only two brief examples since a similar topic is already tackled for instance in Elkin and Kurlin (2020), Smith and Kurlin (2022), Kanari et al. (2020),Curry et al. (2021) and Curry et al. (2022).

2.4.1 Point Clouds

Given a point cloud $C = \{x_1, \dots, x_n\}$ in $\mathbb{R}^n$ there are many ways in which one can build a family of simplicial complexes (Edelsbrunner and Harer, 2008) whose vertices are given by C itself and whose set of higher dimensional simplices gets bigger and bigger. A standard tool to do so is the Vietoris-Rips filtration of C (Edelsbrunner and Harer, 2008), as are α filtrations, Céch filtrations etc..

As we are interested only in path connected components we restrict our attention to 0 dimensional simplices (points) and 1 dimensional simplices (edges). With such restrictions, many of the aforementioned filtrations become equivalent and amount to having a family of graphs $\{C_t\}_{t\geq 0}$ such that the vertex set of C_t is C and the edge between x_i and x_j belongs to C_t if and only if $d(x_i, x_j) < t$. Thus, the set of edges of $C_{t'}$ contains the set of edges of C_t, with $t \leq t'$; while the set of vertices is always C. Note, for instance, that the path connected components of C_t are equivalent to the ones of $X_{t/2}$ with $X_\bullet$ being the Céch filtration built in Section 2.2.1. Along this filtration of graphs, the closest points become connected first and the farthest ones at last. It is thus reasonable to interpret the path connected components of C_t as clusters of the point cloud C. In order to choose the best "resolution" to look at clusters, i.e. in order to choose t and use C_t to infer the



(a) Sublevel sets of a function (b) A function with its associated merge tree.

Figure 2.4: Merge Trees of Functions

clusters, statisticians look at the merge tree $\mathcal{M}(\pi_0(C_t)_{t\geq 0})$, which is called hierarchical clustering dendrogram. More precisely, $\mathcal{M}(\pi_0(C_t)_{t\geq 0})$ is the *single linkage hierarchical clustering dendrogram*. Note that $\pi_0(X_\bullet)$ is a regular abstract merge tree.

Suppose, instead, that we have the persistence diagram obtained from $\{\pi_0(C_t)\}_{t\in\mathbb{R}_{\geq 0}}$. Persistence diagrams are made of points in $\mathbb{R}^2$ whose coordinates (b, d) represent the value of t at which a certain path-connected component appears and the value of t at which that component merges with a component which appeared before b. Each point in the point cloud is associated to a path connected component but, in general, we have no way to distinguish between points of the diagram associated to path connected components which are proper clusters and points of the diagrams associated to outliers.

Now, consider the single linkage dendrograms and the zero dimensional PDs obtained from point clouds as in Figure 2.3. The persistence diagrams (in Figure 2.3c and Figure 2.3f) are very similar, in fact they simply record that there are four major clusters which merge at similar times across the Vietoris-Rips filtrations of the two point clouds. The hierarchical dendrograms, instead, are clearly very different since they show that in the first case (Figure 2.3a, Figure 2.3b, Figure 2.3c) the cluster with most points is the one which is more separated from the others in the point cloud; while in the second case (Figure 2.3d, Figure 2.3e, Figure 2.3f) the two bigger clusters are the first that get merged and the farthest cluster of points on the right could be considered as made by outliers. In many applications it would be important to distinguish between these two scenarios, since the two main clusters get merged at very different heights on the respective dendrograms.

These observations are formalized in Curry et al. (2021), with the introduction of the *tree realization number* with is a combinatorial description of how many merge trees share a particular persistence diagram. With hierarchical clustering dendrograms with n leaves, such number is $n!$: all leaves are born at height 0, and so, at the first merging point, each of the n leaves can merge with any of the $n-1$ remaining ones. At the following merging step we have $n-1$ clusters and each one of them can merge with the other $n-2$ etc..

2.4.2 Real Valued Functions

Given a continuous function $f : [a, b] \to \mathbb{R}$ we can extract the merge tree $\mathcal{M}(\pi_0(X_\bullet))$, with $X_\bullet$ being the sublevel set filtration (see Section 2.2.1 and Section 2.3.2): we obtain a merge tree that tracks the evolution of the path connected components of the sublevel sets $f^{-1}((-\infty, t])$. For a visual example see Figure 2.4b. Chapter 4 shows that $\pi_0(X_\bullet)$ is a regular merge tree.

We use this example to point out two facts. First PDs may not be able to distinguish functions one may wish to distinguish, as made clear by Figure 2.5. Second, Proposition 1 of Chapter 4 states that if one changes the parametrization of a function by means of homeomorphisms, then, both the associated merge tree and persistence diagram do not

change. A consequence of such result is that one can shrink or spread the domain of the function $f : [a, b] \to \mathbb{R}$ with reasonably regular functions, without changing its merge tree (and PD). There are cases in which such property may be useful but surely there are times when one may want to distinguish if an oscillation lasted for a time interval of 10^{-5} or 10^{5}. As we will see in the following section, being able to embed in a topological representation some kind of additional information about such oscillations, could help in dealing with those situations.

2.5 Functions Defined on Display Posets

Now we formalize how we want to deal with functions defined on merge trees, devoting much care to setting up a framework in accordance with the equivalence relationships introduced in Section 2.2.2.

2.5.1 Metric Spaces

Following Burago et al. (2022), we briefly report the definitions related to metric geometry that we need in the present work.

Definition 2.21. *Let X be an arbitrary set. A function $d : X \times X \to \mathbb{R}$ is a (finite) pseudo metric if for all $x, y, z \in X$ we have:*

1. $d(x, x) = 0$
2. $d(x, y) = d(y, x)$
3. $d(x, y) \leq d(x, z) + d(z, y)$.

The space (X, d) is called a pseudo metric space.

Given a pseudo metric d on X, if for all $x, y \in X$, $x \neq y$, we have $d(x, y) > 0$ then d is called a metric or a distance and (X, d) is a metric space.

Proposition 2.22 (Proposition 1.1.5 Burago et al. (2022))**.** *For a pseudo metric space (X, d), $x \sim y$ iff $d(x, y) = 0$ is an equivalence relationship and the quotient space $(X, d)/\sim$ is a metric space.*

Definition 2.23. *Consider X, Y pseudo metric spaces. A function $f : X \to Y$ is an isometric embedding if it is injective and $d(x, y) = d(f(x), f(y))$. If f is also bijective then it is an isometry or and isometric isomorphism.*

Definition 2.24. *A pseudo metric d on X induces the topology generated by the open balls $B_\varepsilon(x) := \{y \in X \mid d(x, y) < \varepsilon\}$.*

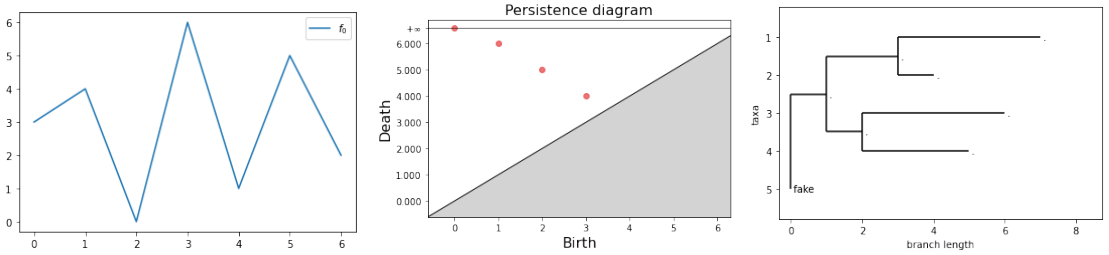
2.5.2 The Display Poset as a Pseudo-Metric Space

Now we start the proper discussion to build function spaces on display posets. We begin by giving the notion of *common ancestors* for subsets of the display poset of an abstract merge tree.

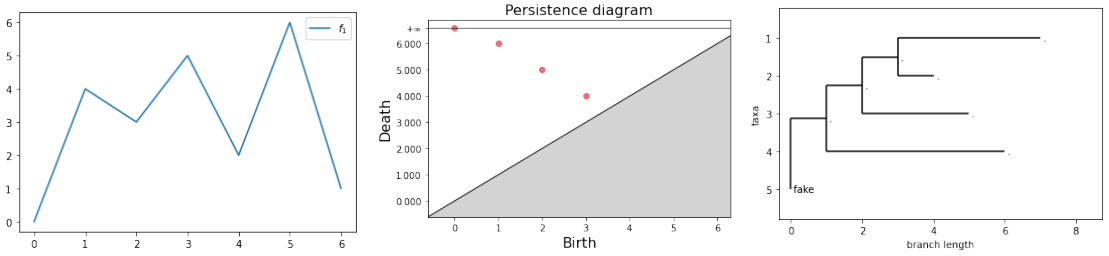
Definition 2.25. *Given $Q \subset D_{\pi_0(X_\bullet)}$, with $\sup h(Q) < \infty$, then common ancestors of Q is the set $\mathrm{CA}(Q)$ defined as:*

$$\mathrm{CA}(Q) = \{p \in D_{\pi_0(X_\bullet)} \mid p \geq Q\}$$

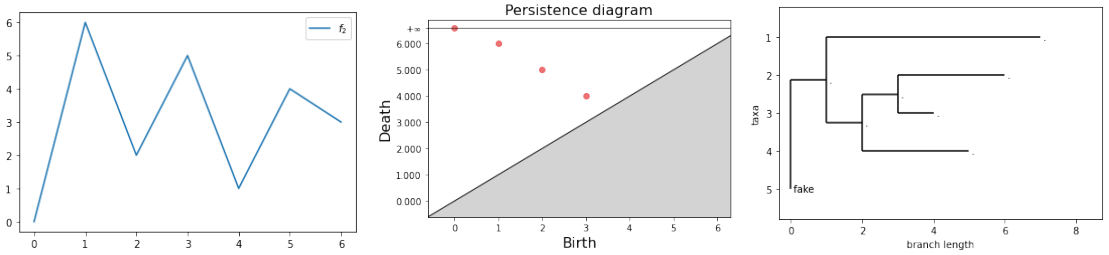
If $\pi_0(X_\bullet)$ is regular then we have a well defined element $\min \mathrm{CA}(Q)$ which we call the least common ancestor $\mathrm{LCA}(Q)$.



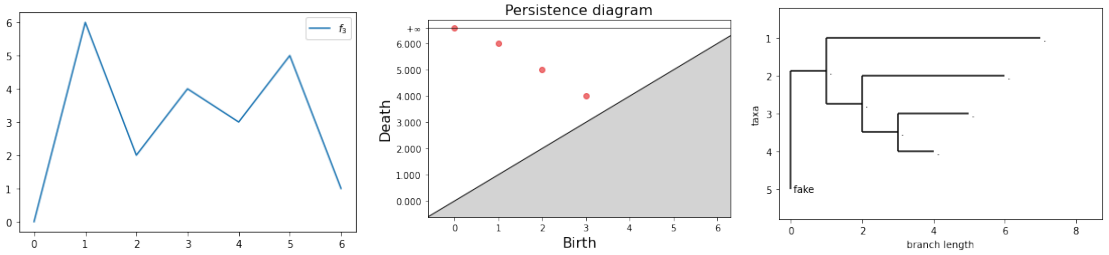
(a) A function f_0. (b) The 0-dimensional persistence diagram of f_0. (c) The merge tree of f_0.



(d) A function f_1. (e) The 0-dimensional persistence diagram of f_1. (f) The merge tree of f_1.



(g) A function f_2. (h) The 0-dimensional persistence diagram of f_2. (i) The merge tree of f_2.



(j) A function f_3. (k) The 0-dimensional persistence diagram of f_3. (l) The merge tree of f_3.

Figure 2.5: We compare four functions; they are all associated to the same PD but to different merge trees. Functions are displayed in the first column and on each row we have on the centre the associated PD and on the right the merge tree.

The definition is well posed since $\{p \in D_{\pi_0(X_\bullet)} \,|\, p \geq Q\}$ is non empty if $\sup h(Q) < \infty$. Moreover it is bounded from below in terms of h. Clearly, if $Q \subset Q' \subset D_{\pi_0(X_\bullet)}$ then $\inf \mathrm{CA}(Q) \leq \inf \mathrm{CA}(Q')$.

Proposition 2.26. *The display poset $D_{\pi_0(X_\bullet)}$ of any abstract merge tree can be given a pseudo-metric structure with the following formula:*

$$d((a,t),(b,t')) = \tilde{t} - t + \tilde{t} - t'$$

with $\tilde{t} = \inf\{h(p) \,|\, p \in \mathrm{CA}(\{(a,t),(b,t')\})\}$. If $\pi_0(X_\bullet)$ is regular then d is a metric.

Proof. First note that even if $\inf \mathrm{CA}(Q)$, with $Q \subset D_{\pi_0(X_\bullet)}$ and $\sup h(Q) < \infty$, may be a set with more than one element, $\inf\{h(p) \text{ with } p \in D_{\pi_0(X_\bullet)} \,|\, p \geq Q\}$ is uniquely defined. Moreover, consider $p = (b, t_b), q = (c, t_c) \in \inf \mathrm{CA}(Q)$. For every $(a,t) \in \mathrm{CA}(Q)$ we know $\pi_0(X_{t_b \leq t})(b) = \pi_0(X_{t_c \leq t})(c) = a$. Clearly t_b and t_c must be critical values otherwise we can consider $p' > p$ and $q' > q$ with $q', p' \leq Q$, which is absurd. But the same holds if $t_b \neq t_c$: suppose $t_b < t_c \leq h(Q)$ then $p' = (\pi_0(X_{t_b < t_b + \varepsilon})(b), t_b + \varepsilon)$ (with $\varepsilon > 0$ small enough) satisfies $p < p'$ and $p' \leq Q$, which is absurd. Thus $t_b = t_c = t_i$ critical value.

The map $d : D_{\pi_0(X_\bullet)} \times D_{\pi_0(X_\bullet)} \to \mathbb{R}_{\geq 0}$ is symmetric. For what have said before $d(p,q) = 0$ if and only if $p, q \in \inf \mathrm{CA}(\{p,q\})$ and $h(p) = h(q) = t_i$ critical value. This is equivalent to $p = (b, t_i), q = (c, t_i) \in D_{\pi_0(X_\bullet)}$ such that $\pi_0(X_{t_i < t_i + \varepsilon})(b) = \pi_0(X_{t_i < t_i + \varepsilon})(c)$ for every $\varepsilon > 0$.

Thus, if $\pi_0(X_\bullet)$ is regular we have $d((b,t_i),(c,t_i)) = 0$ if and only if $p = q$; in fact $X_{t_i < t_i + \varepsilon}$ is an isomorphism for $\varepsilon > 0$ small enough.

Now we check the triangle inequality. Let $p_1, p_2, p_3 \in D_{\pi_0(X_\bullet)}$. And let $t_i = h(p_i)$, $Q_{ij} = (p_i, p_j)$, $q_{ij} = \inf\{h(p) \text{ with } p \in \mathrm{CA}(Q_{ij})\}$ and $q = \inf\{h(p) \text{ with } p \in \mathrm{CA}(\{p_1, p_2, p_3\})\}$.

Consider $P_1 = \mathrm{CA}(\{p_1\})$. Clearly $\inf \mathrm{CA}(\{p_1, p_2\}) \subset P_1$ and $\inf \mathrm{CA}(\{p_1, p_3\}) \subset P_1$. Thus either (1) $q_{13} \leq q_{12}$ (and $q_{23} = q_{12}$) or (2) $q_{12} < q_{13}$ (and $q_{13} = q_{23}$) hold.

In case (1) holds:

$$q_{12} - t_1 = q_{12} - q_{13} + q_{13} - t_1 \leq q_{12} - q_{13} + q_{13} - t_1 + 2q_{13} - 2t_3 = q_{13} - t_1 + q_{13} - t_3 + q_{23} - t_3$$

Thus:

$$q_{12} - t_1 + q_{12} - t_2 \leq q_{13} - t_1 + q_{13} - t_3 + q_{23} - t_3 + q_{23} - t_2$$

The proof in case (2) holds is analogous. □

See Figure 2.6 for an example of a display poset with its pseudo metric structure.

Remark 2.27. *Proposition 2.26 states that if $\pi_0(X_\bullet)$ is a regular abstract merge tree, then via $E_T \hookrightarrow D_{\pi_0(X_\bullet)}$, we can induce a metric on (T, h_T). It is not hard to see that this is the shortest path metric on E_T, with the length of an edge $e = (v, v')$ being given by $h_T(v') - h_T(v)$.*

Remark 2.28. *Given $\pi_0(X_\bullet)$ abstract merge tree, we have that the quotient of $D_{\pi_0(X_\bullet)}$ under the relationship $x \sim y$ iff $d(x,y) = 0$, is isometric as a metric space to $D_{R(\pi_0(X_\bullet))}$.*

2.5.3 Functions Spaces on the Display Poset

Thanks to Proposition 2.26 any display poset of an abstract merge tree inherits the topology generated by the open balls of the (pseudo) metric.

Consider now an abstract merge tree $\pi_0(X_\bullet)$ with critical set $\{t_1, \ldots, t_n\}$ and let $t \neq t_i$ for all $i = 1, \ldots, n$. Consider $p = (a,t) \in D_{\pi_0(X_\bullet)}$. We call $t_p = \max\{h(q) \in$

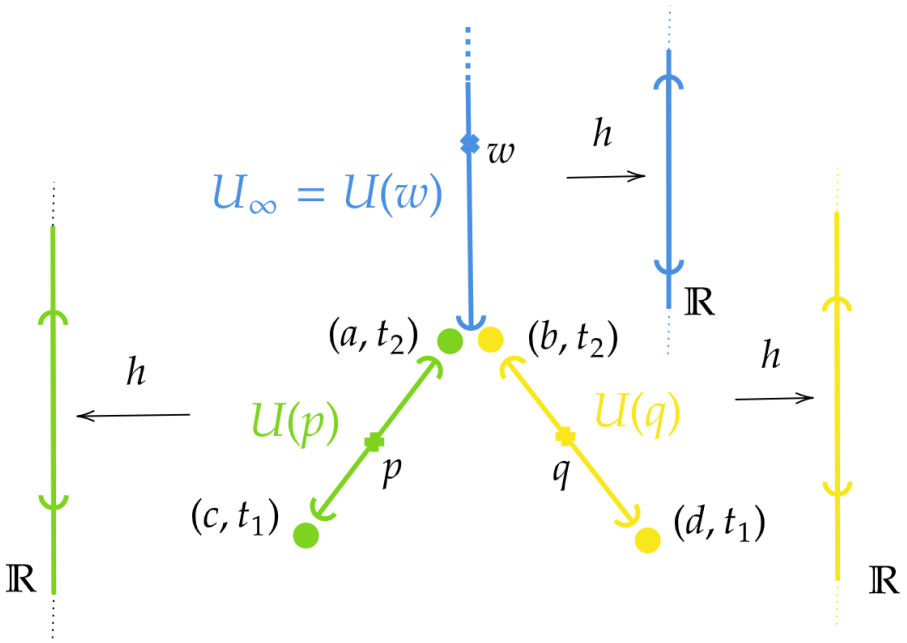


Figure 2.6: A graphical representation of the display poset, with its a.e. covering - see Section 2.5.2 - highlighted by the brackets at the extremes of the edges. Each such covering is the mapped homeomorphically to $\mathbb{R}$ via the height function h. Note that $d((a,t_2),(b,t_2)) = 0$ and $\{(a,t_2),(b,t_2)\} = CA((c,t_1),(d,t_1))$. The color scheme is coherent with the one in Figure 2.1.

$\{t_1,\dots,t_n\}$ with $q < p\}$ and $t^p = \min\{h(q) \in \{t_1,\dots,t_n\}$ with $q > p\}$. An open ball of radius $\varepsilon > 0$ is by definition:

$$B_\varepsilon(p) := \{q \in D_{\pi_0(X_\cdot)} \,|\, d(p,q) < \varepsilon\}.$$

Consider now $\varepsilon > 0$, with $t_p \le t - \varepsilon < t + \varepsilon \le t^p$. Let $p = (a,t)$ be a point such that for every $\eta > 0$ small enough $\#X^{-1}_{t-\eta<t}(p) = 1$ and $\#X^{-1}_{t<t+\eta}(X_{t<t+\eta}(p)) = 1$. The ball of radius ε around p is:

$$B_\varepsilon(p) := \{q \in \mathrm{CA}(\{p\}) \,|\, h(q) < t + \varepsilon\} \bigcup \{q \,|\, p \in \mathrm{CA}(\{q\}) \text{ and } h(q) > t - \varepsilon\}.$$

Thus, for any point such $p = (a,t)$ we can define the set:

$$U(p) := \{q \in \mathrm{CA}(\{p\}) \,|\, h(q) < t^p\} \bigcup \{q \,|\, p \in \mathrm{CA}(\{q\}) \text{ and } h(q) > t_p\}$$

which is an open neighbor of p. If $t > t_n$, then $t^p = \infty$ and so we have:

$$U_\infty := U(p) = \{q \in \mathrm{CA}(\{(\star, t_n)\}) \,|\, h(q) > t_n\}.$$

Refer to Figure 2.6 to have a visual intuition for the following proposition.

Proposition 2.29. *The map $h : D_{\pi_0(X_\cdot)} \to \mathbb{R}$ is monotone, continuous and $h_{|U(p)} : U(p) \to (t_p, t^p)$ is an homeomorphism and an isometry.*

Proof. Using the same notation of Proposition 2.26, we have:

$$|\,h((a,t)) - h((b,t))\,| \;=\; |\,t - t'\,| \;\le \tilde{t} - t + \tilde{t} - t' = d((a,t),(b,t')).$$

Thus h is continuous. Monotonicity is trivial. Suppose now we have $p = (a, t)$ and $(b, t'), (c, t'') \in U(p)$ such that $t' = t''$ and $b \neq c$. This is absurd since it implies that either $\#X^{-1}_{t-\varepsilon<t}(p) > 1$ or $X^{-1}_{t<t+\varepsilon}(X_{t<t+\varepsilon}(p)) > 1$ depending on whether $t > t'$ or $t < t'$, respectively. Moreover $h_{|U(p)}$ is clearly surjective for $h(X_{t<t'}(p)) = t'$. Thus $h_{|U(p)}$ is a bijective map. If $(b, t'), (c, t'') \in U(p)$, $\tilde{t} = \inf\{h(q) \,|\, q \in \mathrm{CA}(\{(b, t'), (c, t'')\})\} = \min\{t', t''\}$, which implies that $h_{|U(p)}$ is an isometry. And thus an homeomorphism. □

Definition 2.30. *The set* $\mathcal{U}(D_{\pi_0(X_\bullet)}) := \{U \subset D_{\pi_0(X_\bullet)} \,|\, U = U(p)$ *for some* $p \in D_{\pi_0(X_\bullet)}\}$ *is called the a.e. canonical covering of* $D_{\pi_0(X_\bullet)}$.

Remark 2.31. *Recall that the sets* $U(p)$ *are defined only for points* $p = (a, t)$ *for which there is* $K > 0$ *such that for every* $0 < \varepsilon < K$, *we have* $\#X^{-1}_{t-\varepsilon<t}(p) = 1$ *and* $\#X^{-1}_{t<t+\varepsilon}(X_{t<t+\varepsilon}(p)) = 1$.

Note that $\mathcal{U}(D_{\pi_0(X_\bullet)})$ is finite by the finiteness of $\pi_0(X_\bullet)$. Moreover, if $U(p), U(q) \in \mathcal{U}(D_{\pi_0(X_\bullet)})$ then either $U(p) = U(q)$ or $U(p) \bigcap U(q) = \emptyset$.

In fact, for every $t, t' \in h(U)$, $U \in \mathcal{U}(D_{\pi_0(X_\bullet)})$, the map $\pi_0(X_{t\leq t'})$ is injective on $U \cap X_t$ and $U \cap X_{t'}$. But having $(c, t') \in U(p) \bigcap U(q)$ implies $\pi_0(X_{t\leq t'})(a) = \pi_0(X_{t\leq t'})(b) = (c, t')$ for some $(a, t) \in U(p)$ and $(b, t) \in U(q)$. But then $t' \geq t_p, t_q$, which is absurd.

With the help of $\mathcal{U}(D_{\pi_0(X_\bullet)})$ we want to induce a measure on the sigma algebra generated by the open sets of $D_{\pi_0(X_\bullet)}$. For a display poset $D_{\pi_0(X_\bullet)}$ we define the measure $\mu_{\pi_0(X_\bullet)}$ as:

$$\mu_{\pi_0(X_\bullet)}(Q) = \sum_{U \in \mathcal{U}(D_{\pi_0(X_\bullet)})} \mathcal{L}(h(U \bigcap Q))$$

A graphical representation of such measure can be found in Figure 2.7a. Note that, if we call $D^\circ_{\pi_0(X_\bullet)} = \bigcup_{U \in \mathcal{U}(D_{\pi_0(X_\bullet)})} U$, we have $\mu_{\pi_0(X_\bullet)}(D_{\pi_0(X_\bullet)} - D^\circ_{\pi_0(X_\bullet)}) = 0$.

Proposition 2.32. $\mu_{\pi_0(X_\bullet)}(Q) = \sum_{U \in \mathcal{U}(D_{\pi_0(X_\bullet)})} \mathcal{L}(h(U \bigcap Q))$ *induces a measure on the sigma algebra generated by the open sets of* $D_{\pi_0(X_\bullet)}$.

Proof. We prove that $\mu_{\pi_0(X_\bullet)}$ is σ-additive. Let X_i, $i \in \mathbb{N}$, be disjoint sets in the Borel sigma algebra of $D_{\pi_0(X_\bullet)}$; we need to prove that $\mu_{\pi_0(X_\bullet)}(\bigcup_{i\in\mathbb{N}} X_i) = \sum_{i\in\mathbb{N}} \mu_{\pi_0(X_\bullet)}(X_i)$.

We have:

$$(\bigcup_{i\in\mathbb{N}} X_i) \bigcap U = \{p \in D_{\pi_0(X_\bullet)} \,|\, p \in X_i \text{ for some } i \text{ and } p \in U\} = \bigcup_{i\in\mathbb{N}} (X_i \bigcap U)$$

and so we are finished since $\mathcal{L}$ is σ-additive on $h(U \bigcap X_i)$. Note that, if Q is in the Borel sigma algebra of $D_{\pi_0(X_\bullet)}$, being h an homeomorphism on U (due to Proposition 2.29), $h(U \bigcap Q)$ is always Lebesgue measurable in $\mathbb{R}$.

□

In a similar fashion, consider a function $f : D_{\pi_0(X_\bullet)} \to \mathbb{R}$: by construction we have that f is $\mu_{\pi_0(X_\bullet)}$-*measurable* if $f \circ (h_{|U})^{-1}$ is $\mathcal{L}$-measurable on $\mathbb{R}$ for every $U \in \mathcal{U}(D_{\pi_0(X_\bullet)})$. So, given a $\mu_{\pi_0(X_\bullet)}$-measurable function $f : D_{\pi_0(X_\bullet)} \to \mathbb{R}$ we can define:

$$\int_{D_{\pi_0(X_\bullet)}} f d\mu_{\pi_0(X_\bullet)} = \sum_{U(p) \in \mathcal{U}(D_{\pi_0(X_\bullet)})} \int_{t_p}^{t^p} f \circ (h_{|U(p)})^{-1} d\mathcal{L}.$$

Leveraging on this definition, we want to define a framework to work with functions defined in some metric space (E, d_e). For reasons which will be clarified in the next section, we want that inside the metric space E there is a reference element 0 such that the amount of information contained in the value $f(p)$ can in some sense be quantified as the distance $d_e(f(p), 0)$. So we make the following assumption.

Assumption 2.33. *We always assume that (E, d_e) is a metric space and that $(E, *, 0)$ is a monoid, i.e. that $*$ is an associative operation with neutral element 0.*

We establish the following notation for any measure space (M, μ):

$$L_p(M, E) := \left\{ f : M \to E \,|\, d(f(\cdot), 0) : M \to \mathbb{R} \text{ measurable and } \int_M d_e(f(\cdot), 0)^p d\mu < \infty \right\} / \sim$$

with $\sim$ being the usual equivalence relationship between functions identifying functions up to μ-zero measure sets. This space becomes a monoid and a metric space with $(f+g)(p) := f(p) * g(p)$ and:

$$d_{L_p}(f, g) = \int_M d_e(f(\cdot), g(\cdot))^p d\mu.$$

To verify that d_{L_p} is a metric is enough to see that $d_{L_p}(f, g) = 0$ if and only if f and g differ on μ-zero measure sets and prove the triangle inequality using that $L_p(M, \mathbb{R})$ is a normed space.

For the sake of brevity, in the following we do not write explicitly the request that $d(f(\cdot), 0)$ is measurable and we imply it in the existence of its integral. Thus, we are interested in the spaces:

$$L_p(\pi_0(X_\bullet), E) := \left\{ f : D_{\pi_0(X_\bullet)} \to E \,|\, \int_{D_{\pi_0(X_\bullet)}} d_e(f(\cdot), 0)^p d\mu_{\pi_0(X_\bullet)} < \infty \right\} / \sim$$

Consider now $\pi_0(X_\bullet)$ and $\pi_0(Y_\bullet)$ such that $\pi_0(X_\bullet) \cong_{a.e.} \pi_0(Y_\bullet)$. Let $Z \subset \mathbb{R}$ such that $\alpha : \pi_0(X_{\bullet\,|Z}) \to \pi_0(Y_{\bullet\,|Z})$ is a natural isomorphism and $\mathcal{L}(\mathbb{R} - Z) = 0$. Then α induces a bijection between the display posets:

$$D_{\pi_0(X_{\bullet\,|Z})} := \bigcup_{t \in Z} \pi_0(X_t) \times \{t\}$$

and

$$D_{\pi_0(Y_{\bullet\,|Z})} := \bigcup_{t \in Z} \pi_0(Y_t) \times \{t\}.$$

With an abuse of notation we call such bijection $\alpha : D_{\pi_0(X_{\bullet\,|Z})} \to D_{\pi_0(Y_{\bullet\,|Z})}$.

Given $f : D_{\pi_0(Y_\bullet)} \to E$ we can clearly restrict it to $D_{\pi_0(Y_{\bullet\,|Z})}$ and thus we can pull it back on $D_{\pi_0(X_{\bullet\,|Z})}$ with α:

$$D_{\pi_0(X_{\bullet\,|Z})} \xrightarrow{\alpha} D_{\pi_0(Y_{\bullet\,|Z})} \hookrightarrow D_{\pi_0(Y_\bullet)} \xrightarrow{f} E$$

We call such function $\alpha^* f$.

Proposition 2.34. *The rule $f \mapsto \alpha^* f$ induces map $\alpha^* : L_p(D_{\pi_0(X_\bullet)}, E) \to L_p(D_{\pi_0(Y_\bullet)}, E)$ which is an isometry and a map of monoids.*

Proof. Since $\mathcal{L}(\mathbb{R} - Z) = 0$ then both $f \in L_p(D_{\pi_0(Y_{\bullet\,|Z})}, E)$ and $\alpha^* f \in L_p(D_{\pi_0(X_{\bullet\,|Z})}, E)$ identify a unique equivalence class, respectively, in $L_p(D_{\pi_0(Y_\bullet)}, E)$ and $L_p(D_{\pi_0(X_\bullet)}, E)$. Moreover, it is easy to see that the map α^* is such that $\alpha^*(f + g) = \alpha^* f + \alpha^* g$ and $d_{L_p}(f, g) = d_{L_p}(\alpha^* f, \alpha^* g)$. Lastly, because α is a natural isomorphism, then $(\alpha^{-1})^*$ yields the opposite correspondence. □

Proposition 2.34 implies that, for our purposes, we can always restrict ourselves to considering regular abstract merge trees. Thus we make the following assumption.

Assumption 2.35. *From now on we will always suppose that any abstract merge tree we consider is regular.*

2.5.4 Local Representation of Functions

When comparing two functions f, g defined on different display posets, we face the problem of combining together two kinds of variability: using language borrowed from functional data analysis (see the Special Section on Time Warpings and Phase Variation on the Electronic Journal of Statistics, Vol 8 (2), and references therein) and shape analysis (Kendall, 1977, 1984; Dryden and Mardia, 1998) we have an "horizontal" variability, due to the different domains (i.e. display posets), and a "vertical" variability which depends on the actual values that the functions assume. It is reasonable that both kinds of variability contribute to the final distance value: we have a cost given by the aligning the two display posets - horizontal variability - and a cost arising from the different amplitudes of the functions - vertical variability. In particular, we would like the horizontal variability to be measured in a way which is suitable for abstract merge trees (for instance, it should posses some kind of stability properties) and, similarly, the way in which the amplitude variability is measured should assume a somehow natural form, related to the spaces $L_p(D_{\pi_0(X_\bullet)}, E)$.

In other words, given $f : D_{\pi_0(X_\bullet)} \to E$ and $g : D_{\pi_0(Y_\bullet)} \to E$ we want to align, deform the display posets by locally comparing the information given by f and g and matching the display posets in the more convenient way. The word *locally* is on purpose vague at this stage of the discussion and should be thought as in some neighborhood of points of the posets. To compare local information carried by functions, we need to embed such objects in a common space so that differences can be measured.

First we formalize the procedure of obtaining local information from a function $f : D_{\pi_0(X_\bullet)} \to E$ - Figure 2.7b can help in the visualization of such idea. Given $D_{\pi_0(X_\bullet)}$ display poset and its a.e. canonical covering, we have an isomorphism of metric spaces and monoids:

$$L_p(\pi_0(X_\bullet), E) \cong \bigoplus_{U \in \mathcal{U}(D_{\pi_0(X_\bullet)})}^{p} L_p(h(U), E)$$

where $\bigoplus^p$ means that the norm of the direct sum is the p-th root of the sum of the p-th powers of the elements in the direct sum.

In this way we split up a function f on open disjoint subsets, without losing any information. However, as in Figure 2.7, to compare different functions one may need to represent this information on a finer scale and thus $\mathcal{U}_{D_{\pi_0(X_\bullet)}}$ may not be the correct way to split up f, which may need to be partitioned in smaller pieces. Thus we allow $\mathcal{U}_{D_{\pi_0(X_\bullet)}}$ to be refined with particular collections of open sets.

Definition 2.36. *A collection of open sets of $D_{\pi_0(X_\bullet)}$ is an a.e. covering of $D_{\pi_0(X_\bullet)}$ if it covers $D_{\pi_0(X_\bullet)}$ up to $\mu_{\pi_0(X_\bullet)}$-zero measure set. An a.e. covering of $D_{\pi_0(X_\bullet)}$ is regular if it is made by disjoint, connected open sets, each contained in some $U \in \mathcal{U}(D_{\pi_0(X_\bullet)})$.*

Given $\mathcal{O}'$ regular a.e. covering of $D_{\pi_0(X_\bullet)}$, a refinement of $\mathcal{O}'$ is a regular a.e. covering $\mathcal{O}$ such that for every $U \in \mathcal{O}$ there is $U' \in \mathcal{O}'$ such that $U \subset U'$.

Given the display poset $D_{\pi_0(X_\bullet)}$ of an abstract merge tree $\pi_0(X_\bullet)$ we collect all the refinements of its a.e. canonical covering in the set $\mathrm{Cov}(\pi_0(X_\bullet))$.

Proposition 2.37. *The set $\mathrm{Cov}(\pi_0(X_\bullet))$ is a lattice. It is a poset with the relationship $\mathcal{O} < \mathcal{O}'$ if $\mathcal{O}$ is a refinement of $\mathcal{O}'$ and for every couple of elements $\mathcal{O}, \mathcal{O}'$ there is a unique least upper bound $\mathcal{O} \vee \mathcal{O}'$ and a unique greater lower bound $\mathcal{O} \wedge \mathcal{O}'$. The operations are defined as follows:*

$$\mathcal{O} \vee \mathcal{O}' := \pi_0 \left(\bigcup_{U \in \mathcal{O}' \text{ or } U \in \mathcal{O}} U \right)$$

$$\mathcal{O} \wedge \mathcal{O}' := \{U \cap U' \,|\, U' \in \mathcal{O}' \text{ and } U \in \mathcal{O}\}.$$

Proof. Let's start with $\mathcal{O} \vee \mathcal{O}'$. It is clearly an a.e. covering. Moreover $\bigcup_{U \in \mathcal{O}' \text{ or } U \in \mathcal{O}} U$ is clearly contained in $\bigcup_{U \in \mathcal{U}(D_{\pi_0(X_\bullet)})} U$, and by functoriality we have that the set $\pi_0\left(\bigcup_{U \in \mathcal{O}' \text{ or } U \in \mathcal{O}} U\right)$ is included in $\pi_0\left(\bigcup_{U \in \mathcal{U}(D_{\pi_0(X_\bullet)})} U\right)$. And the latter is equal to $\mathcal{U}(D_{\pi_0(X_\bullet)})$. Thus $\mathcal{O} \vee \mathcal{O}'$ is regular and clearly is refined by $\mathcal{O}$ and $\mathcal{O}'$. Lastly, consider any $\mathcal{O}, \mathcal{O}' < \mathcal{O}''$. Since the sets of $\mathcal{O}''$ are disjoint and path connected (by construction), then any $U'' \in \mathcal{O}''$ contains all the sets of $\mathcal{O}$ and $\mathcal{O}'$ it intersects. Thus it contains a path connected component of their union.

Now we turn to $\mathcal{O} \wedge \mathcal{O}'$. All the sets in $\mathcal{O} \wedge \mathcal{O}'$ are disjiont, open and path connected. And they form an a.e. cover of $D_{\pi_0(X_\bullet)}$ - otherwise a positive-measure set would be left out by $\mathcal{O}$ or $\mathcal{O}'$. Thus $\mathcal{O} \wedge \mathcal{O}'$ is a regular a.e. covering which refines $\mathcal{O}$ and $\mathcal{O}'$. Consider $\mathcal{O}''$ such that $\mathcal{O}, \mathcal{O}' > \mathcal{O}''$. Take $U'' \in \mathcal{O}''$. By construction there are $U \in \mathcal{O}$ and $U' \in \mathcal{O}'$ with $U'' \subset U', U$. Thus $U'' \subset U \cap U'$. So $\mathcal{O}'' < \mathcal{O} \wedge \mathcal{O}'$. □

Given $\mathcal{O} \in \mathrm{Cov}(\pi_0(X_\bullet))$ we have:

$$L_p(\pi_0(X_\bullet), E) \cong \bigoplus_{U \in \mathcal{O}}^{p} L_p(h(U), E)$$

As already mentioned, to compare functions defined on different abstract merge trees we want to embed all these representations of functions into one common metric space, shared by all abstract merge trees. What we do is to consider $L_p((a,b), E)$, for some $(a,b) \subset \mathbb{R}$ and embed it into $L_p(\mathbb{R}, E)$ by extending $f : (a,b) \to E$ to $\mathbb{R}$ with $0 \in E$ outside (a,b). In this way we have an isometric embedding $L_p((a,b), E) \hookrightarrow L_p(\mathbb{R}, E)$.

In the next definition we need the notion of the essential support of a function $f : (M, \mu) \to E$ defined on a measure topological space (M, μ) and with values in $(E, *, 0)$:

$$\mathrm{supp}(f) = M - \bigcup \{U \subset M \text{ open } \mid f_{|U} = 0 \ \mu - \text{a.e.}\}$$

Definition 2.38. *Given $D_{\pi_0(X_\bullet)}$ and $\mathcal{O} \in \mathrm{Cov}(\pi_0(X_\bullet))$, a local representation of a function in $L_p(D_{\pi_0(X_\bullet)}, E)$ on $\mathcal{O}$ is a function $\varphi_{\mathcal{O}} : \mathcal{O} \to L_p(\mathbb{R}, E)$ such that $\mathrm{supp}(\varphi(U)) \subset h(U)$ for every $U \in \mathcal{O}$.*

Note that if, instead of splitting f on a finer scale, we want to look at the function on a coarser level, we can do that. Consider $\mathcal{O}'$ refinement of $\mathcal{O}$; then for every $V \in \mathcal{O}$:

$$\varphi_{\mathcal{O}}(V) = \sum_{U \in \mathcal{O}' \text{ such that } U \subset V} \varphi_{\mathcal{O}'}(U)$$

2.5.5 Functions Defined on Merge Trees

Thanks to Proposition 2.34 we have seen that to work with functions defined on display posets we can reduce to the case of regular abstract merge trees. This makes the upcoming discussion much easier since, thanks to Proposition 2.19, we can associate a merge tree to any regular abstract merge tree.

We have already seen that the metric d defined on the display poset $D_{\pi_0(X_\bullet)}$ induces the shortest path metric on the graph $(T, h_T) = \mathcal{M}(\pi_0(X_\bullet))$ via the inclusion $E_T \hookrightarrow D_{\pi_0(X_\bullet)}$ - see Proposition 2.19. Similarly, we can establish a correspondence between the edges E_T and the a.e. canonical covering $\mathcal{U}(D_{\pi_0(X_\bullet)})$: informally speaking, each edge $(v, v') \in E_T$ corresponds to the open set $U = \{p \in D_{\pi_0(X_\bullet)} \mid v < p < v'\}$ or $U_\infty = \{p \in D_{\pi_0(X_\bullet)} \mid v < p\}$ if $v' = r_T$ - as in Figure 2.2c.

As a consequence, a local representation of a function on $\mathcal{U}(D_{\pi_0(X_\bullet)})$, i.e. $\varphi_{\mathcal{U}(D_{\pi_0(X_\bullet)})} : \mathcal{U}(D_{\pi_0(X_\bullet)}) \to L_p(\mathbb{R}, E)$ induces a unique function $\varphi_T : E_T \to L_p(\mathbb{R}, E)$, and viceversa.

Before proceeding we point out a fact which will be discussed with more details in Section 2.7 and which brings together the idea of local representation of a function and the need to work up to order 2 vertices. Consider $\mathcal{M}(\pi_0(X_\bullet)) = (T, h_T)$ and $(T', h_{T'})$ such that T' and T are equivalent up to order two vertices. We know that $\pi_0(X_\bullet) = \mathcal{F}((T', h_{T'})) \cong \mathcal{F}((T, h_T))$ and that $V_T \hookrightarrow D_{\pi_0(X_\bullet)}$ induces the a.e. canonical covering $\mathcal{U}(D_{\pi_0(X_\bullet)})$. The inclusion $V_{T'} \hookrightarrow D_{\pi_0(X_\bullet)}$ induces another regular a.e. covering, always with the rule $(v, v') \in E_T \mapsto U = \{p \in D_{\pi_0(X_\bullet)} \,|\, v < p < v'\}$. This regular a.e. covering is a refinement of $\mathcal{U}(D_{\pi_0(X_\bullet)})$. Thus via splittings and ghostings we are able to change the local representation of a function.

At this point we face the problem of defining a suitable (pseudo) metric framework for objects of the form (T, h_T, φ_T), with $\varphi_T : E_T \to L_p(\mathbb{R}, E)$ such that $\mathrm{supp}(\varphi_T(e)) \subset [h_T(a), h_T(b)]$ for $e = (a, b)$. Before dealing with such problem we take a brief detour to make some examples of local representation of functions on merge trees.

2.5.6 Examples

In what follows, we present some examples of functions defined on display posets, to show how they can be used to capture useful information about a filtration $X_\bullet$.

The general structure of the following examples is to consider a subcategory $\mathcal{B}$ of Top and pick a function $\Theta : \mathcal{B} \to E$. Then, $f : D_{\pi_0(X_\bullet)} \to E$ is obtained as $f(a, t) = \Theta(a)$. We call φ_T^Θ the local representation of such function, and we prove that in all our examples the information contained in the functions generalizes, in some sense, the notion of merge trees. More formally, if $(T, \varphi_T^\Theta) \cong (T', \varphi_{T'}^\Theta)$ then $(T, h_T) \cong (T', h_{T'})$. Under such hypotheses a metric to compare (T, φ_T^Θ) and $(T', \varphi_{T'}^\Theta)$ can be pulled back to compare objects of the form (T, h_T, φ_T).

We immediately stress that many of the upcoming functions do not lie in $L_p(D_{\pi_0(X_\bullet)}, E)$, for some $D_{\pi_0(X_\bullet)}$, as:

$$\lim_{x \to +\infty} d(f \circ (h_{|U_\infty})^{-1}(x), 0) > 0.$$

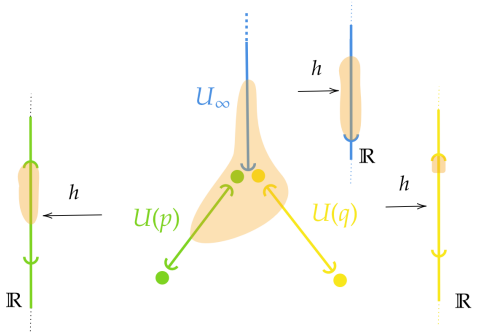
However, in Section 2.7 we discuss how these examples can be modified to fit into the proposed framework.

2.5.6.1 Merge Trees

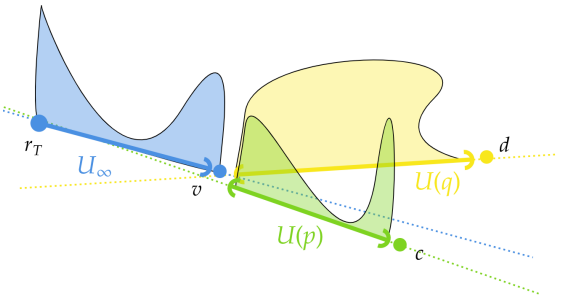
Consider the special case of the constant function $\Theta_1 : \mathrm{Top} \to \mathbb{R}_{\geq 0}$, such that $\Theta_1(s) = 1$ for all sets s. That is $f : D_{\pi_0(X_\bullet)} \to \mathbb{R}_{\geq 0}$ is defined by $f((a, t)) = 1$. As a consequence we have $f \circ (h_{|U(p)})^{-1} = \chi_{(t_p, t^p)}$, for some $p \in D_{\pi_0(X_\bullet)}$ and with χ_I being the characteristic function over the set $I \subset \mathbb{R}$. Consider now $\mathcal{M}(\pi_0(X_\bullet)) = (T, h_T)$; given $e = (v, v') \in E_T$, we have $\varphi_T(e) = \chi_{(t_p, t^p)} = \chi_{(t_i, t_j)}$ with $h_T(v) = t_i$ and $h_T(v') = t_j$. We call $\varphi_T^{\Theta_1}$ the local representation of the function f induced by Θ_1 on $D_{\pi_0(X_\bullet)}$.

Consider two functions $(T, \varphi_T^{\Theta_1})$ and $(T', \varphi_{T'}^{\Theta_1})$. Suppose there is $\eta : V_T \to V_{T'}$ isomorphism of tree structures such that $\varphi_T \circ \eta = \varphi_{T'} \circ \eta$. Then $(T, h_T) \cong (T', h_{T'})$. In fact via η, the support of $\varphi_T^{\Theta_1}(e)$ coincide with the support of $\varphi_{T'}^{\Theta_1}(\eta(e))$. But the support is given by the critical values of the filtration, that is, the value of the height function h_T on the extremes of the edge e.

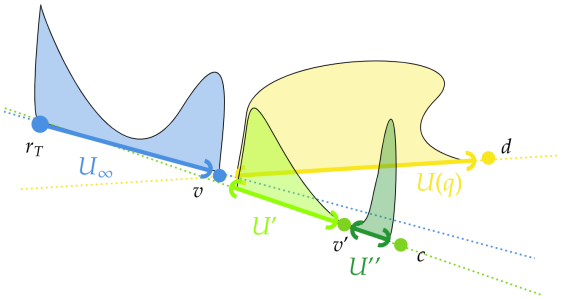
Thus, via the function $\Theta_1 \equiv 1$, we have found another way to represent the information contained in merge trees: $(T, h_T) \cong (T', h_{T'})$ if and only if $(T, \varphi_T^\Theta) \cong (T', \varphi_{T'}^\Theta)$.



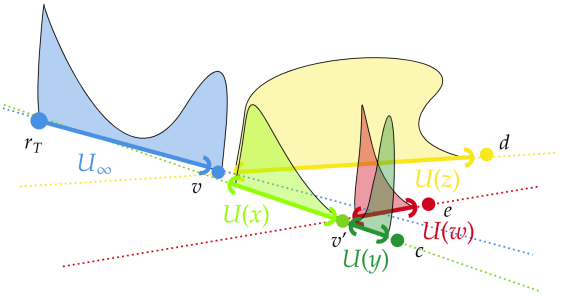
(a) A display poset $D_{\pi_0(X_\cdot)}$ with the measure $\mu_{D_{\pi_0(X_\cdot)}}$. The orange shaded set is first intersected with the open sets U_∞, $U(p)$ and $U(q)$ and then its Lebesgue measure is taken in $\mathbb{R}$ via the height function h.



(b) A function $f : D_{\pi_0(X_\cdot)} \to \mathbb{R}$ defined on the display poset $D_{\pi_0(X_\cdot)}$. With different colors we have highlighted the restrictions of the function on the different open sets of the canonical a.e. covering.



(c) A function $f : D_{\pi_0(X_\cdot)} \to \mathbb{R}$ defined on the display poset $D_{\pi_0(X_\cdot)}$ represented with the restrictions on a regular a.e. covering which refines the canonical one.



(d) A function $g : D_{\pi_0(Y_\cdot)} \to \mathbb{R}$ defined on the display poset $D_{\pi_0(Y_\cdot)}$ along with its restrictions on the canonical a.e. covering of $D_{\pi_0(Y_\cdot)}$. The refinement of the canonical a.e. covering of $D_{\pi_0(X_\cdot)}$ which appears in Figure 2.7c is much more suited than the canonical a.e. covering in Figure 2.7b to compare the two functions: on U_∞ and U_∞ the functions are very similar, as the are on U' and $U(x)$, on U'' and $U(y)$ and on $U(q)$ and $U(z)$.

Figure 2.7: Measures and real valued functions defined on display posets. In every plot but the upper left one, for visualization purposes the posets are represented as embedded on the horizontal plane in $\mathbb{R}^3$ and plotted with thick lines. The vertical axis represents the value of the functions. With different colors we have highlighted the restrictions of the functions on different open sets. The colored dotted lines are are a qualitative visual representation of the embedding $(f : (a, b) \to \mathbb{R}) \mapsto (f' : \mathbb{R} \to \mathbb{R})$ where f' extends f with 0 outside (a, b).

2.5.6.2 Cardinality of Clusters

Consider now the case of a merge tree $\mathcal{M}(\pi_0(X_\bullet)) = (T, h_T)$, with $X_\bullet$ being the Céch filtration of the point cloud $\{x_1, \dots, x_n\}$. Sensible information that one may want to track down along $\pi_0(X_\bullet)$ is the cardinality of the clusters. Thus we can take $\Theta_c : FSets \to \mathbb{R}_{\geq 0}$, defined on all finite sets ($Fsets$) considered with the discrete topology, defined as $\Theta_c(\{x_{j,1}, \dots, x_{j,n_j}\}) = n_j$. As a consequence, we have $\varphi_T^{\Theta}(e) = m\chi_{[t_i,t_j)}$, for some positive cardinality m and some critical points t_i, t_j. Note that, clearly, $\text{supp}(\varphi_T^{\Theta_c}(e)) = [t_i, t_j]$. Thus if we have $(T, \varphi_T^{\Theta_c}) \cong (T', \varphi_{T'}^{\Theta_c})$ then $(T, h_T) \cong (T', h_{T'})$.

We now make a concrete example - Figure 2.8a and Figure 2.8b. Consider the finite set $\{v_{-1} = -1, v_0 = 0, v_2 = 2\}$ and build the abstract merge tree and the single linkage hierarchical clustering dendrogram. Abstract merge tree is given by $a_t = \{\{v_{-1}\}, \{v_0\}, \{v_2\}\}$ for $t \in [0, 1)$, $a_t = \{\{v_{-1}, v_0\}, \{v_2\}\}$ for $t \in [1, 2)$ and $a_t = \{\{v_{-1}, v_0, v_2\}\}$ for $t \geq t^+ = 2$. With maps given by $a \mapsto b$ with $a \subset b$.

The associated merge tree (T, h_T) - see Figure 2.8a - can be represented with the vertex set $V_T = \{\{v_{-1}\}, \{v_0\}, \{v_2\}, \{v_{-1}, v_0\}, \{v_{-1}, v_0, v_2\}, r_T\}$. The leaves are $\{v_{-1}\}, \{v_0\}$ and $\{v_2\}$; the children of $\{v_{-1}, v_0\}$ are $\{v_{-1}\}$ and $\{v_0\}$, and the ones of $\{v_{-1}, v_0, v_2\}$ are $\{v_{-1}, v_0\}$ and $\{v_2\}$. The height function h_T is given by $h_T(\{v_i\}) = 0$ for $i = -1, 0, 2$, $h_T(\{v_{-1}, v_0\}) = 1$, $h_T(\{v_{-1}, v_0, v_2\}) = 2$ and $h_T(r_T) = +\infty$.

Consider Θ_c. The local representation $\varphi_T^{\Theta_c}$ of the induced function is thus the following: $\varphi_T^{\Theta_c}(\{v_i\}) = \chi_{[0,1)}$ for $i = -1, 0$, $\varphi_T^{\Theta_c}(\{v_2\}) = \chi_{[0,2)}$, $\varphi_T^{\Theta_c}(\{v_{-1}, v_0\}) = 2\chi_{[1,2)}$ and $\varphi_T^{\Theta_c}(\{v_{-1}, v_0, v_2\}) = 3\chi_{[2,+\infty)\}}$. See Figure 2.8b.

2.5.6.3 Measure of Sublevel Sets

Now consider $U \subset \mathbb{R}^m$ convex bounded open set, with $\overline{U}$ being its topological closure, and let $\mathcal{L}$ be the Lebesgue measure in $\mathbb{R}^m$. Let $f : \overline{U} \to \mathbb{R}$ be a *tame* (Chazal et al., 2016) continuous function. Consider the sublevel set filtration $X_t = f^{-1}((-\infty, t])$ with $\pi_0(X_t) = \{U_1^t, \dots, U_n^t\}$. Here the tameness condition is simply asking that $\pi_0(X_\bullet)$ is a finite constructible persistent set, and thus a finite abstract merge tree. Call $\psi_t^{t'}$ the functions $\psi_t^{t'} = X_{t \leq t'}$. We set $\Theta_{\mathcal{L}} = \mathcal{L} : \mathcal{B}(\mathbb{R}^n) \to \mathbb{R}_{\geq 0}$ with $\mathcal{B}(\mathbb{R}^n)$ being the Borel σ-algebra of $\mathbb{R}^n$. So that we can always take: $\Theta_{\mathcal{L}}(U_i^t) = \mathcal{L}(U_i^t)$.

Proposition 2.39. *If we have* $(T, \varphi_T^{\Theta_{\mathcal{L}}}) \cong (T', \varphi_{T'}^{\Theta_{\mathcal{L}}})$ *then* $(T, h_T) \cong (T', h_{T'})$.

Proof. Let (T, h_T) being the merge tree representing $\pi_0(X_\bullet)$, and $\varphi_T^{\Theta_{\mathcal{L}}}$ the local representation of the associated function. Since f is continuous, for and edge $e = (v, v') \in E_T$ spanning from height $h_T(v) = t_i$ to $h_T(v') = t_j$, we can prove that $\text{supp}(\varphi_T^{\Theta_{\mathcal{L}}}(e)) = [t_i, t_j]$. We know that v is associated to a connected component $U_k^{t_i}$, for some k. If v represents the merging of two or more path connected components $U_{k_1}^{t_i-\varepsilon}$ and $U_{k_2}^{t_i-\varepsilon}$, for some small $\varepsilon > 0$, with $\mathcal{L}(U_{k_1}^{t_i-\varepsilon}), \mathcal{L}(U_{k_2}^{t_i-\varepsilon}) > 0$, then, since $U_{k_1}^{t_i-\varepsilon}, U_{k_2}^{t_i-\varepsilon} \subset U_k^{t_i}$, we have $\mathcal{L}(U_k^{t_i}) > 0$. Thus if we prove the statement for v leaf, we are done. So, suppose v is a leaf and consider $x_0 \in U_k^{t_i}$. We know $f(x_0) = t_i$. By the continuity of f, for every $\varepsilon > 0$ there is $\delta > 0$ such that if $|\,|x - x_0|\,| < \delta$, then $f(x_0) \leq f(x) < f(x_0) + \varepsilon$. Since $\{x \in \overline{U} \,|\, |\,|x - x_0|\,| < \delta\}$ is convex (and so path connected), then it is contained in $\psi_{t_i}^{t_i+\varepsilon}(U_k^{t_i})$. Moreover, since it contains the non-empty open set $\{x \in U \,|\, |\,|x - x_0|\,| < \delta\}$, we have $\mathcal{L}(\psi_{t_i}^{t_i+\varepsilon}(U_k^{t_i})) > 0$ for every $\varepsilon > 0$. As a consequence, $\text{supp}(\varphi_T^{\Theta_{\mathcal{L}}}(e)) = [t_i, t_j]$. □

Again we make a quick hands-on example. Consider the function $f = |\,|x| - 1\,|$ defined on the interval $[-2, 2]$. Let $\pi_0(X_t) = \pi_0(f^{-1}((-\infty, t]))$. Let (T, h_T) be the merge tree associated to the sequence $\pi_0(X_\bullet)$. Now we obtain the local representation $\varphi_T^{\Theta_{\mathcal{L}}}(e_i)$.

We have $\varphi_T^{\Theta_\mathcal{L}}(e_1) = |1 + t - 1 + t| = 2t$ for $t \in [0, 1)$, and 0 otherwise. Clearly $\varphi_T^{\Theta_\mathcal{L}}(e_1) = \varphi_T^{\Theta_\mathcal{L}}(e_2)$. Lastly $\varphi_T^{\Theta_\mathcal{L}}(r_T) = 4\chi_{[2,+\infty)}$.

2.5.6.4 Merge Trees with Homological Information

Lastly we propose a function Θ_p to combine homological information of different dimensions (Hatcher, 2000) obtaining dendrograms which are closely related to the barcode decorated merge trees defined by Curry et al. (2022). We consider the topological spaces with p-th homology of finite type, that is, their p-th homology group is finitely generated, and collect all the spaces with finitely generated $1, \dots, p$-th homology groups in the set FTop_p. Consider $\Theta_p : \mathrm{FTop}_p \to \mathbb{N} \times \ldots \times \mathbb{N}$ defined on a topological space U as $\Theta_p(U) = (\dim(H_0(U; \mathbb{K}), \dots, \dim(H_p(U; \mathbb{K}))$, with $H_p(U; \mathbb{K})$ being the p-th homology group of U with coefficients in the field $\mathbb{K}$. Note that, by definition, generators of homology groups of U lie inside a connected component. In this way we are able to track if in a path connected component there are some kind of holes arising or dying, and thus collecting a more complete set of topological invariants which capture the shape of each connected component, which could be useful in situations like the one depicted in Figure 2.8e. From another point of view, at every step along a filtration, we are decomposing homological information of a topological space by means of its connected components.

Note that we clearly have $(T, \varphi_T^{\Theta_p}) \cong (T', \varphi_{T'}^{\Theta_p})$ implying $(T, h_T) \cong (T', h_{T'})$. In fact $\mathrm{FTop}_p \xrightarrow{\Theta_p} \mathbb{N} \times \ldots \times \mathbb{N} \xrightarrow{\pi_1} \mathbb{N}$ is Θ_1 - with π_1 being the projection on the first component.

2.6 Dendrogram Edit Distance

The aim of the remaining part of the chapter is to define a (pseudo) metric structure that allows us to work with all the machinery we defined up to this point. Since the properties of the spaces $L_p(D_{\pi_0(X_\bullet)}, E)$ and $L_p(\mathbb{R}, E)$ are very much dependent on E, instead of making assumptions on E and then evaluating their consequences on $L_p(D_{\pi_0(X_\bullet)}, E)$, we take a step back and consider general functions $\varphi_T : E_T \to W$, trying to identify which properties we need to impose on W to build our metric. In Section 2.6.2, starting from such properties, we will recover information on E and $L_p(D_{\pi_0(X_\bullet)}, E)$.

2.6.1 Editable Spaces and Edits of Dendrograms

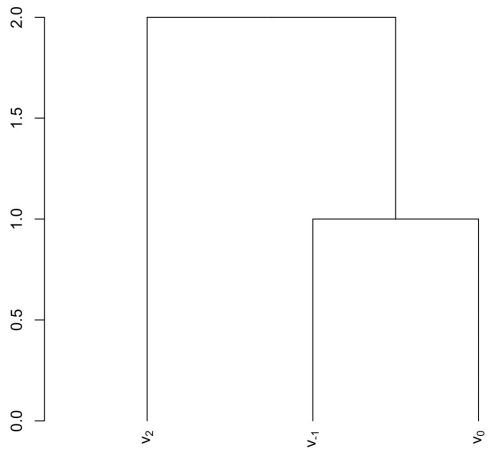
The approach we follow is to define a distance, for objects of the form $(T, \varphi_T : E_T \to W)$, which is inspired by the Tree Edit Distances (Tai, 1979), but with substantial differences in the edit operations. The philosophy of these distances is to allow certain modifications of the base object, called edits, each being associated to a cost, and to define the distance between two objects as the minimal cost that is needed to transform the first object into the second with a finite sequence of edits. In this way, up to properly setting up a set of edits, one can formalize the deformation of a tree comparing the local information induced by the function defined on the edges. On top of that, edit distances frequently enjoy some decomposition properties which simplify the calculations (Hong et al., 2017), which are notoriously very heavy (Hein et al., 1995).

To begin with, let us make some hypotheses on the codomain of the functions $\varphi_T : E_T \to W$.

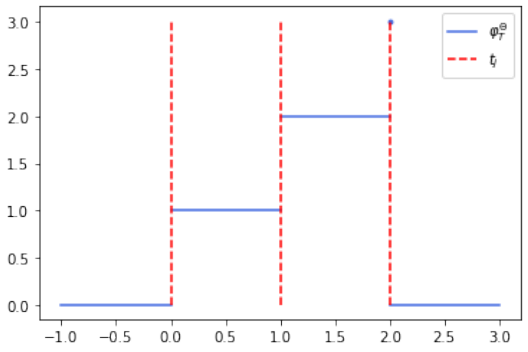
Definition 2.40. *A set W is called editable if the following conditions are satisfied:*

(P1) (W, d) *is a metric space*

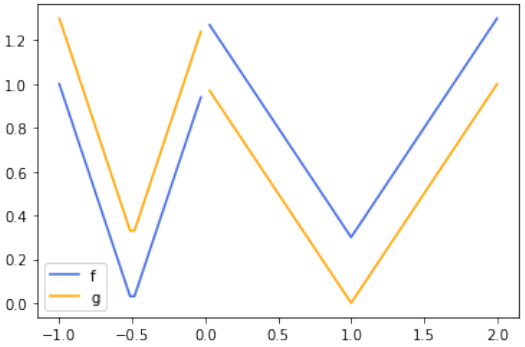
(P2) $(W, *, 0)$ *is a monoid (that is W has an associative operation $*$ with zero element 0)*



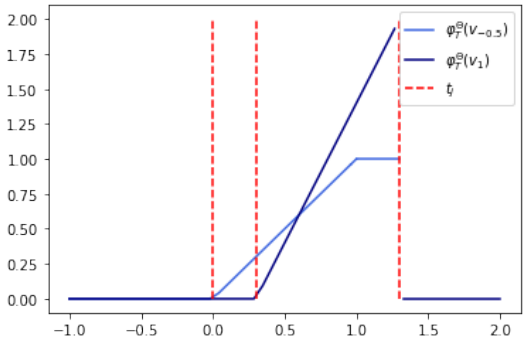
(a) Single linkage clustering dendrogram referring to the example in Section 2.5.6.2.



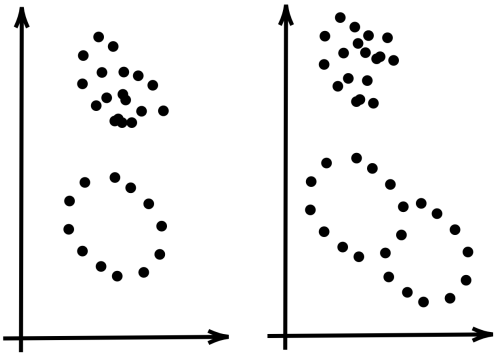
(b) In the context of the example in Section 2.5.6.2, we see the sum of the weight functions of the vertices going from v_0 to the root r_T: $\varphi_T^{\Theta_c}(\{v_0\}) + \varphi_T^{\Theta_c}(\{v_0, v_{-1}\})$. The dotted lines represent critical values.



(c) The functions f and g in Section 2.7.4; with $\varepsilon = 0.3$.



(d) In the context of the example in Section 2.7.4, we report $\varphi_T^{\Theta_\mathcal{L}}(v_1)$ and $\varphi_T^{\Theta_\mathcal{L}}(v_{-0.5})$. The dotted lines represent critical values.



(e) Two point clouds made by two clusters each which cannot be separated by zero dimensional homology, but present different within-cluster homological information and can be distinguished by Θ_1 defined in Section 2.5.6.4.

Figure 2.8: Plots referring to the examples in Section 2.5.6 and Section 2.7.4.

(P3) *the map* $d(\cdot, 0) : W \to \mathbb{R}$ *is a map of monoids between* $(W, *)$ *and* $(\mathbb{R}, +)$*:* $d(x*y, 0) = d(0, x) + d(0, y)$.

(P4) d *is* $*$ *invariant, that is:* $d(x, y) = d(z * x, z * y) = d(x * z, y * z)$

Note that in property (P3), $d(x * y, 0) = d(x, 0) + d(y, 0)$, implies that $x * y \neq 0$. Moreover (P3)-(P4) imply that the points 0, x, y and $x * y$ form a rectangle which can be isometrically embedded in an Euclidean plane with the Manhattan geometry (that is, with the norm $|\,|\cdot|\,|_1$): $d(x, x*y) = d(0, y)$, $d(y, x*y) = d(0, x)$ and $d(x*y, 0) = d(0, x)+d(0, y)$.

Remark 2.41. *The rationale behind properties (P1)-(P4) is mainly contained in the proof of Theorem 2.49 and is linked with the properties of mappings, introduced in Section 2.6.4. Some further insights on these properties is given in Section 2.7.*

With these additional pieces of structure there are situations which we want to avoid because they represent "degenerate" functions which introduce formal complications.

Definition 2.42. *Given an editable space* W *and a tree-structure* T*, a proper weight function is a weight function* φ_T *such that* $\varphi_T : E_T \to W$ *and* $0 \in \varphi(E_T)$ *if and only if* $E_T = \emptyset$ *and* $V_T = \{\star\}$*. The couple* (T, φ_T) *is an editable dendrogram.*

Assumption 2.43. *From now on we only work with editable spaces and we want to consider exclusively proper weight functions. To lighten the notation, however, we omit to write "proper" explicitly. Similarly we omit the word editable when referring to editable dendrograms.*

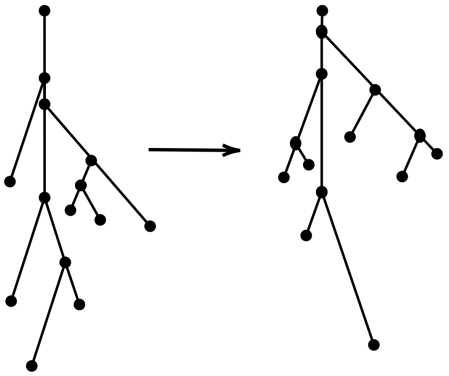
Definition 2.44. *Given an editable space* W *the editable dendrogram space* $(\mathcal{T}, W)$ *is given by the set of (editable) dendrograms* (T, φ_T) *with (proper) weight functions such that* $\varphi_T : E_T \to W$*.*

Given an editable dendrogram space $(\mathcal{D}, W)$, with $(W, *, 0)$ editable space, we can define our edits. All the edits are operations that can be done on the edges E_T, or, equivalently, on the vertices $V_T - \{r_T\}$, via the bijection $E_T \cong V_T - \{r_T\}$ given by $(v, v') \mapsto v$.

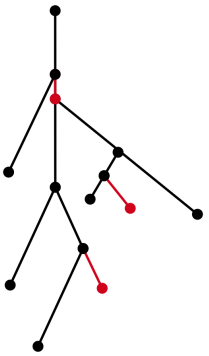
- We call *shrinking* of a vertex/edge a change of the weight function. The new weight function must be equal to the previous one on all vertices, apart from the "shrunk" one. In other words, for an edge e, this means changing the value $\varphi(e)$ with another non zero value in W.

- A *deletion* is an edit with which a vertex/edge is deleted from the dendrogram. Consider an edge (v_1, v_2). The result of deleting v_1 is a new tree structure, with the same vertices a part from v_1 (the smaller one), and with the father of the deleted vertex which gains all of its children. The inverse of the deletion is the *insertion* of an edge along with its lower vertex. We can insert an edge at a vertex v specifying the name of the new child of v, the children of the newly added vertex (that can be either none, or any portion of the children of v), and the value of the weight function on the new edge.

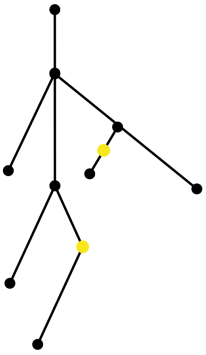
- Lastly, we generalize Definition 2.17, defining a transformation which eliminates an order two vertex, connecting the two adjacent edges which arrive and depart from it. Suppose we have two edges $e = (v_1, v_2)$ and $e' = (v_2, v_3)$, with $v_1 < v_2 < v_3$. And suppose v_2 is of order two. Then, we can remove v_2 and merge e and e' into a new edge $e'' = (v_1, v_3)$, with $\varphi(e'') := \varphi(e) * \varphi(e')$. This transformation is called the *ghosting* of the vertex. Its inverse transformation is called the *splitting* of an edge.



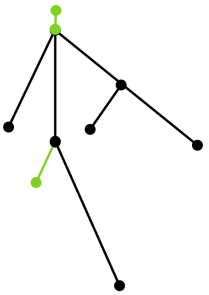
(a) Starting and target weighted trees.



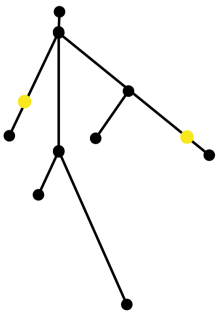
(b) Deletions in red.



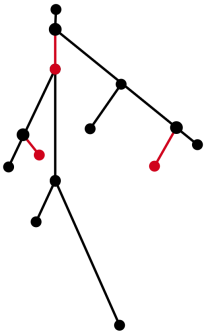
(c) Ghostings in yellow.



(d) Shrinkings in green.



(e) Splittings in yellow.



(f) Insertions in red.

Figure 2.9: (b)→(e) form an edit path made from the left weighted tree in Figure 2.9a to the right one. Each time the edges involved in the editing are highlighted with different colors. In the following plot such vertices return black. This edit path can clearly be represented with a mapping - Section 2.6.4 - made by couples $(v, "D")$ for all the red vertices in Figure 2.9b, $(v, "G")$ for all the yellow vertices in Figure 2.9c, (v, w) for all the vertices associated via the green color in Figure 2.9d, $("G", w)$ for all the yellow vertices in Figure 2.9c and $("D", w)$ for all the red vertices in Figure 2.9f.

A dendrogram T can be edited to obtain another dendrogram, on which one can apply a new edit to obtain a third dendrogram and so on. One can think of this as composing two edits e_0, e_1 which are not defined on the same dendrogram, since the second edit is defined on the already edited dendrogram. This is what we mean by composition of edits. Any finite composition of edits is referred to as an *edit path*. The notations we use are functional notations, even if the edits are not operators, since an edit is not defined on the whole space of dendrograms but on a single dendrogram. For example $e_1 \circ e_0(T)$ means that T is edited with e_0, and then $e_0(T)$ with e_1. See Figure 2.9 for an example of an edit path in the case of weighted trees - Section 2.6.2.3.

Lastly, note that, even if shrinking and deletions are classical edits, with shrinking being usually referred to as *relabeling*, the ghosting edit is completely unusual and we are not aware of any previous work employing it. However, such edit is fundamental for our purposes and for the stability results contained in Section 2.7.3 and Chapter 4.

Definition 2.45. *Dendrograms are equal up to order* 2 *vertices if they become isomorphic after applying a finite number of ghostings or splittings. We write* $(T, \varphi_T) \cong_2 (T', \varphi_{T'})$.

Definition 2.45 induces an equivalence relationship. The set of dendrograms inside $(\mathcal{T}, W)$ that we want to treat as equal are exactly the equivalence classes given by Definition 2.45. We call $(\mathcal{T}_2, W)$ the space of equivalence classes of dendrograms in $(\mathcal{T}, W)$, equal up to order 2 vertices.

2.6.2 Examples of Editable Spaces

Now we give some examples of editable spaces, including all the spaces which appear in Section 2.5.6.

2.6.2.1 Curves in Editable Spaces

Consider an editable space E. Then the space of functions $W := L_1(\mathbb{R}, E)$ is editable.

The function d is always non negative, so if properties (P3) and (P4) hold pointwise, then they hold also for integrals. For instance we verify (P3) as follows:

$$d_W(f *_W g, 0) = \int_{\mathbb{R}} d_e(f(t) *_e g(t), 0) dt = \int_{\mathbb{R}} d_e(f(t), 0) + d_e(g(t), 0) dt = d_W(f, 0) + d_W(g, 0).$$

This first example is pivotal, as it paves the way for treating the function spaces built in Section 2.5.

2.6.2.2 Finite Products of Spaces

Consider two editable spaces E and E', that is $(E, \odot, 0_E)$ and $(E', \diamond, 0_{E'})$ satisfying properties (P1)-(P4). Then $(E \times E', *, (0_E, 0_{E'}))$ is an editable space, with $*$ being the component-wise operations $\odot$ and $\diamond$, and the metric d on $E \times E'$ being the (possibly weighted) sum of the component-wise metrics of E and E'.

2.6.2.3 Positive Real Numbers

Clearly the set $(\mathbb{R}_{\geq 0}, +, |\cdot|)$ is an editable space, as well as its subsets which are monoids, like $\mathbb{N}$. We call $(\mathcal{T}, \mathbb{R}_{\geq 0})$ the space of weighted trees. Such space is studied and used in Chapter 3 to define a metric for merge trees, which are turned into weighted trees via the rule $w_T((v, v')) = h_T(v') - h_T(v)$, plus some non-trivial technical modifications.

Remark 2.46. *This fact, along with the previous examples, implies that all the functions* Θ *used in Section 2.5.6 have as codomain an editable space. However we still have the problem that the local representation of functions* φ_T^{Θ} *do not take values in* $L_1(\mathbb{R}, \mathbb{R}_{\geq 0})$, $L_1(\mathbb{R}, \mathbb{N})$, $L_1(\mathbb{R}, \mathbb{N}) \times L_1(\mathbb{R}, \mathbb{N})$ *because their integral is not finite on* U_∞.

2.6.3 Costs of Edit Operations

Now we associate to every edit a cost, that is, a length measure in the space $(\mathcal{T}, W)$. The costs of the edit operations are defined as follows:

- if, via shrinking, an edge goes from weight x to weight y, then the cost of such operation is $d(x, y)$;
- for any deletion/insertion of an edge with weight x, the cost is equal to $d(x, 0)$;
- the cost of ghosting operations is $\mid d(x * y, 0) - d(x, 0) - d(y, 0) \mid = 0$.

Definition 2.47. *Given two dendrograms T and T' in $(\mathcal{T}, W)$, define:*

- *$\Gamma(T, T')$ as the set of all finite edit paths between T and T';*
- *$cost(\gamma)$ as the sum of the costs of the edits for any $\gamma \in \Gamma(T, T')$;*
- *the dendrogram edit distance as:*

$$d_E(T, T') = \inf_{\gamma \in \Gamma(T,T')} cost(\gamma)$$

By definition the triangle inequality and symmetry must hold, but, up to now, this edit distance is intractable; one would have to search for all the possible finite edit paths which connect two dendrograms in order to find the minimal ones. On top of that, having an edit which is completely "for free", it is not even obvious that $d_E(T, T') > 0$ for some dendrograms. However, it is clear that d_E induces a pseudo-metric on classes of dendrograms up to order two vertices.

2.6.4 Mappings

Now we introduce a fundamental tool, called *mapping*, that, by parametrizing certain sets of edit paths, makes d_E computable and its properties more readily available. The idea of mappings is not novel (Tai, 1979) and often it is a the key ingredient both for proofs and calculations in Tree Edit Distances (Hong et al. (2017); Sridharamurthy et al. (2020); Wetzels et al. (2022) and references therein), but we employ it with some key modifications. From now on "D" and "G" will be used to indicate "deletion" and "ghosting". Recall that E_T identifies the vertices $V_T - \{r_T\}$.

A *mapping* between T and T' is a set $M \subset (E_T \cup \{"D", "G"\}) \times (E_{T'} \cup \{"D", "G"\})$ with the following properties:

(M1) consider the projection of the Cartesian product $(E_T \cup \{"D", "G"\}) \times (E_{T'} \cup \{"D", "G"\}) \to (E_T \cup \{"D", "G"\})$; we can restrict this map to M obtaining $\pi_T : M \to (E_T \cup \{"D", "G"\})$. The maps π_T and $\pi_{T'}$ are surjective on $E_T \subset (E_T \cup \{"D", "G"\})$ and $E_{T'} \subset (E_{T'} \cup \{"D", "G"\})$ respectively;

(M2) π_T and $\pi_{T'}$ are injective;

(M3) $M \cap (V_T \times V_{T'})$ is such that, given (a, b) and $(c, d) \in M \cap (V_T \times V_{T'})$, $a > c$, if and only if $b > d$;

(M4) if $(a, "G")$ (or $("G", a)$) is in M, let $child(a) = \{b_1, .., b_n\}$. Then there is one and only one i such that for all $j \neq i$, for all $x \in V_{sub(b_j)}$, we have $(x, "D") \in M$ (respectively $("D", x)$); and there is one and only one c such that $c = \max\{x' \in sub(b_i) \mid (x', y) \in M \text{ for any } y \in V_{T'}\}$.

Conditions (M1)-(M2) are asking that every vertex in $V_T - \{r_T\}$ is assigned to one and only one "transformation"; (M3) ensures that the associations induced by $M \cap (V_T \times V_{T'})$ respect the tree structures of T and T'; lastly (M4) means that, once all vertices v appearing in the couples $(v, "D")$ or $("D", v)$ in M are deleted, the points which are coupled with G (that is $(a, "G")$ or $("G", a)$) are all vertices of order two and therefore they can be ghosted.

Remark 2.48. *Properties (M1)-(M4) are not canonical properties of mappings, which usually satisfy properties (M2) and (M4) (see Hong et al. (2017); Sridharamurthy et al. (2020); Wetzels et al. (2022) and references therein). And this is caused by the introduction of the ghosting edit. Accordingly, the way in which mappings parametrize edit paths in the present work is completely novel and, for instance, it establishes some (partial) ordering between the edits.*

In what follows we use the properties of M to parametrize a set of edit paths in the dendrogram space, starting from T and ending in T', which are collected under the name γ_M. We call:

- γ_d^T a path made by the deletions to be done in T, that is, the couples $(v, "D")$, executed in any order. So we obtain $T_d^M = \gamma_d^T(T)$, which, instead, is well defined and not depending on the order of the deletions.
- One then proceeds with ghosting all the vertices $(v, "G")$ in M, in any order, getting a path γ_g^T and the dendrogram $T_M := \gamma_g^T \circ \gamma_d^T(T)$.
- Since all the remaining points in M are couples, the two dendrograms T'_M (defined in the same way as T_M, but starting from T') and T_M must be isomorphic as tree structures. This is guaranteed by the properties of M. So one can shrink T_M onto T'_M, and the composition of the shrinkings, executed in any order is an edit path γ_s^T.

By definition:

$$\gamma_s^T \circ \gamma_g^T \circ \gamma_d^T(T) = T'_M,$$

and:

$$(\gamma_d^{T'})^{-1} \circ (\gamma_g^{T'})^{-1} \circ \gamma_s^T \circ \gamma_g^T \circ \gamma_d^T(T) = T'$$

.

where the inverse of an edit path is thought as the composition of the inverses of the single edit operations, taken in the inverse order.

Lastly, we call γ_M the set of all possible edit paths:

$$(\gamma_d^{T'})^{-1} \circ (\gamma_g^{T'})^{-1} \circ \gamma_s^T \circ \gamma_g^T \circ \gamma_d^T.$$

obtained by changing the order in which the edit operations are executed inside γ_d, γ_g and γ_s. Observe that, even if γ_M is a set of paths, its cost is well defined:

$$cost(M) := cost(\gamma_M) = cost(\gamma_d^T) + cost(\gamma_s^T) + cost(\gamma_d^{T'}).$$

See Figure 2.9 for an example of a mapping between weighted trees.

Before moving on, we fix some notation and call $\mathrm{Mapp}(T, T')$ the set of all mappings between T and T'. This set is never empty, in fact $M = \{(v, "D") : v \in E_T\} \cup \{("D", v') : v' \in E_{T'}\}$ is always a mapping between T and T'. In other words one can always delete all the edges of a dendrogram, and then insert all the edges of the other.

Theorem 2.49 (Main Theorem)**.** *Given two dendrograms T and T', for every finite edit path γ, exists a mapping $M \in \mathrm{Mapp}(T, T')$ such that $cost(M) \leq cost(\gamma)$.*

A first corollary immediately follows.

Corollary 2.50. *Since $\mathrm{Mapp}(T, T')$ is a finite set we have the following well defined pseudo-metric:*

$$d_E(T, T') = \inf\{cost(\gamma) \,|\, \gamma \in \Gamma(T, T')\} = \min\{cost(M) \,|\, M \in \mathrm{Mapp}(T, T')\}$$

which we will refer to as the edit distance between T and T'.

A second corollary is obtained observing that, if a mapping has cost equal to zero, then it must contain only ghostings.

Corollary 2.51. *Given T and T' dendrograms, $d_E(T, T') = 0$ if and only if T and T' are equal up to order 2 vertices. In other words d_E is a metric for dendrograms considered up to order 2 vertices.*

We close this section with a very useful couple of results which will be applied in the next sections to work functions defined on merge trees and solve the issues presented in Section 2.5.6 with the local representation of functions obtained in such section. We make use of $sub_T(v)$ as defined in Definition 2.14.

Proposition 2.52 (Extension/Truncation)**.** *Take (T, φ_T) and $(T', \varphi_{T'})$. Suppose r_T and $r_{T'}$ are of order 1 and there is a splitting $\{(v, r_T)\} \to \{(v, v'), (v', r_T)\}$ and $\{(w, r_{T'})\} \to \{(w, w'), (w', r_{T'})\}$ giving the dendrograms (G, h_G) and $(G', h_{G'})$. Suppose moreover that $\varphi_G((v', r_T)) = \varphi_{G'}((w', r_G))$, then $d_E(T, T') = d_E(sub_G(v'), sub_{G'}(w'))$.*

Proof. Consider a minimizing mapping M between G and G'.

Apply the deletions described by M both on G and on G' obtaining the merge trees G_M and G'_M. After such deletions the vertices r_G and $r_{G'}$ are still in the resulting trees, for they cannot be removed in any way. Moreover, if $(v', w') \notin M$ then neither v' nor w' can be deleted. In fact, for any 4 positive numbers n_1, n_2, n_3, n_4 we have:

$$|\, n_1 + n_2 - (n_3 + n_4) \,| \leq n_1 + n_3 + |\, n_2 - n_4 \,|$$

thus instead of deleting v' with cost n_1 and w' with cost n_3 and then shrinking two edges of the form $e = (a, r_G)$ and $e' = (b, r_{G'})$ with weights n_2 and n_4 is better to merge (a, v') with (v', r_G) and (b, w') with $(w', r_{G'})$ and then shrink them.

Thus, whatever edge of the form $e = (a, r_G)$ remains contains, as a merged edge, also (v', r_G). And the same for $e' = (b, r_{G'})$. By construction e is matched with e'. Since $\varphi_{G_M}(e) = \varphi_G((v', r_G)) * \ldots$ and $\varphi_{G'_M}(e) = \varphi_G((w', r_{G'})) * \ldots$, when computing the cost of shrinking e on e', by (P4), $\varphi_G((v', r_G))$ and $\varphi_G((w', r_{G'}))$ cancel out.

Thus $d_E(T, T') = d_E(G, G') = d_E(sub_G(v'), sub_{G'}(w'))$. □

The proof of Proposition 2.52 also yields the following corollary.

Corollary 2.53. *Given (T, φ_T) and $(T', \varphi_{T'})$. If r_T and $r_{T'}$ are of order 1, for any minimizing mappings M, then neither (v, r_T) or $(w, r_{T'})$ are deleted and we have $\#\max M = 1$.*

2.7 Edit Distance Between Local Representation of Functions

This section has the multi-faceted role of connecting Section 2.6 to the initial goal of comparing functions defined on merge trees and, possibly, further clarifying some points which may require additional explanations or are yet to be addressed. Thus we touch on the following topics:

1. in Section 2.7.1 we take a closer look at the correspondence between the edit operations and the local representations of functions;
2. in Section 2.7.2 we return on the issues we left unsolved in Section 2.5.6 and present different ways in which the examples therein introduced can be framed into our framework;
3. in Section 2.7.3 we briefly introduce, relying on results proved in Chapter 4, some stability properties of d_E when applied to functions defined on merge trees;
4. we close this section with Section 2.7.4, which contains a brief example showcasing a very different behaviour between PDs and merge trees caused by the *elder rule*.

The remaining sections of the chapter, namely Section 2.8 and Section 2.9, are then devoted to the simulations and to the computational aspects of the edit distance d_E.

2.7.1 Edits of Local Representation of Functions

Consider now $\mathcal{M}(\pi_0(X_\bullet)) = (T, h_T)$ and a proper weight function φ_T with values in the editable space $L_1(\mathbb{R}, \mathbb{R}_{\geq 0})$. We know that φ_T is equivalent to a local representation of a function on $\mathcal{U}(D_{\pi_0(X_\bullet)})$ - see Section 2.5.5 and Definition 2.38, which, in turns, amounts to the datum of a function f in $L_1(D_{\pi_0(X_\bullet)}, \mathbb{R}_{\geq 0})$. By construction $\varphi_T(e) = f \circ (h_{|U})^{-1} : (t, t') \to \mathbb{R}_{\geq 0}$, with $U \in \mathcal{U}(D_{\pi_0(X_\bullet)})$ being associated to the edge e and h being the height function of $D_{\pi_0(X_\bullet)}$.

Suppose that we want to split the edge e into $e_1 = (v, v'')$ and $e_2 = (v'', v')$. We define the novel weight function $\varphi_{T'}$ as: $\varphi_{T'}(e_1) = (\varphi_T(e))_{|[t,t'']}$ and $\varphi_{T'}(e_2) = (\varphi_T(e))_{|[t'',t']}$ for some $t'' \in (t, t')$. As already mentioned, $E_{T'} \hookrightarrow D_{\pi_0(X_\bullet)}$ induces another regular a.e. cover $\mathcal{O}$ given by the replacement of the open set U associated to e, with the open sets $(h_{|U})^{-1}((h_T(v), h_T(v'')))$ and $(h_{|U})^{-1}((h_T(v''), h_T(v')))$, which are clearly contained in U. In other words, it induces a refinement of $\mathcal{U}(D_{\pi_0(X_\bullet)})$. Accordingly, the weight function $\varphi_{T'}$ is by construction obtained by restricting $\varphi_T(e)$ onto such open sets. Thus the splitting that we defined with the weight function $\varphi_{T'}$ is equivalent to the local representation of f obtained via the refinement $\mathcal{O}$ of $\mathcal{U}(D_{\pi_0(X_\bullet)})$ induced by $(T', h_{T'})$. See also Figure 2.7.

Viceversa, if we start from $(T', \varphi_{T'})$ and we ghost the vertex v'' we return to the tree structure T. The weight function φ_T on the new edge $e = (v, v')$, resulting from the merging of $e_1 = (v, v'')$ and $e_2 = (v'', v')$, is given by $\varphi_{T'}(e_1) * \varphi_{T'}(e_2)$ which is exactly passing from the local representation of f on $\mathcal{O}$ to the one on $\mathcal{U}(D_{\pi_0(X_\bullet)})$.

As we have just seen, we have defined a way of deforming dendrograms which accounts for refinements or "coarsements" of local representation of functions.

Remark 2.54. *We point out that, in general, shrinkings and splittings do not guarantee that, starting from a local representation of $f \in L_1(D_{\pi_0(X_\bullet)}, E)$, we end up with a local representation of some other function after an edit. This may be a point which could be improved by future works, but it does not represent a problem in terms of defining a metric structure to compare $f \in L_1(D_{\pi_0(X_\bullet)}, E)$ and $g \in L_1(D_{\pi_0(Y_\bullet)}, E)$.*

2.7.2 Normalizing and Truncating Functions

We devote this section to describe a way in which the functions defined in Section 2.5.6 can fit into our framework.

Consider $f, g : D_{\pi_0(X_\bullet)} \to E$. If $\| f_{|U_\infty} - g_{|U_\infty} \|_{L_1(U_\infty,E)} = \infty$, then there is really no point in comparing such functions and any attempt to embed those functions into $L_1(\mathbb{R}, E)$ implies losing infinite variability/information at least for one of the two functions. In fact, at least one between $f_{|U_\infty}$ and $g_{|U_\infty}$ has norm equal to ∞ and any approximation we make of that function with a function of finite norm, would be at infinite distance from the original function. However, if we deem that the information contained in $f_{|U_\infty}$ and $g_{|U_\infty}$ after a certain height is negligible, we can always extend f with $0 \in E$ after some $K \geq \max h_T$, with $\mathcal{M}(\pi_0(X_\bullet)) = (T, h_T)$. We indicate extension with $f_{|U_\infty} \cdot \chi_{[\max h_T, K]}$ with an abuse of notation, and we refer to it as the truncation of f at height K. Then, call $f_{|K}$ the function obtained as $(f_{|K})_{|U} := f_{|U}$ for all $U \in \mathcal{U}(D_{\pi_0(X_\bullet)})$, $U \neq U_\infty$, and $(f_{|K})_{|U_\infty} := f_{|U_\infty} \cdot \chi_{[\max h_T, K]}$. Clearly $f_{|K} \in L_1(D_{\pi_0(X_\bullet)}, E)$.

The examples in Section 2.5.6, however allow also for a different approach. Suppose we have $f : D_{\pi_0(X_\bullet)} \to E$, $g : D_{\pi_0(Y_\bullet)} \to E$ and $K \in \mathbb{R}$ such that:

$$f \circ (h^f_{|U^f_\infty})^{-1}(x) = g \circ (h^g_{|U^g_\infty})^{-1} g(x)$$

for $x > K$. That is, f, g are definitively equal going upwards towards the roots. Then, let (T, h_T) and (G, h_G) be the merge trees associated to the sublevel set filtrations of f and g respectiely. We can split $e_f = (v, r_T) \in E_T$ into $e'_f = (v, v')$, $e''_f = (v', r_T)$ and $e_g = (w, r_G) \in E_G$ into $e'_g = (w, w')$, $e''_g = (w', r_G)$, so that $h^f(v') = h^g(w') = K$. Let $(T', h_{T'})$ and $(G', h_{G'})$ be the merge trees obtained with such splittings. If we call $\varphi_{T'}$ and $\varphi_{G'}$ the local representations of f on T' and g on G', respectively, we have: $\| \varphi_{T'}(e''_f) - \varphi_{G'}(e''_g) \|_{L_1(\mathbb{R},E)} = 0$. Thus we are in the position to apply Proposition 2.52 to T' and G'.

In other words, if we can modify the functions Θs so that

$$\Theta \circ (h^f_{|U^f_\infty})^{-1}(x) = \Theta \circ (h^g_{|U^g_\infty})^{-1} g(x)$$

for some K and $x > K$ then $d_E(f, g)$ can be defined as $d_E(f_{|K}, g_{|K})$. We will do so requiring that Θ is definitively equal to some fixed constant.

Now we consider the different Θ employed in Section 2.5.6:

- Θ_1: in this case we have $\Theta_1 \equiv 1$ and thus Θ^1 is definitively constant and equal to 1;
- Θ_c: Θ_c is employed when we build clustering dendrograms and so definitively it is equal to the cardinality of the starting point cloud. Thus we can normalize Θ_c obtaining Θ^n_c which expresses the cardinality of the clusters as a percentage of the cardinality of the point cloud i.e. the measure of the clusters wrt the uniform measure on the point cloud. Clearly such function is definitively equal to 1;
- $\Theta_{\mathcal{L}}$: when we start from a function $f : X \to \mathbb{R}$ which is bounded and defined on $X \subset \mathbb{R}^n$ bounded, then $X_t = X$ for t big enough and so $\Theta_{\mathcal{L}}$ is definitively constant and equal to $\Theta_{\mathcal{L}}(X)$. Again we can normalize $\Theta_{\mathcal{L}}$, obtaining $\Theta^n_{\mathcal{L}}$ which expresses the measure of path connected components as a percentage of the measure of $\mathcal{L}(X)$.
- Θ_p: for this function it really depends on the chosen filtration and, in particular, if there is the possibility of having homology classes with death time $+\infty$ in p-dimensional homology, $p > 0$. However, if $H_p(U; \mathbb{K}) = 0$, $U \in \pi_0(X_t)$ for all t big enough, as is the case, for instance, with the Céch filtration, or other filtrations of a

simply connected space, we have no issues. In fact we know that, by construction, $H_0(U;\mathbb{K}) = \mathbb{K}$, $U \in \pi_0(X_t)$ for t big enough. Thus there is K big enough so that $\Theta_p \equiv (1,0)$ for $x > K$.

For what we have said previously then we can choose K big enough and define $d_E(f,g) := d_E(f_{|K}, g_{|K})$ for any f induced by the normalized functions Θ_1, Θ_c^n and $\Theta_{\mathcal{L}}^n$.

In other words, suppose that we want to work, for instance, with $\Theta_{\mathcal{L}}^n$ to analyze a data set of functions. For any couple of functions $f : X \to \mathbb{R}$ and $g : Y \to \mathbb{R}$ we obtain the abstract merge trees $\pi_0(X_\bullet)$ and $\pi_0(Y_\bullet)$ with sublevel set filtrations and the corresponding functions:

$$f^{\Theta_{\mathcal{L}}^n}(p) = \Theta_{\mathcal{L}}^n((a,t)) = \mathcal{L}(a)/\mathcal{L}(X)$$

for $p = (a,t) \in D_{\pi_0(X_\bullet)}$ and

$$g^{\Theta_{\mathcal{L}}^n}(q) = \Theta_{\mathcal{L}}^n((b,t)) = \mathcal{L}(b)/\mathcal{L}(Y)$$

for $q = (b,t) \in D_{\pi_0(Y_\bullet)}$. Then we choose K big enough $\Theta_{\mathcal{L}}^n((a,t)) = 1 = \Theta_{\mathcal{L}}^n((b,t))$ for $t \geq K$. Thus we can truncate these functions from K upwards and obtain the local representations $\varphi_T^{\Theta_{\mathcal{L}}^n}$ and $\varphi_G^{\Theta_{\mathcal{L}}^n}$ of the truncated functions. Note that $\mathrm{supp}(\varphi_T^{\Theta_{\mathcal{L}}^n}((v,r_T))) = [\max h_T, K]$ and $\mathrm{supp}(\varphi_G^{\Theta_{\mathcal{L}}^n}((w,r_G))) = [\max h_G, K]$.

By Proposition 2.52, we are guaranteed that this truncation process:

1. does not depend on K, in the following sense. Suppose we have a third function $r : H \to \mathbb{R}$ such that $r^{\Theta_{\mathcal{L}}^n}(u) = \Theta_{\mathcal{L}}^n((c,t)) < 1$ for some $t > K$. While $f^{\Theta_{\mathcal{L}}^n}$ and $g^{\Theta_{\mathcal{L}}^n}$ can be truncated at height K, for $r^{\Theta_{\mathcal{L}}^n}$ we must consider some $K' > K$ to compute $d_E(f^{\Theta_{\mathcal{L}}^n}_{|K'}, r^{\Theta_{\mathcal{L}}^n}_{|K'})$. However, we have :

$$d_E(f^{\Theta_{\mathcal{L}}^n}_{|K'}, g^{\Theta_{\mathcal{L}}^n}_{|K'}) = d_E(f^{\Theta_{\mathcal{L}}^n}_{|K}, g^{\Theta_{\mathcal{L}}^n}_{|K});$$

2. moreover, comparing the truncated functions $f^{\Theta_{\mathcal{L}}^n}_{|K}$ is exactly the same as comparing the original functions $f^{\Theta_{\mathcal{L}}^n}$ with d_E.

2.7.3 Stability

In this section we establish some stability properties for the metric d_E when applied to functions defined on merge trees. To do so, we leverage on the proof of Theorem 1 in Chapter 4.

Developing stability results tailored to the different pipelines we present to obtain functions defined on merge trees is a very broad topic which is outside the aim of the present work. Such results, in fact, require to establish sufficient conditions both for the merge trees to be similar and for Θ to be similar on portions of the merge trees which can be matched together via low-cost mappings.

In this context we will deal with the more general of the two issues, removing the problem about similarity of the values of Θ, which is very application-dependent, and focus on how the functional framework we designed is able to handle similarity between merge trees. In other words we consider Θ_1, which is constant everywhere on the trees and assess the stability properties of some functions $\varphi_T^{\Theta_1}$ and $\varphi_{T'}^{\Theta_1}$ obtained as topological summaries of uniformly close scalar fields.

Being the edit distance a summation of the costs of local modification of trees, we expect that the stability properties of d_E are quite different from the ones of the bottleneck distance between persistence diagrams, which is defined as the biggest modification ones needs to match two persistence diagrams. Instead, we expect the edit distance to be

dependent on the number of vertices in the merge trees but, at the same time, that the cost of the local modifications we need to match the two merge trees go to 0. For this reason we give the following definitions.

Definition 2.55. *Given a constructible persistence module $S : \mathbb{R} \to \mathrm{Vec}_{\mathbb{K}}$, we define its rank as* $\mathrm{rank}(S) := \#PD(S)$ *i.e. the number of points in its persistence diagram. When S is generated on $\mathbb{K}$ by an abstract merge tree $\pi_0(X_\bullet)$ we have* $\mathrm{rank}(S) := \#PD(S) = \#L_T$, *with $(T, h_T) = \mathcal{M}(\pi_0(X_\bullet))$ and may refer to* $\mathrm{rank}(S)$ *also as the rank of the merge tree* $\mathrm{rank}(T)$. *We also fix the notation* $\dim(T) := \#E_T$.

Definition 2.56. *Let f, g be tame functions defined on a path connected topological space X. Define $X_t = f^{-1}((-\infty, t])$ and $Y_t = g^{-1}((-\infty, t])$. Let $T = \mathcal{M}(\pi_0(X_\bullet))$ and $T' = \mathcal{M}(\pi_0(Y_\bullet))$ be the merge trees associated to f and g respectively. A metric for merge trees is locally stable if:*

$$d(T, T') \leq K(\mathrm{rank}(T) + \mathrm{rank}(T')) \parallel f - g \parallel_\infty$$

for some $K > 0$.

Now we prove that d_E induces a locally stable metric on merge trees, via the relationship $(T, h_T) \cong (T', h_{T'})$ if and only if $(T, \varphi_T^{\Theta_1}) \cong (T', \varphi_{T'}^{\Theta_1})$ that we find in Section 2.5.6.1.

Corollary 2.57 (of Theorem 1 in Chapter 4)**.** *Let f, g be tame functions defined on a path connected topological space X and such that*

$$sup_{x \in X} |f(x) - g(x)| \leq \varepsilon.$$

Define $X_t = f^{-1}((-\infty, t])$ and $Y_t = g^{-1}((-\infty, t])$. Let $T = \mathcal{M}(\pi_0(X_\bullet))$ and $T' = \mathcal{M}(\pi_0(Y_\bullet))$ be the merge trees associated to f and g respectively.

Then, there exists a mapping M between V_T and $V_{T'}$ such that:

- *any deletion of an edge $e = (v, v')$ is such that $h_T(v') - h_T(v) \leq 2\varepsilon$;*
- *for any edge (v, v') which is shrunk on (w, w') after all ghostings and deletions on T and on T' we have $|\, h_T(v) - h_{T'}(w) \,| < \varepsilon$ and $|\, h_T(v') - h_{T'}(w') \,| < \varepsilon$ (if $v' \neq r_T$ and $w' \neq r_{T'}$).*

Theorem 2.58. *Let f, g be tame functions defined on a path connected topological space X and such that*

$$sup_{x \in X} |f(x) - g(x)| \leq \varepsilon.$$

Define $X_t = f^{-1}((-\infty, t])$ and $Y_t = g^{-1}((-\infty, t])$. Lastly, let $T = \mathcal{M}(\pi_0(X_\bullet))$ and $T' = \mathcal{M}(\pi_0(Y_\bullet))$ be the merge trees associated to f and g respectively. And let $(T, \varphi_T^{\Theta_1})$ and $(T', \varphi_{T'}^{\Theta_1})$.

Then, there exists a mapping M such that $cost(e_i) < 2 \cdot \varepsilon$, for $e_i \in M$.

Proof. The proof is largely based on Theorem 1 in Chapter 4.

First notice that $\parallel \varphi_T^{\Theta_1}((v, v')) \parallel_{L_1(\mathbb{R})} = h_T(v') - h_T(v) = w_T((v, v'))$ - see Section 2.6.2.3. Thus the cost of deleting v in (T, w_T) is the same as in $(T, \varphi_T^{\Theta_1})$.

Second, consider the following cases:

1. if $h_T(v) < h_{T'}(w) < h_T(v') < h_{T'}(w')$:

$$\parallel \varphi_T^{\Theta_1}((v, v')) - \varphi_{T'}^{\Theta_1}((w, w')) \parallel_{L_1(\mathbb{R})} = |\, h_T(v) + h_T(w) - h_{T'}(w) - h_{T'}(w') \,|\,;$$

2. if $h_T(v) < h_T(v') < h_{T'}(w) < h_{T'}(w')$:
$$\| \varphi_T^{\Theta_1}((v,v')) - \varphi_{T'}^{\Theta_1}((w,w')) \|_{L_1(\mathbb{R})} = w_T((v,v')) + w_{T'}((w,w'));$$

3. if $h_T(v) < h_{T'}(w) < h_{T'}(w') < h_T(v')$:
$$\| \varphi_T^{\Theta_1}((v,v')) - \varphi_{T'}^{\Theta_1}((w,w')) \|_{L_1(\mathbb{R})} = w_T((v,v')) - w_{T'}((w,w')).$$

Consider the same mapping M constructed in the proof of Theorem 1 in Chapter 4. As in Corollary 2.57 M is such that:

- the deletions $(v, "D")$ and $("D", w)$ are always such that $w_{T'}(w), w_T(v) \leq 2\varepsilon$;
- for any edge (v, v') which is shrink on (w, w') after all ghostings and deletions on T and on T' we have $|\, h_T(v) - h_{T'}(w) \,| < \varepsilon$ and $|\, h_T(v') - h_{T'}(w') \,| < 2\varepsilon$.

As a consequence:

1. if $h_T(v) < h_{T'}(w) < h_T(v') < h_{T'}(w')$: $cost_M((v,w)) \leq 2\varepsilon$;
2. if $h_T(v) < h_T(v') < h_{T'}(w) < h_{T'}(w')$: since $h_{T'}(w) - h_T(v) < \varepsilon$, then $w_T((v,v')), w_{T'}((w,w')) < \varepsilon$. So $cost_M((v,w)) \leq 2\varepsilon$.
3. if $h_T(v) < h_{T'}(w) < h_{T'}(w') < h_T(v')$: $cost_M((v,w)) \leq 2\varepsilon$.

□

2.7.4 Example: Merge Trees vs PDs

We close this section with one example which shows how the elder rule, via the instability of the persistence pairs, makes very difficult to add pieces of information to persistence diagrams in a stable way.

Consider the following functions, plotted in Figure 2.8c, defined on $[-1, 2]$:

$$f(x) = |\, x - 1 \,| + \varepsilon \text{ if } x \geq 0$$
$$f(x) = |\, 2x - 1 \,| \text{ if } x < 0$$

and

$$g(x) = |\, x - 1 \,| \text{ if } x \geq 0$$
$$g(x) = |\, 2x - 1 \,| + \varepsilon \text{ if } x < 0$$

for a fixed $\varepsilon > 0$.

Let (T, h_T) and $(T', h_{T'})$ be the merge trees associated to the sublevel set filtrations of f and g; moreover let $\varphi_T^{\Theta_\mathcal{L}}$ and $\varphi_{T'}^{\Theta_\mathcal{L}}$ the two respective local representations of the induced functions with $\Theta_\mathcal{L}$ being the Lebesgue measure on $\mathbb{R}$. Note that $\| f - g \|_\infty = \varepsilon$. The local minima of the functions are the points $\{-0.5, 1\}$, with $f(-0.5) = 0$, $f(1) = \varepsilon$, $g(-0.5) = \varepsilon$ and $g(1) = 0$. Thus the merge trees have isomorphic tree structures: we represent T with the vertex set $\{v_{-0.5}, v_1, v_0, r_T\}$ and edges $\{(v_{-0.5}, v_0), (v_1, v_0), (v_0, r_{T'})\}$; and T' with vertices $\{v_{-0.5}, v_1, v_0, r_{T'}\}$ and edges $\{(v_{-0.5}, v_0), (v_1, v_0), (v_0, r_{T'})\}$. The height functions are the following: $h_T(v_{-0.5}) = 0$, $h_{T'}(v_{-0.5}) = \varepsilon$, $h_T(v_1) = \varepsilon$, $h_{T'}(v_1) = 0$ and $h_T(v_0) = h_{T'}(v_0) = 1 + \varepsilon$.

Having truncated both functions at height $1 + \varepsilon$, the weight functions (see Figure 2.8d) are given by: $\varphi_T^{\Theta_\mathcal{L}}(v_{-0.5})(t) = t\chi_{[0,1)} + \chi_{[1,1+\varepsilon)}$, $\varphi_T^{\Theta_\mathcal{L}}(v_1)(t) = 2(t-\varepsilon)\chi_{[\varepsilon,1+\varepsilon)}$ and $\varphi_{T'}^{\Theta_\mathcal{L}}(v_{-0.5})(t) = (t - \varepsilon)\chi_{[\varepsilon,1+\varepsilon)}$ and $\varphi_{T'}^{\Theta_\mathcal{L}}(v_1)(t) = 2t\chi_{[0,1)} + 2\chi_{[1,1+\varepsilon)}$.

The zero-dimensional persistence diagram associated to f (we name it $PD_0(f)$) is given by a point with coordinates $(0, +\infty)$, associated to the connected component $[-t/2 - 0.5, t/2 - 0.5]$ which is born at $t = 0$, and the point $(\varepsilon, 1+\varepsilon)$, associated to the component $[1-(t-\varepsilon), 1+(t-\varepsilon)]$, born at level $t = \varepsilon$ and "dying" at level $t = 1+\varepsilon$, due to the elder rule, since it merges an older component, being the other component born at a lower level.

For the function g, the persistence diagram $PD_0(g)$ is made by the same points, but the situation is in some sense "reversed". In fact, the point $(0, +\infty)$ is associated to the connected component "centered" in 1, which is $[1-t, 1+t]$, and the point $(\varepsilon, 1+\varepsilon)$, is associated to the component "centered" in 0.5, that is $[-(t-\varepsilon)/2 - 0.5, (t+\varepsilon)/2 - 0.5]$.

The consequence of this change in the associations between points and the components originating the points of the diagrams is that the information regarding the two components, end up being associated to very different spatial locations in the two diagrams: $(0, +\infty)$ and $(\varepsilon, 1+\varepsilon)$. And this holds for every $\varepsilon > 0$. Thus it seems very hard to design a way to "enrich" $PD_0(f)$ and $PD_0(g)$ with additional information, originating the "enriched diagrams" D_f and D_g, respectively, and design a suitable metric d, so that $d(D_f, D_g) \to 0$ as $\varepsilon \to 0$.

Instead, if we consider the mapping $M = \{(v_{-0.5}, v_{-0.5}), (v_1, v_1), (r_T, r_{T'})\}$ we have $d_E((T, \varphi_T^{\Theta_\mathcal{L}}), (T', \varphi_{T'}^{\Theta_\mathcal{L}})) \leq cost(M) = 3\varepsilon$.

2.8 Simulations

Now we use two simulated data sets to put to work the frameworks defined in Section 2.5 and Section 2.6 and the algorithm developed in Section 2.C. The examples are basic, but suited to assert that dendrograms and the metric d_E capture the information we designed them to grasp. In particular, since examples in Section 2.4.1 and Section 2.4.2 already give insights into the role of the tree-structured information, we want to isolate and emphasize the key role of weight functions. We also deal with the problem of approximating the metric d_E when the number of leaves in the tree structures in the data set is too big to be handled. The examples presented concern hierarchical clustering dendrograms and dendrograms representing scalar fields.

In the implementations, dendrograms are always considered with a binary tree structure, obtained by adding negligible edges, that is edges e with arbitrary small $d(\varphi(e), 0)$, when the number of children of a vertex exceeds 2.

2.8.1 Pruning

In this section we propose a way of approximating the edit distance when the number of leaves of the involved tree structures is too high.

If one defines a proper weight function with values in an editable space E coherently with the aim of the analysis, then the value $d(\varphi_T(e), 0)$ can be thought as the amount of information carried by the edge e. The bigger such value is, the more important that edge will be for the dendrogram. In fact such edges are the ones most relevant in terms of d_E. A sensible way to reduce the computational complexity of the metric d_E, losing as little information as possible, is therefore the following. Given $\varepsilon > 0$ and a dendrogram (T, φ_T), define the following 1-step process:

($\mathcal{P}_\varepsilon$) Take a leaf l such that $d(\varphi_T(l), 0)$ is minimal among all leaves; if two or more leaves have minimal weight, choose l at random among them. If $d(\varphi_T(l), 0) < \varepsilon$, delete l and ghost its father if it becomes an order 2 vertex after removing l.

We set $T_0 = T$ and we apply operation ($\mathcal{P}_\varepsilon$) to obtain T_1. On the result we apply again ($\mathcal{P}_\varepsilon$) obtaining T_2 and, for $n > 2$, we proceed iteratively until we reach the fixed point

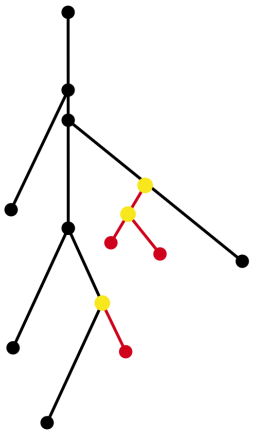


Figure 2.10: Pruning of a weighted tree: in red the deletions and in yellow the ghostings.

of the sequence $\{T_n\}$, which we call $P_\varepsilon(T)$. In this way we define the pruning operator $P_\varepsilon : \mathcal{T} \to \mathcal{T}$. Note that the fixed point is surely reached in a finite time since the number of leaves of each tree in the sequence is finite and non increasing along the sequence. More details on such pruning operator applied on merge trees representing the path connected components of the sublevel sets of real valued functions can be found in Chapter 4, showing in the case of merge trees that the pruning operation can be interpreted quite naturally in terms of function deformations.

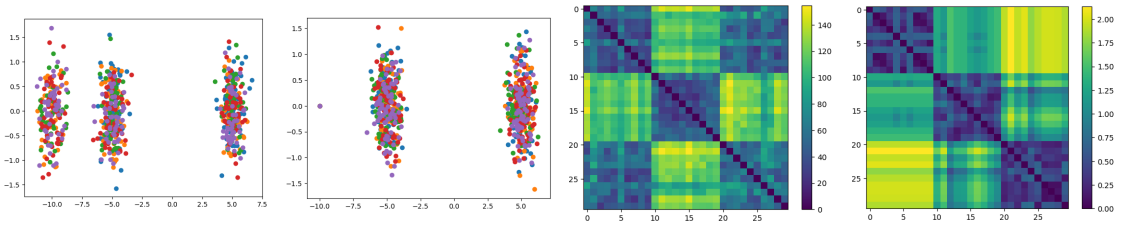
If we define $\| T \|$ as $\| T \| = \sum_{e \in E_T} d(\varphi(e), 0)$, we can quantify the (normalized) lost information with what we call *pruning error* (*PE*): $(\| T \| - \| P_\varepsilon(T) \|)/ \| T \|$.

2.8.2 Hierarchical Clustering Dendrograms

We consider a data set of 30 points clouds in $\mathbb{R}^2$, each with 150 or 151 points. Point clouds are generated according to three different processes and are accordingly divided into three classes. Each of the first 10 point clouds is obtained by sampling independently two clusters of 75 points respectively from normal distributions centered in $(5, 0)$ and $(-5, 0)$, both with $0.5 \cdot Id_{2\times 2}$ covariance. Each of the subsequent 10 point clouds is obtained by sampling independently 50 points from each of the following Gaussian distributions: one centered in $(5, 0)$, one in $(-5, 0)$ and one in $(-10, 0)$. All with covariance $0.5 \cdot Id_{2\times 2}$. Lastly, to obtain each of the last 10 point clouds, we sample independently 150 points as done for the first 10 clouds, that is 75 independent samples from a Gaussian centered $(5, 0)$ and 75 from one centered in $(-5, 0)$, an then, to such samples, we add an outlier placed in $(-10, 0)$.

Some clouds belonging to the second class and third classes are plotted respectively in Figure 2.11a and Figure 2.11b. We obtain dendrograms induced by the single linkage hiercarhical clsutering dendrograms, with the cardinality functions induced by Θ_c^n and then resort to pruning because of the high number of leaves, but we still expect to be able to easily separate point clouds belonging to the first and third classes (that is, with two major clusters) from clouds belonging to the second class, which feature three clusters, thanks to the cardinality information function defined in Section 2.5.6.2. All dendrograms have been pruned with the same threshold, giving an average pruning error of 0.15.

We can see in Figure 2.11c that this indeed the case. It is also no surprise that persistence diagrams do not perform equally good in this classification task, as displayed in Figure 2.11d. In fact PDs have no information about the importance of the cluster, making it impossible to properly recognize the similarity between data from the first and



(a) Data from the second class.

(b) Data from the third class.

(c) Pairwise distances with dendrograms.

(d) Pairwise distances with PD.

Figure 2.11: Data and pairwise distance matrices involved in the hierarchical clustering example.

third class. They are, however, able to distinguish clouds belonging to class two from clouds belonging to class three since the persistence of the homology class associated to the leftmost cluster in clouds belonging to class two is smaller compared to what happens in clouds from the third class. The cluster centered in $(-10, 0)$ and the one in $(-5, 0)$ are in fact closer when the first one is a proper cloud, than when it is a cluster made by a single point.

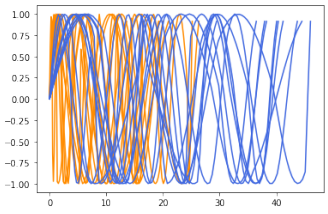
2.8.3 Dendrograms of Functions

This time our aim is to work with dendrograms obtained from functions, adding the (truncated) weight function induced by the Lebesgue measure of the sublevel sets $\Theta_{\mathcal{L}}$ and using them to discriminate between two classes in a functional data set.

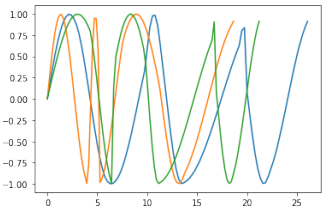
We simulate the data set so that the discriminative information is contained in the size of the sublevel sets and not in the structure of the critical points. To do so, we reproduce a situation which is very similar to the one shown by Sangalli et al. (2010) for the Berkeley Growth Study data, where all the variability between groups in a classification task is explained by warping functions. We fix a sine function defined over a compact $1D$ real interval (with the Lebesgue measure) and we apply to its domain 100 random non linear warping functions belonging to two different, but balanced, groups. Warpings from the first group are more likely to obtain smaller sublevel sets, while in the second groups we should see larger sublevel sets and so "bigger" weight functions defined on the edges. Note that, being the Lebesgue measure invariant with the translation of sets, any horizontal shifting of the functions would not change the distances between dendrograms.

The base interval is $I = [0, 30]$ and the base function is $f(x) = \sin(x)$. The warping functions are drawn in the following way. Pick N equispaced control points in I and then we draw N samples from a Gaussian distribution truncated to obtain only positive values. We thus have $x_1, \dots x_N$ control points and $v_1, \dots, v_N$ random positive numbers. Define $y_i := \sum_{j=1}^{i} v_j$. The warping is then obtained interpolating with monotone cubic splines the couples (x_i, y_i). Being the analysis invariant to horizontal shifts in the functions, for all statistical units we fix $x_0 = y_0 = 0$ for visualization purposes.

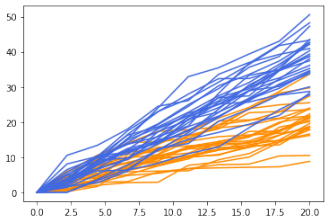
The groups are discriminated by the parameters of the Gaussian distribution from which we sample the positive values v_i to set up the warpings. For the first class we sample $N = 10$ positive numbers from a truncated Gaussian with mean 3 and standard deviation 2; for the second the mean of the Gaussian is 5 and the standard deviation is 2. Thus we obtain each of the first 50 functions sampling 10 values v_i from the truncated Gaussian centered in 3, building the warping function as explained in the previous lines, and then reparametrizing the sine function accordingly. The following 50 functions are obtained with the same pipeline but employing a Gaussian centered in 5. Note that, by construction, all the functions in the data set share the same merge tree. We truncate the



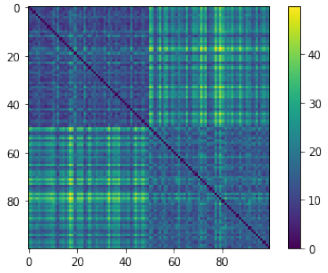
(a) Subset of functions colored by classes.



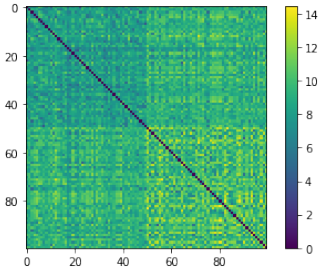
(b) Few functions from the first class.



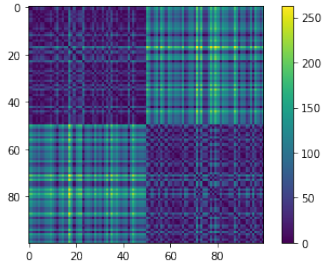
(c) Subset of warping functions colored by classes.



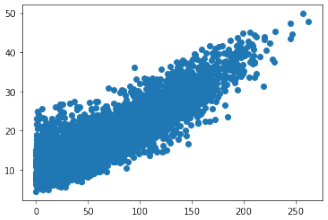
(d) Pairwise distances with dendrograms.



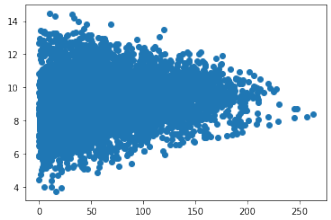
(e) Pairwise distances with L_2 metric.



(f) Pairwise distances with L_2 metric on warping functions.



(g) Correlation between dendrograms and warping functions metric.



(h) Correlation between naive L_2 and warping functions metric.

Figure 2.12: Overview of the example of Section 2.8.3.

functions induced by $\Theta_{\mathcal{L}}$ at height 1.

Examples of the warping functions can be seen in Figure 2.12c; the resulting functions can be seen in Figure 2.12a. The key point here is that we want to see if the dendrograms can retrieve the information contained in the warping functions. For this reason we compare the L_2 pairwise distances between such functions (see Figure 2.12f) and the pairwise distances obtained with dendrograms (see Figure 2.12d). The visual inspection confirms the close relationships between the two sources of information. Moreover, if we vectorize the arrays given by the two matrices (considering only entries above the diagonal) and compute the Fisher correlation, we get a score of 0.85 (see Figure 2.12g). Instead, a naive approach with the L_2 metric applied directly to the data set would capture no information at all, as we can observe from Figure 2.12e and the Fisher correlation with the matrix obtained from warping functions is 0.15 (see Figure 2.12h).

Note that, in general, the problem of finding warping functions to align functional data is deeply studied and with no easy solution (see, for instance, the special issue of the Electronic Journal of Statistics dedicated to phase and amplitude variability - year 2014, volume 8 or Srivastava et al. (2011a)) especially for non-linear warping of multidimensional or non-euclidean domains. Instead, dendrograms less sensitive to such dimensionality issues, in the sense that they only arise in calculating the connected components and measure of the sublevel sets.

2.9 Computing the Edit Distance: Decomposition Properties

In this last section we develop some results and formulations needed to obtain the algorithm presented in the supplementary material, Section 2.C. These theoretical results allow to recursively split up the calculations (following ideas found in Hong et al. (2017)) and lead to the integer optimization problems defined in Section 2.B. The key point is that the properties of editable spaces imply that we can locally look at differences between subtrees and then use those pieces of information to compute the distance between two dendrograms.

Name T_2 the only representative without order 2 vertices inside the equivalence class of T. One can always suppose that a dendrogram is given without order 2 vertices. Thus, for notational convenience, from now on we suppose $T = T_2$ and $T' = T'_2$. To help us in the calculations define a particular subset of mappings $M_2(T, T') \subset \mathrm{Mapp}(T, T')$.

Definition 2.59. *A mapping $M \in \mathrm{Mapp}(T, T')$ has maximal ghostings if $(v, "G") \in M$ if and only if v is of order 2 after the deletions in T and, similarly $("G", w) \in M$ if and only if w is of order 2 after the deletions in T'.*

A mapping $M \in \mathrm{Mapp}(T, T')$ has minimal deletions if $(v, "D") \in M$ only if v is not of order 2 after applying all the other deletions in T and, similarly, $("D", w) \in M$ only if w is not of order 2 after applying all the other deletions in T'.

We collect all mappings with maximal ghostings and minimal deletions in the set $M_2(T, T')$.

In other words we are always eliminating all the order 2 vertices which arise from deletions and we are not deleting edges which we can shrink. The following lemma then applies.

Lemma 2.60.

$$\min\{cost(M) \,|\, M \in \mathrm{Mapp}(T, T')\} = \min\{cost(M) \,|\, M \in M_2(T, T')\}$$

In addition to that, we consider some particular subsets of $E_T \times E_{T'}$ which play a fundamental role in what follows. Recall that, using $E_T \cong V_T - \{r_T\}$, we can induce $\pi_T : E_T \times E_{T'} \to V_T$.

A set $M^* \subset E_T \times E_{T'}$ is in $\mathcal{C}^*(T, T')$ if:

(A1) the points in $\pi_T(M^*)$ form antichains in V_T (and the same for $\pi_{T'}(M^*)$ in $V_{T'}$), with respect to the partial order given by $father > son$. This means that any two distinct vertices of T (respectively of T') which appear in M^* are incomparable with respect to ">";

(A2) the projections $\pi_T : M^* \to V_T$ and $\pi_{T'} : M^* \to V_{T'}$ are injective.

Consider now $M^* \in \mathcal{C}^*(T, T')$. Starting from such set of couples we build a set of edits which form a "partial" mapping between T and T': each couple $(x, y) \in M^*$ means that we do not care of what lies below $x \in V_T$ and $y \in V_{T'}$ and we need to define edits only for the other vertices. The vertices below x and y will be taken care separately. In this way M^* is used as a "dimensionality reduction tool": instead of considering the problem of finding directly the optimal mapping between T and T', we split up the problem in smaller subproblems, and put the pieces together using M^*. To formally do that, some other pieces of notation are needed.

Let $v \in E_T$. One can walk on the graph of the tree-structure T going towards any other vertex. For any $v \in E_T$, ζ_v is the shortest graph-path connecting v to r_T. Note that this is the ordered set $\zeta_v = \{v' \in V_T \,|\, v' > v\}$. Similarly, denote with $\zeta_x^{x'}$ the shortest path on the graph of T connecting x and x'. Note that $\min \zeta_x \cap \zeta_{x'}$ is the least common ancestor between x and x'. This is because v is an ancestor of x iff $v > x$. In other words we have: $LCA(x, x') = \min \zeta_x \cap \zeta_{x'}$.

By Property (A1), given $x \in V_T \cap \pi_T(M^*)$, there exist a unique $\Omega_{M^*}(x) \notin \pi_T(M^*)$ such that:

$$\Omega_{M^*}(x) = \min\{LCA(x, x') \,|\, x' \in \pi_T(M^*) \text{ and } x \neq x'\}$$

And the same holds for $y \in V_{T'} \cap \pi_{T'}(M^*)$. For ease of notation we will often avoid explicit reference to M^* and write directly $\Omega(x)$.

With these bits of notation, given $M^* \in \mathcal{C}^*(T, T')$, we build the "partial" mapping $\alpha(M^*)$: is a mapping that ignores all the vertices which lie below every x, y such that $(x, y) \in M^*$. Figure 2.13 may help in following the upcoming paragraph. Consider $v \in V_T$:

1. if $(v, w) \in M^*$, then $(v, w) \in \alpha(M^*)$;
2. if there is not $x \in V_T$ such that $v < \Omega(x)$ or $v > \Omega(x)$, then $(v, "D") \in \alpha(M^*)$;
3. if there is $x \in V_T$ such that $v > \Omega(x)$ then $(v, "D") \in \alpha(M^*)$;
4. if there is $x \in V_T$ such that $v < \Omega(x)$:
   (a) if $v \in \zeta_x^{\Omega(x)}$ then $(v, "G") \in \alpha(M^*)$
   (b) if $v < v_i$ for some $v_i \in \zeta_x^{\Omega(x)} = \{v_0 < v_1 < \ldots < v_n\}$ then $(v, "D") \in \alpha(M^*)$;
   (c) if $v < x$ no edit is associated to v.

Remark 2.61. *By Properties (A1) and (A2), the conditions used to build $\alpha(M^*)$ are mutually exclusive. This means that each $v \in V_T$ satisfies one and only one of the above conditions and so $\alpha(M^*)$ is well defined.*

The idea behind $\alpha(M^*)$ is that, for all couples $(x, y) \in M^*$, we want to turn $\zeta_x^{\Omega(x)}$ and $\zeta_y^{\Omega(y)}$ into single edges of the form $(x, \Omega(x))$ and $(y, \Omega(y))$ respectively, and then shrink one in the other. As we already anticipated, $\alpha(M^*)$ takes care of all the vertices in V_T and $V_{T'}$, a part from the sets $\cup_{(x,y)\in M^*}\{x' \in E_T \,|\, x' < x\}$ and $\cup_{(x,y)\in M^*}\{y' \in E_{T'} \,|\, y' < y\}$. For this reason we say that $\alpha(M^*)$ is a partial mapping.

We state this formally with the next proposition.

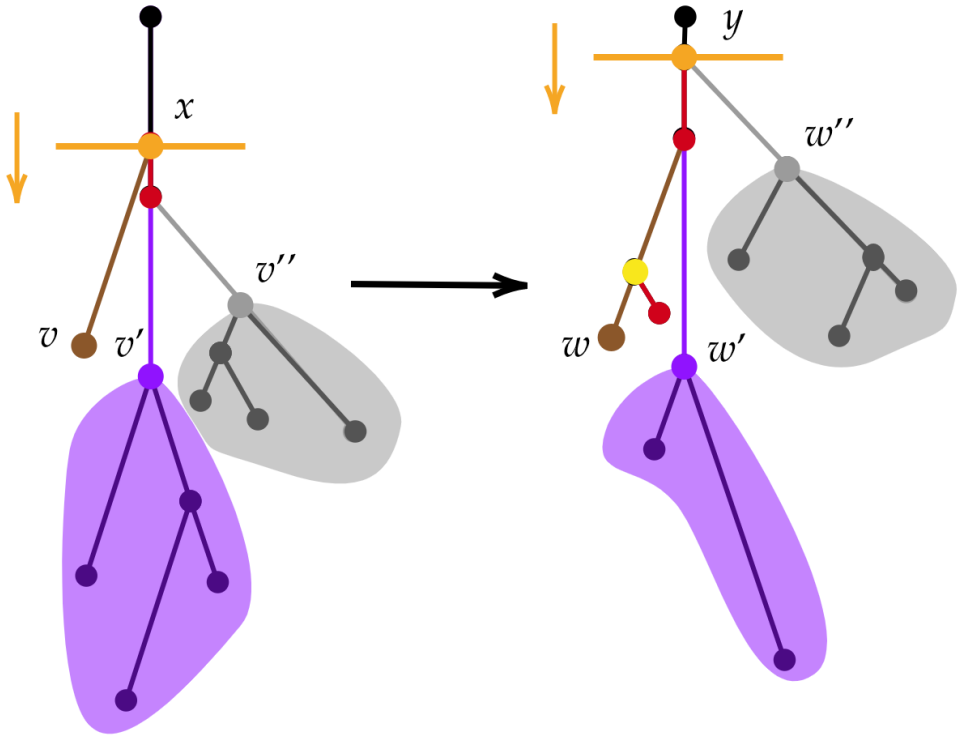


Figure 2.13: Given two weighted trees T (left) and T' (right) - which are the same of Figure 2.9a - we consider $T_x = sub_T(x)$ and $G_y = sub_{T'}(y)$ and we use Theorem 2.63 to compute $d_E(T_x, T_y)$, as in the algorithm in Section 2.B. The set $M^* = \{(v, w), (v', w'), (v'', w'')\}$ satisfies (A1),(A2). The set $\alpha(M^*)$ is made by the deletions/insertions indicated by the red edges, the ghostings/splittings indicated by the yellow and the shrinkings given by edges of the same color, different from red. To obtain $\widetilde{T}_x$ and $\widetilde{T}'_y$ as in Proposition 2.62 all the black vertices covered by shaded regions must be deleted.

Proposition 2.62. *Consider T and T' and $M^* \in \mathcal{C}^*(T, T')$. We obtain from such dendrograms, respectively, the dendrograms $\widetilde{T}$ and $\widetilde{T}'$ by deleting all the vertices $\bigcup_{(x,y)\in M^*}\{x' \in E_T \,|\, x' < x\}$ and $\bigcup_{(x,y)\in M^*}\{y' \in E_{T'} \,|\, y' < y\}$. The set $\alpha(M^*)$ is a mapping in $M_2(\widetilde{T}, \widetilde{T}')$.*

Now we have all the pieces we need to obtain the following key result.

Theorem 2.63 (Decomposition)**.** *Given T, T' dendrograms:*

$$d_E(T, T') = \min_{M^*\in\mathcal{C}^*(T,T')} \sum_{(x,y)\in M^*} d_E(sub_T(x), sub_{T'}(y)) + cost(\alpha(M^*)) \tag{2.1}$$

This result is the foundation of the bottom-up algorithm developed in the supplementary material, which is used in the upcoming simulations.

2.10 Discussion

In this chapter we develop a framework to work with functions defined on merge trees. As motivated throughout the manuscript, we argue that these kinds of topological summaries can succeed in situations where persistence diagrams and merge trees alone are

not effective. They also provide a great level of versatility because of the wide range of additional information that can be extracted from data. We define a metric structure on some spaces of functions defined on merge trees which has suitable stability properties and can be calculated by solving a set of smaller and easier subproblems. This metric proves to be feasible if the number of leaves is not too high and we carry out some examples to showcase its effectiveness in situations which are of interest in different branches of data analysis.

The are however some drawbacks in the framework presented in this chapter:

- the computational complexity involved in computing the metric is surely an issue. Such complexity however is justified by the stability properties satisfied by the metric and it is lower than exact methods available for other stable metrics;

- the deformation between two functions is not guaranteed to always produce a function at the intermediate steps i.e. the metric is not intrisic in the space of functions defined on merge trees. This may limit the statistical tools that can be defined in this space: should Frechét means exists, for instance, it is not guaranteed that they are functions.

The generality of the work however opens up many possible research, some of which are already investigated in other works:

- we think that the properties of the editable spaces can be relaxed; however, the algorithm presented in this manuscript may need to be adapted to the properties of the chosen weight space;

- starting from the edit distance presented in this chapter a metric for merge trees via the edit distance between weighted trees is developed and studied - also in relationship with other metrics for merge trees - in Chapter 3. Its stability properties are assessed in Chapter 4;

- interactions with the more general case of Reeb Graphs can be investigated, possibly following the decomposition presented in Stefanou (2020);

- applications of data analysis with merge trees via the framework defined in this chapter and in Chapter 3 can be found for instance in Chapter 4 and Chapter 5. Such analyses show that interpretable statistics able to discriminate between persistence diagrams and merge trees could be very useful and could provide new topological characterization of data which is inaccessible via PDs.

Appendix

Outline of the Appendix

In Section 2.A the are the proofs of the results appearing in this chapter. The remaining part of the Appendix is devoted to the algorithm which computes d_E: in Section 2.B we develop a LIP formulation of the metric d_E and use it in Section 2.C to write a bottom-up algorithm to compute the distance; the feasibility of the algorithm is then assessed with some simulations in Section 2.D.

2.A Proofs

Proof of Theorem 2.49.

To lighten the notation we use the following symbols:

- the edit induced by $(v, "D")$ is called v_d and v_d^{-1} stands for $("D", v)$.
- the edit induced by $(v, "G")$ is called v_g and v_g^{-1} stands for $("G", v)$.
- the edit induced by (v, v') is called $v_{\varphi,\varphi'}$ with φ being the original weight function, and φ' the weight function after the shrinking.

We know that the set of finite edit paths between two dendrograms is nonempty.

Suppose that γ is a finite edit path. This means that γ is the composition of a finite set of edits. We indicate such ordered composition with $\gamma = \prod_{i=0}^{N} e_i$ with e_i edit operation. We would like to change the order of the edit operations without raising the cost and changing the extremes of the edit path. This is not always possible. However we can work it around in the useful cases using properties (P1)-(P4). In particular, we would like to know when we can commute a generic edit e_i in the following situations:

- $v_d \circ e_i$ and $e_i \circ v_d^{-1}$
- $v_g \circ e_i$ and $e_i \circ v_g^{-1}$.

Moreover we want to reduce the edit path to max one edit for any vertex of T and T'.

We divide the upcoming part of the proof in subsections, each devoted to different combinations of edits.

v_d and v_d^{-1}

When we delete or insert one vertex, we are modifying the tree structure at the level of its father and its children. Therefore, we are only taking into account operations on the father, on the vertex himself or on the children of the deleted/inserted vertex.

- $v_d \circ v'_g$ with v son of v', can be safely replaced with $v_d \circ v'_d$. Instead of ghosting the father and then deleting the whole edge, we can delete both edges one by one; conserving the length of the path (P3). If v is father of v' then we can safely commute the operations.

- $v_d \circ v_g^{-1}$ can be replaced with $v'_{\varphi,\varphi'}$ with v' father of v (after the insertion) and φ' properly defined not to raise the cost of the path. In fact we are inserting v on an edge and then deleting it. This can obviously be achieved by shrinking the original edge (without changing the path length - (P4)).

- similarly, $v_d \circ v_g'^{-1}$ with v' to be father of v can again be replaced safely by a proper shrinking: instead of inserting a point in an edge, and deleting then the edge below, we can directly shrink the original edge (P4). If v' is to be inserted below v this is the same situation, but seen from the point of view of the son of v.

- $v_d \circ v_{\varphi,\varphi'}$ can be replaced by v_d potentially diminishing the length of the path, but surely not raising it (P1).

- $v'_g \circ v_d^{-1}$. If v' is the father of v, this edit can be replaced with just $v'_{\varphi,\varphi'}$ with appropriate weights: we are inserting an edge under a vertex which (in this case) becomes of order two and is ghosted. We can directly modify the edge without changing the length of the path (P4). If v' is the vertex which would become son of v, we can simply shrink v to obtain the same result without raising the cost (P4).

- $v_g'^{-1} \circ v_d^{-1}$, with v' to appear on the edge inserted with v_d^{-1} cannot commute (otherwise can always commute), but can be replaced by two insertions: instead of inserting an edge and then splitting it, we can directly insert two smaller edges; without changing the cost of the path (P3).

- $v_{\varphi,\varphi'} \circ v_d^{-1}$ can be replaced with an insertion directly with weight φ', possibly shortening the path (P1).

- consider $v_d'^{-1} \circ v_d$ with v' to be inserted with, as father, the father of v; if the children of v' are different from the children of v, this operation cannot commute. If the children are the same, it can be changed with a shrinking of v, reducing the length of the path by at most $cost(v_d'^{-1}) + cost(v_d)$ (P1).

v_g and v_g^{-1}

Like in the previous case, we only take into account transformations concerning the father and the son of the added/ghosted order two vertex.

- $v_g \circ v'_g$, with v and v' being on adjacent edges, can commute (P2).

- $v_g \circ v_g'^{-1}$, with v and v' being on adjacent edges, can commute provided we define carefully the splitting $v_g'^{-1}$ (P2).

- $v_g \circ v_{\varphi,\varphi'}$ means that we are shrinking a vertex before ghosting it. However, we can achieve the same result, without increasing the path length, by ghosting the vertex at first, and then shrinking its son (P1)-(P4).

- $v'_{\varphi,\varphi'} \circ v_g^{-1}$ either with $v' = v$, or with v father of v', can be replaced with an appropriate shrinking of the (future) son of v, and then an appropriate insertion of v' without changing the length of the path (P3)-(P4).

- $v_g \circ v'_d$ with v father of v' cannot be commuted and cannot be replaced by a similar operation which inverts ghosting and deletion.

$v_{w,w'}$

- $v_{\varphi',\varphi''} \circ v_{\varphi,\varphi'}$ can be replaced by $v_{\varphi,\varphi''}$ which is either conserving or shortening the path (P1).
- $v_{\varphi,\varphi'}^{-1} = v_{\varphi',\varphi}$.

Thanks to these properties we can take a given path $\gamma = \prod_{i=0,\dots,N} e_i$ and modify the edit operations in order to obtain the following situation:

- the first operations are all in the form v_d; this can be achieved because $v_d \circ -$ can be always rearranged, potentially by changing the path as shown before and shortening it. Of course there can be only one deletion for each vertex of T;
- then we have all the edits in the form v_g; since $v_g \circ -$ is exchangeable any time but when we have $v_g \circ v_d'$, this is not a problem. Observe that all order two vertices which were not deleted can be ghosted (at most one time);
- in the same way we can put last all the paths in the form v_d^{-1} and before them v_g^{-1}. All the new vertices appearing with the insertion of edges and the splitting of edges with order two vertices are all nodes which remain in T' and which are not further edited;
- in the middle we are left with the shrinking paths. Since we can substitute $v_{\varphi,\varphi'} \circ v_{\varphi',\varphi''}$ with $v_{\varphi,\varphi''}$, we can obtain just one single transformation on a vertex.

Thus

$$\overline{\gamma} = (\gamma_d^{T'})^{-1} \circ (\gamma_g^{T'})^{-1} \gamma_s^T \circ \gamma_g^T \circ \gamma_d^T.$$

with:

- $\gamma_d^T = \prod v_d$
- $\gamma_g^T = \prod v_g$
- $\gamma_s^T = \prod v_{\varphi,\varphi'}$
- $(\gamma_g^{T'})^{-1} = \prod v_g^{-1}$
- $(\gamma_d^{T'})^{-1} = \prod v_d^{-1}$

is such that $\gamma(T) = \overline{\gamma}(T) = T'$ and $cost(\overline{\gamma}) \leq cost(\gamma)$. The key point is that $\overline{\gamma}$ can be easily realized as a mapping in the following way:

- $(v, \text{"}D\text{"})\ \forall v_d \in \gamma_d^T$
- $(v, \text{"}G\text{"})\ \forall v_g \in \gamma_g^T$
- $(v, v')\ \forall v_{\varphi,\varphi'} \in \gamma_s^T$, where v' is the renaming of v, with weight given by φ'.
- $(\text{"}G\text{"}, v)\ \forall v_g^{-1} \in (\gamma_g^{T'})^{-1}$
- $(\text{"}D\text{"}, v)\ \forall v_d^{-1} \in (\gamma_d^{T'})^{-1}$

■

Proof of Lemma 2.60.
Any order 2 vertex which is not ghosted is paired with another order 2 vertex. Ghosting both of them does not increase the cost of the mapping.

■

Proof of Proposition 2.62.

Condition (M2) coincide with condition (A2). Condition (M3) is clearly satisfied because of the antichain condition (A1). Consider a vertex $v \in E_T$. The only case in which v is not edited is when $v < x$ with $x \in v_T \cap \pi_T(M^*)$. However, in this case, v does not appear in $\widetilde{T}$, and thus (M1) is satisfied. Moreover, all and only order 2 vertices, after the deletions, are ghosted, and (M4) follows .

■

Proof of Theorem 2.63.
Let $M \in M(T, T')$ such that $d_E(T, T') = cost(M)$.

We note that $father > son$ induces a partial order relationship also on the pairs given by coupled points in M: $(x, y) > (v, w)$ if $x > v$ and $y > w$. In fact, by property (M3), $x > v$ if and only if $y > w$. So we can select (x_i, y_i), the maxima with respect to this partial order relationship. Thus, we obtain (x_0, y_0),...,(x_n, y_n) which form an antichain (both in V_T and $V_{T'}$).

Clearly $M^* = \{(x_0, y_0),...,(x_n, y_n)\} \in \mathcal{C}^*(T, T')$. Now we build $\alpha(M^*)$ and compare the cost of its edits with the ones in M. Let $\bar{x} = LCA(x_i, x_j)$. Since $\bar{x} > x_i, x_j$, it is not coupled in M. Since x_i and x_j are coupled, $\bar{x}$ cannot be ghosted, so it is deleted in M. Any point x above $\bar{x}$ is deleted for the same reasons. So the edits above $\bar{x}$ are shared between $\alpha(M^*)$ and M.

In $\alpha(M^*)$ we ghost any point between $\bar{x}$ and x_i (and the same for x_j) and this is not certain to happen in M (some points could be deleted). Nevertheless, even in the worst case, these ghostings are guaranteed not to increase the distance. For instance, suppose $x_i < x < \bar{x}$ is deleted in M and ghosted by $\alpha(M^*)$, then:

$$d(x_i * x, y_i) \leq d(x_i * x, y_i * x) + d(y_i * x, y_i) = d(x_i, y_i) + d(x, 0)$$

by properties (P1)-(P4). Since $\alpha(M^*) \in M_2(\widetilde{T}, \widetilde{T}')$ by Proposition 2.62, we have:

$$\sum_{(x,y)\in M^*} d_E(sub_T(x), sub_{T'}(y)) + cost(\alpha(M^*)) \leq cost(M)$$

Now we prove the other inequality.

Consider M^* which realizes the minimum of the right side of Equation (2.1), and M_i which realizes $d_E(sub(x_i), sub(y_i))$ with $(x_i, y_i) \in M^*$. We build a mapping M collecting edits in the following way: for every $x' \in E_T$ if $x' \in sub(x_i)$, we take the edit associated to it from M_i, otherwise we know that it is edited by $\alpha(M^*)$, and we take it from there; the set of these assignments gives $M \in M_2(T, T')$ whose cost is exactly $\sum_{(x_i,y_i)\in M^*} cost(M_i) + cost(\alpha(M^*))$. This gives the second inequality.

■

2.B Computing the Edit Distance: Dynamical Integer Linear Programming problems

We want to use Theorem 2.63 to write a dynamical, integer linear optimization algorithm to calculate d_E: by translating Theorem 2.63 into a Integer Linear Programming (ILP) problem, we obtain a single step in a bottom-up procedure.

2.B.1 Notation

We are given two dendrograms T, T' inside some dendrogram space $(\mathcal{T}, E)$. Our objective is to write down Equation (2.1) as a function of some binary variables.

Consider $x \in V_T$ and $y \in V_{T'}$. Along with keeping the notation defined in Section 2.9, define $T_x := sub_T(x)$ and $T_y := sub_{T'}(y)$, $N_x := dim(T_x) = \#E_T$ and $N_y := dim(T_y) = \#E_{T'}$. In particular, given $v \in V_{T_x}$, the sequence $v_0 = v < v_1 < \ldots < r_T$ indicates the points in ζ_v. Thus v_i will be a vertex $v_i > v$. The same with $w \in V_{T_y}$.

2.B.2 Relaxing the Optimization Problem

We would like to find $M^* \in \mathcal{C}^*(T_x, T_y)$ minimizing Equation (2.1) for T_x and T_y, but this is a difficult task. In fact, as evident in the construction of $\alpha(M^*)$, a set $M^* \in \mathcal{C}^*(T_x, T_y)$ has the role of pairing segments of dendrograms: if $(v, w) \in M^*$, then the paths $\zeta_v^{\Omega(v)}$ and $\zeta_w^{\Omega(w)}$ are paired and then shrunk one on the other by $\alpha(M^*)$. However, the points $\Omega(v)$ and $\Omega(w)$ depend on the whole set M^*, and not simply on the couple (v, w). Modeling such global dependence gives rise to non-linear relationships between coupled points, and so leading to a non linear cost function, in terms of points interactions, to be minimized. For this reason we "weaken" the last term in Equation (2.1), allowing also mappings different from $\alpha(M^*)$ to be built from M^*. In other words we minimize over $M^* \in \mathcal{C}^*(T_x, T_y)$ the following equation:

$$\sum_{(v,w)\in M^*} d_E(sub_{T_x}(v), sub_{T_y}(w)) + cost(\beta(M^*)) \tag{2.2}$$

where $\beta(M^*)$ is such that:

- $\beta(M^*) \in M_2(\widetilde{T}_x, \widetilde{T}_y)$ (with the notation obtained from Proposition 2.62 replacing T and T' with T_x and T_y respectively);
- the set of vertices coupled by $\beta(M^*)$ is exactly M^*: $M^* = \beta(M^*) \cap V_{T_x} \times V_{T_y}$.

Since, by construction $M^* = \alpha(M^*) \cap V_T \times V_{T'}$ and by Proposition 2.62, $\alpha(M^*) \in M_2(\widetilde{T}_x, \widetilde{T}_y)$, minimizing Equation (2.1) or Equation (2.2) gives the same result.

2.B.3 Setup and Variables

Suppose we already have W_{xy} which is a $N_x \times N_y$ matrix such that $(W_{xy})_{v,w} = d_E(T_v, T_w)$ for all $v \in E_{T_x}$ and $w \in E_{T_y}$. Note that:

- if x and y are leaves, $W_{xy} = 0$.
- if v, w are vertices of T_x, T_y, then W_{vw} is a submatrix of W_{xy}.

The function to be optimized is defined on the following set of binary variables: for every $v \in E_{T_x}$ and $w \in E_{T_y}$, for $v_i \in \zeta_v$, $v_i < r_{T_x}$, and $w_j \in \zeta_w$, $w_j < r_{T_y}$, take a binary variable $\delta_{i,j}^{v,w}$. We use δ to indicate the matrix of variables $(\delta_{i,j}^{v,w})_{v,w,i,j}$.

The mapping $\beta(M^*)$ is built according to the variables $\delta_{i,j}^{v,w} = 1$: we write a constrained optimization problem such that having $\delta_{i,j}^{v,w} = 1$ means pairing the segments $\zeta_v^{v_{i+1}}$ (that is, the sequence of edges which starts with (v, v_1) and ends with (v_i, v_{i+1})) and $\zeta_w^{w_{j+1}}$, and shrinking one in the other in the induced mapping.

In order to pair and shrink the segments $\zeta_v^{v_{i+1}} = \{v = v_0, v_1, \ldots, v_{i+1}\}$ and $\zeta_w^{w_{j+1}}$ we need to define a set of edits $\beta(M^*)$ adding the following edits:

- all the points $v_k \in \zeta_v^{v_{i+1}}$ with $0 < k < i + 1$ are ghosted, that is $(v_k, G) \in \beta(M^*)$;

- if $v' < v_k$ for some $0 < k < i+1$, then $(v', D) \in \beta(M^*)$;
- if $v' \geq v_{i+1}$ and $v' \neq r_T$, then $(v', D) \in \beta(M^*)$
- $(v, w) \in \beta(M^*)$.

Of course analogous edits must be induced on vertices in T_y. Thus, the edit $(v, w) \in \beta(M^*)$, along the edit paths induced by $\beta(M^*)$, means: shrinking the edge (v, v_{i+1}) onto (w, w_{j+1})). Recall that, if $\delta_{i,j}^{v,w} = 1$, we do not need to define edits for $sub_{T_x}(v)$ and $sub_{T_y}(w)$ since, by assumption, we already know $d_E(T_v, T_w)$.

2.B.4 Constraints

Clearly, not all combinations of $\delta_{i,j}^{v,w}$ are acceptable, in that the set $\beta(M^*)$ is not always a mapping with $M^* \in \mathcal{C}^*(T_x, T_y)$: for instance segments could be paired multiple times. To avoid such issues, we build a set of constraints for the variable δ.

For each $v' \in V_{T_x}$ we call $\Phi(v') := \{(v'', i) \in V_T \times \mathbb{N} \,|\, v' = v_i'' \in \zeta_{v''}^{v''_{i+1}}\}$. In an analogous way we define $\delta(w')$ for $w' \in V_{T_y}$. Call $\mathcal{K}$ the set of values of δ such that for each leaf l in V_{T_x}:

$$\sum_{v' \in \zeta_l} \left(\sum_{(v'',i) \in \Phi(v')} \sum_{w,j} \delta_{i,j}^{v'',w} \right) \leq 1 \tag{2.3}$$

and for each leaf l' in V_{T_y}:

$$\sum_{w' \in \zeta_{l'}} \left(\sum_{(w'',j) \in \Phi(w')} \sum_{v,i} \delta_{i,j}^{v,w''} \right) \leq 1 \tag{2.4}$$

The following proposition clarifies the properties of any value of $\delta \in \mathcal{K}$.

Proposition 2.64. *If $\delta \in \mathcal{K}$:*

- *the couples (v, w) such that $\delta_{i,j}^{v,w} = 1$ define a set $M^* \in \mathcal{C}^*(T_x, T_y)$;*
- *the edits induced by all $\delta_{i,j}^{v,w} = 1$ give a mapping $\beta(M^*)$ in $M_2(\widetilde{T}_x, \widetilde{T}_y)$. With $\widetilde{T}_x, \widetilde{T}_y$ being obtained from T_x and T_y as in Proposition 2.62.*

Proof. Having fixed a leaf l, the constraint in Equation (2.3) allows for at most one path $\zeta_v^{v_{i+1}} \subset \zeta_l$ to be kept after the editing induced by all the variables equal to 1. Moreover if $(v'', i) \in \Phi(v) \cap \Phi(v')$, then $v = v_i'' = v'$. Thus, variables are added at most one time in Equation (2.3) and Equation (2.4). Which means that for any $a \in V_{T_x}$, we are forcing that a can be an internal vertex or lower extreme of at most one path $\zeta_v^{v_{i+1}}$ such that $\delta_{i,j}^{v,w} = 1$. In other words if two "kept" segments $\zeta_v^{v_{i+1}}$ and $\zeta_{v'}^{v'_{i'+1}}$ (i.e. with $\delta_{i,j}^{v,w} = \delta_{i',j}^{v',w} = 1$) intersect each other, it means that they just share the upper extreme $v_{i+1} = v'_{i'+1}$. These facts together imply that (if the constraints are satisfied) the edits induced on T_x by $\delta_{i,j}^{v,w} = 1$ and $\delta_{i',j'}^{v',w'} = 1$ are always compatible: if $v'' \in V_{sub_T(v_i)}$ then it is not touched by (the edits induced by) $\delta_{i',j'}^{v',w'} = 1$ (and the same exchanging the role of v' and v), if v'' is equal or above v_{i+1} and/or $v'_{i'+1}$, then it is deleted in any case. Lastly, by noticing that if $\delta_{i,j}^{v,w} \in \Phi(v')$ then $\delta_{i,j'}^{v,w'} \in \Phi(v')$ for all other possible w' and j', we see that every segment $\zeta_v^{v_i}$ is paired with at most one segment $\zeta_w^{w_j}$, and viceversa.

As a consequence, for any vertex v' in any of the tree structures, at most one point on the path $\zeta_{v'}$ is coupled in M^*, guaranteeing the antichain condition. Moreover, any point

of T_x which is in $\pi_{T_x}(M^*)$ is assigned to one and only one point of T'_y and viceversa. The edits induced by $\delta = 1$ clearly satisfy properties (M2)-(M4). Passing to $\widetilde{T}_x$ and $\widetilde{T}_y$, also (M1) is satisfied. □

Remark 2.65. *If for every $\delta_{i,j}^{v,w} = 1$, $v_{i+1} = \Omega(v)$, then $\beta(M^*) = \alpha(M^*)$.*

2.B.5 Objective Function

Having built a mapping $\beta(M^*)$ using the binary variables, we want to define a cost functions which computes the cost of such mapping depending on δ.

Consider $v \in E_{T_x}$ and interpret $\delta_{i,j}^{v,w} = 1$ as coupling the segments $\zeta_v^{v_{i+1}}$ and $\zeta_w^{w_{j+1}}$; then v is coupled with some $w \in E_{T_y}$ if $C(v) := \sum_{i,w,j} \delta_{i,j}^{v,w} = 1$ and is ghosted if $G(v) := \sum_{\{i,v' \mid v \in \zeta_{v'}^{v'_i}\}} \sum_{w,j} \delta_{i,j}^{v',w} = 1$. The vertex v is instead deleted if $D(v) := 1 - C(v) - G(v) = 1$. We introduce also the following quantities, which correspond to the cost of shrinking $\zeta_v^{v_{i+1}}$ on $\zeta_w^{w_{j+1}}$:

$$\Delta_{i,j}^{v,w} = d\left(\sum_{v' \in \zeta_v^{v_i}} \varphi_{T_x}(v'), \sum_{w' \in \zeta_w^{w_j}} \varphi_{T_y}(w')\right)$$

Note that the above sums are taken inside the editable space E.

The function which computes the cost given by coupled points is therefore:

$$F^C(\delta) := \sum_{v,w,i,j} \Delta_{i,j}^{v,w} \cdot \delta_{i,j}^{v,w}$$

The contribution of deleted points is: $F^D(\delta) - F^-(\delta)$, where

$$F^D(\delta) := \sum_{v \in T_x} D(v) \cdot d(\varphi_{T_x}(v), 0) + \sum_{w \in T_y} D(w) \cdot d(\varphi_{T_y}(w), 0)$$

and

$$F^-(\delta) := \sum_{v \in T_x} C(v) \cdot ||\, sub_{T_x}(v) \,|| + \sum_{w \in T_y} C(w) \cdot ||\, sub_{T_y}(w) \,||$$

where the "norm" of a tree T is $||\, T \,|| = \sum_{e \in E_T} d(\varphi(e), 0)$.

Finally, one must take into account the values of $d_E(T_v, T_w)$, whenever v and w are coupled; this information is contained in $(W_{xy})_{v,w}$:

$$F^S(\delta) := \sum_{v,w} (W_{xy})_{v,w} \cdot \left(\sum_{i,j} \delta_{i,j}^{v,w}\right)$$

Proposition 2.66. *With the notation previously introduced:*

$$d_E(T_x, T_y) = \min_{\delta \in \mathcal{K}} F^C(\delta) + F^D(\delta) - F^-(\delta) + F^S(\delta) \tag{2.5}$$

Proof. The contribution of coupled points is $F^C(\delta)$ and the contribution of deleted points is $F^D(\delta) - F^-(\delta)$.

The cost of $\beta(M^*)$ is: $F^\beta(\delta) := F^C(\delta) + F^D(\delta) - F^-(\delta)$. Lastly, $F^S(\delta)$ takes into account the value of $d_E(T_v, T_w)$, if v and w are coupled. By Theorem 2.63, combined with Proposition 2.64, the solution of the following optimization problem:

$$\min_{\delta \in \mathcal{K}} F^S(\delta) + F^\beta(\delta) \tag{2.6}$$

is equal to $d_E(T_x, T_y)$. □

Remark 2.67. *A solution to Problem Equation* (2.5) *exists because the minimization domain is finite and there are admissible values; it is not unique in general.*

2.C Computing the Edit Distance: Bottom-Up Algorithm

In this section the results obtained in Section 2.9 and the formulation established in Section 2.B are used to obtain the algorithm implemented to compute the metric d_E between dendrograms. Some last pieces of notation are introduced in order to describe the "bottom-up" nature of the algorithm.

Given $x \in V_T$, define $len(x)$ to be the number of vertices in ζ_x and $len(T) = \max_{v \in V_T} len(v)$. Therefore, $lvl(x) = len(T) - len(x)$. Lastly, $lvl_T(n) = \{v \in V_T \,|\, lvl(v) = n\}$

The key property is that: $lvl(x) > lvl(v)$ for any $v \in sub(x)$. Thus, if W_{xy} is known for any $x \in lvl_T(n)$ and $y \in lvl_{T'}(m)$, then for any v, w in V_T, $V_{T'}$ such that $lvl(v) < n$ and $lvl(w) < m$, W_{vw} is known as well. With this notation we can write down Algorithm 1.

Algorithm 1. Bottom-Up Algorithm.

Result: $d_E(T, T')$
initialization: $N = len(T)$, $M = len(T')$, $n = m = 0$;
while $n \leq N$ *or* $m \leq M$ **do**
  for $(x, y) \in V_T \times V_{T'}$ *such that* $lvl(x) \leq n$ *and* $lvl(y) \leq m$ **do**
    Calculate $(W_{r_T r_{T'}})_{x,y}$ solving Problem (2.5);
  end
  $n = n + 1$; $m = m + 1$;
end
return $(W_{r_T r_{T'}})_{r_T, r_{T'}}$

We end up with a result to analyze the performances of Algorithm 1 in the case of dendrograms with binary tree structures.

Proposition 2.68. *Let T and T' be two dendrograms with full binary tree structures with $dim(T) = \#E_T = N$ and $dim(T') = M$.*

Then $d_E(T, T')$ can be computed bu solving $O(N \cdot M)$ ILP problems with $O(N \cdot log(N) \cdot M \cdot log(M))$ variables and $O(N + M)$ constraints.

Proof. In a full binary tree structure, at each level l we have 2^l vertices. Let $L = len(T)$ and $L' = len(T')$. We have that, for any vertex $v \in V_T$ at level l, the cardinality of the path from v to any of the leaves in $sub_T(v)$ is $L - l$ and the number of leaves in $sub_T(v)$ is 2^{L-l}.

So, given $v \in V_T$ at level l and $w \in V_{T'}$ at level l', to calculate $d_E(sub_T(v), sub_{T'}(w))$ (having already W_{vw}) we need to solve a integer linear problem with $2^{L-l} \cdot (L-l) \cdot 2^{L'-l'} \cdot (L'-l')$ variables and $2^{L-l} + 2^{L'-l'}$ linear constraints.

Thus, to calculate $d_E(T, T')$, we need to solve $(2^{L+1} - 1) \cdot (2^{L'+1} - 1)$ linear integer optimization problems, each with equal or less than $2^L \cdot L \cdot 2^{L'} \cdot L'$ variables and equal or less that $2^L + 2^{L'}$ constraints. Substituting $L = log_2(N)$ and $L' = log_2(M)$ in these equations gives the result. □

Note that binary dendrograms are dense (with respect to d_E) in any dendrogram space as long as for any $\varepsilon > 0$, there is $x \in (E, *, 0)$ such that $d(x, 0) < \varepsilon$. So this is indeed a quite general result.

2.C.1 Example

Here we present in details the first steps of the Algorithm 1, used to calculate the distance between two merge trees.

We consider the following couple of merge trees. Let (T, h_T) be the merge tree given by: $V_T = \{a, b, c, d, r_T\}$, $E_T = \{(a, d), (b, d), (d, r_T), (c, r_T)\}$ and $w_T(a) = w_T(b) = w_T(d) = 1$, $w_T(c) = 5$; the merge tree $(T', h_{T'})$ instead, is defined by: $V_{T'} = \{a', b', c', d', r_{T'}\}$, $E_{T'} = \{(a', d'), (b', d'), (d', r_{T'}), (c', r_{T'})\}$ and $w_{T'}(a) = 1$, $w_{T'}(b) = w_{T'}(c) = 2$ and $w_{T'}(d) = 3$.

Step: $n = m = 0$

This step is trivial since we only have couples between leaves, like (a, a'), which have trivial subtrees and thus $d_E(sub_T(a), sub_{T'}(a')) = 0$.

Step: $n = m = 1$

The points $x \in V_T$ with $lvl_T(x) \leq 1$ are $\{a, b, c, d\}$ and the points $y \in V_{T'}$ with $lvl_{T'}(y) \leq 1$ are $\{a', b', c', d'\}$. Thus the couples (x, y) which are considered are: (d, d'), (d, a'), (d, b'), (d, c') and (a, d'), (b, d'), (c, d'). The couples between leaves, like (a, a') have already been considered.

Couple: (d, d') Let $T_d = sub_T(d)$ and $T_{d'} = sub_{T'}(d')$. The set of internal vertices are respectively $E_{T_d} = \{a, b\}$ and $E_{T_{d'}} = \{a', b'\}$. For each vertex $v <$ root in each subtree, where "root" stands for d or d', roots of T_d and $T_{d'}$ respectively, we have $\zeta_v = \{v_0 = v, v_1 =$ root$\}$. Thus, the binary variables we need to consider, are the following: $\delta_{0,0}^{a,a'}$, $\delta_{0,0}^{a,b'}$, $\delta_{0,0}^{b,a'}$ and $\delta_{0,0}^{b,b'}$. The quantities $\Delta_{i,j}^{v,w}$ are given by: $\Delta_{0,0}^{a,a'} = 0$, $\Delta_{0,0}^{a,b'} = 1$, $\Delta_{0,0}^{b,a'} = 0$ and $\Delta_{0,0}^{b,b'} = 1$. Thus:

$$F^C(\delta) = 0 \cdot \delta_{0,0}^{a,a'} + \delta_{0,0}^{a,b'} + 0 \cdot \delta_{0,0}^{b,a'} + \delta_{0,0}^{b,b'}$$

While:

$$F^D(\delta) = (1 - \delta_{0,0}^{a,a'} - \delta_{0,0}^{a,b'}) \cdot 1 + (1 - \delta_{0,0}^{b,a'} - \delta_{0,0}^{b,b'}) \cdot 1 + (1 - \delta_{0,0}^{a,a'} - \delta_{0,0}^{b,a'}) \cdot 1 + (1 - \delta_{0,0}^{a,b'} - \delta_{0,0}^{b,b'}) \cdot 2$$

and:

$$F^-(\delta) = (\delta_{0,0}^{a,a'} + \delta_{0,0}^{a,b'}) \cdot 0 + (\delta_{0,0}^{b,a'} + \delta_{0,0}^{b,b'}) \cdot 0 + (\delta_{0,0}^{a,a'} + \delta_{0,0}^{b,a'}) \cdot 0 + (\delta_{0,0}^{a,b'} + \delta_{0,0}^{b,b'}) \cdot 0$$

and:

$$F^S(\delta) = \delta_{0,0}^{a,a'} \cdot 0 + \delta_{0,0}^{a,b'} \cdot 0 + \delta_{0,0}^{b,a'} \cdot 0 + \delta_{0,0}^{b,b'} \cdot 0$$

Lastly the constraints are:

$$\delta_{0,0}^{a,a'} + \delta_{0,0}^{a,b'} \leq 1;\ \delta_{0,0}^{b,a'} + \delta_{0,0}^{b,b'} \leq 1;\ \delta_{0,0}^{a,a'} + \delta_{0,0}^{b,a'} \leq 1;\ \delta_{0,0}^{a,b'} + \delta_{0,0}^{b,b'} \leq 1$$

A solution is given by $\delta_{0,0}^{a,a'} = \delta_{0,0}^{b,b'} = 1$ and $\delta_{0,0}^{a,b'} = \delta_{0,0}^{b,a'} = 0$, which entails $F^C(\delta) = 1$, $F^D(\delta) = 0$, $F^-(\delta) = 0$ and $F^S(\delta) = 0$ and $d_E(T_d, T_{d'}) = 1$.

Couple: (d, a') Obviously: $d_E(sub_T(d), sub_{T'}(a')) = ||sub_T(d)||$. All the couples featuring a leaf and an internal vertex (that is, a vertex which is not a leaf), such as (d, b'), (a, d') etc. behave similarly.

Step: $n = m = 2$

The points $x \in V_T$ with $lvl_T(x) \leq 2$ are $\{a, b, c, d, r_T\}$ and the points $y \in V_{T'}$ with $lvl_{T'}(y) \leq 2$ are $\{a', b', c', d', r_{T'}\}$. Thus the couples (x, y) which are considered are $(d, r_{T'})$, (r_T, d'), $(r_T, r_{T'})$ and then the trivial ones: (r_T, a'), (r_T, b'), (r_T, c') and $(a, r_{T'})$, $(b, r_{T'})$, $(c, r_{T'})$. Some couples have already been considered and thus are not repeated.

Couple: $(d, r_{T'})$ Let $T_d = sub_T(d)$ and $T' = sub_{T'}(r_{T'})$. The set of internal vertices are respectively $E_{T_d} = \{a, b\}$ and $E_{T_{d'}} = \{a', b', c', d'\}$. Thus, the binary variables we need to consider, are the following: $\delta_{0,0}^{a,a'}$, $\delta_{0,1}^{a,a'}$, $\delta_{0,0}^{a,b'}$, $\delta_{0,1}^{a,b'}$, $\delta_{0,0}^{a,c'}$, $\delta_{0,0}^{a,d'}$, $\delta_{0,0}^{b,a'}$, $\delta_{0,1}^{b,a'}$, $\delta_{0,0}^{b,b'}$, $\delta_{0,1}^{b,b'}$ $\delta_{0,0}^{b,c'}$, and $\delta_{0,0}^{b,d'}$.

The quantities $\Delta_{i,j}^{v,w}$ are given by: $\Delta_{0,0}^{a,a'} = 0$, $\Delta_{0,1}^{a,a'} = 3$, $\Delta_{0,0}^{a,b'} = 1$, $\Delta_{0,1}^{a,b'} = 4$, $\Delta_{0,0}^{a,c'} = 1$, $\Delta_{0,0}^{a,d'} = 2$, $\Delta_{0,0}^{b,a'} = 0$, $\Delta_{0,1}^{b,a'} = 3$, $\Delta_{0,0}^{b,b'} = 1$, $\Delta_{0,1}^{b,b'} = 4$, $\Delta_{0,0}^{b,c'} = 1$ and $\Delta_{0,0}^{b,d'} = 2$. The function $F^C(\delta)$ is easily obtained by summing over $\delta_{i,j}^{v,w} \cdot \Delta_{i,j}^{v,w}$.

While:

$$F^D(\delta) = (1 - \delta_{0,0}^{a,a'} - \delta_{0,1}^{a,a'} - \delta_{0,0}^{a,b'} - \delta_{0,1}^{a,b'} - \delta_{0,0}^{a,c'} - \delta_{0,0}^{a,d'}) \cdot 1 + \ldots + (1 - \delta_{0,0}^{a,d'} - \delta_{0,0}^{b,d'}) \cdot 3$$

and:

$$F^-(\delta) = (\delta_{0,0}^{a,a'} + \delta_{0,1}^{a,a'} + \delta_{0,0}^{a,b'} + \delta_{0,1}^{a,b'} + \delta_{0,0}^{a,c'} + \delta_{0,0}^{a,d'}) \cdot 0 + \ldots + (\delta_{0,0}^{a,d'} + \delta_{0,0}^{b,d'}) \cdot 3$$

and:

$$F^S(\delta) = (\delta_{0,0}^{a,a'} + \delta_{0,1}^{a,a'}) \cdot 0 + (\delta_{0,0}^{a,b'} + \delta_{0,1}^{a,b'}) \cdot 0 + \ldots + \delta_{0,0}^{a,d'} \cdot 3 + \delta_{0,0}^{b,d'} \cdot 3$$

Lastly the constraints are:

$$\delta_{0,0}^{a,a'} + \delta_{0,1}^{a,a'} + \delta_{0,0}^{a,b'} + \delta_{0,1}^{a,b'} + \delta_{0,0}^{a,c'} + \delta_{0,0}^{a,d'} \leq 1$$

$$\delta_{0,0}^{b,a'} + \delta_{0,1}^{b,a'} + \delta_{0,0}^{b,b'} + \delta_{0,1}^{b,b'} + \delta_{0,0}^{b,c'} + \delta_{0,0}^{b,d'} \leq 1$$

$$\delta_{0,0}^{a,a'} + \delta_{0,1}^{a,a'} + \delta_{0,0}^{b,a'} + \delta_{0,1}^{b,a'} + \delta_{0,0}^{a,d'} + \delta_{0,0}^{b,d'} \leq 1$$

$$\delta_{0,0}^{a,b'} + \delta_{0,1}^{a,b'} + \delta_{0,0}^{b,b'} + \delta_{0,1}^{b,b'} + \delta_{0,0}^{a,d'} + \delta_{0,0}^{b,d'} \leq 1$$

$$\delta_{0,0}^{a,c'} + \delta_{0,0}^{b,c'} \leq 1$$

In this case there are many minimizing solutions. One is given by: $\delta_{0,1}^{a,a'} = \delta_{0,0}^{b,c'} = 1$ and all other variables equal to 0. This value of δ is feasible since the variables $\delta_{0,1}^{a,a'}$ and $\delta_{0,0}^{b,c'}$ never appear in the same constraint. This value of δ entails $F^C(\delta) = 3 + 1$, $F^D(\delta) = 2$, $F^-(\delta) = 0$ and $F^S(\delta) = 0$, and thus $d_E(T_d, T') = 6$.

Another solution can be obtained with: $\delta_{0,0}^{a,d'} = \delta_{0,0}^{b,c'} = 1$ and all other variables equal to 0. Also this value of δ is feasible since the variables $\delta_{0,0}^{a,d'}$ and $\delta_{0,0}^{b,c'}$ never appear in the same constraint. This value of δ entails $F^C(\delta) = 2 + 1$, $F^D(\delta) = w_{T'}(a') + w_{T'}(b') = 1 + 2$, $F^-(\delta) = ||sub_{T'}(d')|| = 3$ and $F^S(\delta) = d_E(sub_T(a), sub_{T'}(d')) = ||sub_{T'}(d')|| = 3$, and thus $d_E(T_d, T') = 3 + 3 - 3 + 3 = 6$.

Couple: (r_T, d') This and the other couples are left to the reader.

2.D Computing the Edit Distance: Numerical Simulations

In this last section, the feasibility of the algorithm presented in Section 2.C is assessed by means of some numerical simulations.

To get some concrete ideas of proper runtimes needed to calculate distances, we fix the number of leaves n and for 100 times the following procedure is repeated: generate two random samples of n points from the uniform distribution on a compact, real interval, take their single linkage hierarchical dendrograms (with weight function equal to the weight function w_T) and compare them with d_E. This whole pipeline is repeated for any integer n in the interval $[5, 20]$. In Figure 2.D.1 there are the average runtimes as a function of the

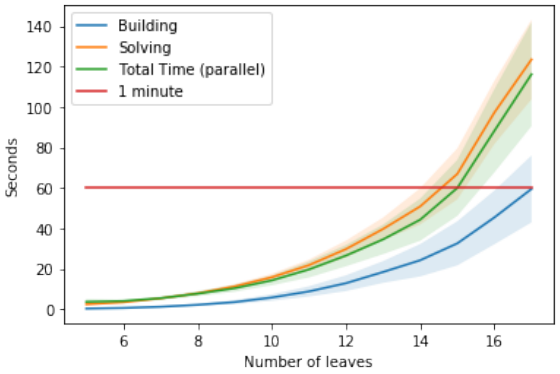


Figure 2.D.1: Graph of the computational times as function of the number of leaves. The curves represent running times to calculated d_E between couples of merge trees, averaged over 100 random couples of trees, with shaded regions including intervals of $+/-$ one standard deviation. "Building time" means the time spent by Python to setup the ILP problems. "Solving time" is the time used by the solver to solve the LIP problems. "Total time" is the time spent computing the distance using parallel computing of the ILP problems: both for the building and solving steps.

number of leaves of the involved binary trees. The standard deviations over the repetitions are also reported, which show a quite large band around the mean. The different curves in Figure 2.D.1 concern the portion of time effectively spent by the solver to compute the solution of the ILP problems, and the amount of time employed to setup such problems. All code is written in Python and thus this second part of the runtimes can likely be greatly reduced by using more performing programming languages. The green line of total time is computed parallelizing the **for** loop in Algorithm 1. Note that dendrograms with the same number of leaves may end up having different tree-structures and so different dimensions. This is the main reason for the big shaded regions around the mean. If the trees were aggregated by dimension, the standard deviation of runtimes would decrease. Nevertheless, in applications, the only thing one can reasonably control is the number of leaves (which is given by the number of minima in the function, the number of clusters in a dendrogram, etc.) and for this reason the trees are aggregated as in Figure 2.D.1.

The computations are carried out on a 2016 laptop with Intel(R) processor Core(TM) i7-6700HQ CPU @ 2.60GHz, 4 cores (8 logical) and 16 GB of RAM. The employed ILP solver is the freely available IBM CPLEX Optimization Studio 12.9.0.

3. A Locally Stable Edit Distance for Merge Trees

Abstract

In this chapter we define a novel edit distance for merge trees. Then we consider the metric space obtained and study the properties of such space obtaining completeness and compactness results, the existence of Frechét Means and local approximations by means of euclidean spaces. Whilst doing so, we explore the geodesic structure of the space of merge trees, stratifying trees by means of their dimension and decomposing geodesics along different strata. All these results contribute to understanding how the different strata interact, which is pivotal for the development of more refined tools and techniques for merge trees.

3.1 Introduction

Topological Data Analysis (TDA) is a particular set of techniques within the field of geometric data analysis which aim at including topological information into data analysis pipelines.

Topological information is usually understood in terms of generators of homology groups (Hatcher, 2000) with coefficients in some field. With *persistent homology* these generators are extracted along a filtration of topological spaces to capture the shape of the initial datum, typically a function or a point cloud, at "different resolutions" (Edelsbrunner and Harer, 2008). To proceed with the analysis, this ordered family of vector spaces is then represented with a topological summary. There are many different kinds of topological summaries such as persistence diagrams (Edelsbrunner et al., 2002), persistence images (Adams et al., 2017), persistence silhouettes (Chazal et al., 2015), and persistence landscapes (Bubenik, 2015). Each of these summaries live in a space with different properties and purposes: for instance persistence diagrams are highly interpretable and live in a metric space, persistence landscapes are embedded in a linear space of functions, but the embedding is not closed under linear combinations, persistence images are instead obtained as vectors in $\mathbb{R}^n$, making them suitable for many Machine Learning techniques.

Along with the aforementioned summaries, there are also tree-shaped objects called merge trees. Merge trees arise naturally within the framework of TDA when dealing with zero dimensional homology groups, as they capture the merging structure of path connected components along a filtration of topological spaces. Originally, such objects stem out of Morse Theory (Milnor, 2016) as a topological summary related to Reeb graphs (Shinagawa et al., 1991b; Biasotti et al., 2008) and are frequently used for data visualization purposes (wu and Zhang, 2013; Bock et al., 2017). Analogously, other different but related kinds of trees like hierarchical clustering dendrograms (Murtagh and Contreras, 2017) or phylogenetic trees (Felsenstein and Felenstein, 2004) have also been used extensively in statistics and biology to infer information about a fixed set of labels. The uprising of TDA, however, has propelled works aiming at using trees and Reeb graphs as topological summaries in data analysis contexts and thus developing metrics and frameworks to analyze populations of such objects (Beketayev et al., 2014; Morozov et al., 2013; Gasparovic

et al., 2019; Touli, 2020; Sridharamurthy et al., 2020; Wetzels et al., 2022; Di Fabio and Landi, 2016; Bauer et al., 2020).

Returning to the typical TDA pipeline, once a topological representation suitable for the analysis is chosen the tools that can be employed are a direct consequence of the properties of the space the summaries live in, as for any other data analysis situation. For instance, many statistical techniques rely on the linear structure of Euclidean spaces: typical examples are linear regression and principal components analysis (PCA). However, there are many examples of analyses carried out on data not lying (at least naively) in $\mathbb{R}^n$ or other vector spaces. Quotient spaces of functions up to reparametrization, point clouds up to isometries, probability distributions, matrices up to some kind of base change etc. all require complex mathematical frameworks to be treated, since even things like moving from one point to another must be carefully defined. For these reasons there has been a lot of effort in generalizing statistical tools for data living outside Euclidean spaces. The most commonly considered generalization is the case of data being sampled from Riemannian manifolds - often Lie Groups - (Pennec, 2006; Pennec et al., 2019; Pennec, 2018; Huckemann and Eltzner, 2018; Huckemann et al., 2010; Jung et al., 2012; Fletcher, 2013; Patrangenaru and Ellingson, 2015), where the richness of the Riemannian structure is fundamental to develop techniques and results. The situation becomes further challenging when one cannot build a differential or even a topological structure which falls into the realm of well-known and deeply studied geometrical objects like manifolds. In this case ad-hoc, meaningful tools and definitions must be carefully obtained in order to be able to work in such spaces (Mileyko et al., 2011; Leygonie et al., 2021; Calissano et al., 2020; Miller et al., 2015; Garba et al., 2021).

Related Works

The works that have been dealing with the specific topic of merge trees can be divided into two groups: the first group is more focused on the definition of a suitable metric structure on merge trees and the second is more focused on the properties of merge trees and their relationships with persistence diagrams. The first group, in turns, splits into works dealing with the interleaving distance between merge trees and other metrics with very strong - *universal* - theoretical properties (Beketayev et al., 2014; Morozov et al., 2013; Touli and Wang, 2018; Gasparovic et al., 2019; Touli, 2020; Curry et al., 2022; Cardona et al., 2021), and metrics which are more focused on the computational efficiency (Sridharamurthy et al., 2020; Wetzels et al., 2022; Pont et al., 2022) at the cost of sacrificing stability properties and trying to mitigate the resulting problems with pre-processing and other computational solutions. The second group of articles, instead, investigates structural properties of merge trees, answering questions like how many merge trees share the same persistence diagram (Kanari et al., 2020; Curry et al., 2021), defining function spaces on merge trees - Chapter 2 - and obtaining other structures which lie in between persistence diagrams and merge trees (Elkin and Kurlin, 2020).

Main Contributions

The aim of this chapter is to propose a novel metric for merge trees and to investigate some geometric properties of the resulting metric space of merge trees. Such investigation is intended as a first step into the development of statistical tools to analyze sets of such objects, in analogy with what has already been done for persistence diagrams (Mileyko et al., 2011; Turner et al., 2014; Bubenik and Wagner, 2020; Che et al., 2021) and other kinds of trees (Billera et al., 2001; Miller et al., 2015; Garba et al., 2021).

The metric which is proposed is based on a more general edit distance between weighted (in a broad sense) trees which is developed in Chapter 2: such chapter contains the main

theoretic results which allow us to define the new distance and the algorithm which is used to compute it. This edit distance is very different from other edit distances which have been proposed for merge trees for two reasons: 1) it does not measure differences between trees in terms of persistence pairs (as opposed to Sridharamurthy et al. (2020) and Pont et al. (2022)) but in terms of edge length, resulting in a different way in which variability between trees is captured; 2) the edit operations which characterize this distance are more flexible than the usual ones, with some operations being a complete novelty with respect to the existing literature on general edit distances. This on one side causes a higher computational cost, but allows for suitable stability properties, which are investigated and proved in Chapter 4.

As already mentioned, the general aim of our geometric investigation is to obtain a better understanding of the local behaviour of the metric space of merge trees, to better grasp the mathematical and statistical possibilities offered by that space. In particular, we focus our geometric analysis on three directions. First we obtain a series of topological results which lead to the definition of 0-dimensional summaries of empirical distributions of merge trees, via their *Frechét means* - sometimes called *Karcher means* - (Karcher, 1977): these objects generalize the widely used euclidean means and can be seen as the most basic statistics employed in data analysis, but that can provide important results even outside the euclidean context (Davis, 2008; Pennec, 2018). Second, we start studying the metric structure of the space of merge trees, with results on the local uniqueness of its geodesic paths. Third we provide some tools to have a local linear approximation of the space of merge trees, via a decomposition of its geodesics along different strata of the space.

Outline

This chapter is organized as follows. Section 3.2, Section 3.3 and Section 3.4 contain the preliminary definitions needed in the chapter, which are collected from previous works in the field. In the latter one, in particular, we review the definition of the edit distance between weighted trees, which we use in Section 3.5 to define a metric structure for merge trees which is therein compared with other established metrics for merge trees. In Section 3.6 we discuss some geometric properties of the metric space obtained, with particular focus on geodesic structure, compact sets, Frechét Means and local approximations of the space of trees via some euclidean space. In Section 3.7 we draw some conclusions.

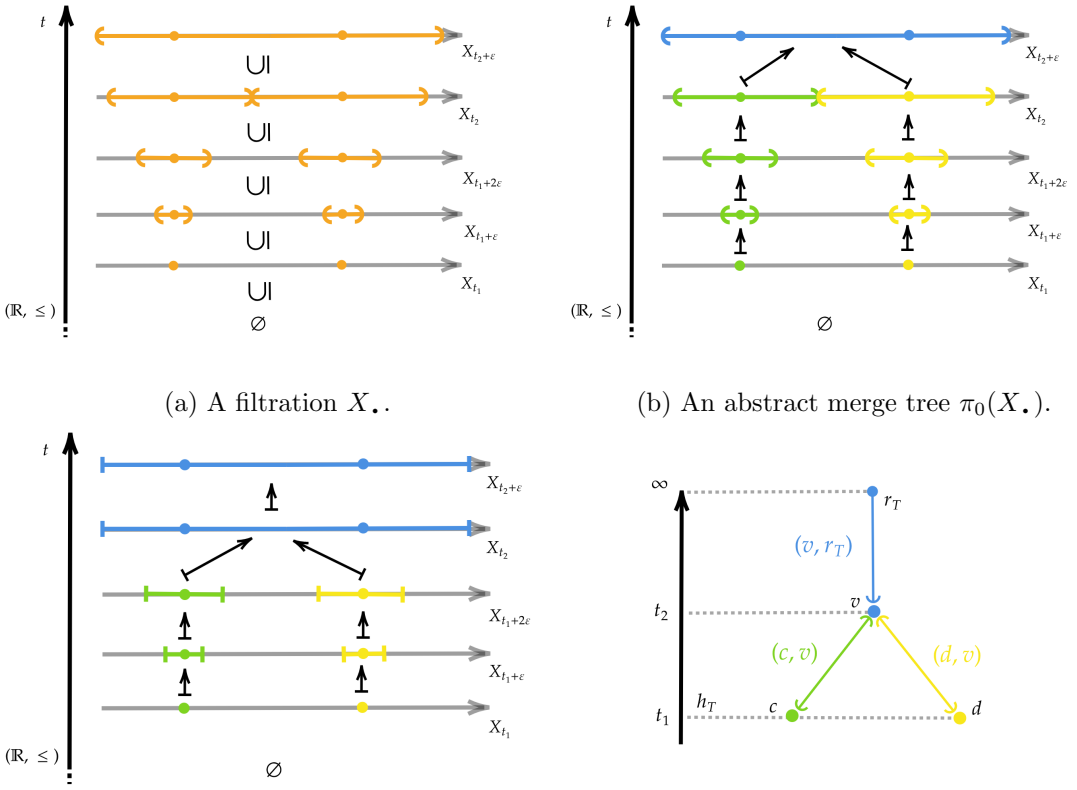
3.2 Preliminary Definitions - Merge Trees

This is the first of three preliminary sections to our work. Here, following Patel (2018), Curry et al. (2022) and Chapter 2 we introduce most of the objects we need for our investigation, reporting definitions which are well established in TDA. The other preliminary sections will focus on recalling metric geometry terminology/results and the edit distance defined in Chapter 2.

Figure 3.2.1 illustrates some of the objects we introduce in this section.

Definition 3.1 (Curry et al. (2022))**.** *A filtration of topological spaces is a (covariant) functor* $X_\bullet : \mathbb{R} \to \mathrm{Top}$ *from the poset* $(\mathbb{R}, \leq)$ *to* Top*, the category of topological spaces with continuous functions, such that:* $X_t \to X_{t'}$*, for* $t < t'$*, are injective maps.*

Example Given a real valued function $f : X \to \mathbb{R}$ the *sublevel set* filtration is given by $X_t = f^{-1}((-\infty, t])$ and $X_{t<t'} = i : f^{-1}((-\infty, t]) \hookrightarrow f^{-1}((-\infty, t'])$.



(a) A filtration $X_\bullet$.

(b) An abstract merge tree $\pi_0(X_\bullet)$.

(c) A regular abstract merge tree $\pi_0(X_\bullet)$.

(d) The merge tree $\mathcal{M}(\pi_0(X_\bullet))$ with $\pi_0(X_\bullet)$ as in Figure 3.2.1c.

Figure 3.2.1: On the first line we see an example of a filtration along with its abstract merge tree. in In the bottom line there are a regular abstract merge tree and the associated merge tree. The colors are used throughout the plots to highlight the relationships between the different objects.

Example Given a finite set $C \subset \mathbb{R}^n$ its the *Céch* filtration is given by $X_t = \bigcup_{c \in C} B_t(c)$. With $B_t(c) = \{x \in \mathbb{R}^n \mid \parallel c - x \parallel < t\}$. As before: $X_{t<t'} = i : \bigcup_{c \in C} B_t(c) \hookrightarrow \bigcup_{c \in C} B_{t'}(c)$.

Given a filtration $X_\bullet$ we can compose it with the functor π_0 sending each topological space into the set of its path connected components. We recall that, according to standard topological notation, $\pi_0(X)$ is the set of the path connected components of X and, given a continuous functions $q : X \to Y$, $\pi_0(q) : \pi_0(X) \to \pi_0(Y)$ is defined as:

$$U \mapsto V \text{ such that } q(U) \subset V.$$

Definition 3.2 (Carlsson and Mémoli (2013); Curry (2018))**.** *A persistent set is a functor $S : \mathbb{R} \to$ Sets. In particular, given a filtration of topological spaces $X_\bullet$, the persistent set of components of $X_\bullet$ is $\pi_0 \circ X_\bullet$. A one dimensional persistent module is a functor $S : \mathbb{R} \to \mathrm{Vec}_{\mathbb{K}}$ with values in the category of vector spaces $\mathrm{Vec}_{\mathbb{K}}$.*

By endowing a persistent set with the discrete topology, every persistence set can be seen as the persistence set of components of a filtration. Thus a general persistent set S can be written as $\pi_0(X_\bullet)$ for some filtration $X_\bullet$.

Based on the notion of constructible persistent sets found in Patel (2018) and Curry et al. (2022) one then builds the following objects.

Definition 3.3 (Chapter 2)**.** *An abstract merge tree is a persistent set $S : \mathbb{R} \to$ Sets such that there is a finite collection of real numbers $\{t_1 < t_2 < \ldots < t_n\}$ which satisfy:*

- *$S(t) = \emptyset$ for all $t < t_1$;*
- *$S(t) = \{\star\}$ for all $t > t_n$;*
- *if $t, t' \in (t_i, t_{i+1})$, with $t < t'$, then $S(t < t')$ is bijective.*

The values $\{t_1 < t_2 < \ldots < t_n\}$ are called critical values of the tree and there is always a minimal set of critical values - Chapter 2. We always assume to be working with such minimal set.

If $S(t)$ is always a finite set, S is a finite abstract merge tree.

Consider an abstract merge tree $\pi_0(X_\bullet)$ and let $t_1 < t_2 < \ldots < t_n$ be its (minimal set of) critical values and let $i_t^{t'} := X_{t \leq t'} : X_t \to X_{t'}$. Take $\varepsilon > 0$ small. We have that at least one between $\pi_0(i_{t_j - \varepsilon}^{t_j})$ and $\pi_0(i_{t_j}^{t_j + \varepsilon})$ is not bijective. So we have the following definition.

Definition 3.4 (Chapter 2)**.** *An abstract merge tree $\pi_0(X_\bullet)$ is said to be regular if $\pi_0(i_{t_j}^{t_j+\varepsilon})$ is bijective for every critical value t_j and for every $\varepsilon > 0$ small enough.*

Assumption 3.5. *From now on we consider only regular abstract merge trees. In Chapter 2 it is shown that this choice is non-restrictive.*

Now we introduce merge trees with the approach found in Chapter 2 - from which we take the upcoming definitions. A combinatorial object, called tree structure, is introduced, and we add to it some height values with a function. Similar approaches can be found in most of the scientific literature dealing with such topics (Gasparovic et al., 2019; Sridharamurthy et al., 2020; Wetzels et al., 2022; Pont et al., 2022).

Definition 3.6. *A tree structure T is given by a set of vertices V_T and a set of edges $E_T \subset V_T \times V_T$ which form a connected rooted acyclic graph. We indicate the root of the tree with r_T. We say that T is finite if V_T is finite. The order of a vertex $v \in V_T$ is the number of edges which have that vertex as one of the extremes, and is called $ord_T(v)$. Any vertex with an edge connecting it to the root is its child and the root is its father: this is the first step of a recursion which defines the father and children relationship for all vertices in V_T. The vertices with no children are called leaves or taxa and are collected in the set L_T. The relation child $<$ father generates a partial order on V_T. The edges in E_T are identified in the form of ordered couples (a, b) with $a < b$. A subtree of a vertex v, called $sub_T(v)$, is the tree structure whose set of vertices is $\{x \in V_T | x \leq v\}$.*

Given a tree structure T, identifying an edge (v, v') with its lower vertex v, gives a bijection between $V_T - \{r_T\}$ and E_T, that is $E_T \cong V_T - \{r_T\}$ as sets. Given this bijection, we often use E_T to indicate the vertices $v \in V_T - \{r_T\}$, to simplify the notation.

To identify merge trees independently of their vertex set the following isomorphism classes are introduced.

Definition 3.7. *Two tree structures T and T' are isomorphic if exists a bijection $\eta : V_T \to V_{T'}$ that induces a bijection between the edges sets E_T and $E_{T'}$: $(a, b) \mapsto (\eta(a), \eta(b))$. Such η is an isomorphism of tree structures.*

Then, we can give the definition of a merge tree.

Definition 3.8. *A merge tree is a finite tree structure T with a monotone increasing height function $h_T : V_T \to \mathbb{R} \cup \{+\infty\}$ and such that 1) $ord_T(r_T) = 1$ 2) $h_T(r_T) = +\infty$ 3) $h_T(v) \in \mathbb{R}$ for every $v < r_T$. The set of all merge trees is called $\mathcal{MT}$.*

Two merge trees (T, h_T) and $(T', h_{T'})$ are isomorphic if T and T' are isomorphic as tree structures and the isomorphism $\eta : V_T \to V_{T'}$ is such that $h_T = h_{T'} \circ \eta$. Such η is an isomorphism of merge trees. We use the notation $(T, h_T) \cong (T', h_{T'})$.

With some slight abuse of notation we set $\max h_T = \max_{v \in V_T \,|\, v < r_T} h_T(v)$ and $\arg\max h_T = \max\{v \in V_T \,|\, v < r_T\}$. Note that, given (T, h_T) merge tree, there is only one edge of the form (v, r_T) and we have $v = \arg\max h_T$.

Definition 3.9. *Given a tree structure T, we can eliminate an order two vertex, connecting the two adjacent edges which arrive and depart from it. Suppose we have two edges $e = (v_1, v_2)$ and $e' = (v_2, v_3)$, with $v_1 < v_2 < v_3$. And suppose v_2 is of order two. Then, we can remove v_2 and merge e and e' into a new edge $e'' = (v_1, v_3)$. This operation is called the* ghosting *of the vertex. Its inverse transformation is called the* splitting *of an edge.*

Consider a merge tree (T, h_T) and obtain T' by ghosting a vertex of T. Then $V_{T'} \subset V_T$ and thus we can define $h_{T'} := h_{T \,|\, V_{T'}}$.

Now we can state the following definition.

Definition 3.10. *Merge trees are equal up to order 2 vertices if they become isomorphic after applying a finite number of ghostings or splittings. We write $(T, h_T) \cong_2 (T', h_{T'})$.*

We report a result summarizing the relationship between abstract merge trees and merge trees. The main consequence of such result is that merge trees considered up to order 2 vertices are an appropriate discrete tool to represent the information contained in regular abstract merge trees. Figure 3.2.1b and Figure 3.2.1d can help the reader going through the following proposition.

Proposition 3.11 (Chapter 2)**.** *The following hold:*

1. *we can associate a merge tree without order 2 vertices $\mathcal{M}(\pi_0(X_\bullet))$ to any regular abstract merge tree $\pi_0(X_\bullet)$;*

2. *we can associate a regular abstract merge tree $\mathcal{F}((T, h_T))$ to any merge tree (T, h_T). Moreover, we have $\mathcal{M}(\mathcal{F}((T, h_T))) \cong_2 (T, h_T)$;*

3. *given two merge trees (T, h_T) and $(T', h_{T'})$, we have $\mathcal{F}((T, h_T)) \cong \mathcal{F}((T, h_T))$ if and only if $(T, h_T) \cong_2 (T', h_{T'})$.*

Remark 3.12. *The explicit construction of $\mathcal{M}$ and $\mathcal{F}$ can be found in the proof of Proposition 4 in Chapter 2.*

3.3 Preliminary Definitions - Metric Spaces

Following Burago et al. (2022), we give a brief overview of the well-established metric geometry definitions we need in the present work.

Definition 3.13. *Let X be an arbitrary set. A function $d : X \times X \to \mathbb{R}$ is a (finite) pseudo metric if for all $x, y, z \in X$ we have:*

1. $d(x, x) = 0$

2. $d(x, y) = d(y, x)$

3. $d(x, y) \leq d(x, z) + d(z, y)$.

The space (X, d) is called a pseudo metric space.

Given a pseudo metric d on X, if for all $x, y \in X$, $x \neq y$, we have $d(x, y) > 0$ then d is called a metric or a distance and (X, d) is a metric space.

Proposition 3.14 (Proposition 1.1.5 Burago et al. (2022))**.** *For a pseudo metric space (X, d), $x \sim y$ iff $d(x, y) = 0$ is an equivalence relationship and the quotient space $(X, d)/\sim$ is a metric space.*

Definition 3.15. *Consider X, Y pseudo metric spaces. A function $f : X \to Y$ is an isometric embedding if it is injective and $d(x, y) = d(f(x), f(y))$. If f is also bijective then it is an isometry or and isometric isomorphism.*

Definition 3.16. *A pseudo metric d on X induces the topology generated by the open balls $B_\varepsilon(x) := \{y \in X \,|\, d(x, y) < \varepsilon\}$.*

Having a topology enables us to talk about continuity properties and in particular we can consider continuous curves $\gamma : [a, b] \to X$.

Definition 3.17. *Given a continuous curve $\gamma : [a, b] \to X$ with (X, d) a pseudo metric space, one can define its length as:*

$$L(\gamma) = \sup_{a=t_0<\ldots<t_{n+1}=b} \sum_{i=0}^{n} d(\gamma(t_i), \gamma(t_{i+1}))$$

If $L(\gamma)$ is finite then γ is rectifiable.

Definition 3.18. *A pseudo metric d on X is intrinsic if $d(x, y) = \inf\{L(\gamma : [a, b] \to X) \,|\, \gamma$ continuous $, \gamma(a) = x, \gamma(b) = y\}$. The space (X, d) is then called a length space.*

Definition 3.19. *Given a length space (X, d) a geodesic is a continuous curve $\gamma : [a, b] \to X$ such that for every t there is an interval $[a', b']$, $a' < t < b'$, such that $d(\gamma(a'), \gamma(b')) = L(\gamma_{|[a',b']})$. Up to reparametrizing γ this can be written as $d(\gamma(t'), \gamma(t'')) = \lambda \, | \, t'' - t' \, |$ for some $\lambda > 0$ and for every $t, t'' \in [a', b']$. If $[a', b'] = [a, b]$ then γ is a minimal geodesic. If there is always a minimal geodesic connecting two points, X is a geodesic space.*

Before proceeding we recall some last topological and metric definitions we use in the following.

Definition 3.20. *Consider a pseudo metric space (X, d).*

- *X is contractible if there is $x \in X$ with $F : [0, 1] \times X \to X$ continuous such that $F(0, p) = p$ and $F(1, p) = x$ for every $p \in X$;*
- *X is totally bounded is for every $\varepsilon > 0$ there is a finite collection of points $x_1, \ldots, x_N$ such that $X = \bigcup_{i=1}^{N} B_\varepsilon(x_i)$;*
- *X is complete if every Cauchy sequence converges; in pseudo metric spaces, a set which is complete and totally bounded is also compact and viceversa (Theorem 1.6.5. in Burago et al. (2022));*
- *X sequentially compact if every sequence in X admits a convergent subsequence. In pseudo metric spaces it is equivalent to compactness (Theorem 1.6.5. in Burago et al. (2022));*
- *X is locally compact if for every point $x \in X$ there is an open set U and a compact set K such that $x \in U$ and $U \subset K$.*

3.4 Weighted Trees Edit Distance

In this section we introduce the last pieces of notation we need to define the merge tree edit distance. In Section 3.4.1 report how Chapter 2 builds a metric for *weighted trees* and in Section 3.4.2 we recall from Chapter 2 a fundamental combinatorial object which will be used throughout the chapter.

3.4.1 Weighted Trees and Edits

Definition 3.21. *A tree structure T with a weight function $w_T : E_T \to \mathbb{R}_{>0}$ is called weighted tree. The set of all weighted trees is called $(\mathcal{T}, \mathbb{R}_{\geq 0})$.*

Given a weighted tree (T, w_T) we can modify its edges $E_T \cong V_T - \{r_T\}$ with a sequence of the following edit operations:

- we call *shrinking* of a vertex/edge a change of the weight function. The new weight function must be equal to the previous one on all vertices, apart from the "shrunk" one. In other words, for an edge e, this means changing the value $w_T(e)$ with another non zero value in W.
- A *deletion* is an edit with which a vertex/edge is deleted from the dendrogram. Consider an edge (v_1, v_2). The result of deleting v_1 is a new tree structure, with the same vertices a part from v_1 (the smaller one), and with the father of the deleted vertex which gains all of its children. The inverse of the deletion is the *insertion* of an edge along with its lower vertex. We can insert an edge at a vertex v specifying the name of the new child of v, the children of the newly added vertex (that can be either none, or any portion of the children of v), and the value of the weight function on the new edge.

- Lastly, we can remove or add order two vertices via the ghosting and splitting edits, which have already been defined in Definition 3.9.

The set of all weighted trees considered up to ghostings and splittings is called $(\mathcal{T}_2, \mathbb{R}_{\geq 0})$.

A weighted tree T can be edited to obtain another weighted tree, on which one can apply a new edit to obtain a third tree and so on. Any finite composition of edits is called *edit path*. See Figure 3.4.1 for an example of an edit path. The set of finite edit paths between T and T' is called $\gamma \in \Gamma(T, T')$.

The cost of the edit operations is defined as follows:

- the cost of shrinking an edge is equal to the absolute value of the difference of the two weights;
- for any deletion/insertion, the cost is equal to the weight of the edge deleted/inserted;
- the cost of ghosting is zero.

The cost of an edit path is the sum of the costs of its edit operations. Putting all the pieces together, the edit distance d_E between weighted trees is defined as:

$$d_E(T, T') = \inf_{\gamma \in \Gamma(T,T')} cost(\gamma)$$

where $\Gamma(T, T')$ indicates the set of edit paths which start in T and end in T'. In Chapter 2 it is proved that d_E is a metric on the space of weighted trees considered up to order two vertices.

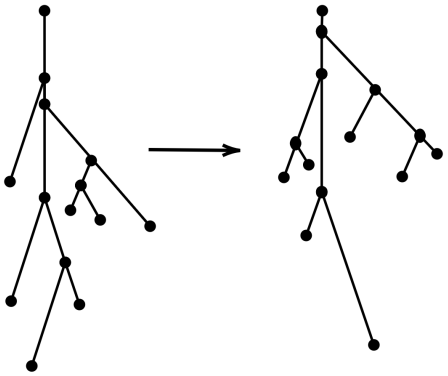
Remark 3.22. *It would be natural to try to define a family of metrics indexed by integers $p \geq 1$ by saying that the costs of an edit path the p-th root of sum of the costs of the edit operations to the p-th power. But now we can easily see that for any $p > 1$ this has no hope of being a meaningful pseudo metric for weighted trees. In fact, consider the case of a weighted tree made by two vertices and one edge with weight 1. The cost of shrinking the p-metric would be $\| 1 \|_p = 1$. At the same time one can split it in half with 0 cost and the cost of shrinking this other tree would be $\| (1/2, 1/2) \|_p < 1$. Splitting the segment again and again will make its shrinking cost go to 0. In other words all weighted trees, if considered up to order 2 vertices, would be at distance zero from the tree with no branches.*

3.4.2 Mappings

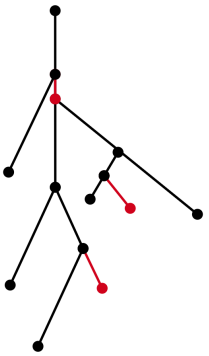
Given an edit path between two weighted trees, its cost is often invariant up to many permutations of the edits. To better work in such environment, we start considering paths up to some permutation of the edits. The objects called *mappings*, as defined in Chapter 2, help us in doing this, as well as making the metric d_E more tractable. For this reason now we report their definition. As in Chapter 2, "D" and "G" are be used to indicate "deletion" and "ghosting". Some novel notation is established and it is highlighted via Remark 3.23 to help the reader.

A *mapping* between T and T' is a set $M \subset (E_T \cup \{"D", "G"\}) \times (E_{T'} \cup \{"D", "G"\})$ satisfying:

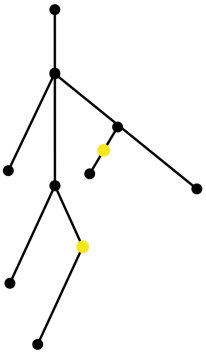
(M1) consider the projection of the Cartesian product $(E_T \cup \{"D", "G"\}) \times (E_{T'} \cup \{"D", "G"\}) \to (E_T \cup \{"D", "G"\})$; we can restrict this map to M obtaining $\pi_T : M \to (E_T \cup \{"D", "G"\})$. The maps π_T and $\pi_{T'}$ are surjective on $E_T \subset (E_T \cup \{"D", "G"\})$ and $E_{T'} \subset (E_{T'} \cup \{"D", "G"\})$ respectively;



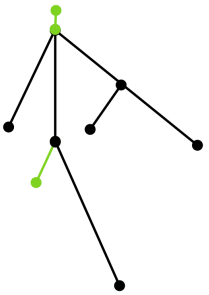
(a) Starting and target weighted trees.



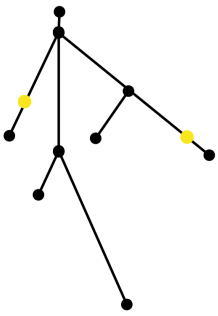
(b) Deletions in red.



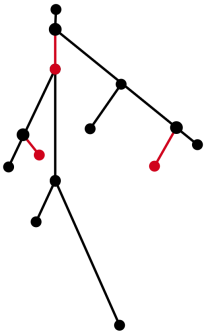
(c) Ghostings in yellow.



(d) Shrinkings in green.



(e) Splittings in yellow.



(f) Insertions in red.

Figure 3.4.1: (b)→(e) form an edit path made from the left weighted tree in Figure 3.4.1a. Each time the edges involved in the editing are highlighted with different colors. In the following plot such vertices return black. This edit path can clearly be represented with a mapping - Section 3.4.2 - made by couples $(v, "D")$ for all the red vertices in Figure 3.4.1b, $(v, "G")$ for all the yellow vertices in Figure 3.4.1c, (v, w) for all the vertices associated via the green color in Figure 3.4.1d, $("G", w)$ for all the yellow vertices in Figure 3.4.1c and $("D", w)$ for all the red vertices in Figure 3.4.1f.

(M2) π_T and $\pi_{T'}$ are injective;

(M3) $M \cap (V_T \times V_{T'})$ is such that, given (a, b) and $(c, d) \in M \cap (V_T \times V_{T'})$, $a > c$, if and only if $b > d$;

(M4) if $(a, "G")$ (or $("G", a)$) is in M, let $child(a) = \{b_1, .., b_n\}$. Then there is one and only one i such that for all $j \neq i$, for all $x \in V_{sub(b_j)}$, we have $(x, "D") \in M$ (respectively $("D", x)$); and there is one and only one c such that $c = \max\{x' \in sub(b_i) \,|\, (x', y) \in M$ for any $y \in V_{T'}\}$. In other words, if $(a, "G") \in M$, then a becomes an order 2 vertex after applying all the deletions induced by $(b, "D") \in M$.

We call $\mathrm{Mapp}(T, T')$ the set of all mappings between T and T'.

As in Chapter 2, we use the properties of $M \in \mathrm{Mapp}(T, T')$ to parametrize a set of edit paths starting from T and ending in T', which are collected under the name γ_M.

- γ_d^T a path made by the deletions to be done in T, that is, the couples $(v, "D")$, executed in any order. So we obtain $T_d^M = \gamma_d^T(T)$, which, instead, is well defined and not depending on the order of the deletions.
- One then proceeds with ghosting all the vertices $(v, "G")$ in M, in any order, getting a path γ_g^T and the dendrogram $T_M := \gamma_g^T \circ \gamma_d^T(T)$.
- Since all the remaining points in M are couples, the two dendrograms T'_M (defined in the same way as T_M, but starting from T') and T_M must be isomorphic as tree structures. This is guaranteed by the properties of M. So one can shrink T_M onto T'_M, and the composition of the shrinkings, executed in any order is an edit path γ_s^T.

By definition:

$$\gamma_s^T \circ \gamma_g^T \circ \gamma_d^T(T) = T'_M,$$

and:

$$(\gamma_d^{T'})^{-1} \circ (\gamma_g^{T'})^{-1} \circ \gamma_s^T \circ \gamma_g^T \circ \gamma_d^T(T) = T'$$

.

where the inverse of an edit path is thought as the composition of the inverses of the single edit operations, taken in the inverse order.

Lastly, we call γ_M the set of all possible edit paths:

$$(\gamma_d^{T'})^{-1} \circ (\gamma_g^{T'})^{-1} \circ \gamma_s^T \circ \gamma_g^T \circ \gamma_d^T.$$

obtained by changing the order in which the edit operations are executed inside γ_d, γ_g and γ_s. Observe that, even if γ_M is a set of paths, its cost is well defined:

$$cost(M) := cost(\gamma_M) = cost(\gamma_d^T) + cost(\gamma_s^T) + cost(\gamma_d^{T'}).$$

Remark 3.23. *For notational convenience, we collect in M_D all the deletions of M, in M_G all the ghostings and in $C(M)$ all the shrinking edits.*

See Figure 3.4.1 for an example of a mapping between weighted trees. We conclude this section by recalling that Chapter 2 proves that given two weighted trees T and T', for every finite edit path $\gamma \in \Gamma(T, T')$, there exists a mapping $M \in \mathrm{Mapp}(T, T')$ such that $cost(M) \leq cost(\gamma)$.

Lastly, consider $M_2(T, T') \subset \mathrm{Mapp}(T, T')$ defined as follows.

Definition 3.24 (Chapter 2)**.** *A mapping $M \in \mathrm{Mapp}(T, T')$ has maximal ghostings if $(v, "G") \in M$ if and only if v is of order 2 after the deletions in T and, similarly $("G", w) \in M$ if and only if w is of order 2 after the deletions in T'.*

A mapping $M \in \mathrm{Mapp}(T, T')$ has minimal deletions if $(v, "D") \in M$ only if v is not of order 2 after applying all the other deletions in T and, similarly, $("D", w) \in M$ only if w is not of order 2 after applying all the other deletions in T'.

We collect all mappings with maximal ghostings and minimal deletions in the set $M_2(T, T')$.

Lemma 3.25 (Chapter 2)**.**

$$\min\{cost(M) \,|\, M \in \mathrm{Mapp}(T, T')\} = \min\{cost(M) \,|\, M \in M_2(T, T')\}$$

3.5 Merge Trees Edit Distance

In this section we finally exploit the notation established in the previous parts of the chapter and results obtained in Chapter 2 to obtain a (pseudo) metric for merge trees.

3.5.1 Truncating Merge Trees

The aim of this part of the chapter is to bridge between merge trees and weighted trees, in order to induce a metric on merge trees by means of the edit distance defined in Section 3.4. the general idea is that we want to truncate the edge going to infinity of a merge tree to obtain a weighted tree. Clearly, many details need to be taken care of, as the metric needs to be well defined and the practical consequences of any truncation process need to be formally addressed.

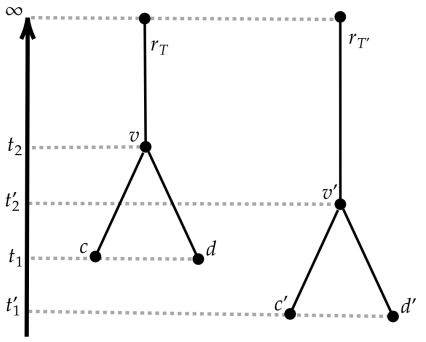
Starting from a merge tree (T, h_T) it is quite natural to turn the height function h_T into a weight function w_T via the rule $w_T((v, v')) := h_T(v') - h_T(v)$. The monotonicity of h_T guarantees that w_T take values in $\mathbb{R}_{>0}$. However, we clearly have an issue with the edge (v, r_T) as $h_T(r_T) = +\infty$. To solve this issue we need some novel tools - see Figure 3.5.1. First consider the set of merge trees $\mathcal{MT}$ and build the subset $\mathcal{MT}_K := \{(T, h_T) \in \mathcal{MT} \,|\, \max h_T \leq K\}$, for some $K \in \mathbb{R}$. Then define the truncation operator as follows:

$$\begin{aligned} \mathrm{Tr}_K : \mathcal{MT}_K &\longrightarrow (\mathcal{T}, \mathbb{R}_{\geq 0}) \\ (T, h_T) &\mapsto (T, h_K) \mapsto (T, w_T) \end{aligned} \tag{3.1}$$

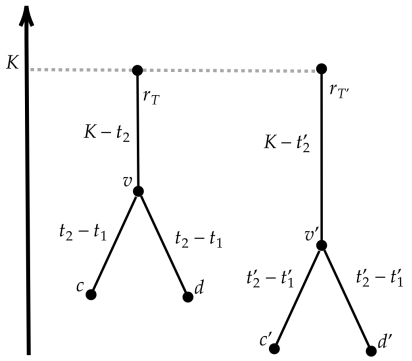
with $h_K(v) = h_T(v)$ if $v < r_T$ and $h_K(r_T) = K$. Then we set $w_T((v, v')) = h_K(v') - h_K(v)$. To avoid $w_T((v, r_T)) = 0$, if $\max h_T = K$, we take $(T, h_T) \mapsto (T', h_{T'}) \mapsto (T', w_{T'})$ with T' obtained from T via the removal of r_T from V_T and (v, r_T) from E_T. The map $h_{T'}$ is $h_{T\,|\,V_{T'}}$. Then $w_{T'}((v, v')) := h_{T'}(v') - h_{T'}(v)$.

In other words with Tr_K we are fixing some height K, truncating the edge $(\arg\max h_T, r_T)$ at height K, and then obtaining a positively weighted tree, as in Figure 3.5.1. To go back with $(\mathrm{Tr}_K)^{-1}$ we "hang" a weighted tree at height K and extend the edge (v, r_T) to $+\infty$. We formally state these ideas in the following proposition. We leave the details of the proof to the reader.

Proposition 3.26. *The operator $\mathrm{Tr}_K : \mathcal{MT}_K \to (\mathcal{T}, \mathbb{R}_{\geq 0})$ can be inverted via $(T, w_T) \mapsto \mathrm{Tr}_K^{-1}((T, w_T)) = (T', h_{T'})$ with the following notation: the tree structure T' is obtained from T via adding $r_{T'}$ to V_T and $(r_T, r_{T'})$ to E_T and ghosting r_T if it becomes an order 2 vertex. Then we have $h_{T'}(r_T) = K$ (if it is not ghosted, i.e. if it is of order greater than 2), $h_{T'}(r_{T'}) = +\infty$ and, recursively, for $(v, v') \in E_T$, $h_{T'}(v) = h_{T'}(v') - w_T((v, v'))$. Clearly $\mathrm{Tr}_K^{-1}(\mathrm{Tr}_K((T, h_T))) \cong (T, h_T)$. Thus $\mathcal{MT}_K \cong (\mathcal{T}, \mathbb{R}_{\geq 0})$ as sets, for every $K \in \mathbb{R}$. Moreover $\mathrm{Tr}_K((T, h_T)) \sim_2 \mathrm{Tr}_K((T', h_{T'}))$ if and only if $(T, h_T) \sim_2 (T', h_{T'})$.*



(a) Two merge trees T and T'.



(b) A graphical representation of the weighted trees $\mathrm{Tr}_K(T)$ and $\mathrm{Tr}_K(T')$. Roughly speaking, the merge trees T and T' are recovered sending the two roots back to infinity.

Figure 3.5.1: A graphical representation of the truncation operator.

3.5.2 Edit Distance For Merge Trees

Consider $(T, h_T), (T', h_{T'}) \in \mathcal{MT}$ and select K such that $(T, h_T), (T', h_{T'}) \in \mathcal{MT}_K$. Let $(G, w_G) = \mathrm{Tr}_K((T, h_T))$ and $(G', w_{G'}) = \mathrm{Tr}_K((T', h_{T'}))$. Lastly, set:

$$d_K((T, h_T), (T', h_{T'})) := d_E((G, w_G), (G', w_{G'})).$$

Despite being a promising and natural pseudo metric, d_K is not defined on the whole $\mathcal{MT}$ and, on top of that, it is not clear if its value depends on the K we choose. Leveraging on the following result taken from Chapter 2 we solve these issues and prove that we can use d_K to define a metric on $\mathcal{MT}$.

Proposition 3.27 (Chapter 2)**.** *Take (T, w_T) and $(T', w_{T'})$. If r_T and $r_{T'}$ are of order 1 and there is a splitting $\{(v, r_T)\} \to \{(v, v'), (v', r_T)\}$ and $\{(w, r_{T'})\} \to \{(w, w'), (w', r_{T'})\}$ giving the weighted trees (G, h_G) and $(G', h_{G'})$, such that: $w_G((v', r_T)) = w_{G'}((w', r_G))$, then $d_E(T, T') = d_E(sub_G(v'), sub_{G'}(w'))$.*

Thus we define the edit distance between merge trees as follows.

Theorem 3.28 (Merge Tree Edit Distance)**.** *Given two merge trees $(T, h_T), (T', h_{T'}) \in \mathcal{MT}_K$, then $d_K(T, T')$ does not depend on K.*

Thus, for any couple of merge trees in $\mathcal{MT}$, we can define the merge tree edit distance

$$d_E((T, h_T), (T', h_{T'})) := d_K((T, h_T), (T', h_{T'}))$$

for any $K \geq \max\{\max h_T, \max h_{T'}\}$.

Proof. Consider (T, h_T) and $(T', h_{T'})$ such that $\max h_T < K'$ and $\max h_{T'} < K'$. Consider $(G, w_G) = \mathcal{T}_{K'}((T, h_T))$ and $(G', w_{G'}) = \mathcal{T}_{K'}((T', h_T))$. For any K such that $\max h_T < K < K'$ and $\max h_{T'} < K < K'$ there is a splitting of $(v, r_G) \in E_G$ with a vertex w and a splitting of $(v', r_{G'})$ with a vertex w' such that the obtained weighted trees (R, w_R) and $(R', w_{R'})$ satisfy: $w_R((w, r_G)) = w_{R'}((w', r_{G'}))$. Thus by Proposition 3.27 we obtain $d_E(G, G') = d_E(R, R') = d_E(sub_R(w), sub_{R'}(w'))$. Moreover, $sub_R(w) = \mathrm{Tr}_K((T, h_T))$ and $sub_{R'}(w') = \mathrm{Tr}_K((T', h_{T'}))$.

Being K, K' arbitrary we have that $d_K((T, h_T), (T', h_{T'}))$ does not depend on K, for $K > \max\{\max h_T, \max h_{T'}\}$. We need to prove the case $K = \max\{\max h_T, \max h_{T'}\}$.

Let $(G, w_G) = \mathcal{T}_K((T, h_T))$ and $(G', w_{G'}) = \mathcal{T}_{K'}((T, h_T))$, for $K' > K$. And consider now the weighted tree $\star = (\{\star\}, w(\emptyset) = 0)$,

We have that $d_E(G, \star) = \sum_{e \in E_G} w_G(e)$ and $d_E(G', \star) = \sum_{e \in E_{G'}} w_{G'}(e)$. Thus by the reversed triangle inequality - Proposition 3.43:

$$| d_E(G, \star) - d_E(G', \star) | = K' - K \leq d_E(G, G'),$$

So we have:

$$d_E(\mathcal{T}_{K'}((T, h_T)), \mathcal{T}_K((T, h_T))) = K' - K.$$

and we can set $K = \max\{\max h_T, \max h_{T'}\}$ and take $K' \to K$ to finish the proof. □

By Proposition 3.26 we also have the following corollary of Theorem 3.28.

Corollary 3.29. *The distance d_E is a pseudo metric on $\mathcal{MT}$ and a metric on $\mathcal{MT}/\sim_2$: given two merge trees (T, h_T) and $(T', h_{T'})$, $d_E((T, h_T), (T', h_{T'})) = 0$ if and only if $(T, h_T) \sim_2 (T', h_{T'})$ if and only if $\mathcal{F}((T, h_T)) \cong \mathcal{F}((T', h_{T'}))$ (by Proposition 3.11) .*

Remark 3.30. *Theorem 3.28 and Corollary 3.29 have a series of important implications. First, they say that $(\mathcal{MT}_K/\sim_2, d_E)$ is isometric and isomorphic to $(\mathcal{T}_2, d_E)$ and thus, if we have a subset of merge trees contained in $\mathcal{MT}_K/\sim_2$, for some K, we can map them in $(\mathcal{T}_2, d_E)$ via Tr_K and carry out our analysis there. Second, suppose we are given a merge tree T'' with $\max h_{T''} > K$. For any two merge trees T, T' with $K \geq \max h_T, \max h_{T'}$, we can consider $K' \geq \max h_{T''}$ and compute $d_E(T, T'') = d_{K'}(T, T'')$ and $d_E(T', T'') = d_{K'}(T', T'')$. But we do not have to compute again $d_{K'}(T, T')$ for we have $d_E(T, T') = d_{K'}(T, T') = d_K(T, T')$. Lastly, putting all the pieces together, the metric d_E can be pulled back as a metric on regular abstract merge trees.*

We close this section with the following remark.

Remark 3.31. *A more naive approach could have been to model each merge tree as a triplet $(G, w_G, h_T(v)) \in (\mathcal{T}, \mathbb{R}_{\geq 0}) \times \mathbb{R}$, with $v = \arg\max h_T$ and $G = sub_T(v)$ - i.e. to record the height of the last merging point in T and then remove (v, r_T) from E_T. Then one could define a pseudo metric on $\mathcal{MT}$ via the rule:*

$$d((T, h_T), (T', h_{T'})) = | h_T(v) - h_{T'}(v') | + d_E((G, w_G), (G', w_{G'})).$$

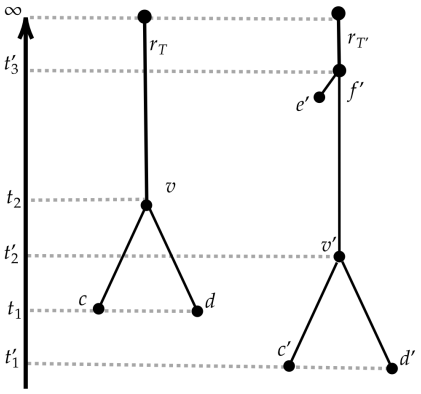
but this leads to unpleasant and unstable behaviors, as in Figure 3.5.2. Such behaviors do not appear with the definition we have given.

Assumption 3.32. *Coherently with Theorem 3.28 and Remark 3.30, to avoid overloading the notation, from now on always assume that we have fixed K big enough for our purposes. For instance, when we take a merge tree (T, h_T) we always imply that we have fixed $K > \max h_T$ and indicate with (T, w_T) the weighted tree $\mathrm{Tr}_K((T, h_T))$.*

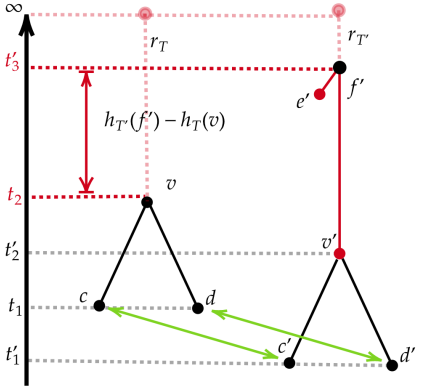
3.5.3 Editing a Merge Tree

We devote this section to explore with some easy examples the definitions given in Section 3.5.1 and Section 3.5.2.

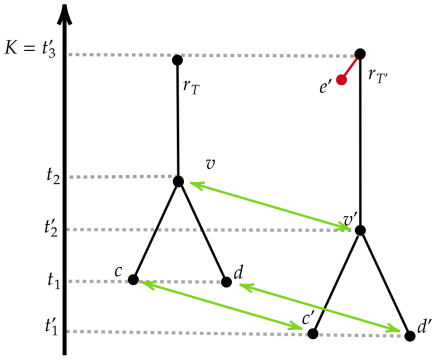
First note that, by construction, $\mathrm{Tr}_K((T, h_T))$ is a representation of the merge tree (T, h_T) coherent with the metric d_E and thus can be used also to visually compare two merge trees. We can then consider a merge tree (T, h_T) and edit (T, w_T) according to the rules in Section 3.4.1: via shrinking, deletions and ghosting of vertices and the inverse operations. We look at the results of the edits in light of the merge tree $(\mathrm{Tr}_K)^{-1}((T, w_T))$.



(a) Two merge trees T (left) and T' (right).



(b) The two merge trees T (left) and T' (right) in Figure 3.5.2a represented as triplets of the form $(G, w_G, h_T(v))$. The shaded red edges represent the merge trees T and T', which are removed to obtained the weighted trees. The height values $h_T(v)$ are colored in red. The red edges represent deletions and the green arrows couple the remaining edges, giving a mapping. The cost of such mapping is given by the height differences $|h_T(v) - h_{T'}(f')|$ plus the two deletions and the weight differences between the coupled edges (which visually appear to be very small). Clearly such distance is inflated by having to account both for $|h_T(v) - h_{T'}(f')|$ and for the deletion of (v', f').



(c) The weighted trees $\mathrm{Tr}_K(T)$ (left) and $\mathrm{Tr}_K(T')$ (right) with $K = t'_3$, obtained from the merge trees T and T' as in Figure 3.5.2a. The red edges represent deletions and the green arrows couple the remaining edges, giving a mapping. The cost of such mapping is given by the weight differences between the coupled edges. Note that the weight difference between (v, r_T) and $(v', r_{T'})$ - i.e. $h_T(v) - h_{T'}(v')$ - is surely smaller than the cost of deleting (v', f')

.

Figure 3.5.2: A situation in which the metric defined in Remark 3.31 shows an unnatural behavior, while d_K still behaves naturally.

Let $(T, h_T) = \mathcal{M}(\pi_0(X_\bullet))$ and consider an edge $e = (v, v') \in E_T$, with $t = h_T(v)$ and $t' = h_T(v')$. Since the height of the root is fixed and equal to K, shrinking e reducing its weight by some value $\varepsilon > 0$ (with $\varepsilon < w_T(e)$) amounts to "moving upwards" $sub_T(v')$ by ε, that is changing $h_T(v'') \mapsto h_T(v'') + \varepsilon$ - as in Figure 3.4.1c→Figure 3.4.1d or in Figure 3.5.4a and Figure 3.5.4c (left). Having $\varepsilon > w_T(e)$ means deleting e. Similarly, increasing $w_T(e)$ by ε, amounts to lowering $sub_T(v')$ by ε - as it partially happens in Figure 3.4.1f inserting the red internal vertex. Consider now the splitting of the edge e into $e_1 = (v, v'')$ and $e_2 = (v'', v')$ with T' being the novel tree structure and $w_{T'}(e_1) = \varepsilon_1$ and $w_{T'}(e_2) = \varepsilon_2$ - as for any of the yellow vertices in Figure 3.4.1e. We must have $\varepsilon_i > 0$ and $w_T(e) = \varepsilon_1 + \varepsilon_2$. This clearly induces a well defined height function $h_{T'}(v'')$. The merge tree $(T', w_{T'})$ differs from (T, w_T) by the order two vertex v'', while the height function on $V_{T'} - \{v''\}$ is still the same. And, accordingly, the associated regular abstract merge trees are the same $\pi_0(X_\bullet) = \mathcal{F}((T, h_T)) \cong \mathcal{F}((T', h_{T'}))$. Thus we have changed the graph structure of T without changing the topological information it represents.

3.5.4 Stability

The behavior of the merge trees edit distance shown in Section 3.5.3 turns out to be fundamental to obtain a stability result for d_E, which is proved in Chapter 4.

Stability results are usually understood in terms of interleavings between persistence modules (Chazal et al., 2008): the distance one measures between (a summary of) the persistence modules should not deviate too much from the interleaving distance between the modules themselves. The rationale behind these ideas follows from the fact that given two functions $f, g : X \to \mathbb{R}$ with suitable properties and such that $\| f-g \|_\infty < \varepsilon$, then their sublevel set filtrations induce ε-interleaved persistence modules via homology functors. Moreover, interleaving distances are taken as reference because they are *universal* among the metrics bounded from above by $\| f - g \|_\infty$ (Lesnick, 2015; Cardona et al., 2021): for any other metric d on persistence modules/multidimensional persistence modules/merge trees such that $d(S_f, S_g) \leq \| f - g \|_\infty$ then $d(S_f, S_g) \leq d_I(S_f, S_g)$, with S_f, S_g being the persistence modules/multidimensional persistence modules/merge trees representing f, g and d_I being the interleaving distance between them. Thus a good behavior in terms of interleaving distance implies a good handling of pointwise noise between functions and so also interpretability of the metric.

For this reason we report the following definitions, which are originally stated by Morozov et al. (2013) with a different notation.

Definition 3.33 (adapted from Morozov et al. (2013))**.** *Given $X_\bullet$ filtration and $\varepsilon > 0$, we define $X^\varepsilon_\bullet$ as $X^\varepsilon_t := X_{t+\varepsilon}$ and $X^\varepsilon_{t \leq t'} := X_{t+\varepsilon \leq t'+\varepsilon}$.*

Consider two abstract merge trees $\pi_0(X_\bullet)$ and $\pi_0(Y_\bullet)$. Two natural transformations $\alpha : \pi_0(X_\bullet) \to \pi_0(Y^\varepsilon_\bullet)$ and $\beta : \pi_0(Y_\bullet) \to \pi_0(X^\varepsilon_\bullet)$ are ε-compatible if:

- $\beta_{t+\varepsilon} \circ \alpha_t = \pi_0(X_{t \leq t+2\varepsilon})$
- $\alpha_{t+\varepsilon} \circ \beta_t = \pi_0(Y_{t \leq t+2\varepsilon})$.

The interleaving distance is thus defined is follows.

Definition 3.34 (Morozov et al. (2013))**.** *Given $\pi_0(X_\bullet)$ and $\pi_0(Y_\bullet)$ abstract merge trees, their interleaving distance is:*

$$d_I(\pi_0(X_\bullet), \pi_0(Y_\bullet)) = \inf\{\varepsilon > 0 \mid \exists \alpha, \beta \ \varepsilon - \textit{compatible}\}.$$

We also say that $\pi_0(X_\bullet)$ and $\pi_0(Y_\bullet)$ are $d_I(\pi_0(X_\bullet), \pi_0(Y_\bullet))$-interleaved.

Being the edit distance a summation of the costs of local modification of trees we expect that d_E cannot be bounded from above by the interleaving distance between merge trees as is the case, for instance, with the bottleneck distance between persistence diagrams (Morozov et al., 2013): such metric in fact is defined as the biggest modification ones needs to optimally match two persistence diagrams. Instead, a suitable stability condition would be for the edit distance to be dependent on the number of vertices in the merge trees but, at the same time, that the cost of the local modifications we need to match the two merge trees goes to 0 as their interleaving distance get smaller and smaller.

Definition 3.35 (Chapter 2)**.** *Given a constructible persistence module $S : \mathbb{R} \to \mathrm{Vec}_{\mathbb{K}}$, we define its rank as* $\mathrm{rank}(S) := \#PD(S)$ *i.e. the number of points in its persistence diagram. When S is generated on $\mathbb{K}$ by a regular abstract merge tree $\pi_0(X_\bullet)$ we have* $\mathrm{rank}(S) := \#PD(S) = \#L_T$*, with $(T, h_T) = \mathcal{M}(\pi_0(X_\bullet))$ and may refer to* $\mathrm{rank}(S)$ *also as the rank of the merge tree* $\mathrm{rank}(T)$*. We also fix the notation* $\dim(T) := \#E_T$*.*

Definition 3.36. *Given $\pi_0(X_\bullet)$ and $\pi_0(Y_\bullet)$ ε-interleaved abstract merge trees; a metric for merge trees is locally stable if:*

$$d(T, T') \leq C(\mathrm{rank}(T) + \mathrm{rank}(T'))\varepsilon$$

for some $C > 0$.

In view of this definitions we can rewrite the following theorem from Chapter 4 and obtain the local stability as an easy corollary.

Theorem 3.37 (Chapter 4)**.** *If there are α, β ε-compatible maps between two abstract merge trees trees $\pi_0(X_\bullet), \pi_0(Y_\bullet)$, then there exist a mapping M between $T = \mathcal{M}(\pi_0(X_\bullet))$ and $G = \mathcal{M}(\pi_0(Y_\bullet))$ such that $cost_M((a, b)) \leq 2\varepsilon$ for every $(a, b) \in M$.*

Corollary 3.38. *Since for a merge tree (T, h_T) we have $\#V_T \leq 2\#L_T$ and* $\mathrm{rank}(T) = \#L_T$*, then d_E is locally stable.*

3.5.5 Comparison with Other Distances For Merge Trees

In this section we want to compare d_E with other definitions of distances between merge trees which appeared in literature. In this way we can better portrait which is the variability between merge trees which is captured by the proposed edit distance.

First, we go through the different definitions which have been employed, some in deeper detail and some in a more qualitative fashion, and in the end we point out two facts which are shared by many of the metrics considered.

3.5.5.1 Interleaving Distance (Morozov et al., 2013) and Metrics for Persistence Diagrams

We start by comparing d_E with the interleaving distance between merge trees, which we have already introduced. Thanks to the relationships between the interleaving distance and the bottleneck distance between persistence diagrams we are also able to interpret the results we obtain in terms of PDs.

Proposition 3.39. *We always have $d_I(T, G) \leq d_E(T, G)$.*

Putting together Proposition 3.39 and Theorem 3.37 we obtain:

$$d_I(T, G) \leq d_E(T, G) \leq 2(\dim(T) + \dim(G))d_I(T, G).$$

This inequalities are in line with our expectations: when editing a merge tree we need to produce a local modification for each vertex of the merge tree adds up all the contributions. While the interleaving distance, in some sense, measures the maximum cost of the modifications that we have to make. The multiplicative factor 2, instead, is caused by the use of weights w_T to compare edges, instead of heights h_T to compare vertices. Topic which we discuss in Section 3.5.5.4. However, we anticipate the following fact: the bound found in Proposition 3.39 cannot be improved. Consider $f : I \to \mathbb{R}$ and $g(x) = f(x) + h$ for some fixed $h \in \mathbb{R}$, and let $X_\bullet$ be the sublevel set filtration of f and $Y_\bullet$ of g. Taking (T, h_T) and (G, h_G) as the merge trees representing $\pi_0(X_\bullet)$ and $\pi_0(Y_\bullet)$, we have $d_I(T, G) = d_E(T, G) = h$.

We can use Proposition 3.39 also to compare the behaviors of the Wasserstein and Bottleneck distance between persistence diagrams and the edit distance d_E between merge trees.

Given two diagrams D_1 and D_2, the expression of such metrics is the following:

$$W_p(D_1, D_2) = \left(\inf_{\gamma} \sum_{x \in D_1} ||x - \gamma(x)||_\infty^p \right)^{1/p}$$

where γ ranges over the functions partially matching points between diagrams D_1 and D_2, and matching the remaining points of both diagrams with the line $y = x$ on the plane (for details see Cohen-Steiner et al. (2007)). In other words we measure the distances between the points of the two diagrams, pairing each point of a diagram either with a point on the other diagram, or with a point on $y = x$. Each point (also called *persistence pair*) can be matched once and only once. The minimal cost of such matching provides the distance. The case $p = \infty$ is usually referred to as the bottleneck distance d_B.

In Morozov et al. (2013) it is shown that:

$$d_B(PD(T), PD(G)) \leq d_I(T, G)$$

with $PD(T)$ being the persistence diagram associated to the merge tree T. Thus, the bottleneck distance is a stable metric for merge trees.

On the other hand we clearly have:

$$d_{W_1}(PD(T), PD(G)) \leq (\text{rank}(T) + \text{rank}(G)) d_B(PD(T), PD(G))$$

which means that the 1-Wasserstein distance between persistence diagrams is locally stable.

Thus we have:

$$d_{W_1}(PD(T), PD(G)) \leq (\dim(T) + \dim(G)) d_E(T, G)$$

and this bound cannot be improved: as with the interleaving distance, consider $f : I \to \mathbb{R}$ and $g(x) = f(x) + h$ for some fixed $h \in \mathbb{R}$, nd let $X_\bullet$ be the sublevel set filtration of f and $Y_\bullet$ of g. Taking (T, h_T) and (G, h_G) as the merge trees representing $\pi_0(X_\bullet)$ and $\pi_0(Y_\bullet)$, we have $d_{W_1}(PD(T), PD(G)) = h(\text{rank}(T) + \text{rank}(G))$ and $d_E(T, G) = h$.

It is thus evident that working with weights $w_T(e)$ instead of persistence pairs, as persistence diagrams do, creates big differences in how the variability between trees is captured by these two metrics. This topic is further discussed in Section 3.5.5.4 as such differences are shared also with other upcoming metrics.

We end this section with a brief simulation study to better showcase the differences between d_I and d_E. The idea behind the simulation can be understood via Figure 3.5.3a. For $i = 0, \ldots, 9$, let $g_i : [0, 11] \to \mathbb{R}$ be such that $g_i \equiv 0$ on $[0, 11] - [i + 1/3, i + 2/3]$ while, on $[i + 1/3, i + 2/3]$, g_i is the linear interpolation of $(i + 1/3, 0)$, $(i + 1/2, 1)$ and $(i + 2/3, 0)$.

Then, for $i = 0, \dots, 9$, define G_i as $G_i \equiv 0$ on $[0, 11] - [i + 2/3, i + 1]$, while, on $[i + 2/3, i]$, G_i is the linear interpolation of $(i + 2/3, 0)$, $(i + 3/4, 5)$ and $(i + 1, 0)$.

Then f_i, $i = 0, \dots, 9$, is obtained as follows:

$$f_i = G_i + \sum_{j=0}^{9} g_j.$$

See Figure 3.5.3a to better visualize this data set: we have a constant set of lower peaks at height 1 and an higher peak with height 5 which is shifting left to right as i increases. In this way we are just changing the left-right distribution of the smaller peaks wrt the highest one.

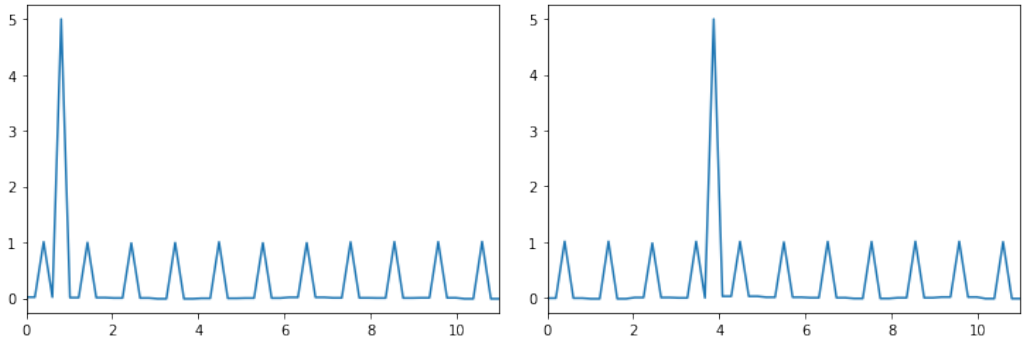
We obtain obtain the associated merge trees and then compute the pairwise distances between the merge trees with d_E. The results are represented in Figure 3.5.3d: the shortest edit path between the i-th merge tree and $i + 1$-th is given by the deletion of one leaf in each tree to make the disposition of leaves coincide between the to trees. The more the peaks' disposition is different between the two trees, the more one needs to delete leaves in both trees to find the path between them. Note that the first function (the one in which the highest peak is the second peak) and the last function (the one in which the highest peak is the second-last peak) can be obtained one from the other via a y-axis symmetry and translation. Similarly, the second function is equal, up to homeomorphisms of the domain, to the third-last one, etc. (see also Chapter 4, Proposition 4.11). Thus the merge trees are the same. To sum up the situation depicted in the first row of Figure 3.5.3d, first we get (left-to-right) farther away from the first merge tree, and then we return closer to it. This intuition is confirmed by looking at the MDS embedding in $\mathbb{R}^2$ of the pairwise distance matrix (see Figure 3.5.3e - note that the shades of gray reflect, from white to black, the ordering of the merge trees). The discrepancies between the couple of points which should be identified are caused by numerical errors.

First, it is very easy to observe that all such functions can't be distinguished by persistence diagrams, since they all share the PD in Figure 3.5.3c. Second, the interleaving distance between any two merge trees representing two functions f_i and f_j is $1/2$ if $i \neq j$ and 0 otherwise. Thus the metric space obtained with d_I from the data set $\{f_i\}_{i=0}^{9}$ is isometric to the discrete metric space on 5 elements, where each point is on the radius 1 sphere of any other point.

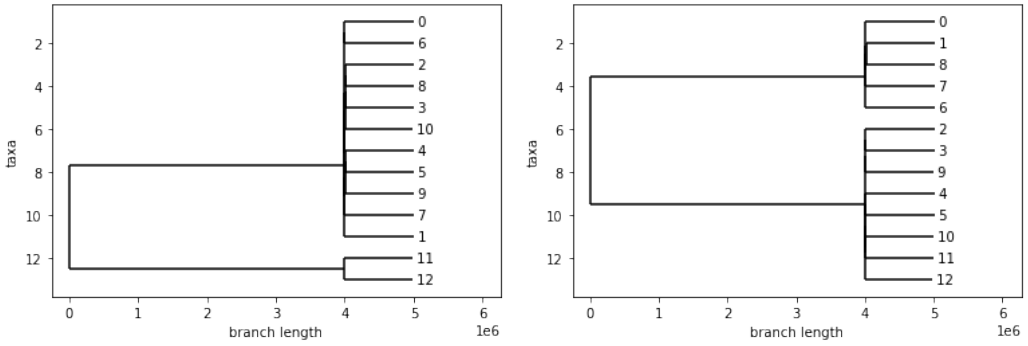
We point out that there are applications in which it would be important to separate f_0 and f_4 more than f_0 and f_1, because they differ by "an higher amount of edits": for instance in Chapter 5 merge trees are used to represent tumors, with leaves being the lesions, and it is well known in literature that the number of lesions is a non-negligible factor in assessing the severity of the illness (Ost et al., 2014), and thus a metric more sensible to the cardinality of the trees is more suitable than d_I.

3.5.5.2 Edit Distance Between Merge Trees (Sridharamurthy et al., 2020) and Wasserstein Distance (Pont et al., 2022)

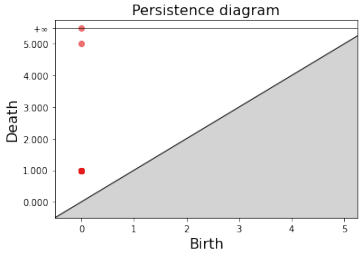
The edit distance in Sridharamurthy et al. (2020) is similar to classical edit distances, with the edit operations being restricted to insertion and deletion of vertices and with a *relabeling* operation which is equivalent to our shrinking operation. There is however the caveat that vertices are in fact understood as persistence pairs (m, s), with m being the leaf representing the local minimum giving birth to the component and s the internal vertex representing the saddle point where the components merge with an earlier born component and thus dies according to the elder rule. There is thus a one-to-one correspondence between persistence pairs in the merge tree and in the associated persistence diagram. Editing a vertex m implies editing also its saddle point s: deleting (m, s) means deleting



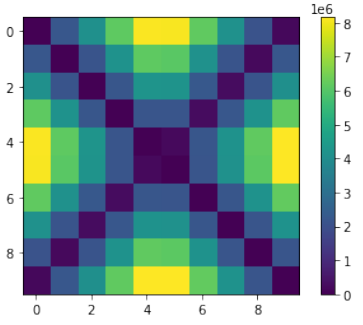
(a) The functions f_0, f_3 belonging to the simulated data set decribed in Section 3.5.5.1.



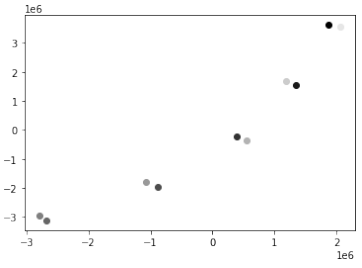
(b) The merge trees (T_{f_0}, h_{f_0}) and (T_{f_3}, h_{f_3}) associated to the functions in Figure 3.5.3a.



(c) The persistence diagram representing the functions in Figure 3.5.3a. The point $(0, 1)$ has multiplicity equal to the number of local minima minus 1.



(d) Matrix of pairwise distances of the merge trees obtained from $\{f_i\}_{i=0}^{10}$.



(e) Multidimensional Scaling Embedding in $\mathbb{R}^2$ of the matrix of pairwise distances shown in Figure 3.5.3d. The shades of gray describe, from white to black, the ordering of the trees.

Figure 3.5.3: Plots related to the simulated scenario presented in Section 3.5.5.1.

all vertices m' such that their persistent pair (m', s') satisfies $s' < s$. If then s becomes of order 2, it is removed. In particular the authors highlight the impossibility to make any deletion - with the word "deletion" to be understood according to our notation - on internal vertices, without deleting a portion of the subtree of the vertex. So they cannot delete and then insert edges to swap father-children relationships. To mitigate the effects of such issue, as a preprocessing step, they remove in a bottom-up fashion all saddles $s \in V_T$ such that $w_T((s, s')) < \varepsilon$ for a certain positive threshold ε. All persistent pairs of the form (m, s) are turned into (m, s'). Such issue is further discussed in Section 3.5.5.5.

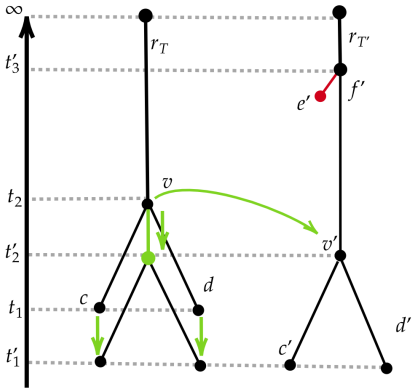
Two merge trees are then matched via mappings representing these edit operations. To speed up the computations, the set of mappings between the trees is constrained so that disjoint subtrees are matched with disjoint subtrees. The cost of the edit operation on an edit pair (m, s) is equal to the edit operation being applied on the corresponding points in the associated persistence diagrams with the 1-Wasserstein distance: deleting a persistent pair has the cost of matching the corresponding point to the diagonal and relabeling a persistent pair with another in the second tree has the cost of matching the two points of the two diagrams - see (Sridharamurthy et al., 2020) Section 4.3.1.

A closely related metric between merge trees is the Wasserstein distance defined in Pont et al. (2022), which extends the metric by Sridharamurthy et al. (2020) producing also further analysis on the resulting metric space of merge trees by addressing the problem of barycenters and geodesics. In this work the authors rely on a particular *branch decomposition* of a merge tree, as defined in Wetzels et al. (2022), from which they induce the branch decomposition tree (BDT - Pont et al. (2022), Section 2.3) used to encode the hierarchical relationships between persistence pairs. A branch decomposition is roughly a partition of the graph T of a merge tree (T, h_T) via ordered sequences of adjacent vertices (Wetzels et al., 2022). The chosen branch decomposition is the one induced by the elder rule and persistence pairs. Edit operations on such BDTs entail improved matchings and deletions between persistence pairs. To obtain the (squared) 2-Wasserstein distance the vertices of two BDTs are match and the resulting costs are squared and then added. However, the authors then explain that with this first definition geodesics cannot by found via linear interpolation of persistence pairs for the hierarchical structure of the merge tree can be broken. To mitigate that, they employ a normalization which shrinks all the branches on $[0, 1]$, irrespective of their original persistence - Pont et al. (2022), Section 4.2 - leading to simple geodesics obtained with linear interpolation between persistence pairs. To mitigate for this invasive procedure they introduce yet another preprocessing step artificially modifying small persistence features to reduce the normalization effects.

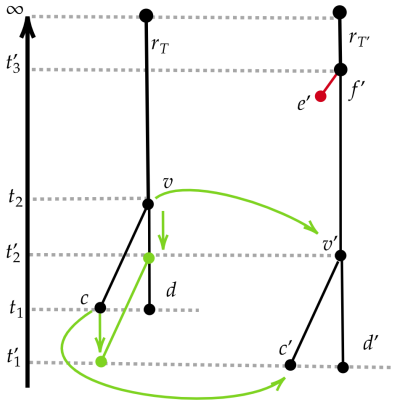
Some of the limitations of this approaches are listed in Pont et al. (2022), Section 7.3. Wetzels et al. (2022), Section 3.3 adds on that with further details and examples. Namely, the restricted space of possible matchings between trees - which is key to obtain the computational performances of the metrics - forces unstable behaviors: issues with saddle swaps (see Pont et al. (2022), Section 4.4 and Fig. 10) and instability of persistence pairs, so that elder ruled-based matchings may force very high persistence features to be matched with other very high persistence features even in situations where this implies making many unreasonable changes in the tree structures as in Figure 3.5.5f (see also Wetzels et al. (2022), Figure 1, Figure 2 b), Section 3.3, and Pont et al. (2022), Section 7.3). Moreover, Pont et al. (2022) does not address the interactions between the normalization and the two preprocessing steps.

3.5.5.3 Branch Decomposition-Independent Edit Distances for Merge Trees (Wetzels et al., 2022)

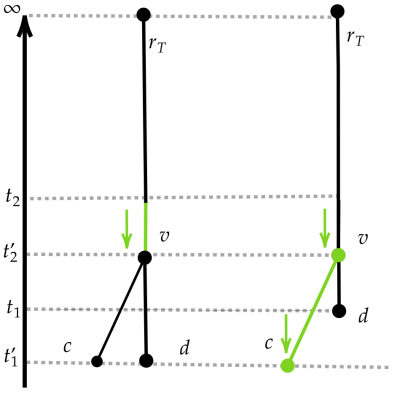
The work Wetzels et al. (2022) starts from the shortcomings of Sridharamurthy et al. (2020) and Pont et al. (2022) trying to overcome them. Namely it defines branch decompositions,



(a) Two merge trees T (left) and T' (right). We first delete (e', f') and then edit T with the shrinking defined by (v, v').



(b) The same merge trees as in Figure 3.5.4a - with some edges drawn at different angles for visualization purposes. We edit T matching the persistence pair (c, v) with (c', v') according to Sridharamurthy et al. (2020); Pont et al. (2022).



(c) The results of the edits applied on T in Figure 3.5.4a (left) and in Figure 3.5.4a (right)

.

Figure 3.5.4: A comparison between the weight based edits on which is based d_E, and the persistence based edits in Sridharamurthy et al. (2020); Pont et al. (2022).

- for instance, the perstence pairs of Sridharamurthy et al. (2020), induce one such branch decomposition - and in order to avoid issues related with the instability of the persistence pairs, Wetzels et al. (2022) introduce also the possibility to optimize the chosen branch decomposition. The only big issue with such approach is that it does not define a metric on the space of merge trees, for the triangle inequality is not satisfied - see Wetzels et al. (2022) Theorem 2 and Figure 3.

3.5.5.4 Heights vs Weights

In this section we try to better understand the different behavior of d_E when compared to the persistence-based metrics presented in the previous sections. The basic idea is that w_T encodes the reciprocal positions of the merging points, instead of having the persistence pairs being free to move independently - at least locally - inside a constrained space.

Consider for instance the merge trees in Figure 3.5.2b: the persistence pairs are (c, v) and (d, r_T). The pairs of the rightmost tree instead are $(c', v'), (e', f')$ and $(d', r_{T'})$. Delet-

ing (e', f') according to Sridharamurthy et al. (2020); Wetzels et al. (2022); Pont et al. (2022) amounts to deleting e' according to our framework. Instead, shrinking (v, r_T) on $(v', r_{T'})$, after the deletion of (e', f'), for us means lengthening (v, r_T) by $t_2 - t_2'$ and so lowering v and all the vertices below v by the same amount, as in Figure 3.5.4a. On the other hand, matching the persistence pair (c, v) with (c', v') for Sridharamurthy et al. (2020); Pont et al. (2022) is equivalent to shifting the edge (c, v) downwards towards (c', v') leaving all other vertices fixed, as in Figure 3.5.4b. We can compare the results of such edits in Figure 3.5.4c.

Thus, given a persistence pair (m, s) inside a merge tree, corresponding to the point (b, d) inside the persistence diagram, moving s upwards by some $\varepsilon > 0$ such that $h_T(s)+\varepsilon < h_T(s')$ costs ε in terms of the other edit distances, of the interleaving distance and in terms of the 1-Wasserstein distance between persistence diagrams and leaves all other points with the same height. In terms of the edit distance d_E, instead, moving s upwards by $\varepsilon > 0$ shortens the edge above s keeping all other edges of a fixed lengths and thus moves the vertices upwards by ε.

We can interpret this fact as the metrics in Morozov et al. (2013), Sridharamurthy et al. (2020) and Pont et al. (2022) better capturing similarities between trees via the location of their vertices in terms of heights; while d_E better captures the "shape" of the tree i.e. relative positions of its vertices, being less sensible on the height variability since you can move more vertices at one. And this is exactly what happens in the example $f : I \to \mathbb{R}$ and $g(x) = f(x) + h$ - as in Figure 3.5.4: we have $d_{W_1}(PD(T), PD(G)) = d_E(T, T')(\dim(T) + \dim(G))$.

3.5.5.5 Stability vs Preprocessing

As already mentioned, the metrics in Sridharamurthy et al. (2020) and Pont et al. (2022) lack stability properties which means that there are certain situations in which such metrics measure variability between the trees which is undesirable as they may perceive as very far trees which are in fact very close in terms of interleaving distance. In particular they are unable to model "saddle swaps", that is, with our terminology, deleting and inserting internal vertices to change father-children relationships. As noted also by the authors themselves, this issue needs to be addressed in some way. To do so authors resort to a computational solution implemented as a preprocessing step: they fix a threshold $\varepsilon > 0$ for any couple of persistence pairs (m, s) and (m', s') with $s' > s$ and $s' - s < \varepsilon$, they change (m, s) into (m, s') merging the two saddle points - in a bottom-up recursive fashion.

In this section, we produce a brief investigation of such procedure, which is absent in the aforementioned works. We represent the possible outcomes of this preprocessing in Figure 3.5.5. All merge trees in Figure 3.5.5 are drawn with colors representing persistence pairs, and similar colors yields persistence pairs with the same persistence throughout the whole figure. They ideally should be matched by the metrics to achieve optimal distances as the differences between edges of different colors could be arbitrary big.

In Figure 3.5.5a (left) we suppose that the edges marked with a red cross represent distances between saddles smaller or equal than $\varepsilon > 0$, thus we recursively merge each saddle point with the higher one, starting from the bottom and going upwards. In this way we obtain the merge tree T' in Figure 3.5.5a (right) which is then used as input for the metric.

In Figure 3.5.5b we see two merge trees T and G such that their interleaving distance is $\varepsilon + \varepsilon'$, for we need to move points of T upwards by $\varepsilon + \varepsilon'$ to match persistence pairs of the same color in G. Their edit distance d_E would be $3(\varepsilon + \varepsilon')$ for we need to delete and then insert back three small edges in T to swap father-children relationships in a suitable way. Note that one can replicate analogous situations to make the interleaving distance between the two merge trees high at will: informally speaking it is enough to add other persistence

pairs as needed and force matchings between pairs in very "different positions".

In Figure 3.5.5c we see a possible output of the preprocessing routine. If the preprocessing threshold is bigger than ε and ε' then the pre-processed trees T' and G' are equal. This in some sense is the desired output of the authors of Sridharamurthy et al. (2020) and Pont et al. (2022) for now their metric can match the colors of the persistence pairs. They suggest that such loss of variability - $d(T', G') = 0$ even if $T \not\cong_2 G$ - could be mitigated by adding to the distance between the preprocessed trees the fixed threshold as many times as there are saddles merging in the procedure. Note that, if this artificial addition approximately matches the variability artificially removed from the pairs attached to the orange vertical pair, it introduces artificial variability at the level of the branches attached to the brown edge, for they do not need any modification to be matched correctly.

In Figure 3.5.5d, instead, we suppose that the preprocessing threshold is smaller than ε and ε'. The metrics then are forced to match persistence pairs with different colors causing an excess of variability which is pictured with red edges in the figure. We point out that, depending on the weights of the persistence pairs involved, these edges could be of arbitrary length.

In Figure 3.5.5e we represent a last possible output of the preprocessing procedure, which is a situation in which the threshold we fix is greater that ε' but smaller then ε. This is possibly the worst scenario: pairs are matched in an optimal way but we have introduced a lot of artificial variability, symbolized with red edges, in G. Again, because of additive phenomena caused by recursive saddle merging, these edges can be arbitrarily long.

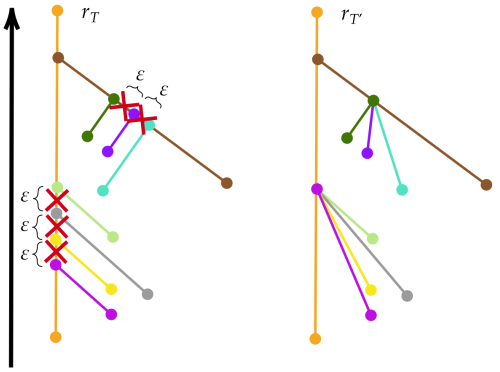
Note that the preprocessing does not fix the issues presented in Figure 3.5.5f, in Wetzels et al. (2022), Figure 1, Figure 2 b), Section 3.3, and Pont et al. (2022), Section 7.3.

To conclude, we point out the following fact. We deem fundamental to have metrics being able to measure different kinds of variability in a data set, especially if the information captured is interpretable, however, any attempt to adapt our edit distance d_E to work with heights instead of weights, would face the problem of coherently define deletion edits. While it seems reasonable to have $w_T((v, v'))$ as a cost to delete and edge - for instance, this cost amounts to the persistence of the feature if (v, v') is a persistence pair - any change in the height of v' changes the weight of (v, v') and thus the cost of its deletion, invalidating all the results about mappings in Chapter 2 and the algorithm therein. Similarly, adapting Sridharamurthy et al. (2020) or Pont et al. (2022) to handle saddle swaps, would mean at least removing from their mappings, the property that would be (M3) in our notation, creating many issues from the theoretical and computational point of view. In fact, internal edges are not really available in the representation used by such metrics, where all points are instead understood as part of a persistence pair.

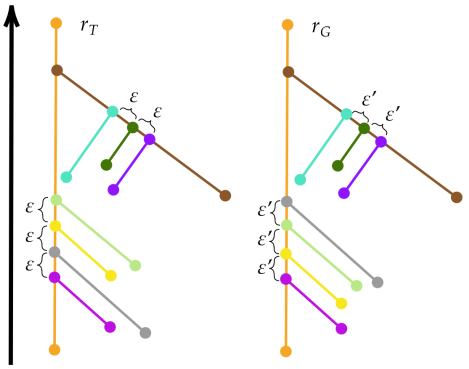
3.6 The geometry of $(\mathcal{MT}, d_E)$

Now we start the investigation of the metric space of merge trees. The structures that we are going to build up throughout the section can all be seen in Figure 3.6.1 and Figure 3.6.2.

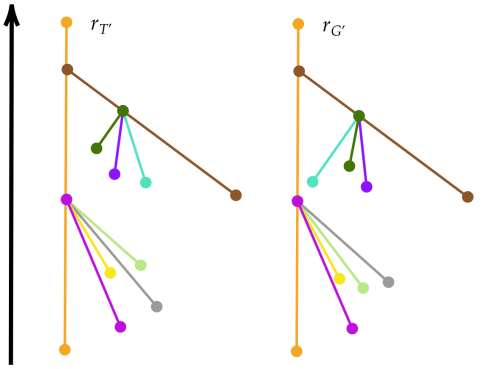
As already stated in the Introduction, this investigation has the general goal of understanding, at least partially, the local geometry of the metric space $(\mathcal{MT}, d_E)$. To achieve this goal, we stratify merge trees according to their dimension and explore the behaviour of the space along these strata and across these strata. As a first result, the topological properties which are obtained in terms of within-in strata completeness and compactness lead to the existence of Frechét Means, which are very important objects for non-Euclidean statistics (see Section 3.6.5). On the other hand, the decomposition of geodesic paths along different strata leads to the local-approximation result in Section 3.6.7. Such result is very promising in terms of the development of statistical techniques like regression and



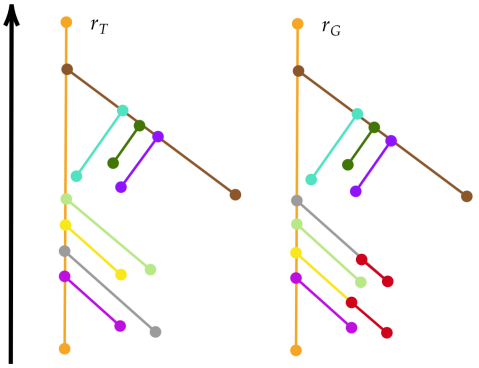
(a) A merge tree T (left) with its pre-processed version (right) according to the procedure in Sridharamurthy et al. (2020); Pont et al. (2022).



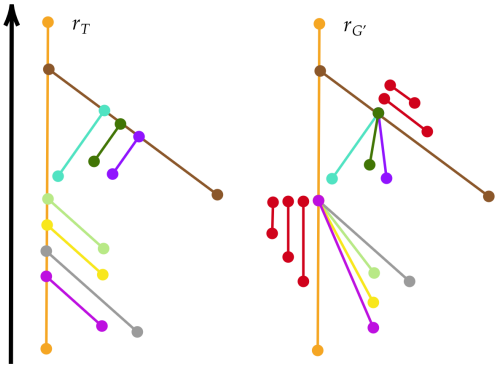
(b) Two merge trees T and G which are going to be preprocessed as in Section 3.5.5.5.



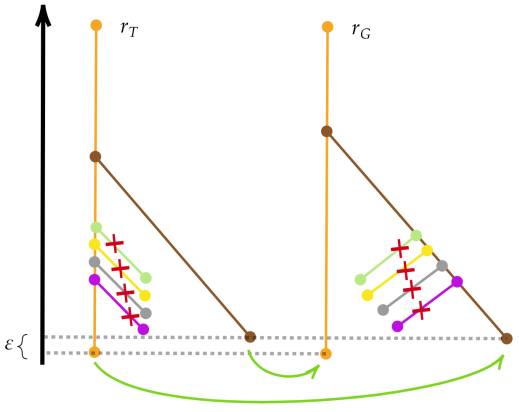
(c) A possible output of the preprocessing: the preprocessing threshold is bigger than ε and ε'.



(d) Another possible output of the preprocessing: the preprocessing threshold is smaller than ε and ε'.



(e) The last possible output of the preprocessing: the preprocessing threshold is smaller than ε and but bigger than ε'.



(f) As noted by Wetzels et al. (2022) and Pont et al. (2022), metrics which are based on persistence pairs may force unnatural scenarios like deleting all the pairs with the red cross, in order to match the orange and the brown persistence pairs of T, respectively, with the orange and the brown of G. Instead, it would be much more natural to match the orange and the brown ones according to the green arrows and then matching the other pairs according to colors.

Figure 3.5.5: Plots related to Section 3.5.5.5

principal components analysis and immediately leads to a simple approximation scheme for Frechét Means - whose properties are not assessed in the present work. Clearly, the picture is far from being complete and many questions remain open: for instance the possibility of doing differential calculus in the space of merge trees or retrieving more global information like the quantification - in some measure-related terms - of the well-behaved trees still need to be studied.

3.6.1 Merge Trees, Weighted Trees and Order 2 Vertices

Recall that $\mathcal{MT}$ is the space of merge trees up to isomorphism, $\mathcal{T}$ (abbreviation for $(\mathcal{T}, \mathbb{R}_{\geq 0})$) is the space of weighted trees, $\mathcal{MT}/\sim_2$ and $\mathcal{T}_2$ are the respective quotient spaces for merge tree and weighted trees up to order 2 vertices. Moreover, by Section 3.5.2, we have the isometry $(\mathcal{MT}_K, d_E) \cong (\mathcal{T}, d_E)$ for any $K \in \mathbb{R}$, with this isomorphism of metric spaces passing to $(\mathcal{MT}_K/\sim_2, d_E) \cong (\mathcal{T}_2, d_E)$.

Both for weighted trees and for merge trees we indicate with T_2 the only representative without order 2 vertices inside the equivalence class of T; if needed in the discussion, we call such equivalence class $[T]$. We have already seen that $d_E(T, T') = 0$ iff $T \sim_2 T'$. This is equivalent to saying that, if we consider the topological spaces $(\mathcal{MT}, d_E)$ and $(\mathcal{MT}/\sim_2, d_E)$ (or $(\mathcal{T}, d_E)$ and $(\mathcal{T}_2, d_E)$), respectively with the pseudo-metric and the metric topology, then $(\mathcal{MT}/\sim_2, d_E)$ is the metric space induced by the pseudometric d_E of $\mathcal{MT}$ via metric identification i.e. $x \sim y$ iff $d(x, y) = 0$. In particular, the projection on the quotient $\Pi : \mathcal{MT} \to \mathcal{MT}/\sim_2$ preserves distances and so open balls. In other words we can focus our attention on $\mathcal{T}$ and $\mathcal{MT}$, to avoid formal complications given by equivalence classes, and most topological result obtained for such space will hold also for $\mathcal{T}_2$ and $\mathcal{MT}/\sim_2$. In particular, since $B_\varepsilon(T) = B_\varepsilon(T')$ for any $T \sim_2 T'$ - recall that $B_\varepsilon(T) = \{T' \big| d_E(T, T') < \varepsilon\}$ - to obtain local topological results we can always assume $T = T_2$.

In a similar fashion we focus our attention on the space of weighted trees and then point out how to extend the results to merge trees via a suitable isometry $(\mathcal{MT}_K, d_E) \cong (\mathcal{T}, d_E)$. On top of that, we also exploit the following trivial fact.

Proposition 3.40. *Given a merge tree (T, h_T) then for every $\varepsilon > 0$ there is $K \in \mathbb{R}$ such that $B_\varepsilon(T) \subset \mathcal{MT}_K$.*

3.6.2 Subspaces

To start things, off there is a natural notion of dimension for weighted trees, which we have already introduced and used. We have defined $\dim(T) = \#E_T$, that is, the number of edges in the tree structure T. To induce a suitable notion of dimension in $\mathcal{T}_2$ we do as follows: $\dim([T]) = \inf_{T' \in [T]} \dim(T')$, which entails $\dim([T]) = \dim(T_2)$.

We take the exact same definitions also for any merge tree (T, h_T). Note that $\dim(T) = \dim \mathrm{Tr}_K(T)$ for any $K > \max h_T$.

Now we can consider trees grouped by dimension.

Definition 3.41. *$\mathcal{T}^N = \{T | \dim(T) \leq N\}$ for any $N \in \mathbb{N}$. The subspaces $\mathcal{T}_2^N$, $\mathcal{MT}^N$ and $\mathcal{MT}^N/\sim_2$ are defined analogously.*

Understanding how these strata interact with each other and how we can navigate between them can shed some light on the structure of these metric spaces.

3.6.3 From Edits to Geodesics

Thanks to Corollary 2.50 we know that there are always edit paths which attain the metric d_E between two weighted trees.

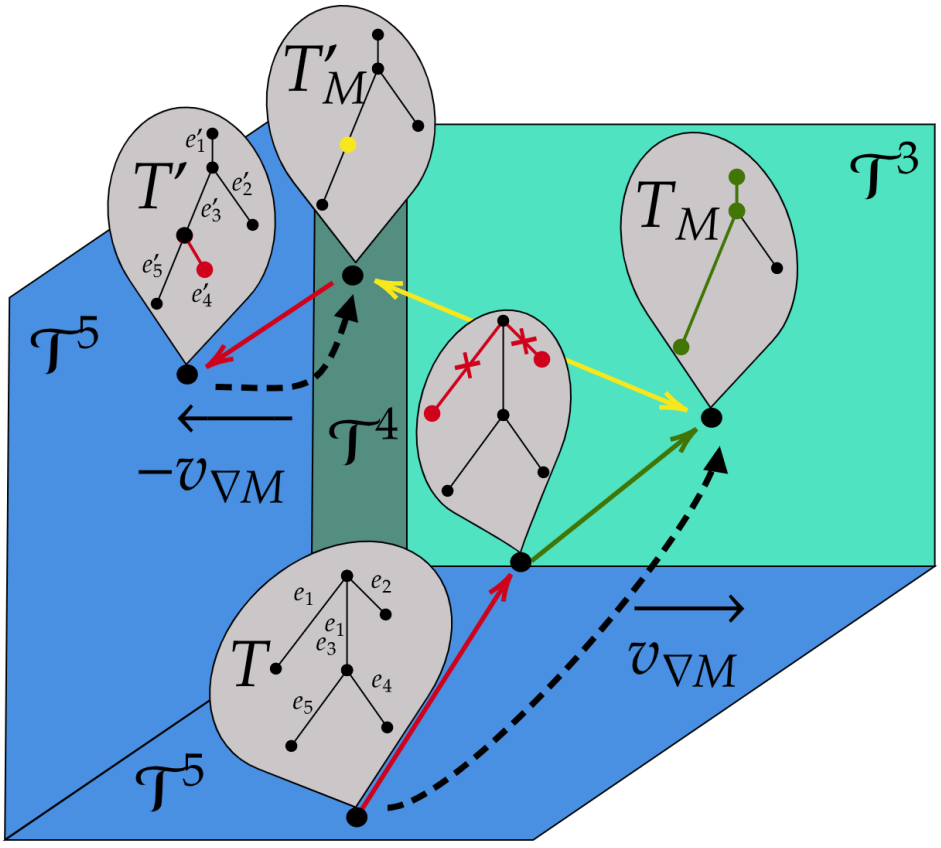


Figure 3.6.1: A mapping M with the associated geodesic and the dashed arrows which represent the $(T \to T')$-decomposition via the canonical representation of M (see Section 3.6.7): starting from T, the red arrow represents the deletion of e_1 and e_2, the following green arrow shrinks the green edges e_3 and e_5, the yellow double headed arrow is a length 0 jump between two trees equal up to the splitting induced by the yellow vertex and then the red arrow pointing in T' represents the insertion of the red edge e'_4. The vector $\overrightarrow{v_{\nabla M}}$ represents the geodesic paths from T for T_M and thus is given by: $-w_T(e_1)\boldsymbol{e_1} - w_T(e_2)\boldsymbol{e_2} - a\boldsymbol{e_3} + 0\boldsymbol{e_4} + b\boldsymbol{e_5} = (-w_T(e_1), -w_T(e_2), -a, 0, b) \in R^T$, with $a, b > 0$ which give the shrinking that ends in T_M. The vector $\overleftarrow{v_{\nabla M}}$, instead, goes from T' to T'_M and accounts just for the deletion of e'_4. The double headed yellow arrow then identifies T_M and T'_M modulo order 2 vertices. So $\overleftarrow{v_{\nabla M}}$ is given by $0\boldsymbol{e'_1} + 0\boldsymbol{e'_2} + 0\boldsymbol{e'_3} - w_{T'}(e'_4)\boldsymbol{e'_4} + 0\boldsymbol{e'_5} = (0, 0, 0, -w_{T'}(e'_4), 0) \in R^{T'}$.

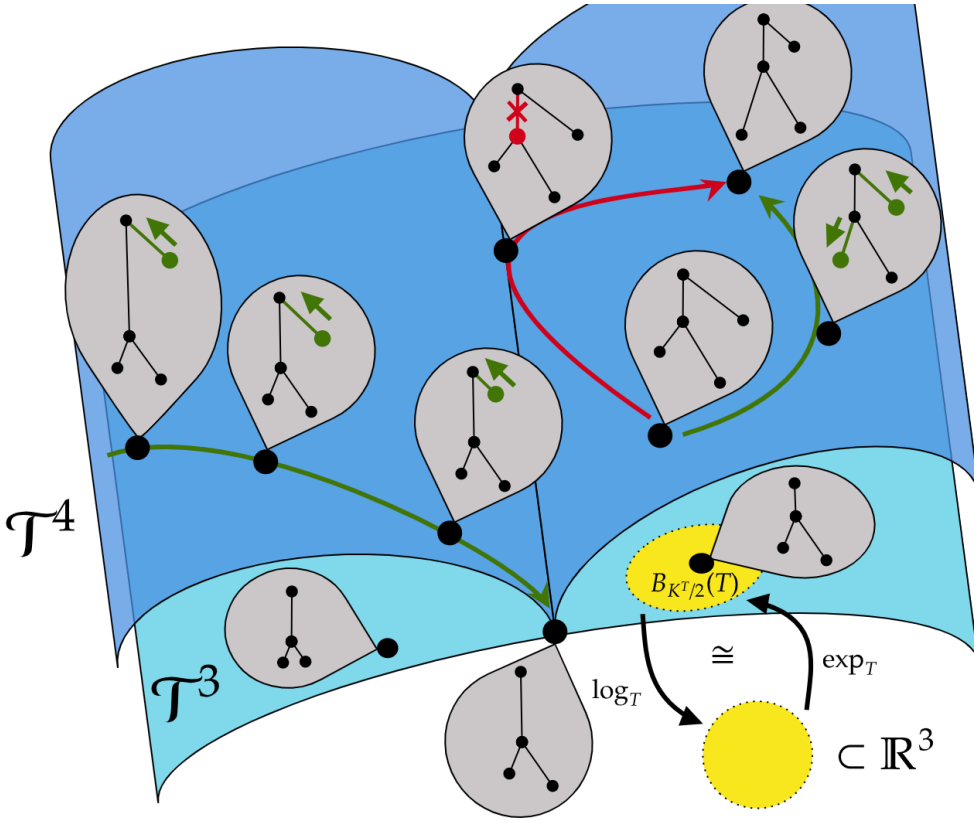


Figure 3.6.2: A qualitative and synthetic representation of the structures introduced in Section 3.6. All the tree are contained in $\mathcal{T}^4$, i.e. they have at most four edges. The two different shades of blue separates the trees in $\mathcal{T}^3$ - light blue, trees with ≤ 3 edge - and the trees with 4 edges - blue. Each tree is represented by a black dot in this space and can be visualized inside a grey cloud. In the left "page" of the blue surface we see a geodesic represented by a green arrow, which progressively shrinks an edge - the green one - up to deleting it and obtaining a tree in $\mathcal{T}^3 \cap \mathcal{T}^4$. On the right "page", instead, we see two trees joined by two optimal mappings: the one represented by the red arrows which deletes the red edge - and so goes through $\mathcal{T}^3 \cap \mathcal{T}^4$ - and the regrows it swapping two children, and the green one which instead shrinks the two green edges with the direction given by the arrows. In the light blue region we see a weighted tree T with a yellow open neighborhood which represents $B_{\mathcal{K}^T/2}(T)$ and is homeomorphic via $\log_T$ and $\exp_T$ to an open neighborhood of the origin in $\mathbb{R}^T_{>T} \subset \mathbb{R}^3$.

Consider an optimal edit path $e_n \circ \ldots \circ e_1(T) = G$; now we show that it can be seen as the discretization of a continuous curve $\gamma : [0,1] \to \mathcal{T}$ at points $t_j = \sum_{i=1}^{j} cost(e_i)/d_E(T,G)$. Set $t_0 = 0$.

Pick an edit e_i and let T' be $T' = e_{i-1} \circ \ldots \circ e_1(T)$. We have $d_E(T', e_i(T')) = cost(e_i)$. Suppose e_i is a deletion or the shrinking of an edge $e = (v, v')$ and let C be the weight of e after e_i or $C = 0$ if e_i is a deletion.

We define $\gamma_i : [t_{i-1}, t_i) \to \mathcal{T}$ as $\gamma_i(t) = T'_t$ with $T'_t \cong T'$ as tree structures and

$$w_{T'_t}(e) = w_{T'}(e) + \frac{(t - t_{i-1})}{t_i - t_{i-1}}(C - w_{T'}(e)).$$

Note that, for $t \in [t_{i-1}, t_i)$ and $\varepsilon > 0$ such that $t_i + \varepsilon \in [t_{i-1}, t_i)$, we have:

$$\begin{aligned} d_E(\gamma_i(t), \gamma_i(t+\varepsilon)) =& \left| w_{T'}(e) + \frac{(t - t_{i-1})}{t_i - t_{i-1}}(C - w_{T'}(e)) - (w_{T'}(e) + \frac{(t + \varepsilon - t_{i-1})}{t_i - t_{i-1}}(C - w_{T'}(e))) \right| \\ =& \frac{\varepsilon}{t_i - t_{i-1}} \left| C - w_{T'}(e) \right|. \end{aligned}$$

Thus for every $t_{i-1} = t^1 < \ldots < t^m = t_i$ we have:

$$\sum_{j=1}^{m-1} d_E(\gamma_i(t^j), \gamma_i(t^{j+1})) \left| C - w_{T'}(e) \right| = \sum_{j=1}^{m-1} \frac{t^{j+1} - t_j}{t_i - t_{i-1}} \left| C - w_{T'}(e) \right| = \left| C - w_{T'}(e) \right| = cost(e_i)$$

As a consequence, the length of γ_i is $cost(e_i)$. Moreover we can define $\gamma_i(1) = T'$. Note that γ_i is continuous.

If instead of e_i being a deletion we consider an insertion, we define $\gamma_i(t) = \gamma'_i(1-t)$ with γ'_i being obtained from the deletion e_i^{-1}. Lastly, if e_i is a ghosting we define $\gamma_i(t) = e_i(T')$ for all $t \in [t_{i-1}, t_i]$. In this case the length of γ_i is 0.

Thus let $\gamma : [0,1] \to \mathcal{T}$ be $\gamma_n \circ \ldots \gamma_1$. Note that:

$$t_i - t_{i-1} = \sum_{j=1}^{i} cost(e_j)/d_E(T,G) - \sum_{j=1}^{j-1} cost(e_j)/d_E(T,G) = cost(e_i)/d_E(T,G)$$

Consider now $t < t' \in [0,1]$. If $t, t' \in [t_{i-1}, t_i]$ we have

$$d_E(\gamma(t), \gamma(t')) = \frac{t' - t}{t_i - t_{i-1}} cost(e_i) = (t' - t) d_E(T,G).$$

If instead $t \in [t_{i-1}, t_i]$ and $t' \in [t_{j-1}, t_j]$, with $i < j$, we have:

$$\begin{aligned} d_E(\gamma(t), \gamma(t')) =& d_E(\gamma(t), \gamma(t_i)) + d_E(\gamma(t_{j-1}), \gamma(t')) + \sum_{k=i}^{j-1} d_E(\gamma(t_k), \gamma(t_{k+1})) \\ =& d_E(T,G) \left(t_i - t + \sum_{k=i}^{j-1} t_{k+1} - t_k + t' - t_{j-1} \right) = d_E(T,G) \left(t' - t \right). \end{aligned}$$

Thus γ is a minimizing geodesic, as in Definition 3.19 and, moreover, it induces a geodesic in the quotient under $\sim_2$. Note that if we have (T, h_T) and (G, h_G) merge trees, then embedding them in the in the space of weighted trees, we can obtain the same results. From now on, with an abuse of notation, with the term geodesic we indicate either a minimal edit path or the curve obtained from the minimal edit path as explained above.

Chapter 2, shows that for any pair of weighted trees the distance between them is given by the length of a path γ_M, with M being a mapping between the two trees. Moreover,

looking at how mappings parametrize finite edit paths, we see that if $T, T' \in \mathcal{T}^N$, then there is at least a geodesic between them which does not exit $\mathcal{T}^N$. By considering T_2 and T_2' we obtain the same consequence for $\mathcal{T}_2^N$. Clearly all these things hold true also for merge tree, via isometries. We sum up these facts with the following proposition.

Proposition 3.42. *The spaces $(\mathcal{T}, d_E)$, $(\mathcal{T}_2, d_E)$, $(\mathcal{T}^N, d_E)$, $(\mathcal{T}_2^N, d_E)$, $(\mathcal{MT}, d_E)$, $(\mathcal{MT}/\sim_2, d_E)$, $(\mathcal{MT}^N, d_E)$ and $(\mathcal{MT}^N/\sim_2, d_E)$ are geodesic spaces.*

3.6.4 Topology

Topology plays a central role when investigating the properties of a space. For instance, being able to characterize or identify open, closed and in particular compact sets is fundamental to work with real valued operators defined on such space.

We establish the following notation for weighted trees: $\| T \| = \sum_{e \in E_T} w_T(e)$, so that the reversed triangle inequality in the case of weighted trees has the following form.

Proposition 3.43. *Given T, T' weighted trees, we have:*

$$| \, \| T \| - \| T' \| \, | \leq d_E(T, T')$$

If we have two weighted tree $T \sim_2 T'$ then $\| T \| = \| T' \|$ and thus Proposition 3.43 holds also in $\mathcal{T}_2$. Instead $\| T \|$ is not well defined for merge trees as it depends on the embedding Tr_K.

Given $C > 0$, we have $B(C) := B_C(\{\star\}) = \{T \in \mathcal{T} \mid \| T \| < C\}$. We also establish the notation: $B(C)^N := B(C) \bigcap \mathcal{T}^N$.

The following result presents some topological properties of the space $\mathcal{T}$ and its subspaces $\mathcal{T}^N$. By Section 3.6.1, all the results which follow hold also for the quotient spaces $\mathcal{T}_2$ and $\mathcal{T}_2^N$.

Theorem 3.44. *For any $N \in \mathbb{N}$:*

1. *$(\mathcal{T}, d_E)$ is contractible.*
2. *$(\mathcal{T}, d_E)$ is not locally compact.*
3. *$(\mathcal{T}^N, d_E)$ is locally compact.*

Theorem 3.44 states that our spaces are "without holes", that is we can continuously shrink the whole space onto the tree with one vertex and no edges and so $\mathcal{T}$ and $\mathcal{T}^N$ are contractible. As predictable $\mathcal{T}$ has at every point issues with losing compactness because of the growing dimension of the trees. However, now we see that these compactness issues can be solved by setting an upper bound on the dimension, which means working in $\mathcal{T}^N$. We can further characterize the subspaces $\mathcal{T}^N$ (and $\mathcal{T}_2^N$) with the following results.

Theorem 3.45. *The metric spaces $(\mathcal{T}^N, d_E)$ and $(\mathcal{T}_2^N, d_E)$ are complete.*

By Section 3.6.1, these results also pass to the quotient via $\sim_2$.

We report the Hopf-Rinow-Cohn-Vossen Theorem which we exploit to conclude this section.

Theorem 3.46 (Theorem 2.5.28 in Burago et al. (2022))**.** *For a locally compact length space X the following are equivalent:*

- *X is complete;*
- *every closed metric ball in X is compact;*

- *every geodesic $\gamma : [0, a) \to X$ can be extended to a continuous path $\bar{\gamma} : [0, a] \to X$;*
- *there is a point $p \in X$ such that every shortest path $\gamma : [0, a) \to X$ with $\gamma(0) = p$ can be extended to a continuous path $\bar{\gamma} : [0, a] \to X$.*

Thanks to Theorem 3.45 we can apply Theorem 3.46 and conclude that the closed balls $\overline{B_\varepsilon(T)^N}$ in $(\mathcal{T}^N, d_E)$ and $\overline{B_\varepsilon(T)^N}$ in $(\mathcal{T}_2^N, d_E)$ are compact for any $\varepsilon > 0$. In particular, the sets $\overline{B(C)^N}$ in $(\mathcal{T}^N, d_E)$ and $\overline{B(C)^N}$ in $(\mathcal{T}_2^N, d_E)$ are compact sets.

Now we take care of merge trees.

Corollary 3.47. *For any $N \in \mathbb{N}$:*

1. *$(\mathcal{MT}, d_E)$ is not locally compact.*
2. *the closed balls $\overline{B_\varepsilon(T)^N}$ in $(\mathcal{MT}^N, d_E)$ and $\overline{B_\varepsilon(T)^N}$ in $(\mathcal{MT}^N/sim_2, d_E)$ are compact for any $\varepsilon > 0$.*

Proof. These results follow from Proposition 3.40.

1. Consider $(T, h_T) \in \mathcal{MT}$ and $B_\varepsilon(T)$. Take K such that $B_\varepsilon(T) \subset \mathcal{MT}_K$. Then $B_\varepsilon(T) \cong B_\varepsilon(\mathrm{Tr}_K(T))$ which is not compact.
2. The result is proven reasoning as in the previous point, observing that $B_\varepsilon(T)^N \cong B_\varepsilon(\mathrm{Tr}_K(T))^N$.

□

3.6.5 Frechét Means

In this section we take the next step in the understanding of the spaces $(\mathcal{T}, d_E)$ and $(\mathcal{MT}, d_E)$, focusing on the Frechét means of a set of trees. Relying heavily on the results obtained in Section 3.6.4 we obtain that these objects exists in $(\mathcal{T}, d_E)$, $(\mathcal{T}_2, d_E)$, $(\mathcal{MT}, d_E)$ and $(\mathcal{MT}/\sim_2, d_E)$.

Frechét means are objects of particular interest in data analysis, as they are defined as the minimizers of operators which look for central points in the distribution of a random variable and thus can be used as 0-dimensional summaries of such distribution. More formally, given X random variable with values in (M, d_M) metric space, a $p-$Frechét mean is defined as $\mathrm{argmin}_{q \in M} \mathbb{E}_X(d_M(q, x)^p)$ - if it exists. Often, this definition is given with $p = 2$ but, at this point we have no reasons to make this choice. In this work we deal with empirical distributions, which amount to considering the case of a finite sets of merge trees.

As generalization of the idea of "average", or 0-dimensional summary of a random variable, Frechét means are among the most used statistics and data analysis for manifold valued data (Davis, 2008) but not only (Turner et al., 2014; Calissano et al., 2020), and are used as starting points to build more refined tools (Pennec, 2018).

Proposition 3.48. *Given $T_1, \ldots, T_n$ weighted trees and $p > 0$; then exist at least one $\overline{T}$ such that:*

$$\overline{T} = \arg\min_T \sum_i d_E^p(T, T_i)$$

Proof. Since $\mathcal{F}(T) := \sum_i d_E^2(T, T_i)$ is a continuous real valued function, if we can restrict the minimization domain in some compact subset of $\mathcal{T}$, we obtain the result, since continuous functions preserve compactness.

First we know that, if $\overline{T}$ exists, then $\| \overline{T} \| \leq \sum_i \| T_i \|^2$. Otherwise $\mathcal{F}(\{\star\}) < \mathcal{F}(\overline{T})$, with $\{\star\}$ being the tree with no edges. Lets call $N_i = \dim(T_i)$ and $N = \sum_i N_i$. Consider T such that $\#L_T = R$ with $R > N$. For all i, any geodesic between T and T_i deletes at least $R - \#L_{T_i}$ edges of T. Since $\#L_{T_i} \leq N_i$, we have $\sum_i R - \#L_{T_i} > \sum_i R - \#N_i > R(n-1)$. This implies that there is at least one leaf of T which is deleted all the times. Then, if we delete it, we obtain T' such that, for all i, $d_E(T', T_i) < d_E(T, T_i)$.

Thus, the number of leaves of $\overline{T}$ cannot exceed N. But this immediately implies that, if it exists, $\dim(\overline{T}) < 2N$.

Since we have a bound on the norm and the dimension of $\overline{T}$, we can restrict the optimization domain on a compact set, which means that $\overline{T}$ exists. □

Corollary 3.49. *Given $T_1, \ldots, T_n$ merge trees and $p > 0$; then exist at least one $\overline{T}$ such that:*

$$\overline{T} = \arg\min_T \sum_i d_E^p(T, T_i)$$

Proof. As in Proposition 3.48 we can obtain $B_C(\{\star\}) \subset \mathcal{MT}$ and N such that $\overline{T} \in \overline{B_C^N}$. Then we can apply Proposition 3.40 and conclude the proof via compactness. □

Thus, for any finite set of trees, we can minimize the function $T \mapsto \sum_i d_E^p(T, T_i)$, obtaining a p-Frechét Mean of the subset.

Being p-Frechét Means important objects for data analysis, it is quite natural to look at the problem of their numerical approximation. In Section 3.6.7.1 we exploit Section 3.6.6 and Section 3.6.7 to give a simple numerical scheme to achieve that, but whose properties are yet to be studied.

Before carrying on, we make the following claim which is still to be investigated.

Claim 3.50. *Given $T_1, \ldots, T_n$ weighted trees, if $T_i \in \mathcal{T}^N$, then there is at least one Frechét mean $\overline{T}$ such that $\overline{T} \in \mathcal{T}^N$.*

This claim is supported by the fact that $\mathcal{T}^N$ are geodesic spaces, and thus is reasonable that we do not need to increase the dimension to find a Frechét mean.

3.6.6 Metric Structure

When working outside linear spaces there are many definitions that must be reinterpreted and generalized to work where no linear structure is available. In the case of manifold, the most common way to do so is exploiting locally the linear structure of the tangent space and to focus on the geodesic nature of straight lines in linear spaces. For instance in Geodesic Principal Component analysis (Huckemann et al., 2010), principal components are replaced by geodesic minimizing the average distance from data points and orthogonality if verified in the tangent space at the barycenter. Moreover the geodesic structure of a space is strictly connected to its *curvature* in the metric geometry sense (Bridson and Haefliger, 2013), also called *Alexandrov's curvature* (Definition 4.1.2. in (Burago et al., 2022)), which in turns is often used to show convergence of statistical estimation algorithms (Sturm, 2003; Miller et al., 2015; Chakraborty and Vemuri, 2015).

For these reasons we want to get a better understanding of the metric structure of the tree space, with particular attention to its geodesic paths, to see if there is, at least locally, there is some regularity/well behaved structure to be exploited for future works.

Since the cost of geodesic paths with the metric d_E is often invariant up to many permutations of the edits, we explore the metric structure of $\mathcal{T}$ assuming the point of view of mappings.

The first thing that we prove is that $(\mathcal{T}, d_E)$, $(\mathcal{T}_2, d_E)$, $(\mathcal{MT}, d_E)$ and $(\mathcal{MT}/\sim_2, d_E)$ are not well behaved in terms of curvature/geodesic structure, in fact we have non-uniqueness issues arising in every neighbor of every tree.

Proposition 3.51. *For every $T \in \mathcal{T}$, for every ε, exists $T' \in \mathcal{T}$ such that T and T' are connected by multiple minimizing mappings and $d_E(T, T') < \varepsilon$.*

Proof. It is enough to attach to any of the leaves of T a pair of equal branches of length less than $\varepsilon/2$ each, obtaining T'. WLOG let l_1, l_2 be the two added leaves and let x be their father in T'.

Then we can build two mappings M and M' such that: $M \cap M' = \{(v, v) \text{ for all } v \in V_T, v \neq x\}$, $("D", l_1), ("D", l_2), (x, x) \in M$ and $("D", l_1), (x, l_2), ("G", x) \in M'$. The cost of both mappings is ε. □

Remark 3.52. *In the present work we are not interested in going into the details of Alexandrov's curvature, however we point out that by Proposition 1.4 in Chapter II of Bridson and Haefliger (2013), $(\mathcal{T}_2, d_E)$ cannot be of bounded curvature as this implies local uniqueness of geodesics. This holds also for $(\mathcal{T}_2^N, d_E)$ as all permutations of deletions and all permutations of shrinkings produce different geodesics.*

The set of points in whose neighbors there are non-unique minimizing mappings (and so geodesics) is therefore the whole space. Now we want to get a better understanding of the origins of these problems. Looking at Figure 3.6.3, we see two reasons which are at the roots of this non-uniqueness:

- similarity between subtrees of the same tree;
- exchange of father-son relationships through the deletion of internal edges.

In Figure 3.6.3d, Figure 3.6.3e and Figure 3.6.3f we find an example of non uniqueness arising because of similar subtrees, and also in the proof of Proposition 3.51 we see this problem in action between subtrees made by a branch each.

In Figure 3.6.3a, Figure 3.6.3b and Figure 3.6.3c on the other hand, we can see uniqueness being broken by topological changes made with internal edges: if we need to change lengths of branches sometimes it can be less expensive to make topological changes like deleting internal edges, and regrowing them to swap children. When this kind of mapping is as expensive as adjusting the children we have of course multiple mappings. To hope to achieve some kind of general uniqueness for mappings we must therefore prevent these things to happen.

We call:

$$k_T = \min_{v,v' \in V_T - L_T, v \neq v'} d_E(sub_T(v), sub_T(v'))$$

Lastly let $m_T = \min_{e \in E_T} w_T(v)$ and $\mathcal{K}_T = \min\{m_T, k_T\}$.

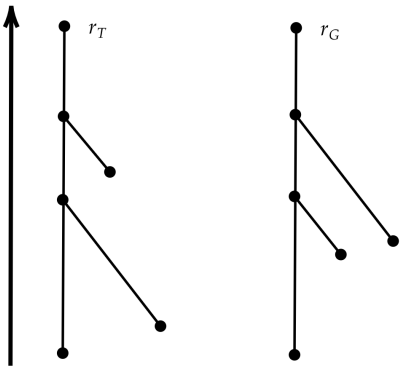
We want to prove that for trees with $\mathcal{K}_T > 0$, if we don't go too far, at least on internal vertices, minimizing mappings with certain properties are uniquely determined. But we need some preliminary results and tools.

Remark 3.53. *Given $M \in \mathrm{Mapp}(T, T')$, thanks to property (M3), $C(M)$ can be given the partial order: $(a, b) \leq (c, d)$ iff $a \leq c$ iff $b \leq c$.*

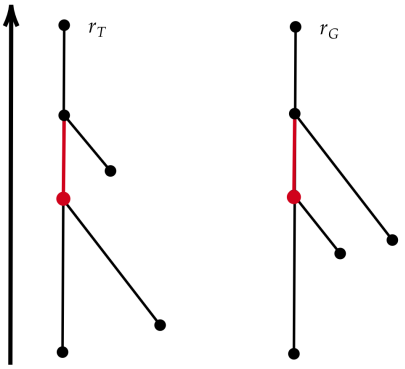
3.6.6.1 Sequences of Edges

We start with the following definition.

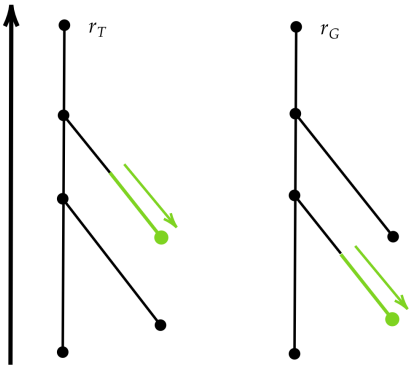
Definition 3.54. *Given a weighted tree T, a sequence of edges is an ordered sequence of adjacent edges $\{e_1, \ldots, e_n\}$. Which means that we have $e_1 < \ldots < e_n$, according to the order induced by the bijection $E_T \leftrightarrow V_T - \{r_T\}$ and that e_i and e_{i+1} share a vertex. We will*



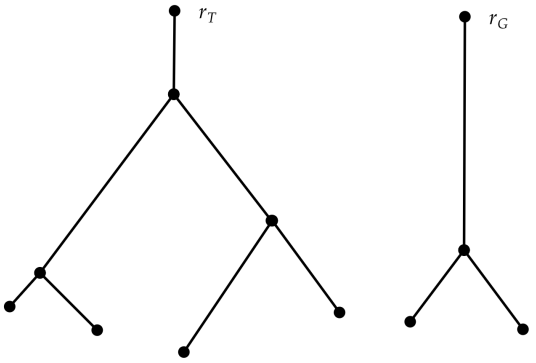
(a) Two merge trees (T, h_T) and (G, h_G).



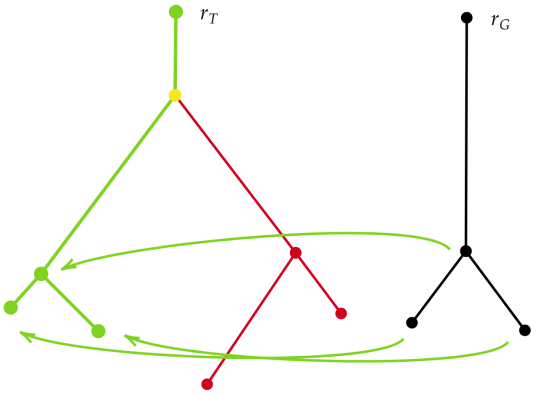
(b) One geodesic between the merge trees in Figure 3.6.3a given by the deletions of the red edges. Such merge trees are identical up to swapping the two "central" branches.



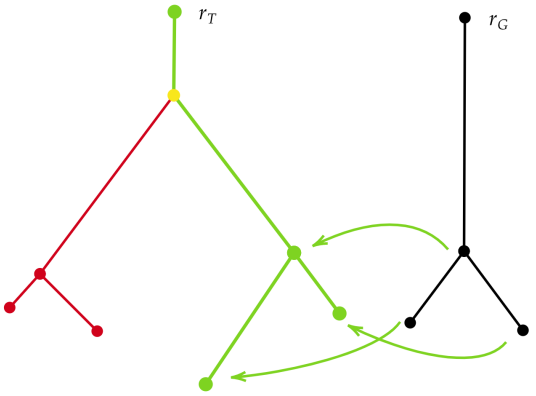
(c) One geodesic between the merge trees in Figure 3.6.3a given by the green shrinkings highlighted in the plot.



(d) Two weighted trees (T, w_T) and (G, w_g).



(e) One geodesic between the weighted trees in Figure 3.6.3d given by the green shrinkings highlighted in the plot and the yellow ghosting.



(f) One geodesic between the weighted trees in Figure 3.6.3d given by the green shrinkings highlighted in the plot and the yellow ghosting.

Figure 3.6.3: Two examples of merge trees or weighted trees which admit different geodesics.

use the notation $[v, v']$ to indicate a sequence of edges which starts in the vertex v and ends with the vertex v', with $v < v'$. Note that, if we write down $[v, v']$ using $E_T \simeq V_T - \{r_T\}$, v is included in the sequence, while v' is the first excluded vertex. The sequence $[v, v']$ is said to be continuous if all the vertices v'' with $v < v'' < v'$ are of order 2. A maximal continuous sequence $[v, v']$ is a continuous sequence such that v is not of order 2 and if v' is of order 2, then v' is the root of the tree.

Now we want to give another definition to help us describing interactions between sequences of edges and mappings. Suppose we have a weighted tree T and we delete an edge. Then we have a natural identification of all the other edges with edges within the newly obtained tree: each edge which is not deleted still has the same extremes and weight in the new tree, a part from the ones which share one vertex with the deleted one. In this latter case we simply have a replacement for the lower or upper extreme of the edge, but the identification with an edge of the new tree is still uniquely determined in a natural way.

Definition 3.55. *Given weighted trees T and T' and a mapping $M \in \mathrm{Mapp}(T, T')$, a sequence $[v, v']$ is said to be (partially) deleted by M if all (some) of its edges are deleted by M. And it is said to be coupled by M if:*

1. *it is not partially deleted by M*
2. *after the deletions it becomes a continuous sequence (note that v' could be deleted and thus replaced by another vertex)*
3. *all vertices v'' with $v < v'' < v'$ are ghosted, but the extremes of the sequence are not*
4. *$v \in \pi_T(C(M))$.*

Note that an edge is a trivial sequence of edges, and an edge is coupled if is it not deleted and its extremes are not ghosted.

Given $(v, w) \in C(M)$, one can restrict M to $sub(v)$ and $sub(w)$, obtaining $M_{|(v,w)} = \{(a, b) \in M | a \leq v \text{ or } b \leq w)\} \in \mathrm{Mapp}(sub(v), sub(w))$ and, as for M, we can consider projections $\pi_T : C(M) \to V_T$. When restricting a mapping intuitively the cost cannot grow, as proven by the following lemma.

Lemma 3.56. *Consider $M \in \mathrm{Mapp}(T', T)$ and $(v, w) \in C(M)$. We can restrict the mapping M to $M_{|(v,w)} \in \mathrm{Mapp}(sub_T(v), sub_{T'}(w))$ and $cost(M_{|(v,w)}) \leq cost(M)$. Moreover if v is an order 2 vertex and there is $v' < v$ with $[v, v']$ being a continuous sequence completely deleted by M, then we can also consider $M_{|(v',w)}$ and we have $cost(M_{|(v',w)}) \leq cost(M)$.*

With these definitions we prove a series of technical lemmas which will lead us to the formulation of Theorem 3.60. These lemmas provide characterizations of how mappings with small cost associate sequences of edges between trees.

Lemma 3.57. *Let $M, M' \in \mathrm{Mapp}(T, T')$ with $cost(M), cost(M') < m_T$. Consider $[w_1, w_2], [w_3, w_4] \subset E_{T'}$ disjoint sequences of edges coupled by M. If $[w_1, w_2]$ is deleted by M', then $[w_3, w_4]$ it cannot be deleted by M'.*

Using Lemma 3.57 we can prove the following stronger result.

Lemma 3.58. *Consider M and M' mappings with $cost(M), cost(M') < m_T$. If a sequence of edges $[w, w'] \subset E_{T'}$ is coupled by M, then it cannot be deleted by M'.*

Moreover the deletions on T' shared between M and M', turn T' to a tree, whose representative without order two vertices has the same tree structure of T (up to isomorphisms of tree structures).

At this point we have enough results to start going into the details of the comparison between two mappings M and M' with "small cost".

Lemma 3.59. *Consider $M, M' \in \mathrm{Mapp}(T', T)$ with $cost(M), cost(M') < \mathcal{K}_T/2$. Let T'' be the tree obtained from T' applying the deletions in $M_D \bigcap M'_D$. Induce mappings N and N' from T'' to T removing respectively from M and M' the deletions already done. Then the followings hold:*

- *given $v \in V_{T'}$, suppose we have $(v, x) \in C(M)$ and $(v, x') \in C(M')$; if at least one between x and x' is not a leaf, then $x = x'$;*
- *if $[v, v'] \subset E_{T''}$ is a maximal continuous sequence then there exist one and only one $(v'', x) \in C(N)$ and one and only one $(v''', x') \in C(N')$ with $v \leq v'', v''' < v'$. If either x or x' is not a leaf, then $x = x'$;*
- *if x is not a leaf in T and we have $(v, x) \in M$ and $(v', x) \in M'$, then $v < v'$ or $v \geq v'$ hold. Suppose $v < v'$; then $[v, v']$ is a continuous sequence in T'';*
- *if we have $(v, x) \in M$ and $(v, x') \in M'$ with $x \neq x'$, both leaves. Then x and x' are siblings.*

3.6.6.2 Main Result

Putting the pieces together, we can prove the following theorem.

Theorem 3.60. *Consider $M, M' \in \mathrm{Mapp}(T', T)$ with $cost(M), cost(M') < \mathcal{K}_T/2$. Then we can obtain a mapping $M'' \in \mathrm{Mapp}(T', T)$ such that $M''_D = M_D \bigcap M'_D$ and $cost(M'') \leq cost(M), cost(M')$. Moreover M'' is uniquely determined on the internal vertices of T.*

We conclude this section with the following corollary which follows easily from Theorem 3.60, representing a uniqueness results for deletion-minimizing mappings.

Corollary 3.61. *Given T, T' with $d_E(T, T') < \mathcal{K}_T/2$ there is at least one mapping M such that for any other mapping M'' - with $cost(M'') < \mathcal{K}_T/2$ - we have $M_D \subset M''_D$ and $cost(M) \leq cost(M'')$. Moreover, if M and M' both satisfy this property, then they coincide on deletions, ghostings and all the couplings with internal vertices of T. Clearly, M, $M' \in M_2(T, T')$ and all leaves of T are coupled with leaves of T'.*

With Corollary 3.61 we have almost obtained what we wanted: understanding at which point around a merge tree the curvature of the metric space starts behaving badly (see Remark 3.52), with metric pathologies as the non-uniqueness of length-minimizing paths arising. By further restricting the neighborhood of T, taking into account also the differences between siblings, we reach the desired uniqueness of mappings.

Corollary 3.62. *Consider T, T' and $M, M' \in M_2(T, T')$ satisfying the properties of Corollary 3.61. Let $\mathcal{K}_{L_T} = \min_{l,l' \in L_T, father(l)=father(l')} |\, w_T(l) - w_T(l') \,|$. Then, if $cost(M), cost(M') < \mathcal{K}_{L_T}/2$, $M = M'$.*

Thus, if we set $\mathcal{K}^T := \min\{\mathcal{K}_T, \mathcal{K}_{L_T}\}$, we have only one mapping in $M_2(T, T')$ satisfying the property in Corollary 3.61 for every $T' \in B_{\mathcal{K}^T/2}(T)$. And such mapping is optimal.

Remark 3.63. *All the results in this section are based on local neighbors of weighted trees. Thus they all extend to merge trees via Proposition 3.40.*

3.6.7 Decomposition of Mappings and Local Isometries

In Section 3.6.6 we have proven that for certain merge trees we have a regular neighborhood where strong uniqueness properties hold. The next step that we want to take is to relate a general geodesic to such neighborhood and to see if the can establish some canonical way in which we can represent how these paths navigate through the different strata in the the space of merge trees. In particular, in this section first we find a way to decompose any mapping via a pair of straight lines each lying in an euclidean space with the 1-norm $\|\cdot\|_1$ and then prove that this decomposition induces an isometry with an euclidean space when we are in regular neighborhood previously mentioned.

We start introducing a partial order on equivalence classes of tree structures up to order 2 vertices: we say that $T' < T$, with $T \sim_2 T'$, if $V_T \subset V_{T'}$. The contravariant way in which this relationship is stated follows what is done in Chapter 2 for coverings of display posets, meaning that T' "refines" T. Given a mapping $M \in \mathrm{Mapp}(T', G')$ with $T' \leq T$ and $G' \leq G$, then we can restrict it to T and G obtaining $M_{|(T,G)} := \{(a, b) \in M \,|\, a \in V_T \text{ or } b \in V_G\}$. Note that in general $M_{|(T,G)}$ is not a mapping between T and G.

Given a tree structure T we define $\mathbb{R}^T := \{v : E_T \to \mathbb{R}\}$. Any element in $\mathbb{R}^T$ may be referred to as a vector. We say the two such sets $\mathbb{R}^T$ and $\mathbb{R}^G$ are isomorphic if and only if $T \cong G$ as tree structures. Similarly we define $\mathbb{R}^T_{\geq 0}$. Any weighted tree (T, w_T) can clearly be seen as $w_T \in \mathbb{R}^T_{\geq}$. Conversely, any $v \in \mathbb{R}^T_{\geq 0}$ gives a weighted tree (T, v).

Consider now a mapping $M \in \mathrm{Mapp}(T, T')$ containing only shrinkings. This implies that $T \cong T'$ as tree structures and thus, up to renaming the vertices in T' via the isomorphism induced by M, such mapping can be represented by the vector $w_{T'} - w_T$ for $w_T + (w_{T'} - w_T) = w_{T'}$ and $\| w_{T'} - w_T \|_1 = cost(M)$.

Consider this time the general case of any mapping $M \in \mathrm{Mapp}(T, T')$; we can obtain T_M and T'_M by applying all deletions and ghostings on T and T' and then $C(M) \in \mathrm{Mapp}(T_M, T'_M)$ contains all the shrinkings. Clearly $C(M)$ can be represented by a vector in $\mathbb{R}^{T_M} \cong \mathbb{R}^{T'_M}$, but, in general, we have no trivial way to bridge consistently between T and T_M in order to establish a relationship between $\mathbb{R}^T$ and $\mathbb{R}^{T_M}$ (and the same for T').

Lets call T_D the tree obtained from T by applying all deletions of the form $(a, "D")$ in M. Then clearly $T_D \leq T_M$. Our next objective is to find a way to extend $C(M)$ on T_D. Recall that, given $T' \leq T$ and $G' \leq G$ and $M' \in \mathrm{Mapp}(T', G')$, $M'_{|(T,G)}$ is the restriction of M' to $(E_T \cup \{"D", "G"\}) \times (E_{T'} \cup \{"D", "G"\})$.

Definition 3.64. *Given a mapping $M \in \mathrm{Mapp}(T, G)$ and $T' \leq T$ and $G' \leq G$ an extension of M is a mapping $M' \in \mathrm{Mapp}(T', G')$ such that $M'_{|(T,G)} = M$. An extension is non-trivial if for every $v \in V_{T'}$ such that $v \notin V_T$, $(v, "G") \notin M'$. And the same for $w \in V_{G'}$ such that $w \notin V_G$.*

We point out that non-trivial extensions are defined as we are not interested in extending a mapping by simply ghosting the added vertices.

To simplify certain situations we take T and T' without order 2 vertices, and later we will use the subset of mappings $M_2(T, T') \subset \mathrm{Mapp}(T, T')$ (see Definition 3.24), since they are defined so that T_M and T'_M are without order 2 vertices.

Proposition 3.65. *Consider a mapping $M \in \mathrm{Mapp}(T, G)$, with $T = T_2$ and $G = G_2$, with no deletions. Given $T' \leq T$, there is always a non trivial extension $M' \in \mathrm{Mapp}(T', G')$ of M, for some $G' \leq G$.*

Proof. Consider $v \in V_{T'}$. Let $u(v) = \min\{v' \geq v \,|\, v' \in V_T\}$ and $l(v) = \max\{v' \leq v \,|\, v' \in V_T\}$. Note that for every $v \in V_{T'}$, $l(v)$ and $u(v)$ always exists: if $v \in V_T$ then $l(v) = u(v) = v$; if $v \notin V_T$ then it is of order 2, and so $l(v), u(v)$ are well defined. Note that $[l(v), u(v)]$

is a maximal continuous sequence: all the vertices $l(v) < v' < u(v)$ are of order 2 and $[l(v), u(v)]$ correspond to an edge in E_T. For every $v' \in [l(v), u(v)]$ set

$$\lambda(v') = \frac{\sum_{l(v)\leq e<v'} w_{T'}(e)}{\sum_{l(v)\leq e<u(v)} w_{T'}(e)} \in [0,1]$$

where and edge $e = (a, b)$ satisfies $e \leq v'$ if $a \leq v'$.

The number $\lambda(v')$ uniquely identifies $v' \in [l(v), u(v)]$ and, moreover, if $v'' < v'$, $\lambda(v'') < \lambda(v')$. Let $(l(v), w), (u(v), w') \in M$. Pick $v' \in [l(v), u(v)]$, $l(v) < v' < u(v)$, we split $(w, w') \in E_G$ with a vertex $w_{v'}$ so that $w((w, w_{v'})) = \lambda(v')$. In a recursive way, for all $v' \in [l(v), u(v)]$, $l(v) < v' < u(v)$, taken in any order, we recursively insert an onrder 2 vertex in $[w, w']$ so that $\lambda(w_{v'}) = \lambda(v')$. And then add the couple $(v', w_{v'})$ to M. Let M' be the mapping obtained at the end of this process and G' the resulting refinement of G.

Consider now $a = \sum_{l(v)\leq e<u(v)} w_{T'}(e)$. Then for every $l(v) \leq v' \leq u(v)$ we have $v' \mapsto \lambda(v')a$ gives a point in $[0, a]$. Similarly let $b = w_G((w, w'))$. Every point $w_{v'}$ can be mapped to $\lambda(w_{v'})b \in [0, b]$. This gives ordered sequences of points $a_1 = 0 < \ldots < a_{n+1} = a$ and $b_1 = 0 < \ldots < b_{n+1} = b$. For every i we have $a_i = \lambda_i a$ with $\lambda_i = \lambda(v_i)$ (assuming v_i corresponds to a_i) and similarly $b_i = \lambda_i b$. Consider now:

$$\sum_{l(v)\leq v'\leq u(v)} cost_{M'}((v', w_{v'})) = \sum_{i=1}^{n} |\, a_{i+1} - a_i - (b_{i+1} - b_i)\,| = \sum_{i=1}^{n} |\,(\lambda i + 1 - \lambda_i)a - (\lambda_{i+1} - \lambda_i)b\,|$$

$$= |\, a - b\,| \sum_{i=1}^{n} (\lambda_{i+1} - \lambda_i) = |\, a - b\,|\,(1 - 0) = cost_M((l(v), w))$$

Thus M' is a non trivial extension of M. □

Note that, if we are in the hypotheses of Proposition 3.65, then M' can be represented by the vector $w_{T'} - w_{G'} \in \mathbb{R}^{T'} \cong \mathbb{R}^{G'}$.

Consider now $M \in M_2(T, T')$. Obtain T_M and T'_M by applying all deletions and ghostings on T and T'. Then we are in the conditions to apply Proposition 3.65 on $C(M) \in \mathrm{Mapp}(T_M, T'_M)$. Let T_D obtained from T by applying all deletions of the form $(a, "D") \in M$. Then $T_D \leq T_M$ and thus we can obtain a mapping $M' \in Mapp(T_D, T''_M)$ - with $T''_D \leq T'_M$ - extending $C(M)$ on T_D as in Proposition 3.65. Thus, if we add to M' all the deletions contained in M, we obtain a mapping $\nabla M \in \mathrm{Mapp}(T, T'')$ for some $T'' \leq T'$. Note that since the construction of ∇M is well defined (see the proof of Proposition 3.65), so is T''.

Definition 3.66. *Given T and T' weighted trees and $M \in M_2(T, T')$, $\nabla M \in \mathrm{Mapp}(T, T'')$ - with $T'' \leq T'$ - is called the canonical representation of M.*

Take now $M \in M_2(T, T')$ and $\nabla M \in \mathrm{Mapp}(T, T'')$; consider $w_T \in \mathbb{R}^T$ and $w_{T'} \in \mathbb{R}^{T'}$. By construction, for every $e = (v, v') \in E_T$, either v is deleted or v is coupled by ∇M and thus we can consider $\overrightarrow{v_{\nabla M}} \in R^T$ such that $\overrightarrow{v_{\nabla M}}(e) = w_T(e)$ if v is deleted or $\overrightarrow{v_{\nabla M}}(e) = w_{T'}(e) - w_T(e)$ if $(v, w) \in \nabla M$. Similarly set the vector $\overleftarrow{v_{\nabla M}} \in \mathbb{R}^{T'}$ as $\overleftarrow{v_{\nabla M}}(e) = w_{T'}(e')$ if $e' = (w, w') \in E_{T'}$ is deleted or $\overleftarrow{v_{\nabla M}}(e') = 0$ otherwise. Using the same notation of the previous paragraph, note that $w_{T'} - \overleftarrow{v_{\nabla M}} \sim_2 T'_M \sim_2 T''_D$. Thus, we have that:

- $cost(M) = cost(\nabla M) = \| \overrightarrow{v_{\nabla M}} \|_1 + \| \overleftarrow{v_{\nabla M}} \|_1$;
- $w_T + \overrightarrow{v_{\nabla M}} \sim_2 w_{T'} - \overleftarrow{v_{\nabla M}} \sim_2 T'_M$.

We point out that we are using ∇M in a very "directional" way: we are using it to infer the vector $\overrightarrow{v_{\nabla M}} \in R^T$, but $\overleftarrow{v_{\nabla M}} \in R^{T'}$ just depends on M. That is the idea driving the following definition.

Definition 3.67. *A $(T \to T')$-decomposition of a mapping $M \in M_2(T, T')$ is a couple of vectors $v \in \mathbb{R}^T$ and $w \in \mathbb{R}^{T'}$ such that:*

- $w_T + v \sim_2 w_{T'} - w \sim_2 T'_M$, *with T'_M obtained from T' applying all deletions of the form $("D", a) \in M$;*
- $cost(M) = \| v \|_1 + \| w \|_1$.

Clearly $\overrightarrow{v_{\nabla M}}$ and $\overleftarrow{v_{\nabla M}}$ give a $(T \to T')$-decomposition of M.

The idea behind $(T \to T')$-decompositions of mappings is to locally approximate $(\mathcal{T}_2, d_E)$ with $\mathbb{R}^n$, for some n and with the 1-norm $\| \cdot \|_1$. For a visual interpretation of this fact see Figure 3.6.1. The approximation around T goes as far as $\mathbb{R}^T$ allows and when the geodesics change strata then also this approximation stops. However, note that not all vectors in $\mathbb{R}^T$ give the decomposition of a geodesic since in general $\| v \|_1 \geq d_E(w_T, w_T + v)$.

If we call $\mathbb{R}^T_{>T} := \{v \in \mathbb{R}^T \mid v + w_T \in \mathbb{R}^T_{>0}\}$ (with $\mathbb{R}^T_{\geq T}$ defined analogously), via the canonical representation of mappings in $M_2(T, T')$ we have found the following correspondence:

$$\log_T : \mathcal{T}^N \times M_2(T, \cdot) \to \mathbb{R}^T_{\geq T}$$

so that $\log_T((T', M)) = \overrightarrow{v_{\nabla M}}$ with $M \in M_2(T, T')$. This correspondence can clearly be reversed via:

$$\exp_T : \mathbb{R}^T_{\geq T} \to \mathcal{T}^N \times M_2(T, \cdot)$$

for any vector in $\mathbb{R}^T_{\geq T}$ identifies a unique couple (T', M), with T' represented by $w_t + v$. We have that:

$$\| \log_T((T', M)) \|_1 \leq cost(M)$$

with the equality holding if $T' \cong w_T + \log_T((T', M))$. While:

$$\| v - v' \|_1 \geq d_E(\exp_T(v), \exp_T(v'))$$

for any $v, v' \in \mathbb{R}^T_{\geq T}$.

Lastly, note that if two mappings M and M' share the same deletions on T then the supports of $w_T + \overrightarrow{v_{\nabla M}}$ and $w_T + \overrightarrow{v_{\nabla M'}}$ - i.e. the components where the vectors are not zero - coincide.

Corollary 3.68. *Consider now $T \in \mathcal{T}^N$ with $\mathcal{K}^T/2 > 0$ and $N = \dim(T)$, $B_{\mathcal{K}^T/2}(T)^N$ and $B_{\mathcal{K}^T/2}(0) := \{v \in R^T_{>T} \mid \| v \|_1 < \mathcal{K}^T/2\}$. Take $T' \in B_{\mathcal{K}^T/2}(T)^N$: since no deletions can be made on T, then also no deletions can be made on T'. Which means that $T'_M = T'$ and, by Corollary 3.61, there is only one mapping in $M_2(T, T')$ with cost less then $\mathcal{K}^T/2$. We call such mapping $M_{T'}$. Thus if we define, with an abuse of notation, $\log_T : B_{\mathcal{K}^T/2}(T)^N \to B_{\mathcal{K}^T/2}(0)$ as $\log_T(T') := \log_T((T', M_{T'}))$ we obtain an injective map.*

Similarly, consider any $v \in B_{\mathcal{K}^T/2}(0)$: v satisfies Corollary 3.61 as it represents a mapping with no deletion and with cost less then $\mathcal{K}^T/2$. Thus v represents the unique minimizing mapping - in $B_{\mathcal{K}^T/2}(0)$ - between T and $w_T + v$.

In other words, $\log_T$ and $\exp_T$ are inverse to each other and give isometries between $B_{\mathcal{K}^T/2}(T)^N$ and $B_{\mathcal{K}^T/2}(0)$.

The idea behind the terminology employed is justified by Corollary 3.68: when we can meaningfully reduce minimizing mappings in $M_2(T, T')$ to a subset with just one element $M_{T'}$, then $\exp_T$ and $\log_T$ can be restricted to $\log_T(T') := \log_T((T', M_{T'}))$. Such maps

are then in analogy to the log and exp maps for Riemannian manifolds as we obtain local homeomorphism (in fact, an isometry) between $B_\varepsilon(T)^N$ and an open set in $\mathbb{R}^n$ - with the $\|\cdot\|_1$-norm - for any weighted tree with $\mathcal{K}^T > 0$. Clearly, 1) we need restrict to trees with bounded dimension, 2) even in this case this does not hold for all trees and so we are far from having a well behaved structure like the Riemannian one. We leave for future works further investigation of such neighborhoods to assess how they can be exploited to establish extrinsic statistical techniques by projecting trees in this "tangent" structure. Such investigation should also aim at studying the relationship between $B_{\mathcal{K}^T/2}(T)^N$ and $B_{\mathcal{K}^T/2}(0)$ when $N > \dim(T)$, to see if it can be stratified in a fruitful way.

3.6.7.1 Approximation of Frechét Means

In this last section we exploit the results previously obtained in Section 3.6.7 to give an approximation scheme for the Frechét mean of a finite set of merge trees.

Consider now a finite set of weighted trees $\{T, T_1, \dots, T_n\}$ and the minimizing mappings $M_i \in M_2(T, T_i)$.

Moreover for every i consider $\overrightarrow{v_{\nabla M_i}}$, $\overleftarrow{v_{\nabla M_i}}$, a $(T \to T_i)$-decomposition of M_i. We know that:

$$\begin{aligned}\mathcal{F}(T) := \sum_{i=1}^{n} d_E(T, T_i)^p &= \sum_{i=1}^{n} (\| \overrightarrow{v_{\nabla M_i}} \|_1 + \| \overleftarrow{v_{\nabla M_i}} \|_1)^p \\ &\geq \min_{v \in \mathbb{R}^T_{\geq T}} \sum_{i=1}^{n} (\| v - \overrightarrow{v_{\nabla M_i}} \|_1 + \| \overleftarrow{v_{\nabla M_i}} \|_1)^p\end{aligned}$$

Thus if we solve:

$$\arg\min_{v \in \mathbb{R}^T_{\geq T}} \sum_{i=1}^{n} (\| v - \overrightarrow{v_{\nabla M_i}} \|_1 + \| \overleftarrow{v_{\nabla M_i}} \|_1)^p \tag{3.2}$$

we obtain a weighted tree T^* in $\mathbb{R}^T_{\geq 0}$ such that $\mathcal{F}(T^*) \leq \mathcal{F}(T)$. Clearly we can recursively repeat the same procedure:

1. compute $M_i \in M_2(T, T_i)$ minimizing mapping for every $i = 1, \dots, n$;
2. compute $\overrightarrow{v_{\nabla M_i}}$, $\overleftarrow{v_{\nabla M_i}}$, a $(T \to T_i)$-decomposition of M_i every $i = 1, \dots, n$;
3. solve Equation (3.2) to obtain T^* and replace T with T^*.

Remark 3.69. *With $p = 1$ we get:*

$$\arg\min_{v \in \mathbb{R}^T_{\geq T}} \sum_{i=1}^{n} \| v - \overrightarrow{v_{\nabla M_i}} \|_1 + \sum_{i=1}^{n} \| \overleftarrow{v_{\nabla M_i}} \|_1 = \arg\min_{v \in \mathbb{R}^T_{\geq T}} \sum_{i=1}^{n} \| v - \overrightarrow{v_{\nabla M_i}} \|_1$$

thus the solution of Equation (3.2) *is the pointwise median of* $\{\log_T((T_i, M_i))\}_{i=1,\dots,n} = \{\overrightarrow{v_{\nabla M_i}}\}_{i=1,\dots,n}$ *in the "tangent space"* $\mathbb{R}^T_{\geq T}$.

We leave to future works a rigorous study of Equation (3.2) and of the properties of the approximation scheme just proposed. In particular, we believe that the approximation scheme can be framed in terms of a gradient descent algorithm upon establishing differential calculus in the space of merge trees, following what has been done for persistent homology (Leygonie et al. (2021), Leygonie et al. (2021)).

3.7 Discussion

In this chapter we introduce a novel edit distance for merge trees, we compare it with existing metrics for merge trees establishing stability properties and then face the problem of geometric characterization of the space of merge trees endued such distance. Our geometric investigation covers the following directions: 1) to establish some compactness results to prove the existence of Frechét means, which are objects of great interest in non-Euclidean statistics, 2) start understanding the geodesic structure of such space 3) working with decomposition of geodesics to obtain local approximation of the space of merge trees via euclidean spaces. Thanks to previously introduced parametrizations of geodesic paths, we obtain the existence of Frechét means, along with introducing an approximation scheme for such objects and prove local uniqueness of geodesics modulo some permutations of the involved edits.

Natural further developments of this work would be to study the possibility of establishing differential calculus on the space of merge trees, deepen the understanding of the logarithm and exponential map we defined, and obtaining a natural measure in such space to quantify now many trees have a well behaved neighborhood and how many not. As a byproduct of those investigations, we believe that the proposed approximation scheme for Frechét Means could be better understood and extrinsic statistical techniques - like regression or L_1-norm principal component analysis - defined via the local approximation of the tree space could be obtained.

Appendix

3.A Proofs

Proof of Proposition 3.39.

Proof. Let $\pi_0(X_\bullet)$ and $\pi_0(Y_\bullet)$ be two regular abstract merge trees which are represented by the merge trees (T, h_T) and (G, h_G), which we represent as weighted trees via the some Tr_K.

Introduction - Display Posets For ease of notation, we introduce the following objects.

Definition 3.70 (Curry et al. (2022))**.** *Given a persistent set $S : \mathbb{R} \to \mathrm{Sets}$ we define its display poset as:*

$$D_S := \bigcup_{t\in\mathbb{R}} S(t) \times \{t\}.$$

The set D_S can be given a partial order with $(a, t) \leq (b, t)$ if $S(t \leq t')(a) = b$.

Let $D_{\pi_0(X_\bullet)}$ and $D_{\pi_0(Y_\bullet)}$ be the display posets induced by $\pi_0(X_\bullet)$ and $\pi_0(Y_\bullet)$. We call $h_{\pi_0(X_\bullet)}$ the height function of $D_{\pi_0(X_\bullet)}$ and $h_{\pi_0(Y_\bullet)}$ the height function of $D_{\pi_0(Y_\bullet)}$.

Introduction - Couplings The content of this part of the proof is taken from Chapter 6.

We leverage on the notion of couplings between merge trees defined in Chapter 6. Before recalling such definition that we highlight a subtle difference in merge trees as defined in Chapter 6 wrt respect to the definition we give here. In Chapter 6 the edge going to infinity which we require in our merge trees - (v, r_T) with $h_T(r_T) = \infty$ - is not needed and thus such edge is removed. In other words, a merge tree (T, h_T) is the sense of Chapter 6 is such that $ord_T(r_T) > 1$ and $h_T(r_T) \in \mathbb{R}$. we state the results in Chapter 6 with the notation of the present chapter.

Given $C \subset V_T \times V_G$ and the projection $\pi_T : V_T \times V_G \to V_T$, we define the multivalued map $\Lambda_C^T : V_T \to V_T$ as follows: $\Lambda_C^T(v) = \max_{v'<v} \pi_T(C)$ if $\#\{v' \in V_T \,|\, v' < v \text{ and } v' \in \pi_T(C)\} > 0$ or $\Lambda_C^T(v) = \emptyset$ otherwise. A coupling between the merge trees (T, h_T) and (G, h_G) - with the definition given in the present chapter - is then a set $C \subset V_T - \{r_T\} \times V_G - \{r_G\}$ such that:

(C1) $\#\max C = 1$ or, equivalently, $\#\max \pi_T(C) = \#\max \pi_G(C) = 1$

(C2) the projection $\pi_T : C \to V_T$ is injective (the same for G)

(C3) given (a, b) and (c, d) in C, then $a < c$ if and only if $b < d$

(C4) $a \in \pi_T(C)$ implies $\#\Lambda_C^T(a) \neq 1$ (the same for G).

Similarly to mappings each couple in C is associated to a cost, and $\| C \|_\infty$ is defined as the highest of such costs. In Chapter 6 it is proven that $d_I(T,G) \leq \| C \|_\infty$ for all couplings C.

Define the following functions as in Chapter 6:

- define $\varphi_C^T : V_T \to V_T$ so that $\varphi_C^T(x) = \min\{v \in V_T \,|\, v > x \text{ and } \#\Lambda(v) \neq 0\}$. Note that since the set $\{v \in V_T | v > x\}$ is totally ordered, $\varphi_C^T(x)$ is well defined;
- similarly, define $\delta_C^T : V_T \to V_T$, defined as $\delta_T(x) = \min\{v \in V_T \big| v \geq x \text{ and } v \in \pi_T(C)\}$;
- set $\gamma_C^T : V_T - D_C^T \to V_G$ to be $\gamma_C^T(x) = \arg\min\{g(w) \,|\, (v,w) \in C, v < x\}$, if $\#\{g(w) \,|\, (v,w) \in C, v < x\} > 0$, or $\gamma_C^T(x) = \emptyset$. If $\#\{g(w) \,|\, (v,w) \in C, v < x\} > 0$, by *(G)*, $\gamma_C^T(x)$ is uniquely defined. Note that $\gamma_C^T(\varphi_C^T(x))$ is well defined for any $v \in V_T$;
- lastly, set $\eta_C^T(x) := \gamma_C^T(\varphi_C^T(x))$.

To lighten the notation, when clear from the context, we may omit subscripts and superscripts. The costs of a coupling are defined in Chapter 6 as follows.

- if $(x,y) \in C$, $cost_C(x) = |h_T(x) - h_G(y)|$;
- if $x \notin\in \pi_T(C)$, we have two different scenarios:
  - if $\#\Lambda(x) = 0$, then $cost_C(x) = \max\{(h_T(\varphi(x)) - h_T(x))/2, h_G(\eta(x)) - f(x)\}$;
  - if $\#\Lambda(x) > 1$, we have $cost_C(x) = |\, h_T(x) - h_G(w) \,|$ with $(\delta(x), w) \in C$;
  - zero otherwise.

Lastly, in Chapter 6 it is shown that, via C one can induce

$$\alpha_C : D_{\pi_0(X_\bullet)} \to D_{\pi_0(Y_\bullet)} \text{ and } \beta_C : D_{\pi_0(Y_\bullet)} \to D_{\pi_0(X_\bullet)}$$

such that

$$cost_C(v) = |\, h_{\pi_0(Y_\bullet)}(\alpha_C(v)) - h_{\pi_0(X_\bullet)}(v) \,|$$

for any $v \in V_T$ and

$$cost_C(w) = |\, h_{\pi_0(X_\bullet)}(\beta_C(w)) - h_{\pi_0(Y_\bullet)}(w) \,|$$

for any $w \in V_G$.

Main body of the proof We now want to establish relations between mappings between $\mathrm{Tr}_K((T,h_T))$ and $\mathrm{Tr}_K((G,h_G))$ - with $K > \max h_T, \max h_G$ - and couplings between (T,h_T) and (G,h_G). When working with $\mathrm{Tr}_K((T,h_T))$ and $\mathrm{Tr}_K((G,h_G))$ we can apply the following result from Chapter 2.

Corollary 3.71 (Chapter 2)**.** *Given (T,w_T) and $(T',w_{T'})$ weighted trees. If r_T and $r_{T'}$ are of order 1, for any minimizing mappings M, then neither (v,r_T) or $(w,r_{T'})$ are deleted and we have $\#\max M = 1$.*

Given a minimizing mapping $M \in M_2(T, T')$, thanks to Lemma 3.25, we have that $a \in \pi_T(M)$ implies $\#\Lambda_M^T(a) \neq 1$ (the same for G). In fact (C4) is equivalent to having no vertices of order two after deletions and ghostings. This means that $C_M := C(M) = \{(a, b) \in M \mid a \in V_T \text{ and } b \in V_G\}$ is a coupling.

Conversely, given a minimizing coupling C, the set

$$\begin{aligned} M_C := C &\bigcup \{(a, "D") \mid a \notin \pi_T(C) \text{ and } \#\Lambda_C^T(a) \neq 1\} \\ &\bigcup \{("D", b) \mid b \notin \pi_G(C) \text{ and } \#\Lambda_C^G(b) \neq 1\} \\ &\bigcup \{(a, "G") \mid a \notin \pi_T(C) \text{ and } \#\Lambda_C^T(a) = 1\} \\ &\bigcup \{("G", b) \mid b \notin \pi_G(C) \text{ and } \#\Lambda_C^G(b) = 1\} \end{aligned}$$

is a mapping. Clearly $M_{C_M} = M$ and $C_{M_C} = C$.

Now we prove the following lemma.

Lemma 3.72. *Let $f : [a, K] \to [b, K]$ be a monotone function such that $f(K) = K$, $K \in \mathbb{R}$. For every $\{x_1 < \ldots < x_{n+1} = K\} \subset [a, K]$:*

$$\max_{i=1,\ldots,n+1} d(f(x_i), x_i) \leq \sum_{i=1}^{n} | \, d(x_i, x_{i+1}) - d(f(x_i), f(x_{i+1})) \, | \tag{3.3}$$

Proof. Let $v_i = f(x_i) - x_i$. And let $m \in \{1, \ldots, n+1\}$ be such that $| \, f(x_m) - x_m \, | = \max | \, f(x_i) - x_i \, |$.

Then we set:

$$\begin{aligned} v_1 &= v_m + \varepsilon_1 \\ v_2 &= v_m + \varepsilon_1 + \varepsilon_2 \\ &\ldots \\ v_i &= v_m + \sum_{j=1}^{i} \varepsilon_j \\ &\ldots \\ v_n &= v_m + \sum_{j=1}^{n} \varepsilon_j. \end{aligned}$$

Thus we can write Equation (3.3), which we need to prove, as:

$$| \, v_m \, | \leq \sum_i | \, x_i - f(x_i) - (x_{i+1} - f(x_{i+1})) \, | = \sum_i | \, v_i - v_{i+1} \, | = \sum_{i=1}^{n} | \, \varepsilon_{i+1} \, | \, .$$

Clearly we have some constraints on ε_i, in fact:

$$x_i + v_m + \sum_{j=1}^{i} \varepsilon_j \leq x_{i+1} + v_m + \sum_{j=1}^{i+1} \varepsilon_j$$

which means:

$$x_i - x_{i+1} \leq \varepsilon_{i+1}.$$

In particular $x_n + v_n = x_n = 1$ means that $v_m + \sum_{j=1}^{n} \varepsilon_j = 0$ i.e. $-v_m = \sum_{j=1}^{n} \varepsilon_j$. Thus:

$$| \, v_m \, | = | \sum_{j=1}^{n} \varepsilon_j \, | \leq \sum_{i=1}^{n} | \, \varepsilon_{i+1} \, | \, .$$

□

Consider now a leaf $l \in L_T$ and take $\zeta_l = \{p \in V_T \,|\, v \geq l\}$. Consider the interval $[h_T(l), K]$. The map $v \in \zeta_l \mapsto h_T(v)$ gives a $1:1$ correspondence between ζ_l and a finite collection of points in $[h_T(l), K]$. We define $f(h_T(v)) = h_{\pi_0(Y_\cdot)}(\alpha_C(v))$ for $v \in \zeta_l - \{r_T\}$. And $f(K) = K$. Note that α_C is constructed such that $\alpha_C(v) = w$ if $(v, w) \in C$. Thus $f(h_T(v)) \leq K$ for $v \in \zeta_l$.

We extend f on $[h_T(l), K]$ via linear interpolation. Since α_C is monotone wrt the partial order on $D_{\pi_0(X_\cdot)}$, then f is monotone on $[h_T(l), K]$.

Now consider T and apply on it all the deletions and ghostings which involve points which are not in ζ_l. We call $\{v_1 = l < \ldots < v_{n+1} = r_T\}$ the coupled points in ζ_l. Then:

$$|\, h_T(v_i) - h_T(v_{i+1}) - f(h_T(v_i)) - f(h_T(v_{i+1}))\,| \leq cost_{M_C}([v_i, v_{i+1}] \mapsto [\alpha_C(v_i), \alpha_C(v_{i+1})]) \tag{3.4}$$

With $cost_{M_C}([v_i, v_{i+1}] \mapsto [\alpha_C(v_i), \alpha_C(v_{i+1})])$ being the cost of the edits associated to points $v_i \leq p \leq v_{i+1}$ and $\alpha_C(v_i) \leq q \leq \alpha_C(v_{i+1})$. In fact we have equality in Equation (3.4) the there are no deletions for any $p \in V_T$ such that $v_i \leq p \leq v_{i+1}$ and any $q \in V_{T'}$ such that $\alpha_C(v_i) \leq q \leq \alpha_C(v_{i+1})$. Otherwise the total cost of the deletions and shrinking exceeds $|\, h_T(v_i) - h_T(v_{i+1}) - f(h_T(v_i)) - f(h_T(v_{i+1}))\,|$ since:

$$|\, n_1 + n_2 - (n_3 + n_4)\,| \leq n_1 + n_3 + |\, n_2 - n_4\,|$$

with $n_i \in \mathbb{R}_{\geq 0}$. Lastly note that $|\, h_T(v_i) - f(h_T(v_i))\,| = cost_C(v)$.

By Lemma 3.72, we have:

$$\begin{aligned} \max cost_C(v_i) =& \max h_T(v_i) - f(h_T(v_i)) \\ \leq& \sum_{i=1}^{n} |\, h_T(v_i) - f(h_T(v_i)) + f(h_T(v_{i+1}) - h_T(v_{i+1})\,| \\ \leq& cost_{M_C}([l, r_T] \mapsto [w, r_G]) \end{aligned} \tag{3.5}$$

with $(v, w) \in M_C$ and $v = \min \zeta_l \cap \pi_T(C)$.

Conclusion Let $\| C \|_\infty = cost_C(x)$. WLOG $x \in V_T$. Then $x \in \zeta_l$ for some $l \in L_T$. By applying Equation (3.5) we obtain the result.

□

■

Proof of Theorem 3.44.

By Section 3.6.1 we can consider only weighted trees such that $T_2 = T$.

1. Given $\lambda \in [0, 1]$ and $T \in \mathcal{T}$ we define $\lambda \cdot T$ to be the tree obtained by shrinking each edge of T by a factor of λ i.e. if w' is the weight function of $\lambda \cdot T$, we have $w'(v) = \lambda \cdot w(v)$.

   Now consider $M \in \mathrm{Mapp}(T, T')$. We call M_λ the same mapping but inside $\mathrm{Mapp}(\lambda \cdot T, \lambda \cdot T')$. Since we can take λ outside the cost of all the edits, then $cost(M_\lambda) = \lambda \cdot cost(M)$. In other words $d_E(\lambda \cdot T, \lambda \cdot T') \leq \lambda \cdot d_E(T, T') < d_E(T, T')$.

   This of course implies that $F : [0, 1] \times \mathcal{T} \to \mathcal{T}$, such that $F(t, T) = t \cdot T$ is continuous, and so $\mathcal{T}$ is contractible.

2. We prove that given for any tree T and any $\varepsilon > 0$, $B_\varepsilon(T)$ is not complete. Build the following Cauchy sequence: let $T_0 = T$ and obtain the tree T_n by attaching to a leaf of T_{n-1} a pair of edges, each with length $\frac{\varepsilon}{2 \cdot 2^n}$. Consider $n > m$, then

$d_E(T_n, T_m) \leq \varepsilon \sum_{k=m}^{n} 2^{-k}$ which is arbitrarily small as n, m increase. Thus $\{T_n\}$ is a Cauchy sequence. Moreover $d_E(T, T_n) \leq \varepsilon \sum_{k=0}^{n} 2^{-k} \leq \varepsilon$ and thus $\{T_n\} \subset B_\varepsilon(T)$.

Suppose that exists T' such that $T_n \to T'$. By construction we know $\#V_{T_n} \to \infty$. Let $M_n \in \text{Mapp}(T_n, T')$ minimizing mapping. Clearly $cost(M_n) \to 0$. For ease of notation we call $C_n := C(M_n)$ and $D_n := (M_n)_D$.

By definition $\#C_n \leq K = dim(T')$. We know that all the vertices in $V_{T_n} - V_T$ have one sibling since they are added as couples. This means that, even if we take away K vertices, at least one out of every pair of the remaining siblings will be deleted once $dim(T_n) > dim(T')$. Of course $cost(M_n) \geq cost(C_n) + cost(D_n)$ and so $cost(D_n) \to 0$.

But $cost(D_n) > \varepsilon \cdot \frac{1}{2} \sum_{K}^{n} \frac{1}{2^n}$ and so it does not go to zero.

This shows that $C = \overline{B_\varepsilon(T)}$ is not compact. In fact $\{T_n\} \subset C$ is a Cauchy sequence but it is not converging.

3. We need to prove local compactness and we do so via sequential compactness.

   Consider $T \in \mathcal{T}^N$ and $\{T_n\} \subset \overline{B_\varepsilon(T)}$ with $\varepsilon < min_{v \in V_T} w_T(v)$.

   If we consider an edge $l = (v, v')$ in T, along the sequence T_n we know that l will never get wholly deleted. It might be split, shrunk but it will never disappear.

   We fix a sequence of mappings M_n such that $M_n \in \text{Mapp}(T, T_n)$ and $cost(M_n) = d_E(T, T_n)$. For any edge $l = (v, v')$ in T, with $v < v'$, we consider the set $E_n^l = \{e_n^k\}$ such that $e_n^k \in M_n$ edits the edge l. With that we mean: shrinkings of l, splittings inserting points w with $v < w < v'$ (note that these appear only after the shrinkings), insertion of edges in points w with $v \leq w \leq v'$. Of course for each n, we might have e_n^1, e_n^2,... acting on the same edge. While there can be at most one shrinking on l for each n, there can be multiple insertions or splittings; of course this number is uniformly bounded because of the dimension constraints.

   For a fixed n we have a natural order between splittings, given by the height at which the new point is inserted, and a similarly induced partial ordering between insertions. Insertion are comparable with respect to this partial order if they happen at different heights. We fix a random order on insertions happening at the same vertex so that all insertions are completely ordered. For a fixed l we partition these edits into Sh^n, S_k^n and $I_{k',k}^n$, which, for each n, collects the elements in E_n^l which are respectively the shrinking, the k-th splitting and the k-th insertion at the k'-th inserted vertex. One can set $k' = 0$ being the index of v, and $k' = -1$ being the index of v'.

   We know that for each edge l:

   - $Sh = \cup_n Sh^n$ is at most countable, and the sequence given by the different weights of l obtained through the shrinkings, is a sequence in $[L - \varepsilon, L + \varepsilon]$, with L the original length of l. Then we can extract a converging subsequence.
   - $S_k = \cup_n S_k^n$ is at most countable, then the ratio of the distance in height between v and the splitting point, over the length of l (after the shrinking of the n-th mapping) form a sequence in $[0, 1]$. So we can extract a converging subsequence in $[0, 1]$. In other words we can find a subsequence of edits which converges to a certain splitting.
   - $I_{k',k} = \cup_n I_{k',k}^n$ is at most countable for every k, k'; the length of the inserted edges form a sequence in $[0, \varepsilon]$. So for every k, k' we can extract a converging subsequence.

If some of the sets defined above are finite we discard from the sequence the indexes n which appear in the collection of edit, being it Sh, S_k or $I_{k,'k}$. Note that, by the dimension bounds, we have a finite number of such sets, for a finite number of edges.

Thus, for every edge l, we proceed in this way:

(a) since Sh is countable we extract a converging subsequence of shrinkings.

(b) starting from $k = 1$, from the subsequence just obtained we extract a converging subsequence of splitting locations. Starting from this sequence we repeat the extraction procedure for $k = 2$, then recursively till we reach k such that S_k is empty. Again if, for some k, S_k is finite, we simply discard elements of the subsequence appearing in S_k.

(c) lastly, from the last set of indexes we obtained, we extract from $I_{k',k}$ a converging subsequence of insertions for every k and k'; working on k and k' as in the previous point. In a finite number of steps we reach $I_{\widehat{k'},\widehat{k}}$ such that for all $k' > \widehat{k'}$ and $k > \widehat{k}$, $I_{k,k'} = \emptyset$.

Given any ordering on the edges, we recursively apply this for every l, and obtain a subsequence which, with an abuse of notation, we still call $\{T_n\}$.

We fix n and take for every edge the edits in $Sh^n \cup_k S_k^n \cup_{k',k} I_{k',k}^n$. If we take the uninio of such edits over the edges l we obtain a mapping N_n for every n. The mapping N_n is defined on T and by construction is contained in M_n. Note that N_n collects all the edits which are applied by M_n on T that is all the shrikings and splitting, plus all the insertions which are made directly on the vertices of T or on the splittings already applied. In other words $M_n - N_n$ is made by insertions which are applied on edges which are obtained by other insertions.

Call ST_n the tree obtained from T with the edits in N_n. Then clearly $N_n \in \mathrm{Mapp}(T, ST_n)$. For what we have said, T_n can be obtained from ST_n with the edits in $M_n - N_n$, which are only a particular case of insertions.

By construction, any edit in N_n is part of a converging sequence of edits: shrinking converge to a final weight value, splittings converge to a final position on the edge and insertions converge to a final weight of the inserted edge. We call $\overline{N}$ the mapping obtained with the limit of the edits of N_n. Clearly $cost(N_n) \to cost(\overline{N})$. Let ST be the tree obtained from T with $\overline{N}$. We have $ST_n \to ST$. In fact, consider one edge $l \in E_T$. Take for instance the sequence of shrinkings in $Sh = \{e_1, \dots, e_n, \dots\}$ parametrized such that $w_{ST_n}(l) = e_n$ and $e_n \to e$. For any fixed ϵ and for n big enough we know $|e_n - e| < \epsilon$. Thus $|w_{ST_n}(l) - w_{ST_{n+1}}(l)| < \epsilon$ is a shrinking with cost less then ϵ. For the same edge l there are at most N splittings, each edit splits l in E_{ST_n} at a certain height. Similarly, the difference in heights between splittings in S_k^n and S_k^m with $m > n$ is going to zero, and thus we can again choose n big enough so that the $k-$th splitting of ST_n can be turned in the $k-$th splitting of ST, for all k, with cost less than ϵ. The same reasoning can be done on the weights of the insertions. Thus we can go from ST_n to ST with a finite set of edits, with cost less that ϵ and whose number is uniformly bound. For instance we know that on every tree we can have at most N shrinkings, N ghostings, and N insertions. Thus, for every ϵ, there is n big enough such that $d_E(ST_n, ST) < 3N\epsilon$. We remark that, since $N_n \subset M_n$, then $cost(N_n) \leq \varepsilon$ for every n, and thus $d_E(T, ST) \leq \varepsilon$.

At this point we have obtained a sequence $\{ST_n\}$ such that: $ST_n \to ST$ in $\overline{B_\varepsilon(T)}$ and $V_{ST_n} \subset V_{T_n}$. Then next natural step is to recursively "enlarge" ST_n.

Take each M_n and substitute each edit in $N_n \subset M_n$ with its limit $\overline{N}$, obtaining $\overline{M_n}$. Note that $cost(M_n) \to cost(\overline{M_n})$ as n grows.

We can obtain a new sequence of trees T'_n and mappings $M'_n \in \text{Mapp}(ST, T'_n)$ in this way:

- T'_n is obtained from T applying the mapping $\overline{M_n}$.
- $M'_n \in \text{Mapp}(ST, T'_N)$ is the mapping induced by the identifications given by the composition of the paths: $T \xrightarrow{\overline{N}} ST \xrightarrow{\overline{M_n}-\overline{N}} T'_n$; which is well defined since $\overline{M_n} - \overline{N}$ is made only by insertions to be done on ST. Thus we can identify M'_n as given by the "identity" on the vertices of ST, and the edits in $M_n - N_n$, which, as already noted, are only insertions. We call id_{ST} the set of couplings embedding ST into T'_n.

The first thing we highlight is that, since $M_n \to \overline{M_n}$, in the sense that their edits converge as already explained, then $d_E(T_n, T'_n) \to 0$. So in $\{T'_n\}$ converges to a limit point, so does $\{T_n\}$. For the same reason we have that also $\{T'_n\}$ is a Cauchy sequence.

We can perform the steps which lead us to obtain $\{ST_n\} \to ST$ starting from T and the sequence $\{T_n\}$, but taking ST as "reference tree" and $\{T'_n\}$ as initial sequence. So we obtain the subsequence of $\{ST_n\}$ which we use to build sequence $\{ST'_n\} \to ST'$ along with mappings $N'_n \in \text{Mapp}(ST, ST'_n)$. For notational convenience, any time we throw away indexes from $\{T'_n\}$, we remove the same indexes also from $\{T_n\}$.

The key point is that $id_{ST} \subset N'_n$ for all n and, in particular, we have $ST \neq ST'_n$ if and only if $id_{ST} \subsetneq N'_n$. Since $M'_n - N'_n$ contains only insertions to be applied on edges which have already been inserted in ST, if there are no such edges, we have $ST = T'_n$.

Now we relate $\{ST'_n\} \to ST'$ with T and $\{ST_n\} \to ST$. To do so, consider: $A_n = (N'_n - id_{ST}) \bigcup N_n$. By construction $A_n \in \text{Mapp}(T, ST'_n)$. For what we have said, either $N_n \subsetneq A_n$ or $ST = T'_n$ and so $A_n = N_n = M_n$. On top of that $A_n \subset M_n$ and thus $d_E(T, ST') \leq \varepsilon$. Now, call SST_n the weighted tree obtained by editing T with the edits contained in A_n: either $V_{ST_n} \subsetneq V_{SST_n}$ or $SST_n = T_n$. Moreover, since $N_n \to \overline{N}$ and $ST'_n \to ST'$ we get $SST_n \to ST'$.

Thus, this further step has produced a a sequence of weighted trees $\{SST_n\}$ "bigger" than $\{ST_n\}$ and such that $SST_n \to ST'$ with $d_E(T, ST') \leq \varepsilon$.

We can recursively repeat this procedure and obtain a converging sequence G_n which, after a finite number of steps (thanks to the bound on the edge number), will have $G_n = T_n$ for all n; and G_n converges inside $\overline{B_\varepsilon(T)}$.

■

Proof of Theorem 3.45.

It's enough to prove it for $(\mathcal{T}^N, d_E)$.

Consider a Cauchy sequence $\{T_n\}_{n\in\mathbb{N}}$. By Proposition 3.43, $\big|||T_n|| - ||T_m||\big| \leq d_E(T_n, T_m)$. Thus, $\{||T_n||\}$ is a Cauchy sequence and thus, it converges in $\mathbb{R}$. In other words, $||T_n|| \to C$. If $C = 0$ then $T_n \to 0$, the tree with one vertex and no edges. Therefore, we can suppose $C > 0$.

Define $\epsilon(T) := \min_{e\in E_T} w_T(e)$. In the proof of Theorem 3.44 we show that $\overline{B_r(T_m)}$ is compact for every $r < \epsilon(T_m)$. We know $\{\epsilon(T_n)\}$ is a bounded sequence in $\mathbb{R}$ and thus, up to taking a subsequence, it converges. If $\epsilon(T_n) \to q > 0$ then, the sequence $\{T_n\}$ is

definitely in a compact ball of the form $\overline{B_{\epsilon(T_m)}(T_m)}$. In fact, fix $0 < \lambda$ such that $\lambda N < q$; we can find m such that $(|\epsilon(T_m) - q| + \lambda)N \leq \epsilon(T_m)$ and $d_E(T_m, T_n) \leq \epsilon(T_m)/2$ for all $n > m$. Thus we can build $\widehat{T}_m$, obtained from T_m by raising the weight of its maximal edges by $|\epsilon(T_m) - q| + \lambda$, so that $q < \epsilon(\widehat{T}_m)$. We now that, at most, we need to shrink N edges by $|\epsilon(T_m) - q| + \lambda$. Thus:

$$d_E(T_m, \widehat{T}_m) < N(|q - \epsilon(T_m)| + \lambda) \leq \epsilon(T_m)$$

Moreover for $n > m$ we have $d(\widehat{T}_m, T_n) \leq d_E(\widehat{T}_m, T_m) + d_E(T_m, T_n) \leq \epsilon(T_m)$ and so $\{T_n\}$ converges.

Suppose then $\epsilon(T_n) \to 0$. For all n, take T_n, and obtain T_n^1 by deleting $\operatorname{argmin}_{e \in E_{T_n}} w_{T_n}(e)$. By construction we know $d_E(T_n, T^1, n) \leq N\epsilon(T_n)$ and thus $d(T_n, T_n^1) \to 0$. So, if $\{T_n^1\}$ converges, also $\{T_n\}$ converges. We repeat the same reasoning as above, considering $\{\epsilon(T_n)\}$; if $\epsilon(T_n^1) \to q > 0$ we are done, otherwise we take $\{T_n^1\}$ and obtain $\{T_n^2\}$ removing the smallest edge and repeat again the procedure. BY $\#E_{T_n} \leq N$, we know $\#E_{T_n^1} \leq N - 1$ and so $\#E_{T_n^j} \leq N - j$ etc. Since $\| T_n^j \| \to C$, $0 < \#E_{T_n^j}$ and thus in a finite number of step we obtain $\{T_n^j\}$ which converges and so does $\{T_n\}$.

■

Proof of Lemma 3.56.

Due to the properties of M, in particular property (M3), M edits $sub_T(v)$ so that it becomes $sub_{T'}(w)$. In other words $M_{|(v,w)} = \{(v', y) \in M \big| v' < v\} \bigcup \{(x, w') \in M \big| w' < w\}$ is again a mapping and the costs of the single edits does not change between going from M to $M_{|(v,w)}$.

Now we turn to the case of $v' > v$, $[v, v'] \subset M_D$. We need to define $M_{|(v',w)}$.

To build $M_{|(v',w)}$ we take $(M_{|(v,w)} - \{(x, y) \,|\, v' \leq x <\leq v\}) \bigcup \{(v', w)\}$. Note that $M_{|(v',w)}$ is indeed a mapping, since all the vertices $[v', v]$ are deleted or ghosted by M and so are not couple by any vertex in $V_{T'}$. Lastly, for all couples $(a, b) \in M_{|(v',w)}$ such that $(a, b) \neq (v', w)$, $cost_M((a, b)) = cost_{M_{|(v',w)}}((a, b))$.

■

Proof of Lemma 3.57.

Suppose $a = \sum_{e \in [w_1, w_2]} w_{T'}(e)$, $b = \sum_{e \in [w_3, w_4]} w_{T'}(e)$ which in M are coupled with sequences of T with length A and B respectively, and thus contributing to the cost of the mapping with the quantity $|A - a| + |B - b|$. If in M' they are both deleted, the contribute to that cost by: $a + b$. This situation gives the following set of equations:

$$a, b, A, B > 0$$

$$A, B > m_T > 0$$

$$|A - a| + |B - b| < m_T$$

$$a + b < m_T$$

This system of course has solution only if $m_T > a$ and $m_T > b$, and so it becomes:

$$a, b, A, B > 0$$

$$A, B > m_T > 0$$

$$A + B - m_T < a + b$$

$$a + b < m_T$$

which is impossible since it gives:

$$2m_T < A + B < 2m_T$$

The roles of M and M' can of course be reversed.

■

Proof of Lemma 3.58.

Consider $M, M' \in \text{Mapp}(T', T)$ with $cost(M), cost(M') < m_T$.

Suppose there is a sequence of edges $[w, w']$ in T' such that $(w, "D") \in M'$, $(w, x) \in M$ for some $x \in V_T$, and $w'' \in M_G \bigcap M'_G$ for all $w < w'' < w'$. By Lemma 3.57 we know that this is not happening for any other sequences of edges. Note that, by hypothesis, w' and w are not ghosted by M.

Apply on T' all the deletions in $M_D \cap M'_D$ to be applied on it, obtaining T''. We induce in a natural way mappings N and N' from T'' to T, simply removing, respectively, from M and M' the deletions already done. Note that the sequences paired in T' are paired also in T''. Because $m_T \geq m_{T''}$, then Lemma 3.58 holds also for T'' and the mappings N and N'. As a consequence, since we already have $[w, w']$ which is paired by one mapping and deleted by the other, all the other disjoint sequences of edges in T'' still to be deleted by N, cannot be paired by N' (nor can they be even partially deleted by N', since these deletions do not lie in $M_D \cap M'_D$). Thus, any time a sequence $[v, v']$ in T'' is deleted by N_D, if after N'_D it becomes a continuous sequence, then it cannot be maximal, i.e. at least one between v and v' is an order two vertex. Otherwise $[v, v']$ would be paired since none of its extremes can be ghosted. In particular, this implies that each deletion we have in N_D or N'_D deletes an edge of T'' which is left with at least one order 2 vertex by the deletions of the other mapping.

To recap, we have obtained T'' by applying all the deletions shared by M and M', we have induced the mappings N and N' taking the remaining edits of, respectively, M and M'; each deletion we still have in N_D or N'_D deletes an edge of T'' which is left with at least one order 2 vertex by the deletions of the other mapping. For both mappings there are no insertions to be done to obtain T, because their cost would be over $\mathcal{K}_T$.

Consider a vertex v' in T''. The order of v' can change thanks to the deletions of N_D or N'_D obtaining different values in T' only if we have some sequence $[v, v']$ which is deleted by one mapping and not by the other.

So consider $[v, v']$ a sequence of edges in T'' which is deleted by N and not by N'. and suppose that v' is not of order 2 in T'' but it becomes of order two after N'_D. We can clearly take $[v, v']$ so that it becomes continuous after N'_D. Let $v_1, .., v_n$ be the children of v' in T''. We know that $sub_i(v')$ is deleted by N'_D for every i but one, be it v_h, such that $v_h \geq v$. This in turns tells us that $sub_i(v')$ with $i \neq h$ are not deleted by N (otherwise these deletions would lie in $M_D \cap M'_D$) and, for what has been said before, all their edges must have at least one vertex of order two in T''. According to the mapping N, once all order 2 vertices in $sub_i(v')$, $i \neq h$, have been removed, all the maximal sequences remaining pass from being deleted by N' (and so by M') to being coupled by N (and so by M). But since the only sequence for which this happens is $[w, w']$, this means that $i \leq 2$, $w' = v'$, $v_1 \geq v$ and $v_2 \geq w$.

Putting the pieces together first we proved that, starting from T'' any further deletion in N_D or N'_D deletes an edge of T'' which is left with at least one order 2 vertex by the deletions of the other mapping. But now we saw that there can be at most one vertex which goes from order different from 2 in T'', to order 2 thanks to N_D or N'_D. And this vertex is w. As a consequence, for any other edge of T'' deleted by N and not by N', the extreme of order two after N'_D can have no siblings in T'', i.e. they are already of order two. And the same reversing the role of N and N'.

To sum up, we have obtained T'' by applying all the deletion shared by M and M', and then we proved that all other deletions but one, delete edges with at least one of the extremes which is an order 2 vertex in T''. So, apart from the deletion of $[w, w']$ by N'_D, the others in N_D or $N'_D - \{(w, "D")\}$ provide no changes in the tree structures, up to order 2 vertices. So the tree structure obtained from the deletions of such edges or the one resulting from the ghosting of their order two extremes is the same, up to order two vertices. Since T has no order 2 vertices, and since there are no insertions to be done on T'', $[w, w']$ is paired with an edge with no vertices of order two, and so its deletion changes the topology of the tree T''. In other words we obtain the same tree structure, which is the one of T, both by removing from T'' the order 2 vertices, and from first removing the order 2 vertices and then deleting the internal edge $[w, w']$ which is absurd.

■

Proof of Lemma 3.59.

- we know that, by definition of $\mathcal{K}_T$, we have $d_E(sub_T(x), sub_T(x')) > \mathcal{K}_T$, and so, thanks to Lemma 3.56 we have

$$\mathcal{K}_T < d_E(sub_{T'}(v), sub_T(x')) + d_E(sub_{T'}(v), sub_T(x')) \leq \mathcal{K}_T$$

  which is absurd. Thus $x = x'$.

- Write down the sequence $[v, v']$ as an ordered sequence of vertices $[v, v'] = \{a_1 = v, \dots, a_r\}$ with $father(a_r) = v'$. By construction, $[v, v']$ cannot be wholly deleted by both mappings, and so each a_i can be deleted or ghosted by N or N', but there must be a vertex a_{i_N} which is coupled by N and one $a_{i_{N'}}$ coupled by N'. In fact if there is one mapping which wholly deletes $[v, v']$ then the other one must couple it, by construction. Which contradicts Lemma 3.58. We want to prove that $(a_{i_N}, x) \in N$ and $(a_{i_{N'}}, x') \in N'$ with $x = x'$. If $i_N = i_{N'}$ we already proved that $x = x'$. Then, suppose $i_N < i_{N'}$.

  Since $v < a_{i_{N'}} < v'$ then $a_{i_{N'}}$ is an order two vertex, coupled with x' which is not an order two vertex. Thus the vertices between v and $a_{i_{N'}}$ must be deleted, a_{i_N} included. This in particular means that we can consider $N_{|(a_{i_N}),x}$ and $N'_{|(a_{i_N}),x'}$ as in Lemma 3.56. Using the triangular inequality on $d_E(sub_T(x), sub_T(x'))$, if at least one between x and x' is not a leaf, we obtain:

$$\mathcal{K}_T < d_E(sub_{T'}(a_{i_N}), sub_T(x')) + d_E(sub_{T'}(a_{i_N}), sub_T(x)) \leq \mathcal{K}_T$$

  Suppose now that there are $(a_{i_{N_1}}, x_1) \in N$ and $(a_{i_{N_2}}, x_2) \in N$. With $a_{i_{N_j}} \in [v, v']$. Suppose $a_{i_{N_1}} > a_{i_{N_2}}$. Then $a_{i_{N_1}}$ is an order 2 vertex in T'' and it is paired with a vertex of order different from 2. Since it cannot be deleted, all the vertices in $[v, v']$ which are below $a_{i_{N_1}}$ must be deleted, so that the order of $a_{i_{N_1}}$ is no more 2. But $a_{i_{N_2}}$ cannot be deleted, which is absurd. Alternatively, from Lemma 3.58 we see that there is a $1:1$ correspondence between maximal continuous sequences of edes in T'' and the set of vertices corresponding to E_T. Since in every sequence there is at least one coupled vertex, the cannot be more that one, because that would break the correspondence.

- consider x an internal vertex of T. We know that there is a vertex $\tilde{v}$ and a maximal continuous sequence $[v, v'] \subset E_{T''}$ such that $\tilde{v} \in [v, v']$ and $(\tilde{v}, x) \in N$. We know by the previous point that there is also $\hat{v} \in [v, v']$ such that $(\tilde{v}, x') \in N'$ for some x'. But since x is not a leaf, $x = x'$. By construction either $\tilde{v} \leq \hat{v}$ or $\tilde{v} > \hat{v}$ hold. Suppose $\tilde{v} \leq \hat{v}$, then $[\tilde{v}, \hat{v}]$ is a continuous sequence.

- We know that $(v_x, father(x)) \in N$ and $(v_{x'}, father(x')) \in N'$ for some $v_x, v_{x'} > v$. Suppose $v_x < v_{x'}$ and there is a vertex $v' \in V_{T''}$ with $v_x < v' < v_{x'}$ and order different from 2. Then $(v', "D") \in N'$. In fact v' cannot be ghosted and it cannot be coupled by N' for it would be paired with a vertex in between x' and $father(x')$, which is absurd.

  Thus there are $v_1, v_2, v_3 \in V_{T''}$ which give the following maximal continuous sequences of edges: $[v_1, v_2]$ which contains v, $[v_2, v']$ which contains v_x and $[v', v_3]$ which contains $v_{x'}$. Thus, for the previous points of this lemma, there is $(v_{N'}, x'') \in N'$ with $v_{N'} \in [v_2, v']$. But since $v < v_{N'} < v_{x'}$ we have $x' < x'' < father(x')$ which is absurd. Then v_x and $v_{x'}$ are part of the same continuous maximal sequence in T'', and thus, by the previous points of this lemma, $father(x) = father(x')$.

■

Proof of Theorem 3.60.

We divide this proof in three steps: first we define M'', then we check that it is a mapping and lastly we verify that it is still a minimal mapping. Let T'' be the tree obtained from T' applying the deletions in $M_D \bigcap M'_D$. Induce mappings N and N' from T'' to T removing respectively from M and M' the deletions already done.

- Clearly $M''_D = M_D \bigcap M'_D$. Then we have no choice but ghosting all order 2 vertices of T''. We need to define the shrinking operations to obtain T. Each of the vertices in T'' with order different from 2 are coupled by at least one between N and N'. In fact they cannot be ghosted because to do so we would have to completely delete some sequence of edges contradicting Lemma 3.58. Consider $v \in V_{T''}$. Suppose $(v, x) \in N$. If $v \notin \pi_{T'}(C(N'))$ then we add (v, x) to M''. If $(v, x') \in N'$, x or x' are not leaves, then $x = x'$ (and so both of them are internal vertices). Thus we can add again (v, x) to M''.

  Note that, up to here, M'' is uniquely determined in order to avoid further deletions, thanks to Lemma 3.59. In fact deletions in T' are fixed by hypothesis, ghostings are fixed as well since all order 2 vertices must be removed in order to be able to couple vertices with T without further deletions. Then also shrinkings are fixed by Lemma 3.59.

  The only situation we are left with is $(v, x) \in N$ and $(v, x') \in N'$, with $x \neq x'$ leaves. By Lemma 3.59 we know that x and x' are siblings, i.e. $father(x) = father(x') = x''$. Note that also v must be a leaf in T'', otherwise there would be $v' < v$ coupled with some vertex below x, which is absurd. Consider $v'' = \{\tilde{v} > v \big| ord(\tilde{v}) \neq 2\}$. We know $(v'', x'') \in M''$. Then $(v'', x'') \in N$ or/and $(v'', x'') \in N'$. By Lemma 3.59 we know that there is a correspondence between the maximal continuous sequences of $sub_{T''}(v'')$ of the form $[v_i, v'']$, such that v_i is a leaf, and the leaves x_i of $sub_T(x'')$. Suppose $(v'', x'') \in N$. The for each i there is $(v'_i, x_i) \in N$, for $v'_i \in [v_i, v'']$. We add to M'' the couples (v_i, x_i) for every i. In this way we define M'' for every vertex in T' and T.

- We verify that M'' is a mapping.

  (M1) M'' contains all the deletions needed to obtain T''. By Lemma 3.59 every mapping M and M' induces a $1:1$ correspondence between maximal continuous sequences of T'' and vertices of T. Note that by Lemma 3.59, on internal vertices, M and M' induce the same correspondence. To build M'' we used those correspondences to pick one point in every maximal sequence and ghost the

other. On leaves we used either M or M' to complete M''. This guarantees that we reach all vertices of T' and T'' with an edit.

(M2) For vertices which are deleted or ghosted we are sure that there is only one edit associated to them. But this is true also for vertices in $C(M'')$ thanks again to the $1:1$ correspondence between maximal continuous sequences of T'' and vertices of T.

(M3) We can say that $[v, v'] < [v'', v''']$ if $v' < v''$. Note that this holds for any couple of vertices in the sequences, since the sequences are completely ordered. With this notation we see that the correspondences $\{\text{maximal continuous sequences of } T''\} \leftrightarrow V_T - \{r_T\}$ induced by N and N' are ordered correspondences because N and N' are mappings. Thus, given $(v, x), (v', x') \in M''$, we have $v < v'$ if and only if $[v, \tilde{v}] < [v', \hat{v}]$ if and only if $x < x'$.

(M4) this holds by construction since we ghost only vertices which are of order 2 in T''.

- Now we prove that $cost(M'') \leq cost(M), cost(M')$. Clearly the cost of the deletions does not increase: $M''_D = M_D \bigcap M'_D$ and thus $cost(M''_D) < cost(M_D), cost(M'_D)$. We need to check the elements in $C(M'')$. Let $(v, x) \in C(M'')$. The couple (v, x) couples a sequence $[v, v']$ with the edge $(x, father(x))$. Note that the mappings M and M' still turn the sequence $[v, v']$ in the edge $(x, father(x))$ but without coupling them, because they apply some deletions on $[v, v']$. It is enough to prove that the cost with which M'' turns $[v, v']$ in $(x, father(x))$ is not greater than the one of M and M'.

  Write down the sequence $[v, v']$ as an ordered sequence of vertices $[v, v'] = \{a_1 = v, \dots, a_r\}$ with $father(a_r) = v'$. We have:

$$|\sum_{i=1}^{r} w_{T'}(a_i) - w_{T'}(x)| < |\sum_{i \in A} w_{T'}(a_i) - w_{T'}(x)| + \sum_{i \in B} w_{T'}(a_i)$$

  for any A, B partition of $\{1, \dots, r\}$. And thus the cost of turning $[v, v']$ into $(x, father(x))$ using M'' not greater than the one of M and M'.

■

Proof of Corollary 3.62.

Suppose we have $(l, w), (l', w') \in M$ and $(l, w'), (l', w) \in M'$. Then we would have:

$$d_E(T, T') \geq |A_l - B_w| + |A_{l'} - B_{w'}| = |A_l - B_{w'}| + |A_{l'} - B_w|$$

with A_l and $A_{l'}$ being the weights of l and l' and B_w, $B_{w'}$ being the weights of w and w' after the deletions in M and M' (which are shared). Similarly:

$$2d_E(T, T') < |A_l - A_{l'}| \leq |A_l - B_w| + |A_{l'} - B_w|$$

and

$$2d_E(T, T') < |A_l - B_{w'}| + |A_{l'} - B_{w'}|.$$

Putting the things together we obtain:

$$4d_E(T, T') < |A_l - B_{w'}| + |A_{l'} - B_{w'}| + |A_l - B_w| + |A_{l'} - B_w| \leq 2d_E(T, T').$$

Which is absurd. ■

4. Functional Data Representation with Merge Trees

ABSTRACT

In this chapter we face the problem of representation of functional data with the tools of algebraic topology. We represent functions by means of merge trees and this representation is compared with that offered by persistence diagrams. We show that these two structures, although not equivalent, are both invariant under homeomorphic re-parametrizations of the functions they represent, thus allowing for a statistical analysis which is indifferent to functional misalignment. We employ a novel metric for merge trees and we prove some theoretical results related to its specific implementation when merge trees represent functions. To showcase the good properties of our topological approach to functional data analysis, we test it on the Aneurisk65 dataset replicating, from our different perspective, the supervised classification analysis which contributed to make this dataset a benchmark for methods dealing with misaligned functional data.

4.1 Introduction

Since the publication of the seminal books by Ramsay and Silverman (Ramsay and Silverman, 2005) and Ferraty and Vieu (Ferraty and Vieu, 2006), Functional Data Analysis (FDA) has become a staple of researchers dealing with data where each statistical unit is represented by the measurements of a real random variable observed on a grid of points belonging to a continuous, often one dimensional, domain D. In FDA these individual data are better represented as the sampled values of a function defined on D and with values in $\mathbb{R}$. Hence, at the onset of any particular functional data analysis stands the three-faceted problem of *representation*, described by: (1) the smoothing of the raw and discrete individual data to obtain a functional descriptor of each unit in the data set, (2) the identification of a suitable embedding space for the sample of functional data thus obtained and, finally, (3) the eventual alignment of these functional data consistently with the structure of the embedding space. As a reference benchmark of the typical FDA pipeline applied to a real world dataset, we take the paper by Sangalli et al. (2009b) where the first functional data analysis of the AneuRisk65 dataset (https://statistics.mox.polimi.it/aneurisk) is illustrated.

Smoothing is the first step of a functional data analysis. For each statistical unit, individual raw data come in the form of a discrete set of observations regarded as partial observations of a function. Smoothing is the process by means of which the analyst generates the individual functional object out of the raw data. This functional object will be the atom of the subsequent analysis, a point of a functional space whose structure is apt to sustain the statistical analysis required by the problem at hand. A common approach to obtain functional representations is to fit the data with a member of a finite dimensional functional space generated by some basis, for instance, splines or trigonometric polynomials. Signal-to-noise ratio and the degree of differentiability required for the functional representation, as well as the structure of the embedding space, drive the smoothing process. Functional representations interpolating the raw data are of no practical use when

the analysis requires to consider functions and their derivatives or, for instance, the natural embedding space is Sobolev's; see, for instance, Sangalli et al. (2009a) for a detailed analysis of the trade-off between goodness of fit and smoothness of the functional representation when dealing with the Aneurisk65 dataset.

Functional data express different types of variability (Vantini, 2009) which the analyst might want to decouple before carrying out the statistical analysis. Indeed the Aneurisk65 dataset is by now considered a benchmark for methods aimed at the identification of *phase* and *amplitude variation* (see the Special Section on Time Warpings and Phase Variation on the Electronic Journal of Statistics, Vol 8 (2), and references therein). In many applications phase variation captures ancillary non-informative variability which could alter the results of the analysis if not properly taken into account (Lavine and Workman, 2008; Marron et al., 2014). A common approach to this issue is to embed the functional data in an appropriate Hilbert space where equivalence classes are defined, based on a notion of *alignment* or *registration*, and then to look for the most suitable representative for any of these classes (Marron et al., 2015). Such approach evokes ideas from shape analysis (Dryden and Mardia, 1998) and pattern theory (Ripley and Grenander, 1995), where configurations of landmark points are identified up to rigid transformations and global re-scalings. In close analogy with what has been done for curves (Michor et al., 2007; Srivastava et al., 2010), functions defined on compact real intervals D are aligned by means of warping functions mapping D into another interval; that is, they are identified up to some re-parametrization. Different kinds of warping functions have been investigated: affine warpings are studied for instance in Sangalli et al. (2010) while more general diffeomorphic warpings have been introduced in Srivastava et al. (2011b). Once the *best* representatives are selected, the analysis is carried out on them leveraging the well behaved Hilbert structure of the embedding space. Classically, the optimal representatives are found by minimizing some loss criterion with carefully studied properties (Sangalli et al., 2014). This approach however has some limitations, arising from the fact that the metric structure of the embedding space might not be compatible with the equivalence classes collecting aligned functions (Yu et al., 2013). An alternative is to employ metrics directly defined on equivalence classes of functions such as the Fisher Rao metric, originally introduced for probability densities (Srivastava et al., 2007), which allows for the introduction of diffeomorphic warpings (Srivastava et al., 2011b). It must be pointed out that all these ways of dealing with the issue of ancillary phase variability encounter some serious challenges when the domain D is not a compact real interval.

A different approach to the problem of phase variation is to capture the information content provided by a functional datum by means of a statistic which is insensitive to the function re-parametrization, but sufficient for the analysis. Algebraic topology can help since it provides tools for identifying information which is invariant to deformations of a given topological space (Hatcher, 2000). Topological Data Analysis (TDA) is a quite recent field in data analysis and consists of different methods and algorithms whose foundations rely on the theory developed by algebraic topology (Edelsbrunner and Harer, 2008). The main source of information collected by TDA algorithms are homology groups (see, for instance, Hatcher (2000)) with fields coefficients which, roughly speaking, count the number of holes (of different kinds) in a topological space. For instance zero dimensional holes are given by path connected components and one dimensional holes are given by classes of loops (up to continuous deformations) which cannot be shrunk to one point. One of the most interesting and effective ideas in TDA is that of *persistent homology* (Edelsbrunner et al., 2002): instead of fixing a topological space and extracting the homology groups from that space, a sequence of topological spaces is obtained along various pipelines, and the evolution of the homology groups is tracked along this sequence. The available pipelines are many, but the one which is most interesting for the purposes of this work is that con-

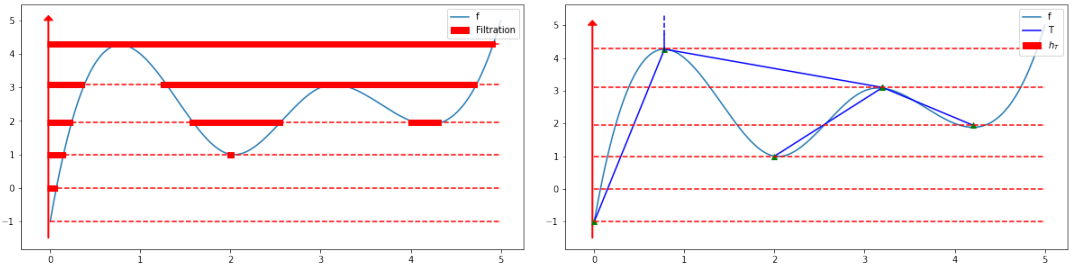
cerning real valued functions. Let the domain D be a topological space X and consider a real valued function defined on X, $f : X \to \mathbb{R}$. One can associate to f the sequence of topological spaces given by the sublevel sets $X_t = f^{-1}((-\infty, t])$, with t ranging in $\mathbb{R}$. The evolution of the connected components along $\{X_t\}_{t\in\mathbb{R}}$ is thus analysed for the purpose of generating a topological representation of f.

In this work, we consider specific topological representations of f constructed along this general scheme and we show that they are invariant with respect to homeomorphic warpings of the domain X. Moreover, these representations are also able to separate big shape features of f from small oscillations; the overall shape of the function captured by the topological representations we will deal with is unaffected by the presence of smaller oscillations, which are captured as well, but separately. These two properties make the TDA approach pursued in this manuscript a candidate for the representation of functional data, indeed a robust competitor able to deal in a natural way with phase variation and insensitive to the fine tuning of the preliminary smoothing phase, since functional features likely generated by overfitted representations are easily identified as ancillary in the subsequent topological representation.

To allow for the statistical analysis of functional data summarised by their topological representations, we need to embed the latter in a metric space. The choice of persistence diagrams (PD) (Cohen-Steiner et al., 2007) as summaries obtained through persistent homology drives many successful applications (Xia et al., 2016; Bhattacharya et al., 2015; Pokorny et al., 2015; Chung et al., 2009; Wang et al., 2018; Kramár et al., 2013), although other topological summaries are in fact known in the literature (Bubenik, 2015; Adams et al., 2017; Chazal et al., 2015). In this work we exploit a topological alternative – not equivalent – to a persistence diagram, called *merge tree.* Merge trees representations of functions are not new (Morozov and Weber, 2013) and are obtained as a particular case of Reeb Graphs (Shinagawa et al., 1991a; Biasotti et al., 2008). Different frameworks have been proposed to work with merge trees (Beketayev et al., 2014; Morozov et al., 2013), mainly defining a suitable metric structure to compare them (Gasparovic et al., 2019; Touli, 2020). However all such metrics have a very high computational cost, causing a lack of examples and applications even when approximation algorithms are available (Touli and Wang, 2018), or they require complex workarounds to be effectively used (Sridharamurthy et al., 2020). We employ the metric for merge trees introduced in Chapter 2, showing that its computational complexity is reasonable when the trees involved are not too large. When working with representations of data, it is fundamental to study the behaviour of the operator which maps the single datum into the chosen representation to assess which kind of information is transferred from the initial data to the space of representations. For this reason we develop a new theoretical analysis on the stability/continuity of merge trees with respect to perturbations of the original functions. We also carry out examples to showcase differences between merge trees and persistence diagrams of functions. Having devoted the initial sections of the chapter to the understanding of the behaviour of these topological tools, we finally tackle, with our TDA approach, the benchmark functional classification case study detailed in Sangalli et al. (2009b). We also complement the classification task with other analyses to better understand the topological description of such data set.

Outline

The chapter is organized as follows. In Section 4.2, we introduce the merge tree representation of a function. In Section 4.3 we briefly recall the definition of persistence diagrams in order to draw, in Section 4.4, some comparison between them and merge trees, before proving the invariance property which holds true for both topological representations. In Section 4.5 we present the metric structure for the space of merge trees which is used



(a) (b)

Figure 4.1.1: Sublevel sets of a function (a); the same function with its associated merge tree (b).

in the examples and in the final case study. In Section 4.6 we investigate the continuity properties of the operator which assigns to a function its merge tree, with respect to the aforementioned metric. Section 4.7 is completely devoted to hands-on examples dealing with further comparisons between PDs and merge trees and pruning.

In Section 4.8, we tackle the functional data classification problem explored in Sangalli et al. (2009b) and we compare their results and conclusions with those obtained following the TDA approach. We complement such results with an unsupervised approach to the same data set and with a brief attempt to interpret the resulting merge trees via some functional statistics. We finally conclude the chapter with a discussion, in Section 4.9, which points out some ideas pertaining our topological approach to functional data analysis.

Section 4.A collects the proofs of the results of the paper.

4.2 Merge Trees of Functions

In this section we define the merge tree representation of a function. Merge trees are an already established tool in topology and, to some extent, also in statistics since clustering dendrograms can be regarded as merge trees. Nevertheless, we are going to spend a few lines to define them, in accordance with the framework of Chapter 2, which differs from the classical one, found, for instance, in Morozov and Weber (2013). Roughly speaking, the pipeline to obtain a merge tree is the following: we transform the given function into a sequence of nested subsets and then we track the topological changes along this sequence. Such information is then turned into a tree.

The details are described in the following subsections.

4.2.1 Sublevel Sets

Consider a function $f : X \to \mathbb{R}$, with X being any topological space. We call sublevel set at height $t \in \mathbb{R}$, the set $X_t := f^{-1}((-\infty, t]) \subset X$. The key property of the family $\{X_t\}_{t\in\mathbb{R}}$ is that such subsets are nested: if $t \leq t'$ then $X_t \subset X_{t'}$. Note that the sequence $\{X_t\}_{t\in\mathbb{R}}$ is fully determined by the shape of the function f; see Figure 4.1.1a. In fact, for $x \in X$, $f(x) = \inf\{t \in \mathbb{R} : x \in X_t\}$, hence no information carried by f is lost by its representation $\{X_t\}_{t\in\mathbb{R}}$.

4.2.2 Path Connected Components

A topological space X is path connected if for every couple of points $x, y \in X$ there is a continuous curve $\alpha : [0, 1] \to X$ such that $\alpha(0) = x$ and $\alpha(1) = y$. The biggest path connected subsets contained in a topological space are called path-connected components.

We call $\pi_0(X)$ the set of the path connected components of X. If $q : X \to Y$ is a continuous function, consider the function $\pi_0(q) : \pi_0(X) \to \pi_0(Y)$ defined as follows:

$$U \mapsto V \text{ such that } q(U) \subset V.$$

π_0 is a (covariant) functor (Mac Lane, 1998) and thus satisfies a number of properties. Among them, we emphasize the following: for two continuous functions p, q that can be composed into the function $p \circ q$, it is true that $\pi_0(p \circ q) = \pi_0(p) \circ \pi_0(q)$.

Path connected components are the source of information we want to track along the family $\{X_t\}_{t\in\mathbb{R}}$, which we will call the *sublevel set filtration* of f. For $t \in \mathbb{R}$, let $\pi_0(X_t) = \{U_i^t\}_{i\in I}$ be the set of the path-connected components of X_t.

Let $i_t^{t'} : X_t \hookrightarrow X_{t'}$; then we have:

$$\pi_0(i_t^{t'}) : \pi_0(X_t) \to \pi_0(X_{t'})$$

such that

$$U_i^t \subset \pi_0(i_t^{t'})(U_i^t)$$

for all $U_i^t \in \pi_0(X_t)$.

4.2.3 Tree Structures, Critical Values and Topological Changes

Coherently with Chapter 2, we now define what we mean with *tree* and with *merge tree*.

Definition 4.1. *A tree structure T is given by a set of vertices V_T and a set of edges $E_T \subset V_T \times V_T$ which form a connected rooted acyclic graph. We indicate the root of the tree with r_T. We say that T is finite if V_T is finite. The order of a vertex $v \in V_T$ is the number of edges which have that vertex as one of the extremes, and is called $ord_T(v)$. Any vertex with an edge connecting it to the root is its child and the root is its father: this is the first step of a recursion which defines the father and children relationship for all vertices in V_T. The vertices with no children are called leaves or taxa and are collected in the set L_T. The relation $child < father$ generates a partial order on V_T. The edges in E_T are identified in the form of ordered couples (a, b) with $a < b$. A subtree of a vertex v, called $sub_T(v)$, is the tree structure whose set of vertices is $\{x \in V_T | x \leq v\}$.*

Definition 4.2. *A finite tree structure T such that r_T is of order 1, coupled with a monotone increasing height function $h_T : V_T \to \mathbb{R} \cup \{+\infty\}$ with $h_T(r_T) = +\infty$ and $h_T(v) \in \mathbb{R}$ if $v < r_T$, is called merge tree.*

Let us now see how we can represent a real valued function $f : X \to \mathbb{R}$ by means of a merge tree. To do that we need to make some key assumptions.

Let us now see how we can represent a real valued function $f : X \to \mathbb{R}$ by means of a merge tree. To do so, we need to make some key assumptions, known in the literature to be apt to produce *constructible* objects (De Silva et al., 2016; Patel, 2018; Curry et al., 2022).

Assumption 4.3. *Given a family of topological spaces $\{X_t\}_{t\in\mathbb{R}}$ with $X_t \subset X_{t'}$, $t \leq t'$, we assume the existence of a finite collection of real numbers $\{t_1 < t_2 < \ldots < t_n\}$, called critical set, such that, given $t < t'$, if $t, t' \in (t_i, t_{i+1})$ or $t, t' > t_n$, then $\pi_0(i_t^{t'})$ is bijective. The values t_i are called critical values. As in Pegoraro (2021c) we always consider a minimal set of critical values, that is, the smallest possible set of critical values. With this condition, for any critical value t_i there is some constant $C > 0$ such that for all $\varepsilon \in (0, C)$, $\pi_0(i_{t_i-\varepsilon}^{t_i+\varepsilon})$ is not bijective. On top of that, we assume that for every $t \in \mathbb{R}$, $\pi_0(X_t)$ is finite. A function f such that its sublevel set filtration $\{X_t\}_{t\in\mathbb{R}}$ satisfies the above set of hypotheses is called* tame *(Chazal et al., 2016). Lastly, we also assume that X is path connected.*

Together with the tameness of f and the path-connectedness of X, we make an extra and simplifying *regularity* assumption - not needed, for the general construction of a merge tree - which implies a strong property of the critical values of the sublevel set filtration $\{X_t\}$. This assumption can be weakened, at the cost of some non trivial topological details (for more details see also Chapter 2). For the sake of clarity, we defer the discussion about this issue to Section 4.2.4.

Let t_j be a critical value of the sublevel set filtration $\{X_t\}_{t\in\mathbb{R}}$ of f. Let $\varepsilon > 0$ be such that $t_j - \varepsilon > t_{j-1}$ and $t_j + \varepsilon < t_{j+1}$. The properties of π_0 imply that $\pi_0(i_{t_j-\varepsilon}^{t_j+\varepsilon}) = \pi_0(i_{t_j-\varepsilon}^{t_j}) \circ \pi_0(i_{t_j}^{t_j+\varepsilon})$. Due to the minimality condition stated in Assumption 4.3 we know that $\pi_0(i_{t_j-\varepsilon}^{t_j+\varepsilon})$ is not bijective.

Assumption 4.4. *We assume that the sublevel set filtration $\{X_t\}_{t\in\mathbb{R}}$ of f is regular: that is, for every critical value t_j of $\{X_t\}_{t\in\mathbb{R}}$, there is a $C > 0$ such that, for all $\varepsilon \in (0, C)$, the map $\pi_0(i_{t_j}^{t_j+\varepsilon})$ is bijective.*

When $\pi_0(i_{t_j}^{t_j+\varepsilon})$ is bijective, we say that the topological changes happen *at* the critical value t_j, as opposed to *across* t_j. Hence we are assuming that all the topological changes of $\{X_t\}$ happen at the critical values.

Remark 4.5 and Remark 4.6 shortly expand on Assumption 4.4; for more details see Section 4.2.4.

Remark 4.5. *From the topological point of view, what lies behind the requirement that the topological changes happen at critical values is the following. Let $U_{t_j} \in \pi_0(X_{t_j})$ and $U_t = \pi_0(i_{t_j}^t)(U_{t_j})$ for $t \in (t_j, t_{j+1})$. By construction $U_{t_j} \subset U = \bigcap_{t\in(t_j,t_{j+1})} U_t$ and $f(p) = t_j$ for all $p \in U$. Which means $U \subset X_{t_j}$. If U is path connected then $U_{t_j} = U$ and we can't have another path connected component $U'_{t_j} \in \pi_0(X_{t_j})$ such that $U'_{t_j} \subset U$.*

All of this implies that, for $t \in (t_j, t_{j+1})$, $\pi_0(i_{t_j}^t)^{-1}(U_t) = \{U_{t_j}\}$ - i.e. $\pi_0(i_{t_j}^t)$ is injective at U_{t_j} and $U_t \in \pi_0(i_{t_j}^t)(\pi_0(X_{t_j}))$. So, if for every path connected component $U_{t_j} \in \pi_0(X_{t_j})$ the set $U = \bigcap_{t\in(t_j,t_{j+1})} \pi_0(i_{t_j}^t)(U_{t_j})$ is non empty and path connected, then $\pi_0(i_{t_j}^t)$ is bijective. However, in general, U need not be non empty and path connected!

Remark 4.6. *Two notable cases where the topological changes happen at critical values are that of a continuous function f defined on a connected compact subset of $\mathbb{R}$ and that of a function f defined on a finite graph.*

Indeed, let $f : X \to \mathbb{R}$ be continuous and $X \subset \mathbb{R}$ be a compact interval. Then, for all $t \in \mathbb{R}$, X_t is closed, since f is continuous, and its path connected components are compact intervals of the form $[a, b]$. Each path connected component U_{t_j} is thus a convex set and any intersection of the form $U = \bigcap_{t\in(t_j,t_{j+1})} \pi_0(i_{t_j}^t)(U_{t_j})$ is non-empty and convex; that is, U is non empty and path connected. Thus, for continuous functions $f : X \to \mathbb{R}$, with $X \subset \mathbb{R}$ being a compact interval, the topological changes always happen at critical values.

Consider now the discrete setting of a finite graph $X = (V, E)$, with vertices V and edges E, such that the sublevel set filtration is well defined (i.e. for any edge $e_{ij} = (x_i, x_j) \in E$ connecting two vertices x_i and x_j, we have $f(e) \geq \max\{f(x_i), f(x_j)\}$). Let $t_1 < t_2 < \ldots < t_n$ be the image of f. Then, $X_t = X_{t_j}$ for all $t \in [t_j, t_{j+1})$. This implies that all topological changes happen at critical values.

In light of Remark 4.6, we point out that all the functions considered in the present work, those illustrated in the examples or those pertaining to the case study described in Section 4.8, do satisfy our Assumption 4.4, and the same is true for all numerical implementations.

The heuristic idea behind the construction of the merge tree representation of f is that, since along the sequence $\{X_t\}_{t\in\mathbb{R}}$ the path-connected components of X_t can only

arise, merge with others (at critical values), or stay the same, it is natural to represent this merging structure with a tree structure T. However, this tree T would not encode the critical values $t_1 < ... < t_n$ of $\{X_t\}_{t\in\mathbb{R}}$; hence we enrich it by defining a monotone increasing height function $h_T : V_T \to \mathbb{R} \cup \{+\infty\}$ encoding them.

The tree structure T and the height function h_T are built along the following rules in a recursive fashion starting from an empty set of vertices V_T and an empty set of edges E_T. We simultaneously add points and edges to T and define h_T on the newly added vertices. From now on, we indicate with $\#C$ the cardinality of a finite set C.

Considering in increasing order the critical values:

- for the critical value t_1 add to V_T a leaf $v_{U_{t_1}}$, with height t_1, for every element $U_{t_1} \in \pi_0(X_{t_1})$;
- for t_i with $i > 1$, for every $U_{t_i} \in \pi_0(X_{t_i})$ such that $U_{t_i} \notin \mathrm{Im}(\pi_0(i^{t_i}_{t_{i-1}}))$, add to V_T a leaf $v_{U_{t_i}}$ with height t_i;
- for t_i with $i > 1$, if $U_{t_i} = \pi_0(i^{t_i}_{t_{i-1}})(U_{t_{i-1}}) = \pi_0(i^{t_i}_{t_{i-1}})(U'_{t_{i-1}})$, with $U_{t_{i-1}}$ and $U'_{t_{i-1}}$ distinct path connected components in $\pi_0(X_{t_{i-1}})$, add a vertex $v_{U_{t_i}}$ with height t_i, and add edges so that the previously added vertices

$$v = \arg\max\{h_T(v'_U) \,|\, v'_U \in V_T \text{ s.t. } U \subset U_{i-1}\}$$

and

$$w = \arg\max\{h_T(w'_U) \,|\, w'_U \in V_T \text{ s.t. } U \subset U'_{i-1}\}$$

connect with the newly added vertex $v_{U_{t_i}}$.

The last merging happens at height t_n and, since X is path connected, at height t_n there is only one point v_U. Thus we can add a vertex r_T and an edge (v_U, r_T) with $h_T(r_T) = +\infty$ to obtain a merge tree.

The reader can look at Figure 4.1.1b for a first example of a merge tree associated to a function. The height function is given by the dotted red lines. We can appreciate that the merge tree of f is heavily dependent on the shape of f, in particular on the displacement of its maxima and minima.

4.2.4 Topological Remark

In Section 4.2.3 we make some selective assumptions on X and f, to make sure that along the filtration $\{X_t\}$ topological changes only happen at critical values, that is: for a critical value t_j, for all $\varepsilon > 0$ small enough, $\pi_0(i^{t_j+\varepsilon}_{t_j})$ is bijective. In Remark 4.5 we point out that this fact boils down to the topology of the sets:

$$\bigcap_{t\in(t_j,t_{j+1})} \pi_0(i^t_{t_j})(U_{t_j})$$

for every critical point t_j and for every $U_{t_j} \in \pi_0(X_{t_j})$. Topological changes happening at the critical values are equivalent to $\bigcap_{t\in(t_j,t_{j+1})} \pi_0(i^t_{t_j})(U_{t_j})$ always being non empty and path connected. This in general does not hold as we can see in the upcoming examples.

Example I Consider the following sequences of topological spaces $\{A_t\}_{t\in[0,\infty)}$ and $\{B_t\}_{t\in[0,\infty)}$. For $t > 0$, let $A_t = (-t, t)\bigcup(1 - t, 1 + t)$ and $B_t = [-t, t]\bigcup[1 - t, 1 + t]$. Moreover, let $A_0 = B_0 = \{0, 1\}$. $\{A_t\}$ and $\{B_t\}$ share the same set of critical values, namely $\{0, 1/2\}$ and they only differ by the number of path connected components at the critical value $1/2$: $\#A_{1/2} = 2$, while $\#B_{1/2} = 1$. In $\{A_t\}$ changes happen across the critical values - $\pi_0(A_{1/3}) \cong \pi_0(A_{1/2})$ and $\pi_0(A_{1/2}) \ncong \pi_0(A_1)$, while in $\{B_t\}$ changes happen at the critical values - $\pi_0(B_{1/3}) \ncong \pi_0(B_{1/2})$ and $B_{1/2} \cong B_1$.

Example II Consider the following sequence of topological spaces $\{A_t\}_{t\in[0,\infty)}$. For $t > 0$, let $A_t = \{(-\infty, +\infty) \times [-1/t, 1/t]\} - \{(0, 0)\}$ and $A_0 = \{(-\infty, 0) \cup (0, +\infty)\} \times \{0\}$. Then $\bigcap_{t>0} A_t = A_0$: however A_0 has two path connected components while, for $t > 0$, the set A_t is path connected.

Example III Let $\gamma : (0, 1] \to \mathbb{R}^2$ defined by $\gamma(t) = (t, \sin(1/t))$. Let T be the closure in the plane of the set $S = \{(t, \gamma(t)) \in \mathbb{R}^2 : t \in (0, 1]\}$, which is given by $T = S \cup \{0\} \times [-1, 1]$. The set S is usually referred to as the topologist's sine curve and the set T as the closed topologist's sine curve. Let S_n be:

$$S_n = \{(t, s) \,|\, t \in [1/n, 1] \text{ and } s \in [\sin(1/t) - 1/n, \sin(1/t) + 1/n]\}$$

That is, S_n is an $1/n$-thickening along the y-axis (second component of $\mathbb{R}^2$), of the graph of γ restricted on the interval $[1/n, 1]$. Thus, if we add to S_n the rectangle $R_n = [-1/n, 1/n] \times [-1 - 1/n, 1 + 1/n]$ we obtain a set $T_n = S_n \bigcup R_n$ such that:

- $T = \bigcap_{n\in\mathbb{N}} T_n$
- T_n is compact and path connected. It is in fact homeomorphic to $R_n \bigcup [1/n, 1] \times [-1/n, 1/n]$ and thus homeomorphic to a closed disk in the plane.

As $n \to \infty$ we obtain a family of compact "disks" whose intersection is T which is not path connected.

Example IV Lastly $A_t = [1/t, +\infty) = f^{-1}((-\infty, t])$ with $f : \mathbb{R}_{>0} \to \mathbb{R}$ being $f(x) = 1/x$. Clearly A_t is closed and path connected, but $\bigcap_{t\geq 0} A_t$ is empty.

In all the examples above we see different situations in which at some critical point t we have a very "unstable" topological scenario, which changes at $t + \varepsilon$ for any small $\varepsilon > 0$:

- in Example I the balls centered in 0 and 1 contained in A_t touch right after $t = 1/2$; in fact their closures (giving $B_{1/2}$) at $t = 1/2$ would intersect;
- in Example II the horizontal stripe given by A_t suddenly disconnects at $t = 0$ because it is no more thick enough to get around the hole in $(0, 0)$;
- we find a very similar situation also in Example III, where every thickening of T would allow us to bridge between its two path connected components;
- lastly, in Example IV, we have a path connected component being born with a minimum "lying" at $+\infty$, thus producing an empty level set at $t = 0$.

The general point of view which we assume, which is formalized in Pegoraro (2021c), is that we deem to be negligible the topological differences between $\{A_t\}$ and $\{B_t\}$ in Example I, as those two filtrations have the same path connected components but for one point, $t = 1/2$, which we may look at as a measure zero subset of the parameter space indexing the filtration (i.e. $\mathbb{R}$). Thus, for all these examples, and, in fact, for all the *tame* filtrations of path connected topological spaces, we propose to build the associated merge tree as if all topological changes happen at critical points: if we have a critical point t_j such that $\pi_0(i_{t_j}^{t_j+\varepsilon})$ is not bijective, instead of looking at the merging information contained in $\pi_0(i_{t_{j-1}}^{t_j})$ - as we do in Section 4.2.3 - one should look at $\pi_0(i_{t_{j-1}}^{t_j+\varepsilon})$, but still recording the topological changes with a vertex at height t_j. In this way, for instance, the merge trees associated to $\{A_t\}$ and $\{B_t\}$ in Example I would be the same, in Example II we would have a single leaf at height 0 - despite A_0 having two path connected components, the same for Example III (upon replacing n with $1/\varepsilon$), and, lastly, Example IV would feature a leaf at height 0 despite A_0 being empty.

4.2.5 Isomorphism classes

Before continuing we must decide on the topological information which we regard as equivalent. In other words, which merge trees we want to distinguish and which we do not. This step is essential and decisive to tackle the phase variation problem presented in the introduction: to select information that is insensitive to some kind of transformation amounts to defining classes of functions which are represented by the same tree. As in Chapter 2 and Chapter 3, we opt for a very general solution: we remove from the vertices of the tree any information regarding the connected components they are associated to, for instance, size, shape, position, the actual points contained etc..

Definition 4.7. *Two tree structures T and T' are isomorphic if there exists a bijection $\eta : V_T \to V_{T'}$ inducing a bijection between the edges sets E_T and $E_{T'}$: $(a, b) \mapsto (\eta(a), \eta(b))$. Such η is an isomorphism of tree structures.*

Definition 4.8. *Two merge trees (T, h_T) and $(T', h_{T'})$ are isomorphic if T and T' are isomorphic as tree structures and the isomorphism $\eta : V_T \to V_{T'}$ is such that $h_T = h_{T'} \circ \eta$. Such η is an isomorphism of merge trees.*

The rationale behind Definition 4.7, and the equivalence classes of isomorphic merge trees it generates, is analogous to that moving the introduction of persistence diagrams, where no specific information about individual path connected components is retained (see Section 4.3 for more details). Moreover, Definition 4.7 does not require any additional structure for the space X. Other choices are possible; for instance, if $X = \mathbb{R}$ the path connected components of f could be given a natural ordering.

4.2.6 Height and Weight Functions

A final step is needed to complete the specific representation of merge trees needed for making use of the metric defined in Chapter 3. The height function h_T of a generic merge tree T takes values in $\mathbb{R}$, but this is not an *editable* space, according to the definition in Chapter 2, which we report here.

Definition 4.9. *Let X be a set endowed with a metric d and an associative operation $*$ with zero element $0 \in X$. Then $(X, d, *, 0)$ is said to be an* editable space *if the following two properties are both satisfied:*

(P1) the map $d(\cdot, 0) : X \to \mathbb{R}$ is a map of monoids between $(X, *)$ and $(\mathbb{R}, +)$, that is: for all $x, y \in X$

$$d(x * y, 0) = d(0, x) + d(0, y);$$

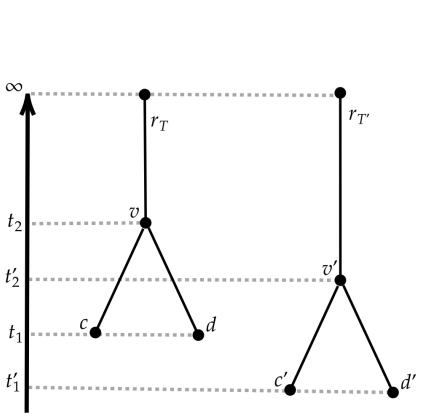
(P2) d is $*$ invariant, that is: for all $x, y, z \in X$,

$$d(x, y) = d(z * x, z * y) = d(x * z, y * z).$$

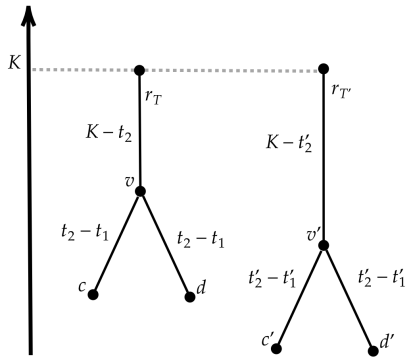
Note that, whereas $\mathbb{R}$ is not editable because $|\,(x + (-x)) - 0\,| \neq |\,x - 0\,| + |\,0 - x\,|$, $\mathbb{R}_{\geq 0}$ is editable.

We thus complement the merge tree T with a transformation of the height function h_T : a weight function w_T defined on $V_T - \{r_T\}$ whose image is a subset of the editable space $\mathbb{R}_{\geq 0}$. To do so, as in Chapter 3, we employ a truncation strategy which takes care of the edge (v, r_T) which goes at infinity. Such strategy relies on the following assumption.

Assumption 4.10. *We assume the existence of a universal constant $K \in \mathbb{R}$ bounding above all the functions for which we will adopt a merge tree representation.*



(a) Two merge trees T and T'.



(b) A graphical representation of the weight functions obtained by truncating the merge trees at height K.

Figure 4.2.1: A graphical representation of the truncation process described in Section 4.2.6.

In Pegoraro (2021d) it is shown that all the upcoming steps in the construction of the specific merge trees considered in this paper, do not depend on K, in the sense that with any $K' > K$ we would obtain the same results. We spend some more words on this issue in the following Remark 4.15.

Given a merge tree (T, h_T), as a first step we define the function $h'_T : V_T \to \mathbb{R}$ as $h'_T(v) = h_T(v)$ for all $v < r_T$ and $h'_T(r_T) = K$. Then for every vertex $v \in V_T - \{r_T\}$, we consider the unique edge between v and its father w and we define $w_T(v) = h'_T(w) - h'_T(v)$; the weight function w_T also codes the weight of the edge (v, w), via the rule $w_T((v, w)) = w_T(v)$, which identifies the set of edges E_T with vertices in $V_T - \{r_T\}$. Note that, because of Assumption 4.10, there is a one-to-one correspondence between h_T and w_T. Finally, the monotonicity of h_T and Assumption 4.10 guarantee that $w_T(v) \in \mathbb{R}_{\geq 0}$, for all $v \in V_T - \{r_T\}$. A visual representation of this procedure can be found in Figure 4.2.1.

The height function introduced in Definition 4.2 turns out to be quite natural for the definition of a merge tree, but from now on along with the height function h_T we also employ the induced weight functions w_T.

4.3 Persistence Diagrams

Persistence diagrams are arguably among the most well known tools of TDA; for a detailed survey see, for instance, (Edelsbrunner and Harer, 2008). We here briefly introduce persistence diagrams since in the following sections we use them to draw comparisons with merge trees.

Loosely speaking a persistence diagram is a collection of points (c_x, c_y) in the first quadrant of $\mathbb{R}^2$, with $c_y > c_x$ and such that: c_x is the t corresponding to the first appearance of an homology class in X_t (birth), while c_y is the t where the same class merges with a different class appeared before c_x (death). Homology classes are a generalization of path-connected components to "holes in higher dimension"; path-connected components can be seen as zero dimensional holes. For more details see Hatcher (2000).

In this work we focus on persistence diagrams associated to path-connected components, since we want to compare them with the merge trees introduced in the previous section. Given a function $f : X \to \mathbb{R}$, we associate to f the zero dimensional persistence diagram ($PD(f)$) of the sequence of sublevel sets $\{X_t\}_{t \in \mathbb{R}}$. We highlight that, in such representation, there is no information about which path-connected component merges with

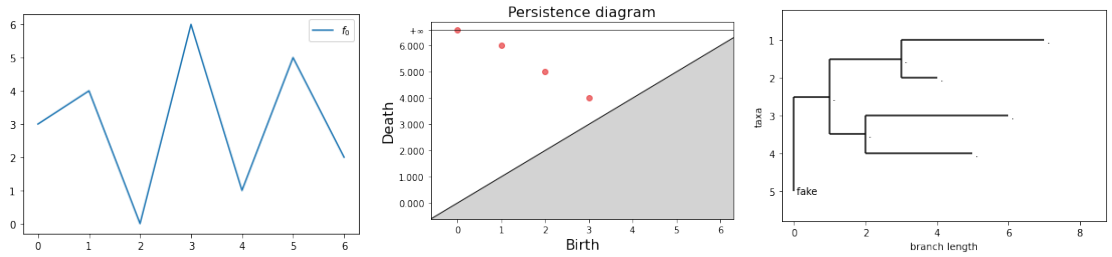


Figure 4.3.1: A function (left) with its associated persistence diagram (centre) and merge tree (right). On the PD axes we see the birth and death coordinates of its points. The plot of the merge tree features the length of its branches (given by the weight function - Section 4.2.6) on the horizontal axis, and the leaves (taxa) are displaced on the vertical axis. The vertical axis scale is only for visualization purposes. The merge tree is truncated at height 7 - see Section 4.2.6.

which; in fact a component represented by the point (c_x, c_y), at height c_y could merge with any of the earlier born and still alive components. Of course this collection of points still depends on the shape of the function and in particular depends on its amplitude and the number of its oscillations. See Figure 4.3.1. Note that, while for merge trees one needs to be careful and consider appropriate isomorphism classes so that the representation does not depend, for instance, on the names chosen for the vertices (that is, the set V_T), this issue does not appear with persistence diagrams. Topological features are represented as points in the plane, without labels or other kinds of set-dependent information. Thus, two persistence diagrams are isomorphic if and only if they are made of the same set of points.

4.4 Properties

In this section we state the main invariance result anticipated in the introduction and we also point out a few differences between persistence diagrams and merge trees.

Proposition 4.11 (Invariance)**.** *The (isomorphism class of the) merge tree and the persistence diagram representations of the function $f : X \to \mathbb{R}$, are both invariant under homeomorphic re-parametrization of f.*

Remark 4.12. *As an immediate consequence of Proposition 4.11 we obtain that, if the functions f and g can be aligned by means of an homeomorphism, that is if $f = g \circ \eta$ being η an homeomorphism, then their associated merge trees T_f and T_g are isomorphic and the same holds for $PD(f)$ and $PD(g)$.*

In other words, we can warp, deform, move the domain X of a function f by means of any homeomorphism, and this will have no effect on its associated PD or merge tree. As a consequence, if each element of a sample of functions is represented by its merge tree, or by its persistence diagram, one can carry out the statistical analysis without worrying about possible misalignements, that is without first singling out, for each function of the sample, the specific warping function, identified by an homeomorphism, which decouples its phase and amplitude variabilities.

Despite sharing this important invariance property, a persistence diagram and a merge tree are not equivalent representations of a function. Indeed, persistence diagrams do not record information about the merging components: as already mentioned, the death of a path connected component could be caused by its merging with any other alive component at the death-time. This implies that, for a given persistence diagram PD, there might be more than one merge tree associated to the diagram: the birth and death of the path

connected components of each merge tree coincide with those of the PD, but the merging structure is different from merge tree to merge tree (Kanari et al., 2020; Curry et al., 2021; Pegoraro, 2021c; Curry et al., 2022; Elkin and Kurlin, 2020; Smith and Kurlin, 2022). In particular, the works of Kanari et al. (2020); Curry et al. (2021) formally address this point, by providing explicit formulas for the *tree realization number*: the number of trees associated to the same persistence diagram. This number can be very high, being $n!$, if n is the number of points in the diagram, for certain configurations of the persistent diagram. This is the case, for instance, with hierarchical clustering dendrograms with n leaves: all leaves are born ath height 0, and so, at the first merging point, each of the n leaves can merge with any of the $n-1$ remaining ones. At the following merging step we have $n-1$ clusters and each one of them can merge with the other $n-2$ and so on. Thus, depending on the structure of the persistence diagram representing a function f, the associated merge tree could contain much more information regarding f; from a different perspective, merge trees can discriminate between functions which are indistinguishable for persistent diagrams. To see some easy examples of how merge trees capture also the local merging structure of the components which persistent diagrams cannot distinguish, see Figure 4.4.1 and Section 4.7. Moreover, with the next proposition we formally state that the information contained in the persistence diagram of a function f can be completely retrieved from the merge tree representing f. The previous part of this paragraph clearly indicates that the converse need not be true, as also shown in Figure 4.4.1.

Proposition 4.13. *For all $f : X \to \mathbb{R}$, the associated $PD(f)$ in dimension 0 can be obtained by the associated merge tree T_f.*

Thus, if two functions induce isomorphic merge trees, they also have the same persistence diagrams. Further details and insights on the differences between PDs and merge trees can be found in Section 4.7 of the supplementary material.

4.5 Metrics

We want to analyze sets of functions using merge trees and PDs, exploiting metrics which have already been defined respectively in Chapter 3 and in Cohen-Steiner et al. (2010). Here we quickly present such metrics, with a special focus on the metric for merge trees, since we use it to develop novel stability results in the next sections.

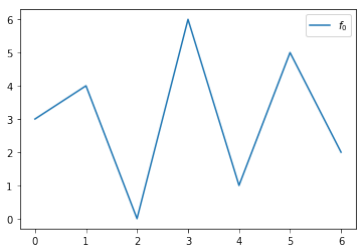
4.5.1 Metrics for Persistence Diagrams

The space of persistence diagrams can be given a metric structure by means of a family of metrics which derives from Wasserstein distances for bivariate distributions.

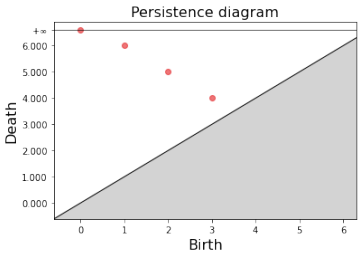
Given two diagrams D_1 and D_2, the expression of such metrics is the following:

$$W_p(D_1, D_2) = \left(\inf_{\gamma} \sum_{x \in D_1} ||x - \gamma(x)||_\infty^p \right)^{1/p}$$

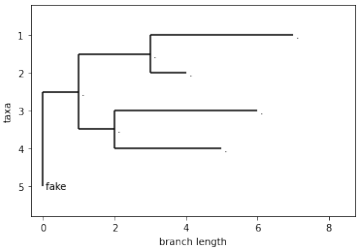
where $p \geq 1$ and γ ranges over the functions partially matching points between diagrams D_1 and D_2, and matching the remaining points of both diagrams with the line $y = x$ on the plane (for details see Cohen-Steiner et al. (2007)). In other words we measure the distances between the points of the two diagrams, pairing each point of a diagram either with a point on the other diagram, or with a point on $y = x$. Each point can be matched once and only once. The minimal cost of such matching provides the distance.



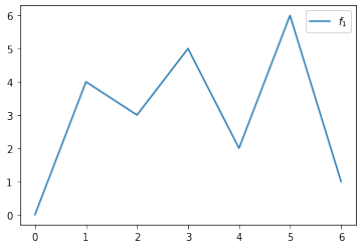
(a) A function f_0.



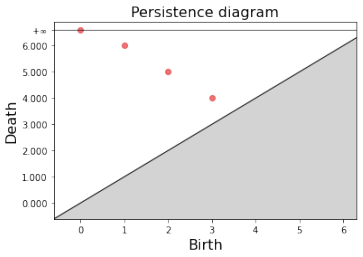
(b) The 0-dimensional persistence diagram of f_0.



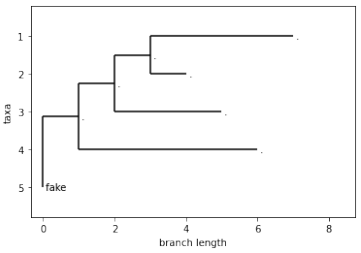
(c) The merge tree of f_0.



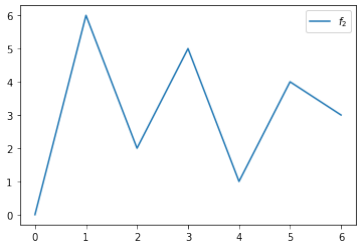
(d) A function f_1.



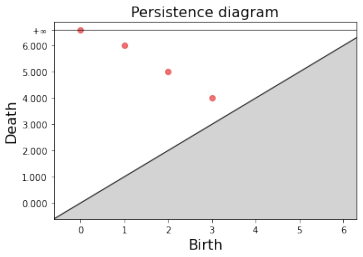
(e) The 0-dimensional persistence diagram of f_1.



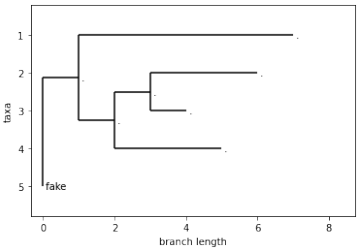
(f) The merge tree of f_1.



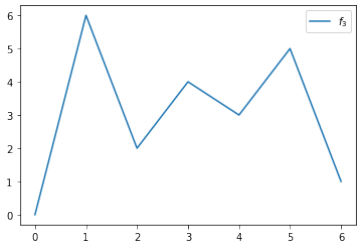
(g) A function f_2.



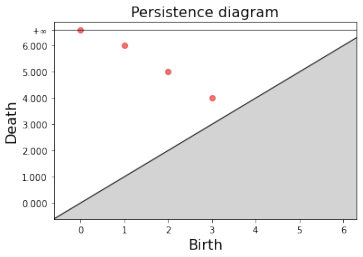
(h) The 0-dimensional persistence diagram of f_2.



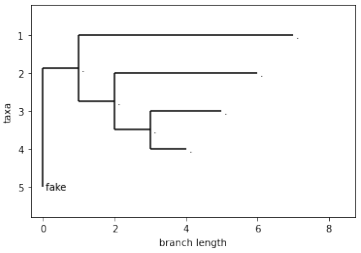
(i) The merge tree of f_2.



(j) A function f_3.



(k) The 0-dimensional persistence diagram of f_3.



(l) The merge tree of f_3.

Figure 4.4.1: We compare four functions; they are all associated to the same PD but to different merge trees. Functions are displayed in the first column and on each row we have on the centre the associated PD and on the right the merge tree. All merge trees are truncated at height 7 - see Section 4.2.6.

4.5.2 Metric for Merge Trees

The metric for tree-like objects defined in Chapter 2 and then adapted to merge trees in Chapter 3 is based on edit distances (Bille, 2005; Hong et al., 2017): they allow for modifications of a starting object, each with its own cost, to obtain a second object. Merge trees equipped with their weight function w_T, as defined in Section 4.2.6, fit into this framework; hence the space of merge trees can be endowed with a metric based on an edit distance and called d_E in the following.

The distance d_E is very different from previously defined edit distances, since it is specifically designed for comparing topological summaries, roughly meaning that all points which are topologically irrelevant can be eliminated by a merge tree without paying any cost. To make things more formal we here introduce the edits, as defined in Chapter 3.

The edits are the followings and can be used to modify any edge (v, v') of a merge tree, or equivalently its lower vertex v:

- *shrinking* an edge means changing the weight value of the edge with a new positive value. The inverse of this transformation is the shrinking which restores the original edge weight.

- *Deleting* an edge (v_1, v_2) results into a new tree, with the same vertices apart from v_1 (the lower one), and with the father of the deleted vertex which gains all of its children. With a slight abuse of language, we might also refer to this edit as the deletion of the vertex v_1, which indeed means deleting the edge between v_1 and its father.

  The inverse of deletion is the *insertion* of an edge along with its child vertex. We can insert an edge at a vertex v specifying the child of v and its children (that can be either none or any portion of the children of v) and the weight of the edge.

- Lastly, we can eliminate an order two vertex v, that is a father with an only child, connecting the two adjacent edges which arrive and depart from v. The weight of the resulting edge is the sum of the weights of the joined edges. This transformation is the *ghosting* of the vertex v. Its inverse transformation is called the *splitting* of an edge.

Remark 4.14. *Edit operations are not globally defined as operators mapping merge trees into merge trees. They are defined on the individual tree. Similarly, their inverse is not the inverse in the sense of operators, but it indicates that any time we travel from a tree T to a tree T' by making a sequence of edits, we can also travel the inverse path going from T' to T and restore the original tree.*

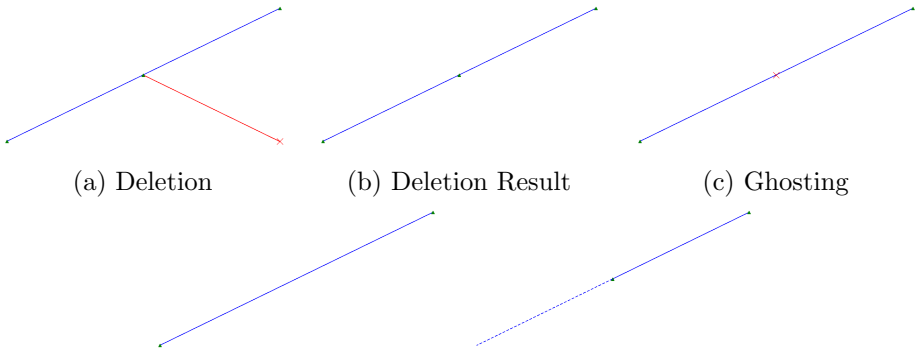
The costs of the edit operations are defined as follows:

- the cost of shrinking an edge is equal to the absolute value of the difference of the two weights;

- for any deletion/insertion, the cost is equal to the weight of the edge deleted/inserted;

- the cost of ghosting is zero.

Given a tree T we can edit it, thus obtaining another tree, on which we can apply a new edit to obtain a third tree and so on. Any finite composition of edits is called an *edit*



(a) Deletion (b) Deletion Result (c) Ghosting

(d) Ghosting Result (e) Shrinking

Figure 4.5.1: (a)→(e) form an edit path made by one deletion, one ghosting and a final shrinking.

path. The cost of an edit path is the sum of the costs of its edit operations. Putting all the pieces together, we can define the edit distance d_E as:

$$d_E(T, T') = \inf_{\gamma \in \Gamma(T,T')} cost(\gamma)$$

where $\Gamma(T, T')$ indicates the set of edit paths which start in T and end in T'.

Remark 4.15. *The results in Chapter 3 show that the metric $d_E(T, T')$ does not depend on the value of K used in the truncation process and introduced with Assumption 4.10. To be sure, if, after having fixed K to analyze a set of merge trees $\mathcal{T}$, we add to the set a new merge tree corresponding to a function f which is not bounded above by K, we proceed by fixing a novel K' bounding f and all the other functions represented in the set $\mathcal{T}$, and compute $d_E(T_f, T_g)$, for all $g \in \mathcal{T}$, after truncating all merge trees in $\mathcal{T} \bigcup \{T_f\}$ at height K'. This won't affect the distances between merge trees in $\mathcal{T}$ computed before the addition of T_f, when the truncation constant was K, since the metric d_E is the same for such merge trees. Thus Assumption 4.10 is in some sense unnecessary, since we do not need to fix K uniformly on our data set but only in a pairwise fashion. However for our applications such assumption is never violated, so we can assume it and avoid some formal complications arising from having to fix K for every couple of functions.*

4.5.2.1 Order Two Vertices

The null cost of ghosting guarantees that order 2 vertices are completely irrelevant when computing the cost of an edit path.

Definition 4.16. *If there is an edit path from the tree T to the tree T' consisting only of ghosting or splitting edits, we say that the two trees are equal up to order 2 vertices. By definition, the length of the edit path starting in T and ending in T' is equal to 0.*

In Chapter 2 it is proved that d_E is a metric on the space of merge trees, identified up to order 2 vertices. As explained in Chapter 2, the fact that order 2 vertices are irrelevant is precisely what makes the metric d_E suitable for comparing merge trees and is fundamental to obtain the results in Section 4.6.

4.6 Pruning & Stability

As stated in the introduction of the paper, any time we use a data representation – or we further transform a representation – it is important to understand and explore the properties of the operators involved. In particular, in this section we establish some continuity

properties for the operator $f \mapsto T_f$, which maps a function to its merge tree. Conditional on the topology endowing the functional space where the function f is embedded, these properties dictate how the variability between functions is captured by the variability between their merge tree representations.

Proposition 4.11 implies that the merge tree representation of a function f is unaffected by a large class of warpings of its domain, which would strongly perturb f if it was embedded, for instance, in an L_p space, with $p \neq \infty$. As an example, if $f : \mathbb{R} \to \mathbb{R}$ is bounded with compact support, shrinking f by setting $f_n(x) = f(x \cdot \lambda_n)$ with $\lambda_n \to +\infty$, produces no effect on the merge tree representation of f since $T_{f_n} = T_f$, while the p-norm of f_n goes to zero.

It might therefore be more natural to study the behavior of $f \mapsto T_f$ endowing the space of functions $f : X \to \mathbb{R}$ with the topology of uniform convergence, which captures pointwise closeness between functions. This topology, available for any domain X, has also the advantage of showing the effect of pointwise noise on merge tree representations.

4.6.1 Pruning

We know that, given f, the merge tree T_f will mostly depend on the critical points of f: as the number of spikes of f grows, also the size of the tree grows, while the weights of its branches grow with the height of the spikes. Similarly, if two functions $f, g : X \to \mathbb{R}$ are pointwise ε close, we can say that the shape of the functions is the same up to spikes of height $2 \cdot \varepsilon$. Each such spike would cause the birth of a leaf whose branch is shorter than $2 \cdot \varepsilon$; the trees must therefore be the similar up to branches of weight $2 \cdot \varepsilon$. These considerations move the idea of pruning, which consists of removing unessential edges from a tree.

Given a merge tree without order 2 vertices, we want to delete the small weight leaves, that is those whose weight is smaller than or equal to a given fixed threshold. However, if two or more small weight leaves are siblings, we only need to remove that of smallest weight, or one of the leaves chosen at random if they have the same weight, and then ghost its father if it becomes an order 2 vertex. To describe this procedure more formally and to make sure that, in the end, no small weight leaves are left in the tree, we need establish some (possibly partial) ordering of the leaves and to resort to recursion, as follows. Given $\varepsilon > 0$ and a merge tree T, define the following 1-step process:

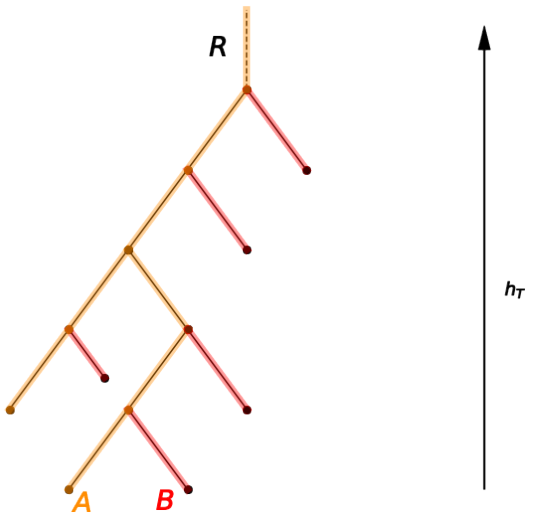
($\mathcal{P}_\varepsilon$) Take a leaf l such that $w_T(l)$ is minimal among all leaves; if two or more leaves have minimal weight, choose l at random among them. If $w_T(l) < \varepsilon$, delete l and ghost its father if it becomes an order 2 vertex after removing l.

For a given function f, we set $T_0 = T_f$, the merge tree representing f, and we apply operation ($\mathcal{P}_\varepsilon$) to obtain T_1. On the result we apply again ($\mathcal{P}_\varepsilon$) obtaining T_2 and, for $n > 2$, we proceed iteratively until we reach the fixed point $P_\varepsilon(T_f)$ of the sequence $\{T_n\}$. Note that the fixed point is surely reached in a finite time since the number of leaves of each tree in the sequence is finite and non increasing along the sequence.

Remark 4.17. *Applying ($\mathcal{P}_\varepsilon$) to a merge tree T representing a function f, is equivalent to removing a point (b, d) with persistence smaller than ε, that is $d - b < \varepsilon$, from the persistence diagram representing f. In fact a leaf l such that $w_T(l)$ is minimal, represents a connected component which, at height $h_T(father(l))$, meets another component which is born "earlier", and so dies at such height. Its persistence is thus $h_T(father(l)) - h_T(l) = w_T(l)$.*

We can thus define the pruning operator:

$$P_\varepsilon : \mathcal{T} \to \mathcal{T}$$



(a) Pruned Tree.

Figure 4.6.1: An example of a pruning operator applied to a merge tree T. The red branches are deleted from the tree, all order 2 vertices are ghosted, and the orange edges are kept and represent the merge tree $P_\varepsilon(T)$. Lastly, note that deleting A instead of B (these are small-weight siblings with same weights) would give isomorphic merge trees.

such that $P_\varepsilon(T)$ is the tree obtained by pruning with threshold ε. See Figure 4.6.1 for a visual example. Notice that P_ε is idempotent, that is $P_\varepsilon(P_\varepsilon(T)) = P_\varepsilon(T)$. On top of that, by construction, we have that $d_E(T, P_\varepsilon(T)) \leq \varepsilon \cdot (\#L_T - \#L_{P_\varepsilon(T)})$.

Remark 4.18. *P_ε is not a continuous operator. Consider T formed by just one edge with weight $\varepsilon > 0$; take $\delta > 0$ and consider T', with the same tree structure as T but made by one edge of weight $\varepsilon + \delta$. Now, $d_E(T, T') = \delta$ and $d_E(P_\varepsilon(T), P_\varepsilon(T')) = \varepsilon + \delta$. If we let $\delta \to 0$, then $d_E(P_\varepsilon(T), P_\varepsilon(T')) \to \varepsilon$.*

Remark 4.17 makes more evident and interpretable what happens at a topological level when pruning a merge tree representing a function f. With the following results, instead, we want to interpret the pruning procedure $(\mathcal{P}_\varepsilon)$ at a functional level, showing that it amounts to removing only a small spike of f at the time, whilst preserving the spikes of f with amplitude larger than the fixed threshold.

Lemma 4.19. *Let $f : X \to \mathbb{R}$ be a continuous function with X being a topological space and let $t \in \mathbb{R}$. Then $f(x) = t$ for every $x \in \partial\{f^{-1}((-\infty, t))\}$.*

Proposition 4.20. *Let X be a path connected and locally path connected topological space and let $f : X \to \mathbb{R}$ be a continuous tame function. Consider the merge tree (T_f, h_f) associated to f. Let $v \in L_{T_f}$ and set $\varepsilon = h_f(father(v)) - h_f(v)$. Then, there exist a function $g : X \to \mathbb{R}$ continuous and tame such that $0 \leq g(x) - f(x) \leq \varepsilon$, for all $x \in X$, and (T_g, h_g) is isomorphic to (T_f, h_f) up to deleting $(v, father(v))$ and - eventually - ghosting $father(v)$.*

Corollary 4.21. *Given (T_f, h_f) merge tree of a continuous tame function $f : X \to \mathbb{R}$ with X being a path connected and locally path connected topological space, for every $\varepsilon > 0$ there is $g : X \to \mathbb{R}$ tame and continuous such that $\| f - g \|_\infty \leq \varepsilon$ and (T_g, h_g) is isomorphic to $P_\varepsilon(T)$.*

For what has been said up to now, the operator P_ε can be considered as a supplementary smoothing operator. We fix some threshold which we think captures meaningful shape changes in a function and then, consistently, we remove what is deemed to be noise or nuisance - due, for instance, to overfitting f in its smoothing pre-processing - from the representation, obtaining a more regular merge tree. This has also the effect of greatly decreasing the number of leaves of the tree, a fact that is important from the computational perspective.

In Section 4.7.4 we carry out a complete example to showcase the effectiveness of pruning when working with a function f presenting some undesirable minor oscillations, along with a possible pipeline to select the pruning threshold.

4.6.2 Stability

Now we study the case of two merge trees T_f and T_g representing functions f and g which are pointwise ε close.

The main result of this section is the following.

Theorem 4.22. *Let f, g be tame functions defined on a path connected topological space X and such that*

$$sup_{x\in X}|f(x) - g(x)| \leq \varepsilon.$$

Let T_f and T_g be the merge trees associated to f and g respectively and let N and M be the cardinalities of V_{T_f} and V_{T_g}.

Then, there exists an edit path $e_1 \circ \ldots \circ e_{N+M} \in \Gamma(T_f, T_g)$ such that $cost(e_i) < 2 \cdot \varepsilon$, for $i = 1, ..., N + M$.

Theorem 3.37 states that if two functions are pointwise close, then we can turn the merge tree associated to the first function into the merge tree associated to the second function using at most one edit per vertex, and each edit has a small cost. Note, however, that if the two functions have a very high number of oscillations, the distance between their merge trees could still be large. Indeed if $||f_n - f||_\infty \xrightarrow{n} 0$ with $\#V_{T_{f_n}} \xrightarrow{n} \infty$, we are not guaranteed that $d_E(T_f, T_{f_n}) \to 0$. Theorem 4.22 however implies that, if the cardinalities $|V_{T_{f_n}}|$ are bounded, then $d_E(T_f, T_{f_n})$ indeed converges to 0.

Problems could then arise when we expect a possibly unbound number of informative spikes, that is spikes which should not be removed by pruning. In this case, however, the computational cost of the metric d_E would also be prohibitive due to the high number of leaves in the trees; indeed this supports the claim that the only practical limitation to the use of the metric d_E is given by its computational cost.

4.6.3 Spline Spaces

We here emphasize for spline spaces the consequences of the results of the previous two subsections, since splines are often used in FDA applications for smoothing the discrete raw data profiling each statistical unit in the sample.

As already noted in the introduction, spline spaces are a preferred tool for smoothing functional data since they provide finite dimensional vector spaces of functions with convenient properties. In particular, spline functions are piecewise polynomials determined by a grid of knots; fixing the knots determines a finite upper bound for the number of critical points of the spline. Consider for instance $\mathcal{S}_n^3$, the space of piecewise cubic polynomials over a grid on $[0, 1]$ with n equispaced knots. On each interval the first derivative of the function is a quadratic polynomial and thus its zero set is composed by at most two points. This means that the number of critical points of $f \in \mathcal{S}_n^3$ is at most $2(n - 1)$; therefore the number of leaves of the tree T_f associated to f cannot be greater than $2(n - 1)$. The following Corollary of Theorem 4.22 is in fact easily obtained:

Corollary 4.23. *Let $\mathcal{S}$ be a space of piecewise polynomial functions of some fixed degree, all defined by means of the same finite grid of nodes. Then the operator $f \in \mathcal{S} \mapsto T_f$ is continuous.*

Smoothing raw data with splines entails a delicate trade-off between being flexible, to capture the salient features of the function the raw data have been sampled from, and avoiding the introduction of artifacts, due, for instance, to noise overfitting or caused by forcing the spline to fit an abrupt spike. Representing the smoothed spline function by means of a merge tree can help in handling this trade-off, by allowing the analyst a certain degree of casualness in the smoothing phase, since the small artifacts generated by a possible overfitting will then be controlled by pruning the tree.

For instance, consider the problem of approximating $f : [0, 1] \to \mathbb{R}$ with a cubic spline function defined by an equispaced grid of knots. Suppose f satisfies some regularity conditions, usually implied by its embedding in a Sobolev space. The parameter which controls the bias-variance trade-off is just the number of knots n. Many results are known in the literature concerning the uniform convergence of spline functions as the step of the grid of knots goes to zero (see for instance De Boor and Daniel (1974)) and most of them are given in terms of a factor $1/n^\alpha$ and the norm of the derivatives or the modulus of continuity of f. In other words, the pointwise error can be reduced as needed by increasing n. When f is approximated by the spline function s_f with an error of ε in terms of uniform norm, this means that whatever happens in intervals of $\pm\varepsilon$ around f is inessential. Stated in different terms, oscillations of s_f taking place in such zone are to be considered uninformative with respect to the analysis. Thus a sensible choice is to represent the function f fitted by the spline s_f by means of the pruned merge tree $\mathcal{P}_{2\varepsilon}(T_{s_f})$. If ε is small enough with respect to the oscillations of f, Corollary 4.21 implies that pruning T_{s_f} by 2ε removes only inessential edges of T_{s_f}, without losing important information about f.

The same argument applies when smoothing observations sampled from a function f. The analyst may allow the spline to slightly overfit the data and then decide that oscillations under a certain amplitude are irrelevant, controlling them by pruning the merge tree associated to the fitted spline.

4.7 Examples

In this section, we present some examples which are intended to put to work the pruning operator and further show the differences between persistence diagrams and merge trees, already highlighted in Section 4.4.

We devote Section 4.7.1 to giving further intuition on the topic of functions being distinguished by merge trees but being represented by the same persistence diagram. Section 4.7.2 and Section 4.7.3 instead give a more qualitative idea of what kind of variability between functions is better captured by PDs and merge trees with, respectively, the 1-Wasserstein metric and the edit distance. Lastly Section 4.7.4 deals with the problem of pruning trees to remove ancillary features.

4.7.1 Example I

In this first example we produce a set of functions which are all described by the same persistence diagram but are distinguished by merge trees.

We want to exploit that, for continuous functions in one real variable, the merging structure of the path connected components (and so the tree structure T) is characterized by how local minima distribute on different sides of local maxima.

We create a very simple toy situation: we define functions which all have a very high peak and a number of smaller peaks of the same height, but with a different disposition of these smaller peaks with respect to the highest one.

For $i = 0, \dots, 9$, let $g_i : [0, 11] \to \mathbb{R}$ be such that $g_i \equiv 0$ on $[0, 11] - [i + 1/3, i + 2/3]$ while, on $[i + 1/3, i + 2/3]$, g_i is the linear interpolation of $(i + 1/3, 0)$, $(i + 1/2, 1)$ and $(i + 2/3, 0)$. Then, for $i = 0, \dots, 9$, define G_i as $G_i \equiv 0$ on $[0, 11] - [i + 2/3, i + 1]$ while, on $[i + 2/3, i]$, G_i is the linear interpolation of $(i + 2/3, 0)$, $(i + 3/4, 5)$ and $(i + 1, 0)$.

Then, for $i = 0, \dots, 9$, f_i is obtained as follows:

$$f_i = G_i + \sum_{j=0}^{10} g_j.$$

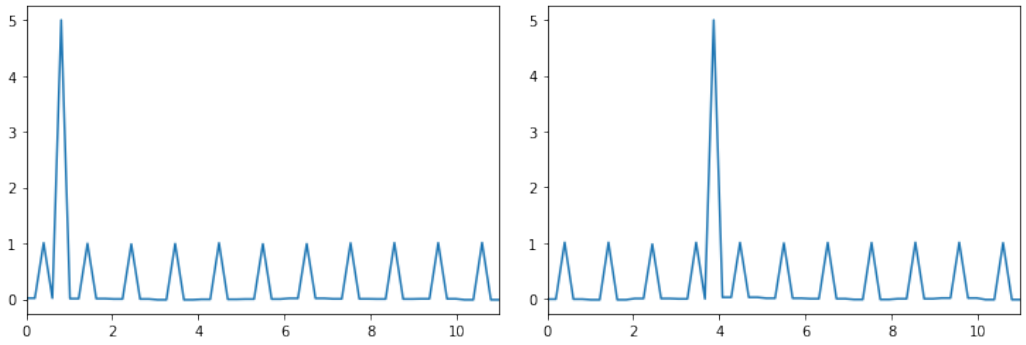
The functions f_0 and f_3 are displayed in Figure 4.7.1a. Note that the first and the last peak of every function, by construction, are always small peaks. The key point is that for every path connected component we are not changing any of the corresponding critical values and thus the associated persistence diagram is always the same (see Figure 4.7.1c).

The shortest edit path between two merge trees T_{f_i} and $T_{f_{i+1}}$ is given be the deletion of one leaf in each tree to make the disposition of leaves coincide between the to trees. The more the peak disposition is different between the two trees, the more one needs to delete leaves in both trees to find a shortest path between them. Thus, if we fix the first line of the matrix in Figure 4.7.1d, we see that going left-to-right the distance at first gradually increases. It is also evident that, from a certain point on, the distance decreases to the point of reaching almost zero. This is because the first function (the one in which the highest peak is the second peak) and the last function (the one in which the highest peak is the second-to-last peak) can be obtained one from the other via a y-axis symmetry and a translation – $x \mapsto -x$ (reflection on the y-axis) and $x \mapsto v + x$, with $v \in \mathbb{R}$ fixed (translation) –, these transformations being homeomorphisms of the abscissa. Similarly, the second function is equal, up to homeomorphic alignment, to the third-to-last one, etc.. Thus by Proposition 4.11 the merge trees are the same. To sum up the situation depicted in the first row of Figure 4.7.1d, first we get (left-to-right) farther away from the first merge tree, and then we return closer to it. This intuition is confirmed by looking at the MDS embedding in $\mathbb{R}^2$ of the pairwise distance matrix (see Figure 4.7.1e - note that the shades of gray reflect, from white to black, the ordering of the merge trees). The discrepancies between the couple of points which should be identified are caused by numerical errors.

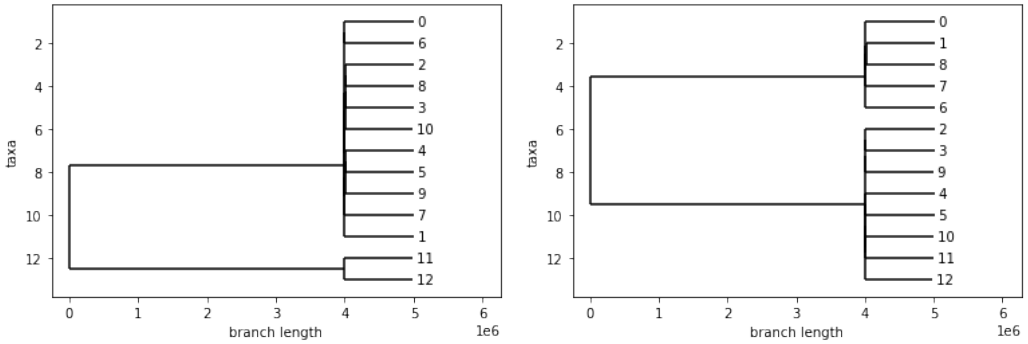
4.7.2 Example II

In this second example we want to produce a situation in which the variability between functional data is better captured by PDs than by merge trees. Accordingly, we generate two clusters of functional data such that the membership of a function to one cluster or the other should depend on the amplitude of its oscillations and not on the merging structure of its path connected components. We then look at the matrices of pairwise distances between functions, comparing merge tree and persistence diagram representations in terms of their goodness in identifying the clustering structure.

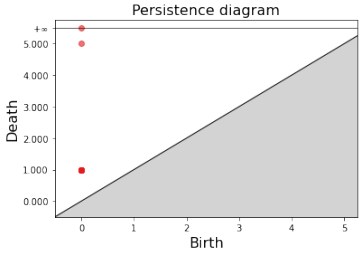
To generate each cluster of functions, we draw, for each cluster, an independent sample of 16 critical points, 8 maxima and 8 minima, from two univariate Gaussian distributions with means equal to $+100$ for maxima and to -100 for minima, respectively. The standard deviations of the two Gaussian distributions are the same and they are set equal to 50. To generate a function inside a cluster, we draw a random permutation of 8 elements and we reorder, according to this permutation, both the set of maxima and the set of minima associated to the cluster. Then, we take a regular grid of 16 nodes on the abscissa axis: on the ordinate axis we associate to the first point on the grid the first minimum, to



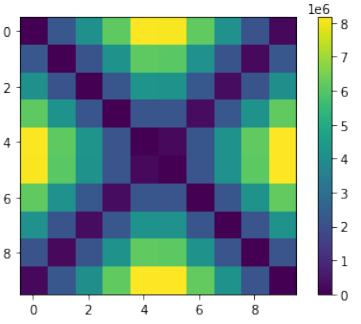
(a) The functions f_0 and f_3 belonging to the data set decribed in Section 4.7.1. Note the changes, between f_0 and f_3, in the disposition of the smaller peaks w.r.t. to the highest one.



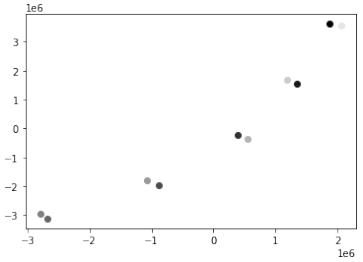
(b) The merge trees (T_{f_0}, h_{f_0}) and (T_{f_3}, h_{f_3}) associated to the functions f_0, f_3 in Figure 4.7.1a. The changes in the tree structures reflect the differet disposition of the disposition of the smaller peaks w.r.t. to the highest one in the associated functions.



(c) The persistence diagram representing the functions in Figure 4.7.1a and all the other functions produced in Section 4.7.1. The point $(0, 1)$ has multiplicity equal to the number of local minima minus 1.

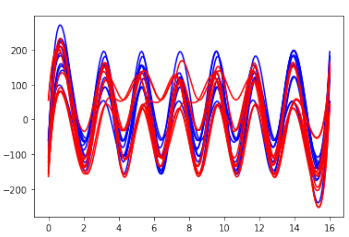


(d) Matrix of pairwise distances of the merge trees obtained in Section 4.7.1.

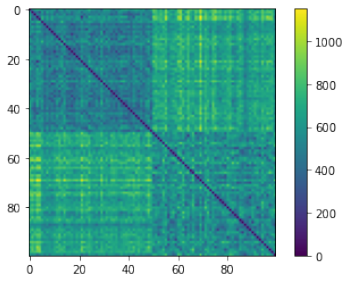


(e) Multidimensional Scaling Embedding in $\mathbb{R}^2$ of the matrix of pairwise distances shown in Figure 4.7.1d. The shades of gray describe, from white to black, the ordering of the trees.

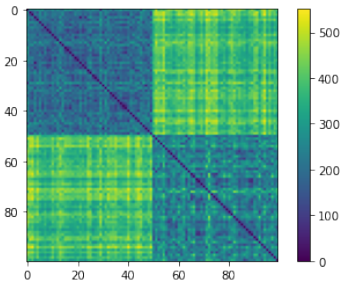
Figure 4.7.1: Plots related to the simulated scenario presented in Section 4.7.1.



(a) Data being plotted with different colours depending on the cluster.



(b) Merge trees pairwise distances.



(c) PDs paiwise distances.

Figure 4.7.2: Example II. In the first row we can see few data from the two clusters. In the second row we see the matrices of pairwise distance extracted with trees and PDs. The data are ordered according to their cluster. It is clear how PDs perform much better in separating the two clusters.

the second the first maximum, to the third the second minimum and so on. To obtain a function we interpolate such points with a cubic spline. We thus generate 50 functions in each cluster. The key point is that, within the same cluster, the critical points are the same but for their order, while the two clusters correspond to two different sets of critical points.

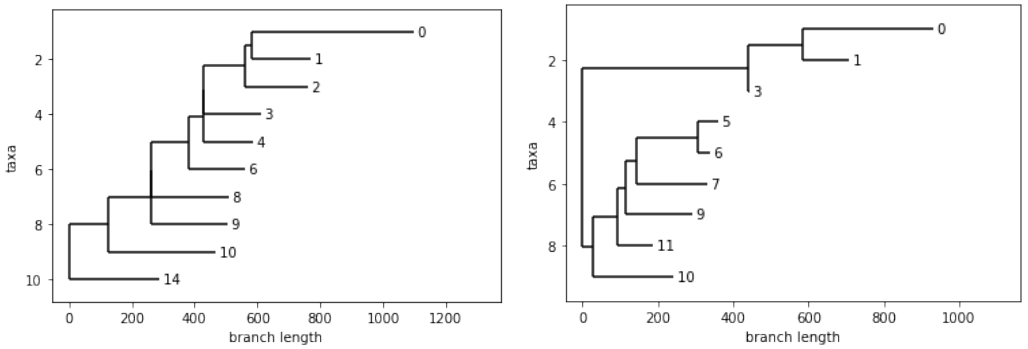
In this example, we expect that the clustering structure carried by the amplitude of the functions will be shadowed by the differences in the merging order, when adopting the merge tree representation; while persistence diagrams should perform much better because they are less sensitive to peak reordering. This is in fact confirmed by inspecting the distance matrices in Figure 4.7.2b and Figure 4.7.2c.

4.7.3 Example III

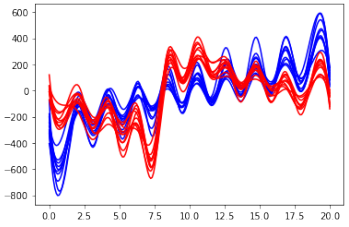
Here we reverse the state of affairs and we set the feature for discriminating between clusters to be the merging structure of the functions. Hence, we generate two clusters of functions: the members of each cluster have the same merging structure which is however different between clusters.

To generate the two clusters of 50 functions each, we first draw an independent sample of 10 critical values, 10 maxima and 10 minima, shared between the clusters. Such samples are drawn from Gaussian distributions with means 100 and -100 respectively and standard deviation 200. Given a regular grid of 20 nodes on the abscissa axis, on the ordinate axis we associate to the first point of the grid a maximum, to the second a minimum, and so on, as is Example I. To generate every member of one cluster or the other, we add to the ordinate of each maximum or minimum critical point a random noise generated by a Gaussian with mean 0 and standard deviation 100. Then we reorder such points following a cluster-specific order. And, lastly, we interpolate with a cubic spline. We remark that the ordering of the maxima and that of the minima now becomes essential. For the two clusters, these orderings are fixed but different and they are set as follows (0 indicates the smallest value and 9 being the largest value):

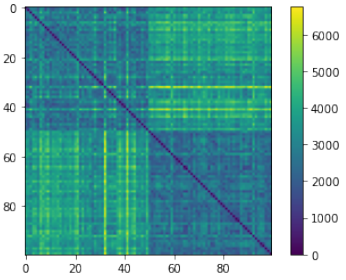
- first cluster: maxima are ordered along the sequence $(0, 1, 2, 3, 4, 5, 6, 7, 8, 9)$, minima along the sequence $(0, 1, 2, 3, 4, 5, 6, 7, 8, 9)$;
- second cluster: maxima are ordered along the sequence$(3, 2, 1, 0, 8, 9, 7, 6, 4, 5)$, minima along the sequence $(3, 2, 1, 0, 8, 9, 7, 6, 4, 5)$.



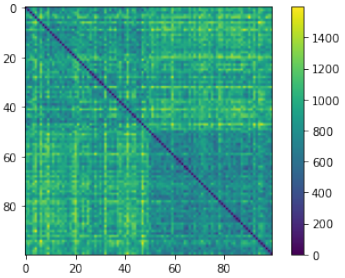
(a) Tree structures of the two clusters: left the first and right the second.



(b) Simulated data.



(c) Merge trees pairwise distances.



(d) PD pairwise distances.

Figure 4.7.3: Example III. In the first row we can find the tree structures associated to the two clusters. In the second row, leftmost plot, we can see a few data from the two clusters. In the central and rightmost column of the second row we see the matrices of pairwise distances between merge tree representations and PDs, respectively. The data are ordered according to their cluster. It is clear how in this example merge trees are more suitable to separate the two clusters.

Such different orderings provide non-isomorphic tree structures for the merge trees associated to the functions of the two clusters, as we can see in Figure 4.7.3a, while keeping a similar structure in terms of persistence diagrams.

In this example, we expect PDs to be unable to recognise the clustering structure among the data; indeed, the only discriminant feature available to PDs is the different height of critical points, but this bears little information about the clusters.

We can visually observe this by comparing Figure 4.7.3c with Figure 4.7.3d.

4.7.4 Pruning

In this section we present a simple example to show the regularization effect of pruning, along with an elbow analysis to identify a suitable pruning threshold. Figures related to this example can be found in Figure 4.7.4.

Consider the function $f(x) = \frac{1}{1+x^2}\sin(10\pi x)$, defined for $x \in [0,1]$, and fix a grid $x_1, \dots, x_n$ on $[0,1]$. Consider g a function which, on the fixed grid, could be considered as a noisy observation of f: $g(x_i) = f(x_i) + \eta_i$ with $\eta_i \sim \mathcal{N}(\mu = 0, \sigma = 0.1)$ i.i.d.. We want to study the relationship between the merge trees built from the samples $\{(x_1, f(x_1)), \dots, (x_n, f(x_n))\}$ and $\{(x_1, g(x_1)), \dots, (x_n, g(x_n))\}$ (via linear interpolation). With an abuse of notation we call such merge trees, respectively, T_f and T_g.

Knowing the generative model, in each point of the grid, with probability 0.95 we expect to see a value of g at a distance at most $2\sigma = 0.2$ from the corresponding value

of f. Thus, a pruning threshold of around 0.4 should be enough to cope with most of the noise in $\{g(x_1), \ldots, g(x_n)\}$. But this information can be retrieved directly from data with an elbow analysis: for every $\varepsilon > 0$ we consider the number of leaves in $P_\varepsilon(T_g)$ and we plot the curve obtained by varying ε. Under the assumption that the signal to noise ratio is high, we should indentify an elbow in the curve ε vs number of leaves of $P_\varepsilon(T_g)$. In fact, this corresponds to having some separation between noisy oscillations and the proper shape of the function. If that is not the case, it means that there are oscillations caused by noise which are at least as big as some oscillations which we would like to retain. Thus the smoothing parameter choice in this case is more arbitrary and some kind of compromise must be achieved. Note that there might be multiple elbows in the aforementioned plot, along with flat regions. A conservative choice would be to look for the leftmost clear elbow, which, for instance, may be recognized at $\varepsilon = 0.25$ in Figure 4.7.4c. We can appreciate that the flat region (meaning that $P_\varepsilon(T)$ does not change) starting in 0.25 contains also $2(2\sigma) = 0.4$. Lastly, we highlight that, despite the pruning procedure, the amplitude of the larger oscillations, which are retained in the pruned tree, is still affected by the noise.

An analogous result can be achieved also working with persistence diagrams, where, instead of an elbow analysis, one has to choose a band close to the diagonal, as shown in Figure 4.7.4f, which contains all the small persistence features which one wants to eliminate. The band in Figure 4.7.4f has width 0.3. As for merge trees, if there is not clear separation between close-to-diagonal noisy points and more persistent features, one needs to make some arbitrary choice to fix a suitable band. For α-filtration of point clouds in $\mathbb{R}^n$ there are bootstrap methods to obtain confidence bands (Fasy et al., 2014).

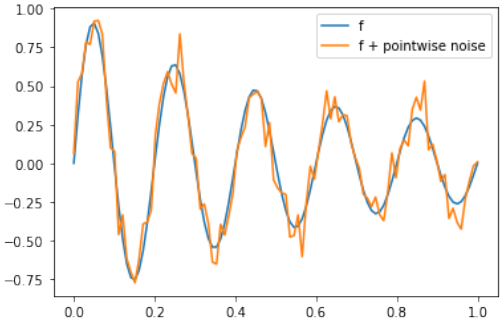
4.8 Case Study

We now run a comparative analysis of the real world Aneurisk65 dataset. This dataset – and the clinical problem for which it was generated and studied – was first described in Sangalli et al. (2009b), but it has since become a benchmark for the assessment of FDA methods aimed at the supervised or unsupervised classification of misaligned functional data (see, for instance, the special issue of the Electronic Journal of Statistics dedicated to phase and amplitude variability - year 2014, volume 8). We then repeat the classification exercise illustrated in Sangalli et al. (2009b) with the double scope of comparing merge trees and persistent diagrams when used as representations of the Aneurisk65 misaligned functional data, and of evaluating the performance of these representations for classification purposes when compared with the results obtained with the more traditional FDA approach followed by Sangalli et al. (2009b).

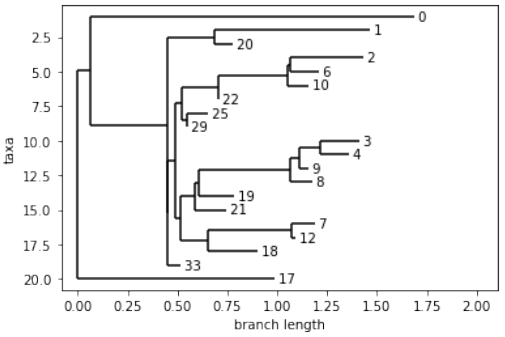
4.8.1 Dataset

The data of the Aneurisk65 dataset were generated by the AneuRisk Project, a multidisciplinary research aimed at investigating the role of vessel morphology, blood fluid dynamics, and biomechanical properties of the vascular wall, on the pathogenesis of cerebral aneurysms. The project gathered together researchers of different scientific fields, ranging from neurosurgery and neuroradiology to statistics, numerical analysis and bioengineering. For a detailed description of the project scope and aims as well as the results it obtained see its web page (https://statistics.mox.polimi.it/aneurisk) and the list of publications cited therein.

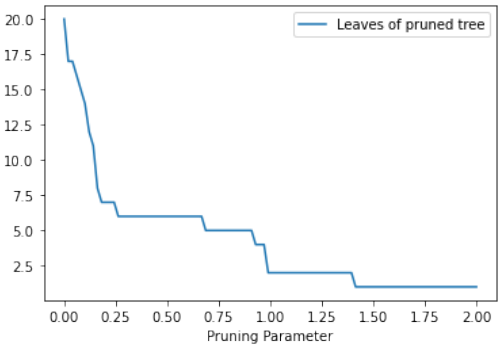
Since the main aim of the project was to discover and study possible relationships between the morphology of the inner carotid artery (ICA) and the presence and location of cerebral aneurysms, a set of three-dimensional angiographic images was taken as part of an observational study involving 65 patients suspected of being affected by cerebral aneurysms and selected by the neuroradiologist of Ospedale Niguarda, Milano. These



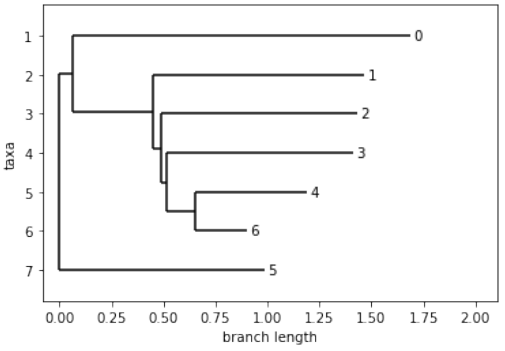
(a) The function f with and without noise (resp. orange and blue).



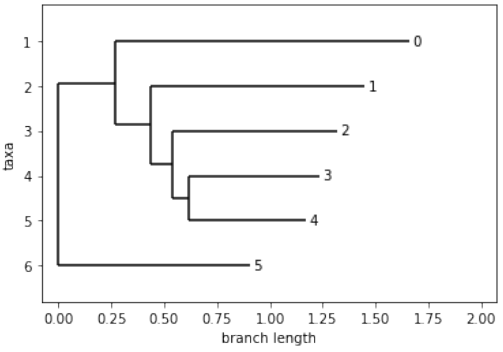
(b) Merge tree of g.



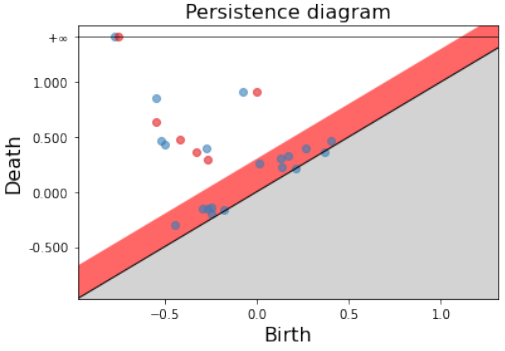
(c) Elbow analysis to identify the pruning threshold.



(d) Pruned merge tree of g.



(e) Merge tree of f.



(f) Persistence diagrams of f (red) and g (blue).

Figure 4.7.4: In the first row we can see the plot of the samples obtained from f and g on the left, and the merge tree obtained from the noisy sample on the right. The second row contains the plot ε vs number of leaves in $P_\varepsilon(T)$ and the merge tree $P_{0.3}(T)$. Lastly, we have the merge tree of f and the persistence diagrams of the samples of f (red) and g (blue), along with the red band identifying the noisy features.

3D images where then processed to produce 3D geometrical reconstructions of the inner carotid arteries for the 65 patients. In particular, these image reconstructions allowed to extract, for the observed ICA of each patient, its centerline "raw" curve, defined as the curve connecting the centres of the maximal spheres inscribed in the vessel, along with the values of the radius of such spheres. A detailed description of the pipeline followed to identify the vessel geometries expressed by the AneuRisk65 functional data can be found in Sangalli et al. (2014).

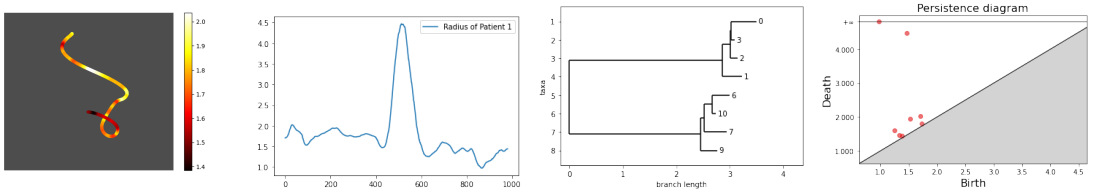
Different difficulties arise when dealing with this data. First, as detailed in Sangalli et al. (2009a), to properly capture information affecting the local hemodynamics of the vessels, the curvature of the centerline must be obtained in a sensible way. Retrieving the salient features of the centerline and of its derivatives is a delicate operation, which is heavily affected by measurement errors and reconstruction errors, due to the complex pipeline involved. Consequently the "raw" curves appear to be very wiggly and it is not obvious how to produce reasonable smooth representations. At the same time the 3D volume captured by the angiography varies from patient to patient. This is due to many factors, such as: the position of the head with respect to the instrument, which in turns depends on the suspected position of the aneurysm, the disposition of the vessels inside the head of the patient, the size of the patient. We can recognize these differences even by visual inspection in Figure 4.8.1: for instance, in Figure 4.8.1a and Figure 4.8.1i we see a longer portion of the ICA than in Figure 4.8.1e. Therefore the reconstructed ICAs cannot be compared directly: we need methods that take into account that the centerlines are not embedded in $\mathbb{R}^3$ in the same way, and that we cannot expect potentially interesting features to appear in exactly the same spots along the centerline. This is the typical situation where one should resort to alignment.

Hence, this dataset is paradigmatic of the three-faceted representation problem highlighted in the introduction; data smoothing, embedding, and alignment present difficult challenges, which propelled a number of original works in FDA.

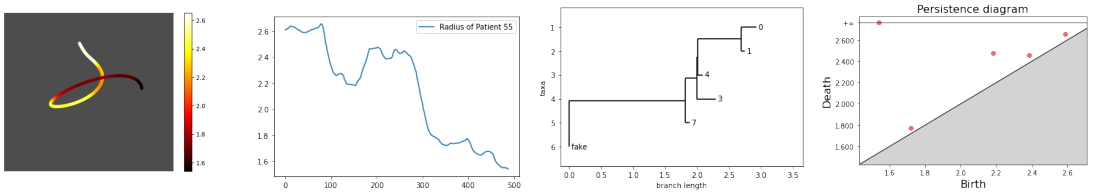
The AneuRisk65 data have been already partially processed; in particular centerlines have been smoothed following the free-knot regression spline procedure described in Sangalli et al. (2009a), and their curvatures were thus obtained after computing the first two derivatives of the smoothed curves. The data relative to the radius of the blood vessel, instead, although measured on a very fine grid of points along the centerline, is still in its raw format. Hence the AneuRisk65 data also allow us to compare the behaviour of tree representations on smoothed data and on raw data.

4.8.2 Analysis - Classification

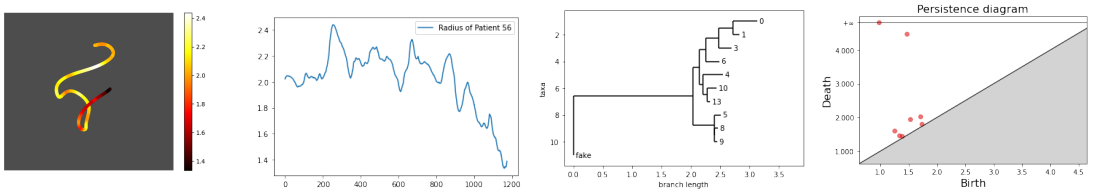
Patients represented in the AneuRisk65 dataset are organized in three groups: the Upper group (U) collects patients with an aneurysm in the Willis circle at or after the terminal bifurcation of the ICA, the Lower group (L) gathers patients with an aneurysm on the ICA before its terminal bifurcation, and finally the patients in the None group (N) do not have a cerebral aneurysm. Our main goal is supervised classification with the aim to develop a classifier able to discriminate membership to the group $L \bigcup N$ against membership to the group U based on the geometric features of the ICA. In Section 4.8.4 and Section 4.8.3, we complement this supervised analysis with a descriptive analysis of the merge trees, aggregated according to their group membership, and an unsupervised exercise which aims at clustering patients solely on the basis of the similarity of geometric features of their ICA, thus recovering a clear structure between the groups listed above and providing further support to the discriminating power of the geometric features of the ICA.



(a) ICA patient 1 (L). (b) Radius along the ICA of patient 1 (L). (c) Merge tree associated to the radius function of patient 1 (L). (d) Persistence diagram associated to the radius function of patient 1 (L).



(e) ICA patient 55 (U). (f) Radius along the ICA of patient 55 (U). (g) Merge tree associated to the radius function of patient 55 (U). (h) Persistence diagram associated to the radius function of patient 55 (U).



(i) ICA patient 56 (U). (j) Radius along the ICA of patient 56 (U). (k) Merge tree associated to the radius function of patient 56 (U). (l) Persistence diagram associated to the radius function of patient 56 (U).

Figure 4.8.1: Three patients in the AneuRisk65 dataset; on the first column of the left, the ICAs of the patients are coloured according to the radius value, on the second column there are the radius functions, on the third column their associated merge trees and on the rightmost column the persistence diagrams. Patient 1 belongs to the Lower group (L), the other two patients to the Upper group (U). Note that the merge trees have been truncated with K equal to the maximum of Figure 4.8.1b.

4.8.2.1 The pipeline for supervised classification

We develop a classification pipeline in close analogy with the one illustrated in Sangalli et al. (2009b) which, after smoothing, reduces the data dimensionality by means of Functional Principal Components Analysis (FPCA) applied to the curvature functions of the ICA centerlines and to the respective radius functions, and then fits a quadratic discriminant analysis (QDA) based on the first two FPCA scores of the curvature functions and of the radius functions respectively.

We interpolate the data points representing the smoothed curvature functions and the raw radius functions provided in the Aneurisk65 dataset with a piecewise linear spline and, for each patient, we consider the merge tree associated to its curvature and the merge tree associated to its radius. We then prune our tree representations; to use a uniform scale across all patients (but of course different for curvature and radius) we parametrize the pruning threshold as a fraction of the total range covered by the curvature and radius functions, respectively, across patients: $I = [\min_f(\min_x(f(x)), \max_f(\max_x(f(x)))]$. For both sets of trees, we then calculate the pairwise distances with the metric d_E and we organize them in two distance matrices. Blending the discriminatory information provided by curvature and radius, we also produce a new distance matrix collecting the pairwise distances obtained by convex linear combination of the distances for curvature and radius, according to the formula:

$$d^2_{\text{mixed}} = w \cdot d^2_{\text{curvature}} + (1 - w) \cdot d^2_{\text{radius}}, \tag{4.1}$$

where $0 \leq w \leq 1$. For lack of references, we prove in Section 4.B that d_{mixed} is a metric, for all $w \in [0, 1]$. We then apply Multi Dimensional Scaling (MDS) to each of the above distance matrices, to map the results in a finite dimensional Euclidean space of dimension m. Lastly, and following Sangalli et al. (2009b), we fit a QDA on such embedded points.

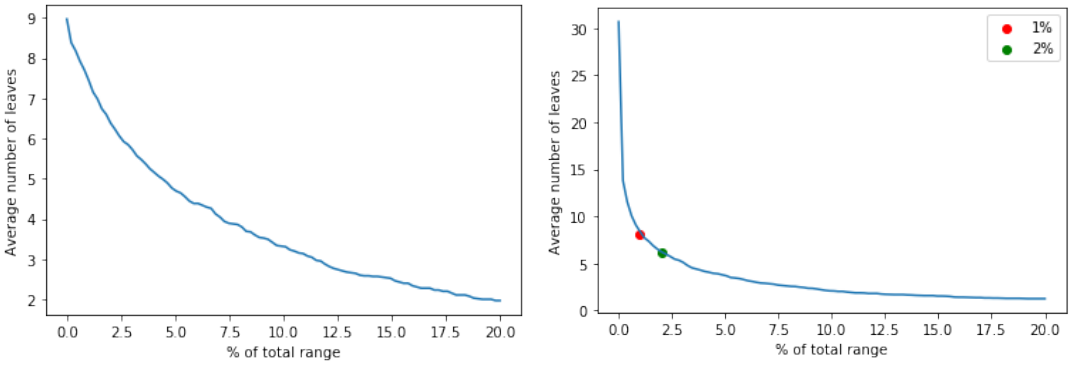
This pipeline requires the setting of three hyperparameters: the pruning threshold, the weight w appearing in Equation (4.1) and, finally, the dimension m of the Euclidean embedding for MDS. While the pruning threshold is chosen with an elbow analysis, see Section 4.8.2.2, the weight w and the dimension m of the multidimensional scaling are selected by maximising the discriminatory power of QDA estimated by means of leave-one-out (L1out) cross-validation.

4.8.2.2 Pruning

In this section, we take a closer look at the smoothing carried out by pruning the merge trees representations of curvature and radius. From the plots in Figure 4.8.1 we see that the radius functions appear to be very wiggly and, given the complex data-generating pipeline, we might assume that some portion of that amplitude variability is uninformative and due to different kinds of errors, which is the same conclusion drawn by Sangalli et al. (2009a) with respect to the raw curvature data.

Assuming, as in Section 4.7.4, that the signal-to-noise ratio is not too low, we expect to find some separation in terms of amplitude between the informative features of the analyzed functions and those to be considered as nuisances. Hence, as in Section 4.7.4, we choose the pruning parameter through an elbow analysis of the curve plotting the number of leaves of the pruned trees, averaged over the whole dataset, against the corresponding threshold. Thus, we look for an elbow in the curves depicted in Figure 4.8.2.

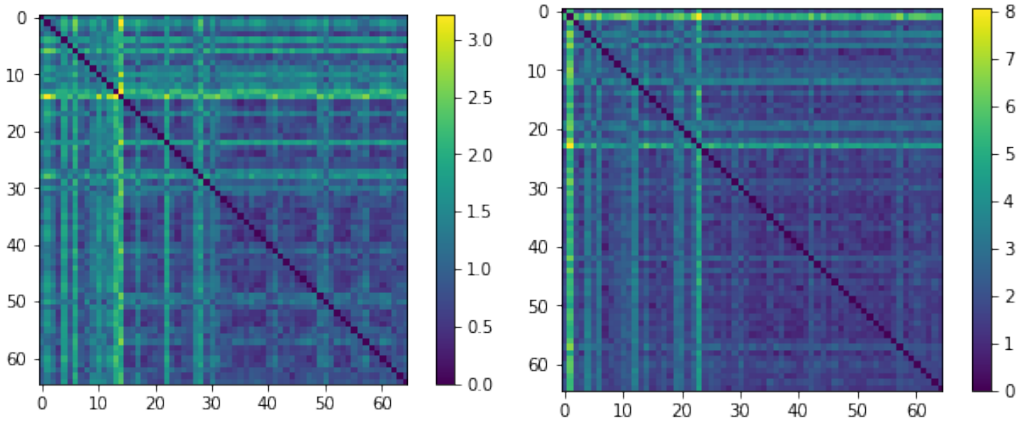
Here we want to emphasise the different behaviours of the curvature trees and of the radius trees. There is no clear elbow in Figure 4.8.2a, showing that there is no reason to believe that data show a large number of small uninformative oscillations. This is not surprising because the curvature functions of the Aneursik65 dataset are the result of a very careful smoothing process. The curve in Figure 4.8.2b, related to radius, has instead



(a) Average number of leaves for curvature

(b) Average number of leaves for radius

Figure 4.8.2: The average numbers of leaves in the merge trees, plotted against the percentage of total range used as pruning threshold.



(a) Pairwise distances between merge trees representing curvatures.

(b) Pairwise distances between merge trees representing radii.

Figure 4.8.3: The distance matrices of merge trees associated to curvature and radius functions. Patients belonging to group L appear in first rows, followed by patients in the N group and patients in the U group.

a clear elbow structure (between 1% and 2%) in accordance with our expectations. Thus, we choose 2% as pruning threshold for the radius curves, whilst we do not prune curvature trees. We later discuss the robustness of our results with respect to these choices.

4.8.2.3 Classification Results

We compare our classification results with those illustrated Sangalli et al. (2009b). The goal is the same: separating the class U from the classes L and N.

Table 4.8.1 reports the prediction errors obtained after L1out cross-validation. As in Sangalli et al. (2009b), we obtain the best classifier by simultaneously considering the combined information conveyed by the couple of curvature and radius functions; the dissimilarity between different couples is measured by the distance in Equation (4.1), where the parameter $w = 0.25$, being this the value which minimizes prediction error computed by L1out.

The same pipeline is followed when curvature and radius functions are represented by merge trees or by persistence diagrams. In the case of PDs', we first removed the points

Merge Trees

Curvature ($w = 1$, $n = 3$)

True \ Predicted	U	L⋃N
U	18	14
L⋃N	3	30

Radius ($w = 0$, $n = 12$)

True \ Predicted	U	L⋃N
U	24	8
L⋃N	7	26

Mixed ($w = 0.25$, $n = 14$)

True \ Predicted	U	L⋃N
U	26	6
L⋃N	5	28

Persistence Diagrams

Curvature ($w = 1$, $n = 3$)

True \ Predicted	U	L⋃N
U	20	12
L⋃N	3	30

Radius ($w = 0$, $n = 9$)

True \ Predicted	U	L⋃N
U	26	6
L⋃N	6	27

Mixed ($w = 0$, $n = 9$)

True \ Predicted	U	L⋃N
U	26	6
L	6	27

Benchmark

True \ Predicted	U	L⋃N
U	26	6
L⋃N	6	27

Table 4.8.1: Confusion matrices for L1out. Below each confusion matrix, the values of the metric coefficient w and of the dimension m for MDS corresponding to the tested classifier are reported. The first row refers to the classifiers receiving as input merge tree representations, the second row PDs. The last row reports the benchmark L1out confusion matrix for the classifier illustrated in Sangalli et al. (2009b).

(that is, the topological features) with persistence lower than a certain threshold (where the persistence is $c_y - c_x$, according to the notation used in Section 4.3). The threshold has been taken equal to the pruning parameter of the merge trees, in accordance with Remark 4.17.

The first two rows in Table 4.8.1 compare prediction errors when merge trees or PD representations are used. From left to right, the table shows the L1out confusion matrices when distances between curvatures, radii or their joint couple are respectively considered. We see that PDs do a slightly better job in extracting useful information from either curvature or radius, when examined separately. This could be due to a situation not dissimilar from that illustrated in the example of Section 4.7.1: the discriminant information contained in the curvature and radius functions lies more in the number and amplitude of oscillations than in their ordering. However, when curvature and radius of the ICA are jointly considered as descriptors and the distance of Equation (4.1) is used, we obtain a better classifier for merge trees while there is no improvement for PDs.

This situation highlights that merge trees and persistence diagrams capture different but highly correlated pieces of information about the current functional data set; note, however, that PDs suggest that most of the information they capture is due to the radius function, while merge trees show some informative interactions between curvature and radius.

For comparison, the third column of Table 4.8.1 reports the prediction errors of the best classifiers based on merge trees and on PDs, respectively, while the last row shows the prediction errors of the classifier described in Sangalli et al. (2009b). The number of patients misclassified by the best classifier based on merge trees is slightly smaller than that of the best classifier based on PDs, but, despite the profound differences between the two topological summaries (see Section 4.4, Section 4.7.1, Section 4.7.2 and Section 4.7.3) the two methods are retrieving similar discriminant information related to the classification task: comparing the two analysis we found that 10 patients were misclassified by both methods.

4.8.2.4 Robustness with respect to the pruning threshold

To argument in favor of the robustness of our results with respect to the choice of the pruning threshold, or, from another point of view, in favor of the robustness of the information conveyed by our tree representation of functions, we go through the same classification pipeline varying the value of the pruning threshold of the radius functions. In Figure 4.8.4 we show the prediction accuracy, estimated by L1out cross-validation, as a function of the pruning threshold. We notice that the accuracy is quite stable and does not oscillate wildly upon perturbing the pruning threshold. This fact, on one hand further supports the elbow analysis approach described in Section 4.8.2.2, on the other is also showing that the results obtained with the information captured by merge trees and persistence diagrams does not depend on a finely tuned choice of the threshold parameters.

4.8.3 Analysis - Clustering

In this section, we present an unsupervised exercise with the aim of clustering patients on the basis of the similarity of geometric features of their ICA. We explore the Aneurisk65 data clustering structure by endowing the merge tree space with the metric d_{mixed} figuring in Equation (4.1), with $w = 0.25$. To get multiple perspectives on this issue, we resort to hierarchical clustering dendrograms with different linkages. The visual inspection of Figure 4.8.3 suggests that, upon blending together the information of radius and curvature, the Upper class should display a low variability while the Lower and None classes should behave more heterogeneously. Thus, a clear clustering structure should not be recogniz-

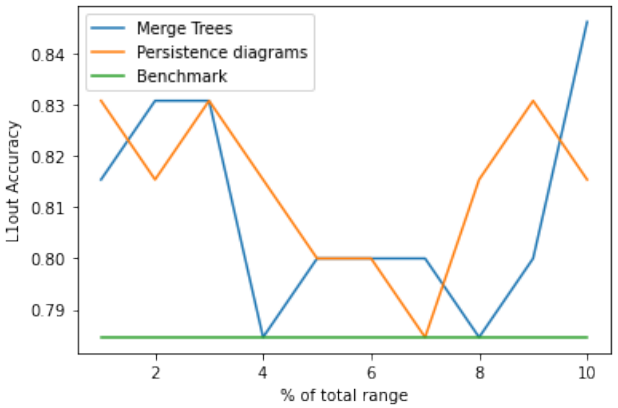


Figure 4.8.4: L1out accuracy of the classification pipeline - both for merge trees and persistence diagrams - with respect of the pruning threshold. The horizontal green line shows the accuracy obtained by Sangalli et al. (2009b). Note that the accuracy of persistence diagrams and merge trees is above or equal to the green line also for large values of the pruning threshold.

able: we expect possibly one cluster made by points belonging to the Upper class and then a series of points scattered around this central nucleus with no easily recognizable pattern.

The hierarchical dendrograms obtained with single, average and complete linkages are displayed in Figure 4.8.5. The first obvious observation is that all three linkages identify the point associated to patient 2 as an outlier. The single linkage dendrogram shows that, as the height on the dendrogram increases, there is only one major cluster which slowly becomes larger and incorporates all points in the data set. No other relevant clusters are found. Average and complete linkages further support this finding: there are no obvious heights where to cut the tree in the average linkage dendrogram; complete linkage instead shows perhaps a two cluster (plus one outlier) structure. The smaller cluster identified by this dendrogram, is also visible with the average linkage and is contained within the group of singletons obtained by cutting the single linkage tree at height 1.3. The overall picture is thus that of a major cluster, with possibly another group of points clustered together, but with much higher heterogeneity.

These findings can indeed be related with the labels declaring membership of the patients to the U, the N and the L group respectively. To grasp if there is an overall pattern in the merging structure of the data point cloud, for each leaf (a patient) of a dendrogram, we collect its merging height defined as the height of its father in the graph, that is the height at which that point is no longer considered as a singleton but instead it is clustered with some other point. In other words, we record the distance between the leaf and the closest cluster in terms of the cophenetic distance induced by the dendrogram. Note that, for the single linkage dendrogram, this is equivalent, for almost all leaves, to the height at which the leaf is merged with the major cluster. Results are shown in Figure 4.8.6. The interpretation of these plots is consistent across the different linkages and is pretty straightforward: the points corresponding to patients of the Upper group get merged within a small range of heights, and the distribution of their merging height is stochastically smaller than the distributions for groups L and N, respectively. The merging heights of the leaves corresponding to patients belonging to the Lower group, instead, display a larger variability and their distribution is stochastically larger than those of the of the leaves belonging to the other two groups. Patients of the class None, merge at heights in between the Upper and the Lower groups and their merging height seems to display a low variability. The plot (d) of Figure 4.8.6 shows the smoothed densities of

the distributions of merging height for leaves belonging to the three groups, in the case of average linkage. Analogous representations could be obtained for the other two linkages; they all confirm the stochastic ordering described above.

This cluster analysis is consistent with our expectations, which, in turn, are in accordance to the findings of Sangalli et al. (2009b). On top of that, we also get two further insights: first, data are scattered around the Upper group with a possibly non uniform structure, as shown by the small and sparse cluster of Lower class patients visible with complete linkage clustering and, second, that the None group of patients lies in a sort of in between situation in the space separating the two other groups of aneurysm-affected patients. This could also explain the good performance of QDA: a quadratic boundary is able to isolate the core of the Upper group of patients from the others, which lie mainly on one side of the quadratic discriminant function.

4.8.4 Reading the trees

Though not strictly related to the classification task presented in Section 4.8.2 and the clustering carried out in Section 4.8.3, we take the opportunity to briefly investigate the population of trees obtained from the Aneurisk data set to see what can be said about the different groups (U,N,L) and to try to explain the effectiveness of the proposed analysis. This is intended as a first step into developing further statistical tools to interpret and understand populations of merge trees.

A path which has been taken to produce summary statistics of a topological representation - in particular a persistence diagram - is to obtain a function for each topological summary and then use consolidated (functional) statistical methods to analyze the obtained curves (see for instance Sørensen et al. (2020)). We will do the same, producing a curve for every merge tree, in three different ways.

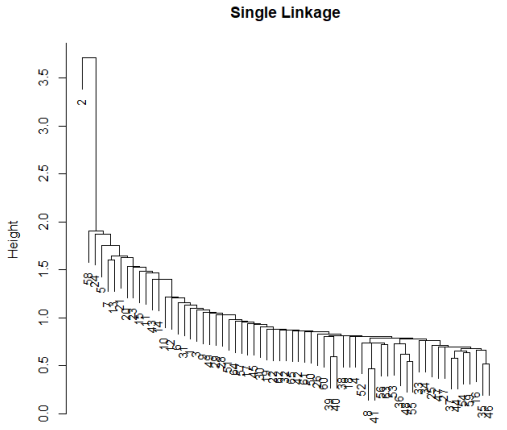
It must be immediately pointed out that the relationship between the topological representation and the chosen summary functional statistic should be formally addressed in a number of ways. For instance, pointwise or uniform convergence of the functional summaries when diagrams/merge trees/etc. are close according to some metric should be studied as well as the injectivity of the functional representation of the topological summary. These aspects of topological data analysis up to now have been hardly explored in the literature; they require efforts and space which are outside the scope of the present work. Our aim here is just to showcase qualitatively the discriminative power, in our case study and examples, of some functional statistics in order to foster and motivate further efforts on the topic.

We make use of the following definition.

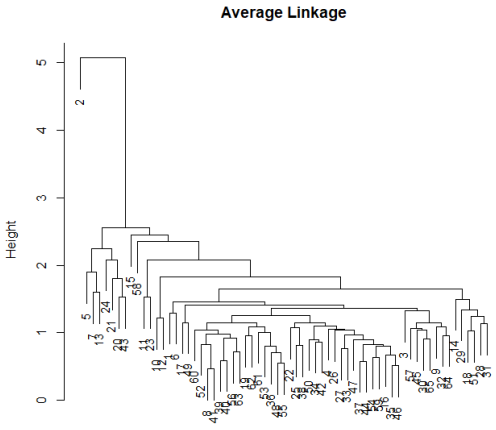
Definition 4.24. *Let (T, h_T) be a merge tree; cutting T at height $h \in \mathbb{R}$ generates the set of merge trees $Cut_T(h) = \{sub_T(v) \,|\, v \in V_T \text{ s.t. } h_T(v) \leq h \text{ and } h_T(father(v)) > h\}$.*

Now, consider a merge tree (T, h_T) associated to a function $f : X \to \mathbb{R}$, with X compact interval in $\mathbb{R}$. Cutting T at height $h \in \mathbb{R}$, means looking locally at the oscillation patterns obtained by removing from the domain of f the super level set $X^h = f^{-1}((h, +\infty))$: the different subtrees represent the function f restricted to the different path connected components of $X - X^h$. Thus each subtree describes locally the function f, with h controlling the resolution of these descriptors. Based on this fact, we introduce the following functional statistics. For each tree (T, h_T) and for each $h \in \mathbb{R}$ we look at:

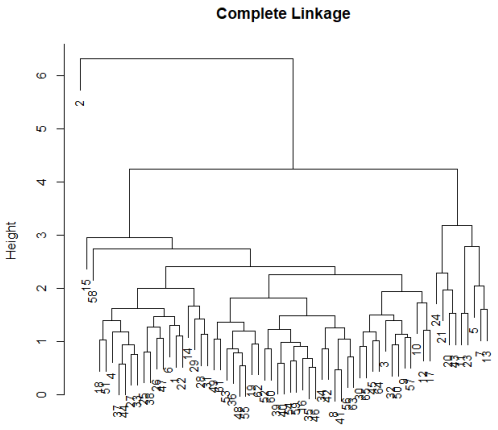
1. $var(T, h)$: the variance of $\{\#L_{T'}/\#L_T \,|\, T' \in Cut_T(h)\}$. With this statistic we capture how the already completed oscillations of f are grouped together, focusing on the homogeneity of these patterns: high variance suggest that the merge tree is very unbalanced below height h, implying that oscillations completed below h are



(a) Single linkage clustering dendrogram.

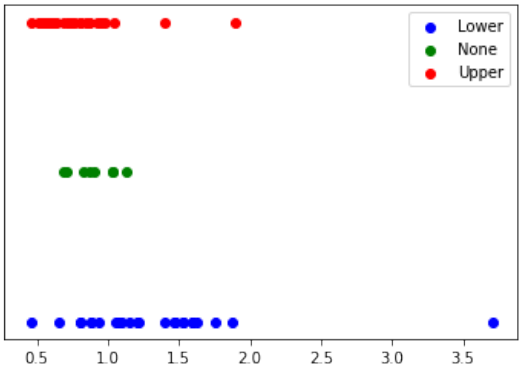


(b) Average linkage clustering dendrogram.

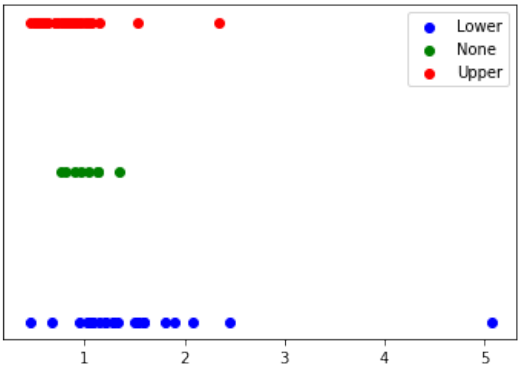


(c) Complete linkage clustering dendrogram.

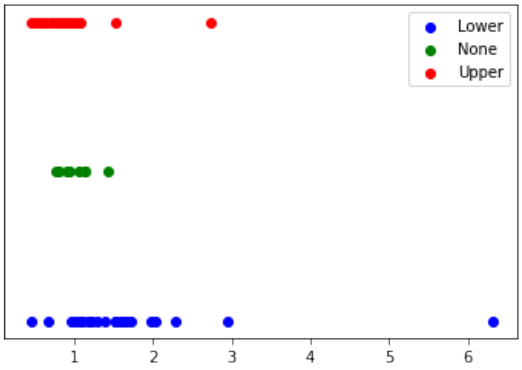
Figure 4.8.5: Hierarchical clustering dendrograms obtained with single, average and complete linkages.



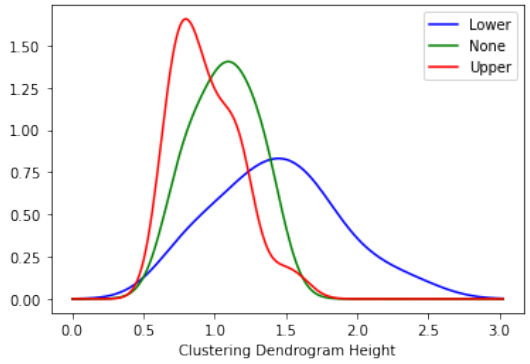
(a) Heights at which leaves get merged in the single linkage clustering dendrogram.



(b) Heights at which leaves get merged in the average linkage clustering dendrogram.



(c) Heights at which leaves get merged in the complete linkage clustering dendrogram.



(d) Density estimate for the distributions in Figure 4.8.6b.

Figure 4.8.6: For each patient belonging to group U, L or N, the plots (a), (b) and (c) represent the merging height at which their corresponding leaf gets merged in the clustering dendrograms, according to single linkage, average linkage and complete linkage, respectively. Plot (d) represents the smoothed densities of merging height for the leaves of the three groups, in the case of average linkage.

more concentrated in some regions of the domain compared to others. Low variance, instead, indicates that oscillations are more evenly distributed;

2. $NLeaves(T,h)$: the statistic $NLeaves(T,h) = \#\{l \in L_T \,|\, h_T(l) \leq h\}$ counts the oscillations which have already started at height h;

3. $NInt(T,h)$: the statistic $NInt(T,h) := \#\{v \in V_T - L_T \,|\, h_T(v) \leq h\}$ counts the oscillations which have been completed before height h.

Note that the first statistic we present is strictly related to information contained in merge trees and so it does not apply to persistence diagrams; the last two, instead, respectively count the number of components born before h and the ones which have died before time h. These statistics can also be computed for persistence diagrams.

Letting h vary in $\mathbb{R}$, for every tree we obtain a curve, a functional statistic. These curves can then be aggregated over different subsets partitioning the population of trees.

4.8.4.1 Examples in Section 4.7

To gain confidence with these tools, we look at the simulated scenarios presented in Section 4.7.1, Section 4.7.2 and Section 4.7.3 through the lenses of these statistical summaries. The reader can follow the next paragraph looking at Figure 4.8.7.

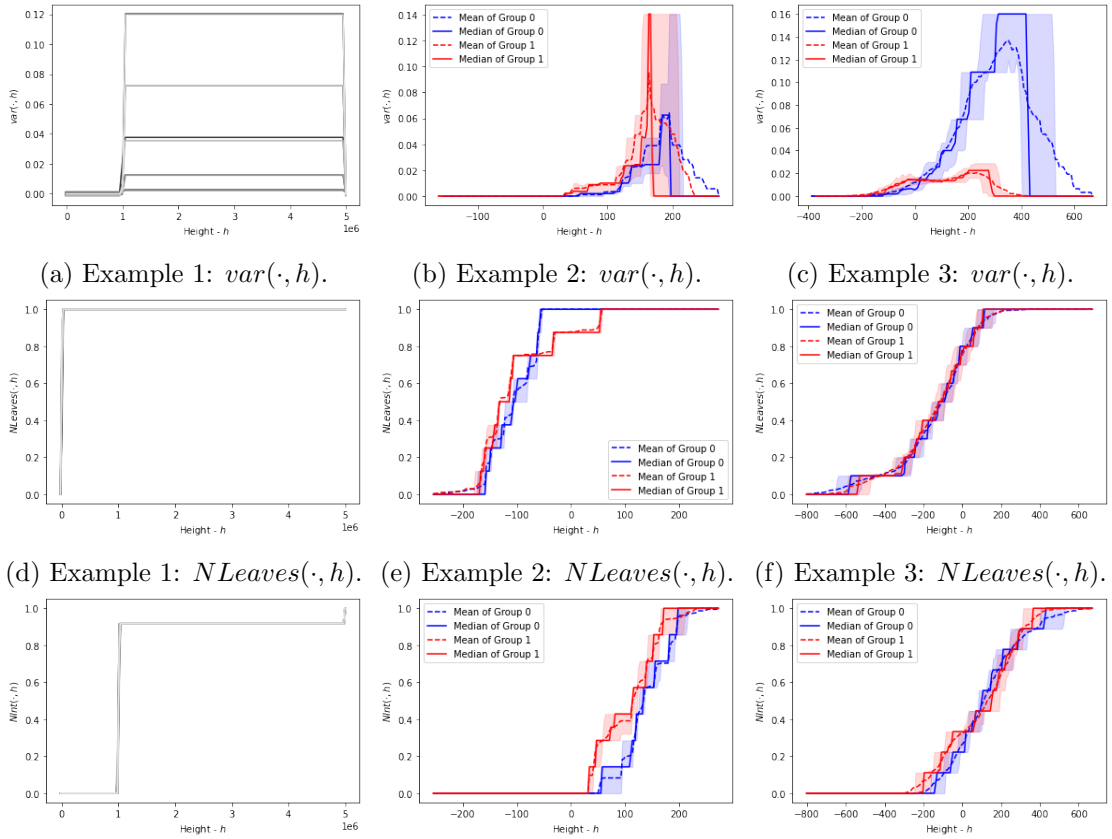
The first example, Section 4.7.1, presents a data set of functions such that the associated merge trees are distinguished just by the tree structures: in fact the associated persistence diagrams are all equal. Each tree as a total of $n = 13$ leaves, the father of each leaf is at height 1 and cutting each tree between heights ≥ 1 and < 5 yields two clusters. Thus, the only difference between the trees lies in the cardinality of such clusters. Accordingly, the only statistic which is able to separate them is the variance between the clusters cardinalities: we clearly see that in Figure 4.8.7a. The other plots, as expected, show perfectly superimposed curves.

The second example, Section 4.7.2, presents data divided into two groups with the discriminant information between the groups lying into the heights of the critical points of the functions. Figure 4.7.2b and Figure 4.7.2c show that both merge trees and persistence diagrams can distinguish, to some extent, the two groups, with persistence diagrams being able to identify them more clearly. Figure 4.8.7e and Figure 4.8.7h show that the groups have two very distinct behaviors in terms of births and deaths of path connected components: as the height parameter h runs through the heights of the merge trees, we can clearly see that shaded areas of the different classes are well separated. Moreover, in Figure 4.8.7b we see that we have a lot of within-group variability generated by tree structures, which is in perfect accordance to the generative process of the data set. This further explains why persistence diagram do such a good job in separating the two groups.

The third example reverses the situation: births and deaths of path connected components should not present different patterns in the two groups, while the tree structures involved should present profound differences between groups. From Figure 4.8.7f and Figure 4.8.7i it is quite evident that heights of local minima and maxima do not contain enough information to separate groups. Instead Figure 4.8.7c shows that while Group 1 presents very balanced tree structures, for $var(T,h)$ is very low for h high, Group 0 has very imbalanced merging structures, with many components being merged together and few standalone ones appearing/merging late.

4.8.4.2 Aneurisk65 Case Study

In this section we illustrate for the Aneurisk65 case study the same analysis obtained in Section 4.8.4.1 for the simulations described in Section 4.7.1, Section 4.7.2 and Section 4.7.3.



(a) Example 1: $var(\cdot, h)$. (b) Example 2: $var(\cdot, h)$. (c) Example 3: $var(\cdot, h)$.

(d) Example 1: $NLeaves(\cdot, h)$. (e) Example 2: $NLeaves(\cdot, h)$. (f) Example 3: $NLeaves(\cdot, h)$.

(g) Example 1: $NInt(\cdot, h)$. (h) Example 2: $NInt(\cdot, h)$. (i) Example 3: $NInt(\cdot, h)$.

Figure 4.8.7: Functional statistics related to the data sets presented in Section 4.7.1, Section 4.7.2 and Section 4.7.3. Each column is devoted to one example. The plotted lines represent the pointwise median (continuous line) and mean (dotted line) of the curves. The shaded regions delimit the pointwise central quartiles Q1 and Q3 of the data set.

4.8.4.3 Curvature

First we focus on the analysis of the merge trees obtained from the functions recording the curvature of the blood vessels centrelines. In the first row of Figure 4.8.8 we see the plots of the statistics we want to discuss.

Figure 4.8.8a tells us that the tree structures of the three classes exhibit quite different behaviors: we see that L tend to be more imbalanced than N and U (especially looking at the median values), but both L and N have many outliers which cause median and mean values to be very different. However, due to the within-group variance it is not possible to discriminate between groups using such statistic, as the shaded area of group L includes the areas of the other classes. Still, we can appreciate that class U shows much less within-group variance compared to N and L. Figure 4.8.8b and Figure 4.8.8c, together, carry the following information: class U tends to have less oscillations compared to class L. Moreover, such oscillation tend to take place at lower values, as the red curves flatten before the blue one. Class N instead lies in between the other two groups.

To conclude, the differences we can spot in the merge trees associated to curvature functions lie mostly in the number of oscillations and in their height, with class U being generally more characterized (i.e. with smaller within group variance) then the other classes. Not much remnant discriminating information seems to be contained into the tree structures of the different groups.

4.8.4.4 Radius

Now we turn to merge trees associated to radius functions - second row of Figure 4.8.8.

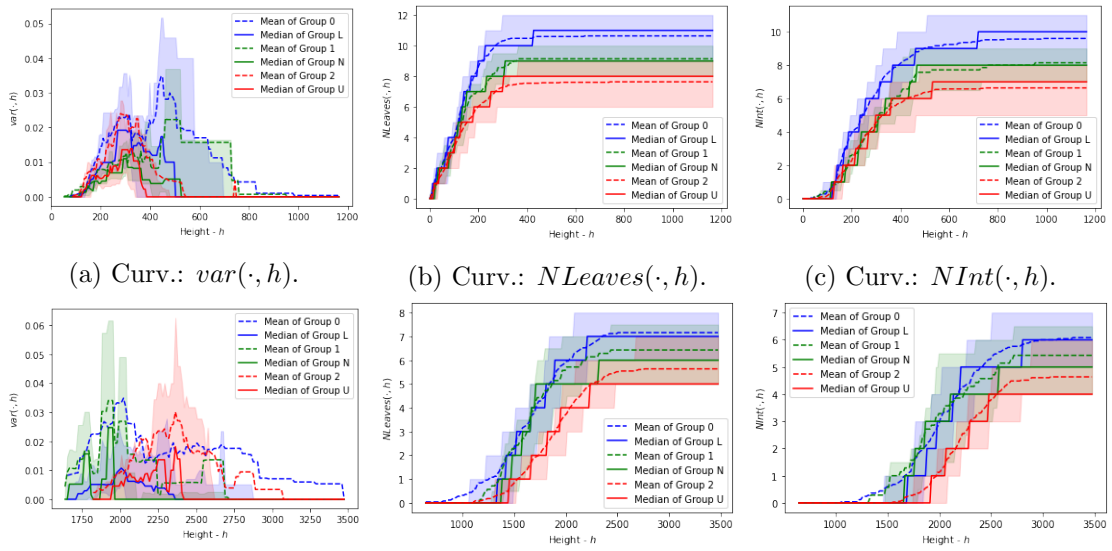
Figure 4.8.8d depicts a situation which is not very different from Figure 4.8.8a, even if the roles of the different labels are exchanged: we have group U showing very high variability for high values of h, with ouliers being present for all groups - the pointwise means end up outside the shaded regions. Both Figure 4.8.8d and Figure 4.8.8a may indicate something related to radius/curvature happening at the extremes of the domain, in particular for class U (radius) and L (curvature). However, as for curvature, these patterns do not discriminate between the groups L, N, U due to the within group variance being very high. Figure 4.8.8e and Figure 4.8.8f tell us that the class U, again, has consistently less oscillations than L and N. On top of that, class L tend to have lower minima and higher maxima than class U.

To sum up, the radius functions of class U have less oscillations, compared to L and N, and the overall amplitude of such oscillations is also smaller.

4.8.4.5 Conclusion

Putting together the information collected in the previous subsections we can say that the curvature functions associated to patients in group U oscillate less then the other groups, with their peaks being lower than the ones of L and N. The radius functions of the group U still oscillate less in terms of numbers of local minima and amplitude of the oscillations.

In other words, blood vessels from group L are more wiggly in the space (and thus their curvature oscillates more, touching higher values) and their diameter oscillates more frequently and by a greater amount compared to class U. These features might indeed be associated to different blood fluid dynamics regimes characterizing patients in group U. Group N lies in between, some patients showing patterns closer to group L and some others to group U. Being the groups L,N,U characterized by this topological information, we can see why both merge trees and persistence diagrams perform well in the pipelines presented in Section 4.8.2. At the same time, the fact that the tree structures do not show particular patterns - Figure 4.8.8a and Figure 4.8.8d - could explain why merge trees and persistence diagrams perform similarly.



(a) Curv.: $var(\cdot, h)$. (b) Curv.: $NLeaves(\cdot, h)$. (c) Curv.: $NInt(\cdot, h)$.

(d) Rad.: $var(\cdot, h)$. (e) Rad.: $NLeaves(\cdot, h)$. (f) Rad.: $NInt(\cdot, h)$.

Figure 4.8.8: Functional statistics related to merge trees obtained from the curvature (Curv.) and the radius (Rad.) of the blood vessels. The shaded regions delimit the pointwise central quartiles Q1 and Q3 of the data set.

The work of Sangalli et al. (2009b) reports that the geometrical features found relevant to describe the data - namely: the overall width of the ICA, its tapering effect, the curvature of the ICA in proximity of the last peak of curvature and the curvature along the segment of the ICA between the two peaks of curvature - show a lower variance for the group U, when compared to group L; and patients in group U tend to have a wider and more tapered ICA's than those in group L. They also present a less curved ICA between the two peaks of curvature. We can appreciate that our findings are in accordance with the ones of Sangalli et al. (2009b): indeed, we found in Section 4.8.3 that the group U exhibits a much smaller variability in terms of merge trees, being its trees clustered together much more than the trees of the other groups. Similarly the information we capture on curvature and radius are coherent. Nevertheless we can also appreciate how the two approaches complement each other: Sangalli et al. (2009b) are able to identify differences in precise locations of the ICA, while the differences in terms of global oscillating behaviors and regimes are much better captured by the topological summaries.

4.9 Discussion

We believe that methods from TDA can be fruitfully added to the toolbox of functional data analysis, especially when non trivial smoothing and alignment are required for data representation. In this chapter we focused on two topological representations of functions: persistence diagrams, which, being the most classical tool in TDA, are regarded as a benchmark, and merge trees, which are rarely used in real data analysis applications. The framework for merge trees is the very recent metric structure defined in Chapter 2, for which we also developed theoretical results specific for the application to functional data.

To support our narrative, we used as paradigmatic real world application the classification analysis of the AneuRisk65 functional data set. This data set poses all the desired challenges: careful smoothing procedures and alignment techniques must be employed to obtain meaningful results. Reanalyzing the seminal case study described in Sangalli et al. (2009b), we show the advantages of having a representation of functional data which is

invariant with respect to homeomorphic transformations of the abscissa – thus lightening the burden of careful alignment – and also allows for agile smoothing – possibly causing some overfitting – thanks to the pruning of the trees which takes care of this aspect of FDA which practitioners often find problematic. Following a classification approach based on QDA applied to proper reduced representations of the data, as in Sangalli et al. (2009b), we obtain robust results with comparable, if not better, accuracy in terms of L1out prediction error, and we confirm some facts about the variability of the data in the groups of patients characterized by the different location of the cerebral aneurysm, consistently with the findings of previous works.

The effectiveness of the simple pipeline proposed in the case study does motivate further research in order to deal with more complicated scenarios including multivariate functional data in which a vector of functions defined on the same domain could be summarized via a topological representation. Similarly, statistical tools to better interpret population of trees should be studied and developed, starting for instance from the brief discussion carried out in Section 4.8.4. This would open up the door for more refined statistical procedures like testing or uncertainty quantification, which are very hard to deal with in general metric spaces. On top of that, optimizing the numeric and computational aspects of the tree-based tools that we introduced would surely make them more viable in applications.

To be sure, we want to stress that careful smoothing is still mandatory when precise differential information about the data is needed, since small oscillations in a function can still cause high amplitude oscillations in the derivatives, which cannot be removed by pruning. Moreover, not all FDA applications are adapted to the representations offered by merge trees or persistent diagrams. Indeed, the information collected by merge trees is contained in the ordering and in the amplitude of the extremal points of a function, and not on their exact abscissa. Hence, if the abscissa carries valuable information for the analysis – for instance, a wavelength, or a precise landmark point in space or time – the TDA approach followed in this work for data representation is not indicated, precisely because of its invariance property with respect to homeomorphic transformations of the abscissa. But this criticism also applies to many alignment procedures proposed in the literature. Similarly, in Section 4.7.2, we point out that there are functions which have equivalent representations in terms of merge trees although the order on the abscissa of their critical points is different, although merge tree are much less sensitive to such issue when compared to persistence diagrams (see also Curry et al. (2021)). If the order of critical points of the function is of importance for the analysis, then surely persistence diagrams, but possibly also merge trees, should be avoided.

Going general, we point out that whenever the datum designating a statistical unit is only a representative of an equivalence class, the analyst must be sure that the variability differentiating the members of the same class is ancillary with respect to the statistical analysis performed on the statistical units. This consideration always applies in FDA, whenever data are aligned according to transformations belonging to a group. Merge trees offer a representation of functional data in terms of equivalence classes whose members are invariant with respect to homeomorphic transformations of the abscissa. Persistence diagrams partition the space of functional data in even coarser equivalence classes, although they could be enough for the analysis, as we saw in the case study illustrated in Section 4.8. Occam's razor should guide the analyst's final choice.

Appendix

4.A Proofs

Proof of Proposition 4.11.

Let $f : X \to \mathbb{R}$ be a bounded function defined on a path connected topological space X and let $\varphi : Y \to X$ be an homeomorphism. We need to prove that the merge tree and the persistence diagram associated to the function f and $f' = f \circ \varphi$ are isomorphic.

We know that:

$$Y_t = \{f'^{-1}((-\infty, t])\} = \{y | f'(y) \leq t\} = \{x = \varphi(y) | f(x) \leq t\}$$

This means that $y \in Y_t$ if and only if $\varphi(y) \in X_t$, and so $Y_t = \varphi^{-1}(X_t)$. Since the restriction of an homeomorphism is still an homeomorphism, we can take its inverse, and by the composition properties of π_0, we obtain that $\pi_0(X_t) \cong \pi_0(Y_t)$. Given $t' < t$, we thus have the following commutative diagram:

$$\begin{array}{ccc} X_{t'} & \longrightarrow & X_t \\ \downarrow{\scriptstyle \varphi} & & \downarrow{\scriptstyle \varphi} \\ Y_{t'} & \longrightarrow & Y_t \end{array}$$

and passing to path connected components/homology:

$$\begin{array}{ccc} \pi_0(X_{t'}) & \longrightarrow & \pi_0(X_t) \\ \downarrow{\scriptstyle \pi_0(\varphi)} & & \downarrow{\scriptstyle \pi_0(\varphi)} \\ \pi_0(Y_{t'}) & \longrightarrow & \pi_0(Y_t) \end{array}$$

$$\begin{array}{ccc} H_p(X_{t'}) & \longrightarrow & H_p(X_t) \\ \downarrow{\scriptstyle H_p(\varphi)} & & \downarrow{\scriptstyle H_p(\varphi)} \\ H_p(Y_{t'}) & \longrightarrow & H_p(Y_t) \end{array}$$

where the vertical arrows in the second diagram are isomorphisms of groups. The first diagram then gives the isomorphism of merge trees, while the last one gives the isomorphism of $PD_0(f)$ and $PD_0(f')$.

■

Proof of Proposition 4.13.

Each leaf in (T, h_f) corresponds to a point in $PD(f)$. The x coordinate of each point is given by its height, which can be retrieved through h_f. Consider $v \in L_T$ and let γ_v, be the ordered set $\{v' \in V_T \big| v' \geq v\}$ i.e. the path from v towards r_T. The y coordinate of the points associated to v is the minimal height at which γ_v intersects another path γ_l, with l being a leaf with height less than v. ■

Proof of Lemma 4.19.

Let $U = f^{-1}((-\infty, t))$. Being f continuous, U is open in X. Therefore $x \notin U$ and $f(x) \geq t$. For every W_x open neighbor of x we know $W_x \bigcap U \neq \emptyset$ and $W_x \bigcap (X - U) \neq \emptyset$. By the continuity of f, for every $\epsilon > 0$ we have W_x open neighbor such that for every $y \in W_x$, $|\, f(y) - f(x) \,| < \epsilon$. In particular, we can always consider $y \in W_x \bigcap U$. Thus, for every $\varepsilon > 0$ we have $f(y) < t \leq f(x)$ and $|\, f(y) - f(x) \,| < \epsilon$. Thus $f(x) = t$.

■

Proof of Proposition 4.20.

The leaf v is associated to a path connected component $U_i^{t_0} \subset X_{t_0} = f^{-1}((-\infty, t_0])$, with $t_0 = h_f(v)$ and $t_1 = h_f(father(v))$. Consider now $i : f^{-1}((-\infty, t_0]) \hookrightarrow f^{-1}((-\infty, t_1))$ and let $U = \pi_0(i)(U_i^{t_0})$. Being f continuous U is open in X.

We can define $g : X \to \mathbb{R}$ such that $g(x) = f(x)$ if $x \notin U$, and $g(x) = t_1$ if $x \in U$.

We have $0 \leq g(x) - f(x) \leq \varepsilon$; in fact $t_0 \leq f(x) < t_1 = g(x)$ for every $x \in U$ and $f(x) = g(x)$ if $x \notin U$. Moreover $f^{-1}((-\infty, t]) = g^{-1}((-\infty, t])$ if $t \geq t_1$ and $g^{-1}((-\infty, t]) = f^{-1}((-\infty, t]) - U$ for $t < t_1$. And this implies the result about merge trees and tameness.

We only need to prove that g is continuous. We need to verify that, for any $x \in X$, for every $\epsilon > 0$ there is W_x open neighbor of x such that for every $y \in W_x$, we have $|\, g(x) - g(y) \,| < \epsilon$. If $x \in U$ this is trivially verified since U is open and so we can always find $W_x \subset U$, where g is constant. If x is an internal point of $X - U$, still the condition is verified for $f(x) = g(x)$ and f is continuous. We are left with the case $x \in \partial U$. For every W_x path connected open neighbor of x, we know that $W_x \bigcap U \neq \emptyset$. Moreover $x \notin g^{-1}((-\infty, t))$ since $g^{-1}((-\infty, t))$ is open in X and it would imply that x an internal point of U. So $x \in \partial g^{-1}((-\infty, t))$ and by Lemma 4.19, $f(x) = t_1$.

So if $|\, f(x) - f(y) \,| < \epsilon$, then $|\, g(x) - g(y) \,| < \epsilon$: if $y \notin U$ then $g(y) = f(y)$ otherwise $g(x) = g(y) = t_1$. Thus, g is continuous. ■

Proof of Corollary 4.21.

Starting from $T_0 = T_f$, each time we apply $(\mathcal{P}_\varepsilon)$ to obtain T_{i+1} from T_i, thanks to Proposition 4.20 we have continuous tame functions f_i and f_{i+1} such that $f(x) \leq f_i(x) \leq f_{i+1}(x) \leq f_i(x) + \varepsilon$. Each f_{i+1} is equal to f_i up to an open set U_i. Thus, if $V_i = \bigcup_{k=1}^{i-1} U_i$, then $f_i = f$ on $X - V_i$. Note that either $U_i \subset U_j$ - with $i < j$ - or they are disjoint. The case $U_i \subset U_j$ occurs when at the j-th step we delete a leaf v_j such that $father(v_j) > v_i$ - with v_i being the leaf deleted at the i-th step and with the father of v_j being taken in T_j, but being compared to v_i in T_f. If this is not the case, the sets U_i and U_j are disjoint.

We want to prove that $\max_X f_{i+1} - f_i = \max_X f_{i+1} - f$.

We know that on U_i we have $\max_{U_i} f_{i+1}(x) - f_i(x) = h_{f_i}(father(v_i)) - h_{f_i}(v_i)$ and on $X - U_i$ we have $f_{i+1} = f_i$. Thus $\max_X f_{i+1} - f_i = h_{f_i}(father(v_i)) - h_{f_i}(v_i)$.

Consider now $U_i - V_i$, which is always non-empty. For $x \in U_i - V_i$ we have $f_i(x) = f(x)$. The key observation is that by construction a local minimum of f over U_i is given by any point in the path connected component associated to v_i, i.e. there is $x \in U_i - V_i$ such that $f_i(x) = h_{f_i}(v_i) = h_f(v_i) = f(x)$ and $f_{i+1}(x) = h_{f_i}(father(v_i))$. Thus $\max_X f_{i+1} - f_i = f_{i+1}(x) - f_i(x) = f_{i+1}(x) - f(x) \leq \varepsilon$. The result on the merge trees follows by construction, since the deletions which lead to each T_i are induced via pruning. ■

Proof of Theorem 3.37.

To prove the theorem, we need some notation and some auxiliary results. To avoid dealing with unpleasant technicalities we work under the hypothesis that for any merge tree (T, h), h is an injective function. We call this assumption (G). It is not hard to see that for any fixed merge tree T, for any $\epsilon > 0$, there is a merge tree T' such that (G) holds and $d_E(T, T') \leq \epsilon$. It is enough to make arbitrarily small shrinkings to the edges.

Similarly for functions: given a continuous function we can always find an arbitrarily close function - in terms of $\|\cdot\|_\infty$ - such that the associated merge tree (T, h_f) satisfies (G).

First we define the *least common ancestor* (LCA) of a set of vertices in a merge tree.

Definition 4.25. *Given a merge tree (T, h_T) and set of vertices $A = \{a_1, \dots, a_n\} \subset V_T$, we define $LCA(a_1, \dots, a_n) = \min \bigcap_{i=1}^n \{v \in V_T \big| v \geq a_i\}$.*

Consider now f, g tame functions on the path connected topological space X such that $\sup_{x\in X} |\, f(x) - g(x)\,| \leq \varepsilon$. Let (F, h_f) and (G, h_g) be their associated merge trees. For $t \in \mathbb{R}$, we set $X_t^f = f^{-1}((-\infty, t])$. Since $|\, f(x) - g(x)\,| \leq \varepsilon$ we have $X_t^f \subset X_{t+\varepsilon}^g$ and of course $X_t^g \subset X_{t+\varepsilon}^f$.

We set $f_t^{t+\varepsilon} := \pi_0(X_t^f \hookrightarrow X_{t+\varepsilon}^f)$, $g_t^{t+\varepsilon} := \pi_0(X_t^g \hookrightarrow X_{t+\varepsilon}^g)$, $\alpha_t^{t+\varepsilon} := \pi_0(X_t^f \hookrightarrow X_{t+\varepsilon}^g)$, and $\beta_t^{t+\varepsilon} := \pi_0(X_t^g \hookrightarrow X_{t+\varepsilon}^f)$. We then call $F_t := \pi_0(X_t^f)$ and $G_t := \pi_0(X_t^g)$. With these pieces of notation we can write down the following commutative diagram:

$$\begin{array}{ccccccc} F_t & \xrightarrow{f_t^{t+\varepsilon}} & F_{t+\varepsilon} & \xrightarrow{\cdots} & F_{t'} & \xrightarrow{f_{t'}^{t'+\varepsilon}} & F_{t'+\varepsilon} \\ & \searrow^{\alpha} \quad \swarrow_{\beta} & & & & \searrow^{\alpha} \quad \swarrow_{\beta} & \\ G_t & \xrightarrow[g_t^{t+\varepsilon}]{} & G_{t+\varepsilon} & \xrightarrow{\cdots} & G_{t'} & \xrightarrow[g_{t'}^{t'+\varepsilon}]{} & G_{t'+\varepsilon} \end{array}$$

Note that the diagonal maps are $\alpha : F_t \to G_{t+\varepsilon}$ and $\beta : G_t \to F_{t+\varepsilon}$. Lastly, if $a_t' = f_t^{t'}(a_t)$, we say that $a_t < a_{t'}$.

If we collect together the path connected components $\{F_t\}_{t\in\mathbb{R}}$ and $\{G_t\}_{t\in\mathbb{R}}$ taking the disjoint unions $\mathbf{F} := \coprod_{t\in\mathbb{R}} F_t$ and $\mathbf{G} := \coprod_{t\in\mathbb{R}} G_t$ we can write down the maps $\alpha : \mathbf{F} \to \mathbf{G}$ and $\beta : \mathbf{G} \to \mathbf{F}$, so that given $a_t \in F_t$, $\alpha(a_t) := \alpha_t^{t+\varepsilon}(a_t)$. Given $a_{t'} \in \mathbf{F}$, we also set $h_f(a_{t'}) = t'$. The same for h_g.

We point out that the vertices of the merge trees F and G are contained in some F_t or G_t, respectively, and thus we have $V_F \hookrightarrow \mathbf{F}$ and $V_G \hookrightarrow \mathbf{G}$. We will often refer to $v \in V_F$ as $v \in \mathbf{F}$, and thus, for instance, take $\alpha(v)$, without explicitly mentioning the inclusion map. Note that the partial order we defined for $\mathbf{F}$ and $\mathbf{G}$ is compatible the the one of the vertices of the merge trees.

In a more technical language, $\mathbf{F}$ and $\mathbf{G}$ are the display posets of the two persistence sets $\pi_0(X^f)$ and $\pi_0(X^g)$ (Curry et al., 2022), but we want to avoid introducing such technical definitions. The work of Beketayev et al. (2014) shows that α and β give an ε-interleaving of merge trees (see Beketayev et al. (2014)), which, by Agarwal et al. (2018), is equivalent to the map α satisfying the following conditions:

(P1) $h_g(\alpha(a_t)) = h_f(a_t) + \varepsilon = t + \varepsilon$ for all $a_t \in \mathbf{F}$

(P2) if $\alpha(a_t) < \alpha(a_{t'})$ then there is $a_{t''}$ such that $a_t < a_{t''}$, $a_{t'} \leq a_{t''}$ and $t'' - t' < \varepsilon$.

(P3) if $b_{t'} \in \mathbf{G} - \alpha(\mathbf{F})$, then, given $b_t = \min\{b_{t''} > b_{t'} \big| b_{t''} \in \alpha(\mathbf{F})\}$, we have $t - t' \leq 2\varepsilon$.

A map which satisfies (P1)-(P3) is called ε-good (Agarwal et al., 2018).

To bridge between the continuous nature of $\mathbf{F}$ and $\mathbf{G}$ and the discrete (F, h_f) and (G, h_g), we define the following maps:

$$L_f : \mathbf{F} \to V_F$$

$L_f(a_t) = \max\{v \in V_F \,|\, v \leq a_t\}$ and similarly for L_g. Leveraging on these definitions we set $\phi := L_g \circ \alpha$ and $\psi := L_f \circ \beta$.

Finally we start building an edit path between (F, h_f) and (G, h_g). To do so we progressively add couples to an empty set M: couples of the form $(v, "D")$ mean that the

vertex $v \in V_F$ is deleted, while $("D", w)$ means that $w \in V_G$ is deleted, $(v, "G")$ means that the vertex $v \in V_F$ is ghosted, $("G", w)$ means that $w \in V_G$ is ghosted and (v, w) means that we shrink $(v, father(v))$ so that the weight of $(v, father(v))$ becomes equal to the one of$(w, father(w))$. The set M is in very close analogy with the *mappings* defined in Chapter 2.

By working simultaneously on F and G, we collect all the "edits" in $M \subset V_F \cup \{"D", "G"\} \times V_G \cup \{"D", "G"\}$ and then, in the last subsection of the proof, we use M on induce an edit path between F and G. We will call $\pi_F : M \to V_F \cup \{"D", "G"\}$ the projection on the first factor, and π_G the projection on the second.

4.A.1 Leaves of F

In this section we take care of the leaves of the merge tree F.

4.A.1.1 Selecting the Coupled Leaves

Consider the following set of leaves:

$$\mathcal{L}_F = \{v \in L_F \,|\, \nexists v' \in L_F \text{ such that } \alpha(v) < \alpha(v')\} \tag{4.2}$$

We name the condition:

(a) $\nexists v' \in L_F$ such that $\alpha(v) < \alpha(v')$

so that we can more easily refer to it during the proof. Observe that we never have $\alpha(v) = \alpha(v')$ thanks to condition (G).

The set $\mathcal{L}_F$ is the set of leaves which will coupled by M: we add to M all the couples of the form $(v, \phi(v))$ with $v \in \mathcal{L}_F$ and add $(v, "D")$ for all $v \in L_F - \mathcal{L}_F$.

Lemma 4.26. *Given $v, v' \in \mathcal{L}_F$, then $\phi(v) \geq \phi(v')$ if and only if $v = v'$. Moreover, for every $v' \in L_F$ such that (a) does not hold, there is $v \in \mathcal{L}_F$ such that $\alpha(v) < \alpha(v')$.*

Proof. The first part of the proof reduces to observing that $\phi(v) \leq \phi(v')$ if and only if $\alpha(v) \leq \alpha(v')$.

Now consider $v' \in L_F$ such that (a) does not hold. We know there is v_0 such that $\alpha(v_0) < \alpha(v')$. If $v_0 \in \mathcal{L}_F$ we are done, otherwise there is v_1 such that $\alpha(v_1) < \alpha(v_0) < \alpha(v')$. Note that $f(v_1) < f(v_0)$. Thus we can carry on this procedure until we find $v_i \in \mathcal{L}_F$. Note that $\arg\min_{a \in V_F} f(a) \in \mathcal{L}_F$, thus, in a finite number of step we are done. □

4.A.1.2 Height Bounds on Couples

Now we want to prove the following proposition which will be used to give an upper bound for the cost of the edits induced by the couples $(v, \phi(v))$ added to M.

Lemma 4.27. *Given $v \in \mathcal{L}_F$, then $|\, h_f(v) - h_g(\phi(v)) \,| \leq \varepsilon$.*

Proof. Suppose the thesis does not hold. Since $h_g(\phi(v)) \leq h_f(v) + \varepsilon$, contradicting the thesis means that we have $v \in \mathcal{L}_F$ such that:

(b) $h_f(v) - h_g(\phi(v)) > \varepsilon$.

Let $w = \phi(v)$. If (b) holds, then $h_g(father(w)) - h_g(w) > h_g(\alpha(v)) - h_g(w) > 2\varepsilon$. Let $v' = \psi(w) \leq \beta(w)$. Note that $h_f(v') < h_f(v)$. We have $\phi(v') \leq \alpha(v') \leq \alpha(\beta(w))$. But since $h_g(father(w)) - h_g(w) > 2\varepsilon$, we also have $\alpha(v') \leq \alpha(v)$ with $v' \neq v$ which is absurd by Lemma 4.26. □

4.A.1.3 Cost Bound on Deletions

In this step we prove a cost bound for some the vertices of F which are going to be deleted. We add to M the couples $(x, "D")$ for every $x \notin \{v' > v \,|\, v \in \mathcal{L}_F\}$.

Lemma 4.28. *Given $x \notin \{v' > v \,|\, v \in \mathcal{L}_F\}$, then $w_F((x, father(x))) \leq 2\varepsilon$.*

Proof. We simply observe that, if $x \notin v \notin \{v' > v \,|\, v \in \mathcal{L}_F\}$ then there is $v \in \mathcal{L}_F$ such that $\alpha(v) < \alpha(x)$. By property (P2) of α, since $h_f(v) < h_f(x)$, we have that $(x, "D")$ has cost at most 2ε. □

4.A.2 Leaves and deletions of G

Similarly we add to M the couples $("D", y)$ for every $y \notin \{w' > w \,|\, w \in \pi_G(M) \cap V_G\}$.

Lemma 4.29. *Given $y \in V_G$ such that $y \notin \{w' > w \,|\, w \in \pi_G(M) \cap V_G\}$, then $w_G((y, father(y))) \leq 2\varepsilon$.*

Proof. Consider $\beta(y)$. Let $v \leq \beta(y)$ leaf. We have $\alpha(\beta(y)) \geq \alpha(v)$ and $\alpha(\beta(y)) \geq y$. If $L_g(\alpha(\beta(y))) \neq y$ we are done since $h_g(\alpha(\beta(y))) = h_g(y) + 2\varepsilon$. Suppose instead $L_g(\alpha(\beta(y))) = y$. This implies $\alpha(v) \leq y$. By construction $v \notin \mathcal{L}_F$, but then there is $v' \in \mathcal{L}_F$ such that $\alpha(v') < \alpha(v) \leq y$. Absurd. □

4.A.3 Internal Vertices

Now we want to take into account the internal vertices of F and G.

Thanks to Lemma 4.28 and Lemma 4.29 we can delete all $x \notin \{v' > v \,|\, v \in \mathcal{L}_F\}$ and $y \in V_G$ such that $y \notin \{w' > w \,|\, w \in \pi_G(M) \cap V_G\}$, each with cost at most 2ε. Note that these deletions do not change the heights of any non deleted vertex.

Call F_1 and G_1 the two trees obtained after such deletions and after the ghosting of all the order 2 vertices arising - and consequently adding $(v, "G")$ or $("G", w)$ to M for all the ghosted vertices respectively in F or G. If we consider $\alpha_{|\mathbf{F}_1}$ then by construction $\alpha_{|\mathbf{F}_1} : \mathbf{F}_1 \to \mathbf{G}_1$. Similarly $\beta_{|\mathbf{G}_1} : \mathbf{G}_1 \to \mathbf{F}_1$. Moreover $\alpha(\mathbf{F}) = \alpha_{|\mathbf{F}_1}(\mathbf{F}_1)$. Thus $\alpha_{|\mathbf{F}_1}$ is still ε-good. In what follows, with an abuse of notation, we avoid explicitly writing the restriction of the maps α and β, implying that these are always considered as defined on the "pruned" trees F_1 and G_1.

4.A.3.1 Results on Internal vertices

We prove the following results.

Lemma 4.30. *Let $x \in V_{F_1} - L_{F_1}$ such that $x = LCA(A)$ with $A = \{v \in V_{F_1} \,|\, v < x\} \bigcap L_{F_1}$. Let $y = LCA(\phi(A))$. Then $|\, h_f(x) - h_g(y) \,| \leq \varepsilon$.*

Proof. For every $a \in A$ we know $\phi(a) < \phi(x)$ and thus $y \leq \phi(x)$. Thus $h_g(y) \leq h_f(\phi(x)) + \varepsilon$. Suppose then $h_g(y) < h_f(x) - \varepsilon$. However, $\beta(y) \geq x$ for $\beta(y) \geq \beta(\phi(a)) \geq a$. Which is absurd for $h_f(\beta(y)) = h_g(y) + \varepsilon < h_f(x)$. □

Lemma 4.31. *Let $x \in V_{F_1} - L_{F_1}$ such that $x = LCA(A)$ with $A = \{v \in V_{F_1} \,|\, v < x\} \bigcap L_{F_1}$. Let $y = LCA(\phi(A)) = LCA(B)$ with $B = \{w \in V_{G_1} \,|\, w < y\} \bigcap L_{G_1}$.*

Then for every $w = \phi(v) \in B - \phi(A)$, let $x' = LCA(A \cup \{v\})$. Then we have $h_f(x') - h_f(x) \leq 2\varepsilon$.

Proof. Since $v \not\leq x$ but $\phi(v) < \phi(x)$, we know $\min\{h_f(x') - h_f(x), h_f(x') - h_f(x)\} \leq 2\varepsilon$. Suppose $h_f(x') - h_f(x) > 2\varepsilon$. We know $\beta(\alpha(x)) > v$ for $\phi(x) > \phi(v)$. But then $\beta(\alpha(x)) \geq x'$ which is absurd since $h_f(\beta(\alpha(x))) = h_f(x) + 2\varepsilon < h_f(x')$. □

Lemma 4.32. *Consider* $e = (x, father(x)) \in E_{F_1}$, $x = LCA(A)$ *with* $A = \{v \in V_{F_1} \,|\, v < x\} \bigcap L_{F_1}$. *Let* $y = LCA(\phi(A)) = LCA(B)$, *with* $B = \{w \in V_{G_1} \,|\, w < y\} \bigcap L_{G_1}$. *Let* $e' = (y, father(y)) \in E_{G_1}$. *Then* $|\, w_{F_1}(e) - w_{G_1}(e') \,| \leq 2\varepsilon$.

Proof. We know by Lemma 4.30 that $|\, h_f(x) - h_g(y) \,| \leq \varepsilon$. Thus we can focus on $x' = father(x)$ and $y' = father(y)$. Let $A' = \{v \in V_{F_1} \,|\, v < x'\} \bigcap L_{F_1}$, $B' = \{w \in V_{G_1} \,|\, w < y'\} \bigcap L_{G_1}$, $w = LCA(\phi(A'))$ and $v = LCA(\psi(B'))$. By Lemma 4.30, again $|\, h_f(x') - h_g(w) \,| \leq \varepsilon$ and $|\, h_f(v) - h_g(y') \,| \leq \varepsilon$.

Since $A \subset A'$, then $B \subset \phi(A')$ and similarly $A \subset \psi(B')$. Which entails $x' \leq v$ and $y' \leq w$. Since $x'' = LCA(\psi(\{w' \in V_{G_1} \,|\, w' < w\})) \geq v$, by Lemma 4.31 we have $h_f(v) - h_f(x') \leq 2\varepsilon$ and, similarly, $h_g(w) - h_g(y') \leq 2\varepsilon$. Putting together $|\, h_f(x') - h_g(w) \,| \leq \varepsilon$, $y' < w$ and $h_g(w) - h_g(y') \leq 2\varepsilon$ and the analogous inequalities for y' we obtain $|\, h_f(x') - h_g(y') \,| \leq \varepsilon$.

Thus $|\, h_f(x') - h_f(x) - (h_g(y') - h_g(y)) \,| \leq 2\varepsilon$.

□

4.A.3.2 Deleting Internal Vertices

Now we proceed as follows: for every $x \in V_{F_1}$ let $A = \{v \in V_{F_1} \,|\, v < x\} \bigcap L_{F_1}$, $y = LCA(\phi(A)) = LCA(B)$, with $B = \{w \in V_{G_1} \,|\, w < y\} \bigcap L_{G_1}$. If $B \neq \phi(A)$, then add $(x, "D")$ to M and delete x. By Lemma 4.31 the cost of deleting x is less then 2ε. We do so for all $x \in V_{T_1}$. Then we follow an analogous process for $y \in V_{G_1}$: let $y = LCA(B)$ with $B = \{w \in V_{G_1} \,|\, w < y\} \bigcap L_{G_1}$, $x = LCA(\psi(B)) = LCA(A)$ with $A = \{v \in V_{F_1} \,|\, v < x\} \bigcap L_{F_1}$. If $A \neq \psi(B)$, then add $("D", y)$ to M and delete y. By Lemma 4.31 the cost of deleting y is less then 2ε.

4.A.3.3 Coupling the Internal Vertices

After the deletions in Section 4.A.3.2 we obtain two merge trees F_2 and G_2, with the same leaves as F_1 and G_1 but with the property that for each $x \in V_{F_2}$ we have a bijection between the leaves $A = \{v \in V_{F_1} \,|\, v < x\} \bigcap L_{F_1}$ and the leaves in $sub_{G_2}(y)$ with $y = LCA(\phi(A))$. Thus for any edge $(x, x') \in E_{F_2}$ we have a unique edge $(y, y') \in E_{G_2}$ such that the leaves of $sub_{F_2}(x)$ and $sub_{G_2}(y)$ are in bijection and the same for x' and y'. Thus we can couple x and y, add (x, y) to M and make the shrinking to pair their weights. Since deleting a vertex doesn't affect the weight of the other edges, then we can still apply Lemma 4.32 which guarantees that the shrinkings cost less then 2ε.

4.A.4 Inducing the Edit Path

To conclude the proof we sum up everything and induce and order edits operations according to the couples contained in M, so that the costs of the edits matches the ones described along the previous subsections of the proof.

First we apply all the deletions on F described in Section 4.A.1.3, with the cost of every edit being at most 2ε. Then we ghost all order 2 vertices. By construction we obtain, from F, the tree F_1. At this point we delete internal vertices of F_1 according to the procedure described in Section 4.A.3.2, obtaining F_2. Then we shrink all the edges of F_2, according to Section 4.A.3.3, obtaining G_2. Then we insert all the edges needed to obtain G_1 from G_2, which are associated to the couples $("D", y)$ mentioned in Section 4.A.3.2. Then we go on with the splittings induced by $("G", w) \in M$, which are needed to subsequently insert the edges which take us from G_1 to G, as explained in Section 4.A.2. In the respective sections it is shown that all the edits we employed have cost less then 2ε.

This concludes the proof.

■

4.B Combining Metrics

To aggregate curvature and radius, we make use of the following proposition.

Proposition 4.33. *Given (X, d_0) and (X, d_1) metric spaces, then $d_{a,b,p} := (a \cdot d_0^p + b \cdot d_1^p)^{1/p}$, with $a, b \in \mathbb{R}_{>0}$ and $p \geq 1$, is a metric on X.*

Proof. $d_{a,b,p}(x, y) = ||(a^{1/p} \cdot d_0(x, y), b^{1/p} \cdot d_1(x, y))||_p$.

Since, given $k > 0$, $k \cdot d_i$ is a metric if and only if d_i is a metric, we can rescale d_0 and d_1 and take $a = b = 1$. We refer to $d_{1,1,p}$ as d_p.

So:

- $d_p(x, y) = 0$ iff $d_0(x, y) = 0 = d_1(x, y)$ and this happens if and only if $x = y$.
- symmetry is obvious
- we use $||h+q||_p \leq ||h||_p + ||q||_p$ with $h = (d_0(x, z), d_1(x, z))$ and $q = (d_0(z, y), d_1(z, y))$.

  Since $d_i(x, y) \leq d_i(x, z) + d_i(z, y)$ we get:

  $||(d_0(x, y), d_1(x, y))||_p \leq ||(d_0(x, z)+d_0(z, y), d_1(x, z)+d_1(z, y))||_p = ||(d_0(x, z), d_1(x, z))+(d_0(z, y), d_1(z, y))||_p \leq ||(d_0(x, z), d_1(x, z))||_p + ||(d_0(z, y), d_1(z, y))||_p$.

  Therefore:

  $d_p(x, y) \leq d_p(x, z) + d_p(z, y)$.

□

5. Imaging-based representation and stratification of intra-tumor Heterogeneity via tree-edit distance

Note to the Reader

As already mentioned in Chapter 1, this chapter of the thesis contains a work which has been developed in close collaboration with other authors (listed in Chapter 1). The spirit of the work is different from the other chapters of the thesis: in the following, a cutting edge medical research problem is tackled by means of an unsupervised analysis, employing a merge tree representation of the patients. The theoretical developments which are carried out in the chapter are motivated by the expertise on the clinical problem of the other collaborators involved in the project, which lead to a novel metric for merge trees, adapted from the one defined in Chapter 3. Even though such metric is of general interest and it may foster similar approaches to be carried out in other TDA pipelines, its rationale is strictly bound to the problem we consider, enhancing the interpretability of the proposed pipeline and thus adding reliability to the unsupervised analysis carried out. For this reason this chapter contains an in-depth presentation of the clinical problem, with its motivations and challenges and it is laid out as a piece of work directed to scientists potentially working in the same field. The developments in terms of merge trees, which make the chapter a perfect example of a non trivial application of the ideas presented in the previous chapters of the thesis, are mainly contained in Section 5.4.2, Section 5.4.3 and from Section 5.B to Section 5.G.

abstract

Personalized medicine is the future of medical practice. In oncology, tumor heterogeneity assessment represents a pivotal step for effective treatment planning and prognosis prediction. Despite new procedures for DNA sequencing and analysis, non-invasive methods for tumor characterization are needed to impact on daily routine. On purpose, imaging texture analysis is rapidly scaling, holding the promise to surrogate histopathological assessment of tumor lesions. In this work, we propose a tree-based representation strategy for describing intra-tumor heterogeneity of patients affected by metastatic cancer. We leverage radiomics information extracted from PET/CT imaging and we provide an exhaustive and easily readable summary of the disease spreading. We exploit this novel patient representation to perform cancer subtyping according to hierarchical clustering technique. To this purpose, a new heterogeneity-based distance between trees is defined and applied to a case study of Prostate Cancer (PCa). Clusters interpretation is explored in terms of concordance with severity status, tumor burden and biological characteristics. Results are promising, as the proposed method outperforms current literature approaches. Ultimately, the proposed methods draws a general analysis framework that would allow to extract knowledge from daily acquired imaging data of patients and provide insights for effective treatment planning.

5.1 Introduction

The current paradigm shifting of modern medical practice sinks its root in providing personalized treatments and improving therapy outcomes. Huge strides have been made in oncology with the uprising of quantitative imaging techniques and new procedures for DNA sequencing and analysis that allow an extensive characterization of cancer subtypes. In particular, recent research has investigated the main causes of cancer progression, resistance to therapy and late recurrence. Among these, tumor heterogeneity has gained special interest and has been recognized to play a crucial role Fisher et al. (2013): defined as complex genetic, epigenetic and protein modifications that can be found within the same patient's disease, tumor heterogeneity behaves as a driver for phenotypic selection. According to Stanta and Bonin and y Cajal et al. Stanta and Bonin (2018); y Cajal et al. (2020), different types of tumor manifestation may exist as a response to microenvironmental and external changing, differing between primary tumor and proximal and distant metastases. As a result, certain tumor phenotypes properly respond to therapies and others become resistant clones, leading to treatments ineffectiveness and cancer progression. Pertinently, detecting at baseline which phenotype will respond and which will not - known as *prognostic cancer subtyping* - represents a pivotal step in personalized medicine.

Although recent findings about heterogeneity suggest that therapy would be improved if guided by the analysis of both primary and metastatic tissues - such as lymph nodes Cummings et al. (2014) -, clinical practice usually relies on primary tumor biomarkers for prognosis definition and treatment planning. Thus, baseline assessment emerges altered by the understimation of intra-tumor heterogeneity which behaves as confounding factor in pre-treatment clinical-pathological prognosis, leading to poor survival rates Esparza-López et al. (2017). This misalignment between research evidence and clinical practice seems mostly due to the lack of non-invasive methods for heterogeneity quantification. Accordingly, current prognostic cancer subtyping cannot be translated into daily clinical practice and therapeutic guidelines.

Over the last two decades, the texture analysis of digital images - such as Magnetic Resonance Imaging (MRI) and Positron Emission Tomography / Computer Tomography (PET/CT) - has arisen as a valuable non-invasive proxy for biological assessment of tumors, eventually growing in a discipline of its own, namely radiomics Mayerhoefer et al. (2020). Specifically, macroscopic appearance of tumors has been acknowledged as a valid tool for guiding clinical decisions in the definition of disease severity and treatment planning. Broadly speaking, image texture analysis consists of extracting descriptors of spatial variation of voxel grey-scale and intensity within the image Volumes Of Interest (VOI), i.e., the tumor lesions. Under the name of radiomic features, such textural descriptors form a high dimensional vector embedding of the VOI and may provide a non-invasive assessment of tumor appearance from routinely acquired imaging studies Gillies et al. (2016). These features are indeed supposed to supply additional predictive and prognostic information, ready to use to postulate the underlying biological mechanisms of disease progression in clinical routine Chicklore et al. (2013). Accordingly, the dissimilarity in the appearance of different lesions, therefore in their texture descriptions, can be regarded as *radiological* heterogeneity, which can be easily quantified and leveraged in the daily practice.

Despite the increasing interest in tumor heterogeneity, imaging-guided therapy currently employs biomarkers for tumor burden that stem from the characterization of the primary tumor, the bigger lesion (often coinciding with the hottest lesion) or the mean lesions' profile. Only recently few radiomics-based approaches have been suggested - for prognosis, treatment outcome and survival prediction - which consider the multi-lesion disease in a comprehensive way. In particular, several researchers Eertink et al. (2022); Ceriani et al. (2020); Burggraaff et al. (2020) proposed different segmentation strategies for feature extraction from patient level VOIs, while Cottereau et al. Cottereau et al. (2020)

evaluated the predictive power of several indicators reflecting the spatial distributions of malignant *foci* spread throughout the whole body. A number of *dissemination* features have been explored and reviewed: the number of lesions, the euclidean distance between crucial or predominant bulks, the largest value of the pairwise sum of the physical distances between lesions, etc.. Stemming from a similar idea, Cavinato et al. Cavinato et al. (2020) proposed a similarity metric for comparing lesions' texture descriptions, defining intra-patient heterogeneity as the normalized average of pairwise distances between lesions' radiomic vectors. This similarity over patient's lesions description has thus been suggested as functional, rather than spatial, dispersion index for tumor burden and disease severeness, with promising results in Hodgkin Lymphoma Sollini et al. (2020) and Prostate Cancer Sollini et al. (2021). Preliminary results represent an insightful starting point in the debate around the proper definition of heterogeneous disease.

In this work, motivated by the need to embed tumor heterogeneity quantification into patients' clinical pathway planning, we propose a novel way for modeling intra-patient tumor heterogeneity in a non-invasive way, leveraging the radiomic framework. Specifically, we perform dimensionality reduction on radiomic vectors, as to remove redundancy and collinearity while preserving the multi-view nature of the texture description. Reduced vectors of peer lesions within the same tumor are then compared via pairwise distances. Representing the patient via the pairwise distance matrix of its lesions makes it laborious to compare patients with different numbers of lesions. For this reason, upon lesions' distance matrix, we build a dendrogram, which hierarchically aggregates peer lesions in a unique combinatorial object. This object-oriented representation summarizes the multi-lesion disease and highlights the relationship among lesions, basing on similarities in their imaging characteristics. In fact, lesions are not independent as they are statistically and semantically connected to the patient they belong to. Accordingly, such relationship shapes and influences the structure of the dendrogram associated to the patient. We then exploit the tree-based patient representation to cluster cancer subtypes according to their imaging heterogeneity. To do so, we define a new *ad hoc* distance between trees. To validate the method, we test the whole pipeline on a dataset of patients affected by metastatic Prostate Cancer (PCa), evaluating the descriptive and stratification performance in terms of disease severeness and outcomes. We associate imaging subtypes to clinically relevant information within and beyond clinical surrogates, with the goal of eventually supporting therapy decisions wherein actions regarding active surveillance, mild treatment or intensified therapy are devised and taken Fisher et al. (2013).

5.2 Results

5.2.1 Case study: Prostate Cancer

Within the personalized medicine framework, Prostate cancer (PCa) is a striking example of the need to exploit an insightful prognostic cancer subtyping for treatment planning. In fact, even if recent studies have reported a decreasing pattern of overall PCa incidence, Culp et al. Siegel et al. (2020) and Siegel et al. Culp et al. (2020) recorded an alarming mortality rate due to an increasing trend of distant stage metastatic disease, even in developed countries. Moreover, the role of imaging-guided therapy for PCa has revealed to be very promising and is consistently spreading in daily practice Giovacchini et al. (2010). Despite these facts, clinical guidelines still relies on primary tumor biomarkers. Besides, very limited methods have been proposed for reliably assessing and quantifying multi-lesion heterogeneity information within the same patient from an imaging point of view. This misalignment between research evidence and clinical routine results in poor disease free survival rates, mostly due to the lack of non-invasive methods for heterogeneity quantification.

The case study analyzed in this work is composed by a set of $N = 333$ lesions belonging to fifty-five patients of Azienda Ospedaliero-Universitaria Pisana with multi-site, multi-lesion, recurrent Prostate Cancer confirmed with a positive PET/CT study. The study was performed in accordance with the Declaration of Helsinki and approved by the local ethics committee. The signature of a specific informed consent and the legal requirements of clinical trials were waived given the observational retrospective study design. During the observational trial, patients showed evidence of biochemical recurrence after first-line treatments, exhibiting metastatic disease. Every patient manifested a different number of tumor lesions n_i, according to the spreading burden of the metastatic tumor. Information about age, sex, lesion site, total tumor volume, Gleason Score Epstein et al. (2016), Prostate Specific Antigen Balk et al. (2003) and therapy treatment was collected per each patient. Personal information and qualitative tumor data are displayed in Table 5.A.1 and Table 5.A.2. Additionally, from PET/CT, volumes of interest, i.e. lesions, were segmented by experienced nuclear medicine physicians and texture features were extracted over VOIs according to the radiomic framework, resulting in forty radiomic features ($p = 40$). Both the segmentation of lesions and radiomic features extraction were performed using LifeX software Nioche et al. (2018), according to the formulas detailed in the software documentation (www.lifexsoft.org).

We fed Prostate Cancer imaging data into the pipeline described in Figure 5.2.1, obtaining a tree-based representation T for each of the patients. The pruned edit distance d_P^{μ}, as defined in the Methods, was implemented and leveraged to compute the patient-to-patient distance matrix. Clustering of patients was thus completed according to hierarchical clustering algorithm with the proposed *ad hoc* distance and *ward* linkage. The number of clusters was selected in the range $[2, 5]$, as a trade off between performance and interpretability, according to silhouette coefficient maximization. The resulting classes could then be intended as groups of patients with similar representations in terms of heterogeneous disease, to be characterized according to exogenous clinical variables and risk assessment.

5.2.2 Clusters characterization

As to profile the clustering, we describe how the stratification procedure captures the differentiation of tumor heterogeneities and provide a clinical/biological interpretation.

Upon pipeline implementation, hierarchical clustering identified three groups: groups 0, 1 and 2 hosted 39, 10 and 6 patients respectively. In Figure 5.2.3 the curves of the heights of the trees' vertices over the three groups can be appreciated: branches present different average heights according to the group their dendrograms belong (see Fig. Figure 5.2.3). Groups are shown to entail different heterogeneity extent, following an ANOVA functional approach Pini and Vantini (2017) Horváth and Kokoszka (2012).

Beside the group-wise characterization of tree conformation as manifestation of tumor heterogeneity, clinical variables were used as exogenous factors to characterize and interpret the groups. We used appropriate tests according to the variable type, normality of data and sample size. Normality was tested according to the Shapiro test. We thus employed Mann-Whitney non-parametric tests for comparing distributions of continuous (non-normal) variables; parametric t-tests for testing the difference of means in continuous (normal) variables; Levene non-parametric tests for comparing variances of continuous (non-normal) variable; Bartlett parametric tests for continuous (normal) variable ratio of variances; $Chi - squared$ tests for independence of categorical variable. P-values are indicated respectively as $p_{m/d}$ for tests on means/distributions, p_{var} for tests on variance and p_{ind} for tests on independence. Pairwise one-sided comparison between groups rather than multivariate analysis was investigated as to provide a group-wise characterization. As to avoid potential Type II errors due to small sample size, value of $\alpha = 0.1$ was considered

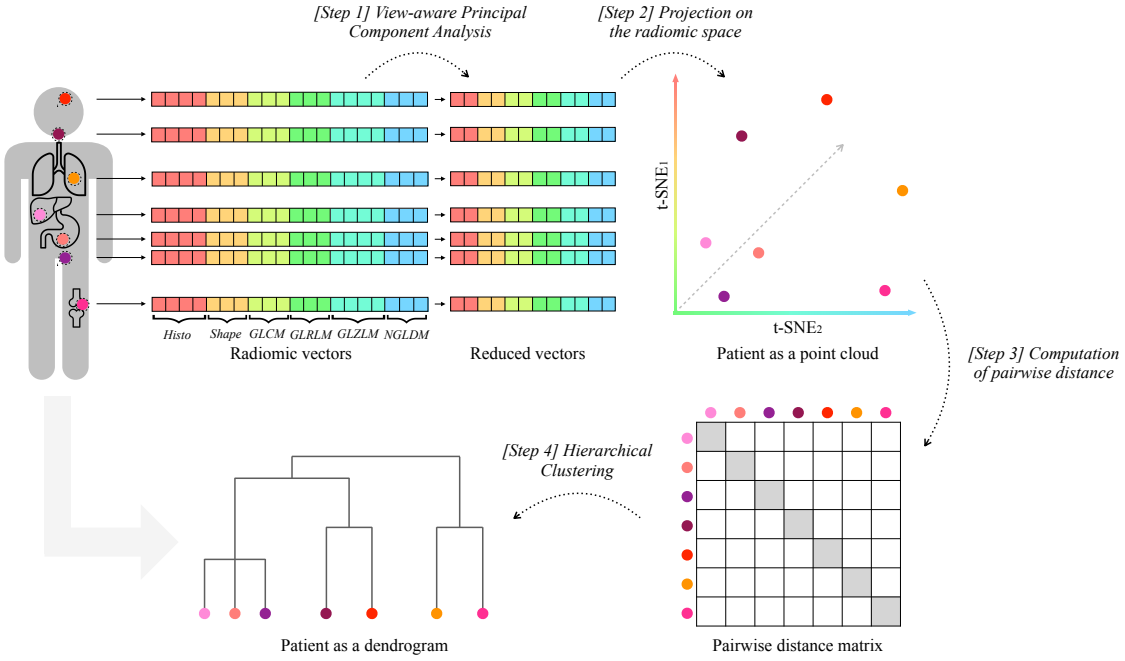


Figure 5.2.1: Patient representation pipeline: lesions' radiomic vectors of each patient are dimensionally reduced according to view-aware Principal Component Analysis. *[Step 1]* Features are grouped according to the six semantic group, or *view*, they are semantically divided into. As to preserve a balanced importance between views, two principal components are kept from the scores of each PCA, leading to different percentages of explained variability. A total of twelve principal components results from the process, which include six orthogonal pairs of linear combinations of original features. *[Step 2]* Accordingly, patients are represented as finite sets of n_i points in $\mathbb{R}^{12}$, that is the reduced radiomic space according to view-aware strategy implementation. In the example, $n_i = 7$. *[Step 3]* Pairwise (Euclidean) distance is compute among patients' lesions and *[Step 4]* hierarchical clustering with *average* linkage is applied to distance matrices, resulting in a dendrogram T representing each patient.

for significance.

We evaluated the differences between the obtained groups in terms of number of oligo/multi-metastatic patients (as classified with two different clinical cut-offs of 3 and 5 lesions), number of patients with bone disease, total tumor volume and number of tumor lesions. Also, the implementation of combined therapy (such as joint radiotherapy and chemotherapy with respect to only chemotherapy) and response to therapy were evaluated in patients of different groups. Additionally, among clinical prognostic tools, tumor aggressiveness is usually assessed with Gleason Grading System (or Gleason Score) Epstein et al. (2016). A Gleason Score (GS) is given to Prostate Cancer based upon its microscopic appearance with respect to cell differentiation. Pathological scores represent the sum of the primary and secondary patterns (each ranging from 1 - well differentiated, like normal cells - and 5 - poorly differentiated, i.e., abnormal cells) and range from 2 to 10. Higher numbers indicate more aggressive disease, worse prognosis and higher mortality Epstein et al. (2016). In particular, patients with Gleason Score exceeding the value of 7 experience extraprostatic extension and biochemical recurrence more frequently than others Draisma et al. (2006). Accordingly, clusters were also analyzed in terms of mean Gleason Score and number of patients exceeding GS of 7.

Besides, Prostate Specific Antigen (PSA) has been proposed for screening, assessment of future risk of prostate cancer development, detection of recurrent disease after local therapy and treatment planning of advanced disease. Often employed as criteria in combination of stage and GS, its role in early stage assessments is still debated due to instability of measurements and the presence of confounding factors. However, PSA is still considered a valid tool for prognosis and treatments in advanced stages of metastatic prostate cancer Pezaro et al. (2014). Moreover, PSA values after cytotoxic regimens has been shown to predict survival. Particularly, the decrease in PSA levels is associated to therapy response in soft tissue lesions and thus could be intended as a proxy of therapy outcome Smith et al. (1998). Accordingly, we recorded PSA levels before the therapy (PSA0), right after the first line of therapy (PSA1) and at the end of the follow up (PSA2). Delta-PSA levels were computed between PSA1-PSA0 and PSA2-PSA0 as proxies of cancer evolution. In the following, they will be referred as PSA, $\Delta PSA_{1,0}$ and $\Delta PSA_{2,0}$.

Table 5.2.1 and Figure 5.2.2 elucidate the results. The profile of the blue and green groups are very similar for what PSA ($p_{m/d} = 0.3787$, $p_{var} = 0.4714$) and $\Delta PSA_{1,0}$ ($p_{m/d} = 0.3477$, $p_{var} = 0.4533$) are concerned, with a very limited range of values concentrated around zero. Different trends are exhibited by the blue and green curves of the $\Delta PSA_{2,0}$ ($p_{m/d} = 0.0591$), where the difference could support the hypothesis of different cancer evolution starting from similar baseline assessments. Yet, they present similar variance ($p_{var} = 0.2159$). The orange group, on the other hand, presents wider ranges and higher intra-group heterogeneity. In particular, orange PSA is significantly higher than the blue group with a much more spread distribution ($p_{m/d} = 0.0116$; $p_{var} = 0.0013$) yet no statistical difference with the green groups is confirmed ($p_{m/d} = 0.3089$; $p_{var} = 0.1845$); orange $\Delta PSA_{1,0}$ is significantly lower than the blue group ($p_{m/d} = 0.0019$) but not than the green one ($p_{m/d} = 0.1810$), however its distribution appears more spread and inhomogeneous, covering both the negative and the positive axis, in both cases ($p_{var} = 0.0003$; $p_{var} = 0.0995$). The $\Delta PSA_{2,0}$ of the orange group does not vary from the one of the blue group ($p_{m/d} = 0.3689$). However, it shows a higher variance than the other, suggesting a heterogeneous long-term tumor prognosis ($p_{var} = 0.0066$). Also, the orange group and the green group do not differ significantly in their average ($p_{m/d} = 0.1855$) but their variances reveal a mild divergence in terms of distribution kurtosis ($p_{var} = 0.1085$).

Regarding the number of lesions, the orange group displays a higher number of metastases than the blue one ($p_{m/d} = 0.0081$). The green group exhibits a behavior very similar to the blue group ($p_{m/d} = 0.4162$), diverging from the orange group with respect to which

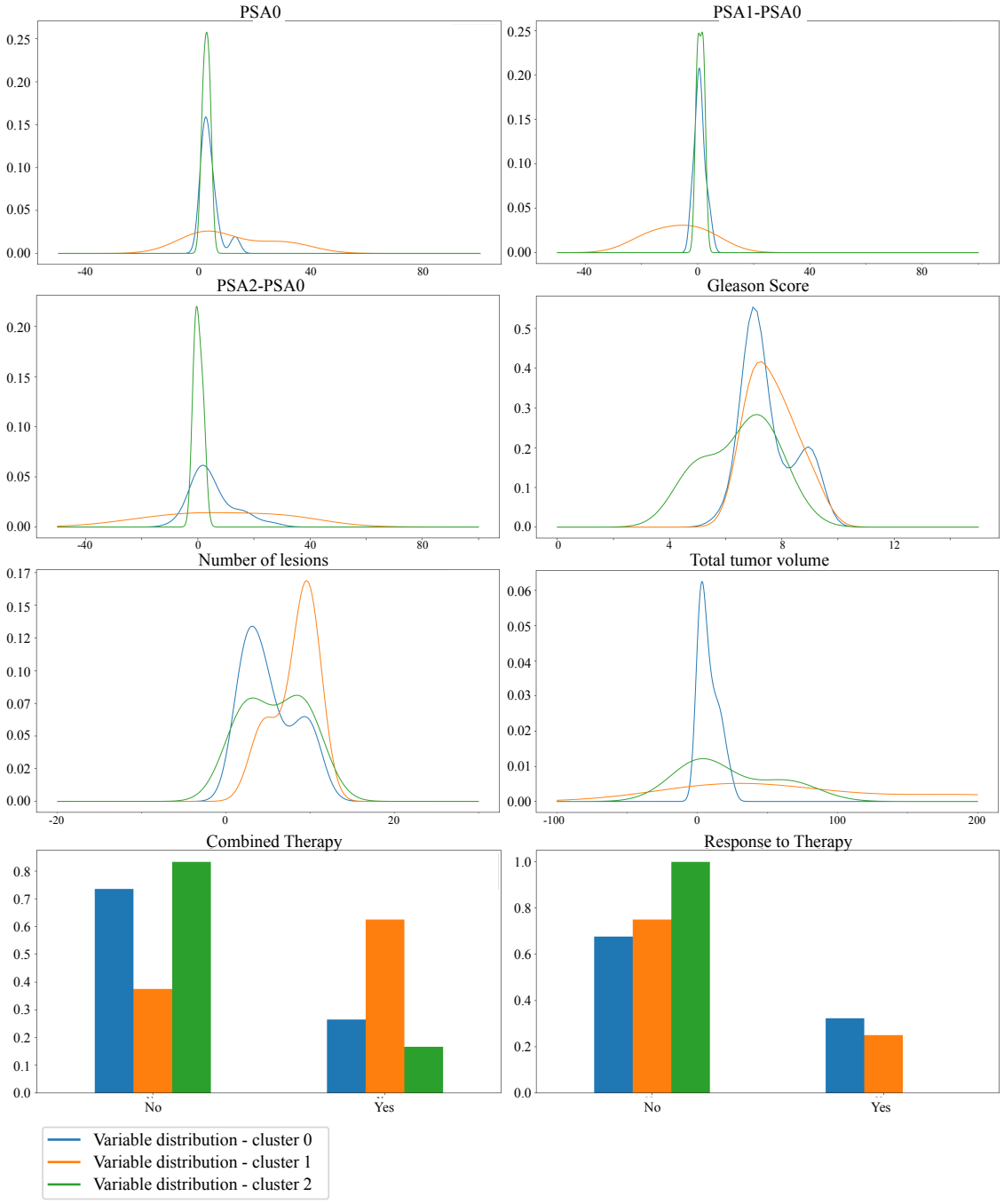


Figure 5.2.2: Results of clustering characterization: first three rows draw the distributions of the numerical clinical variables in the three groups, namely the PSA values, the $\Delta PSA_{1,0}$, the $\Delta PSA_{2,0}$, the number of lesions, Gleason Scores and the total tumor volume; last row shows the proportions of the categorical clinical variables in the three groups, that are the combination of therapy and the response to treatment. For the proportion of skeleton disease and of the oligo/multi-metastatic status as devised by the two clinical cut-offs (3 and 5 lesions) see Section 5.H.

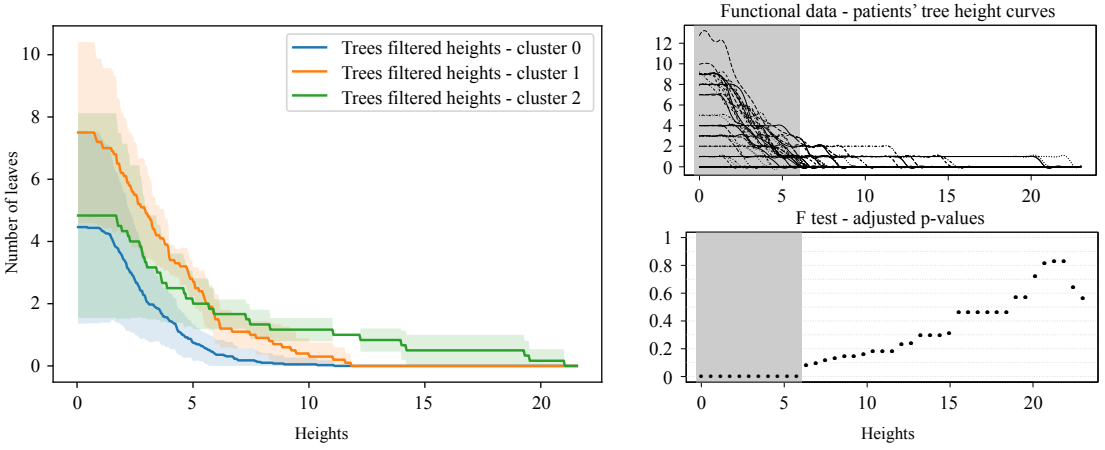
it presents fewer lesions ($p_{m/d} = 0.0722$). Moreover, total volume of the tumor is related to the number of lesions. In fact, the blue group displays a reduced spreading of the tumor over the body with respect to the orange group ($p_{m/d} = 0.0002$) but not to the green group ($p_{m/d} = 0.4917$). The orange and the green groups also exhibit a statistical difference in terms of tumor volume ($p_{m/d} = 0.0306$). Of note, despite the number of metastases in the blue and green groups are very similar, it should be noticed that their tumor spreading appears shifted in the figure, entailing unrelated tumor burden information. Similarly, the orange group, while presenting a greater number of lesions, shows an extension of the tumor visually analogous to the green group. Such discrepancy is imputable to the difference of variances the distributions display.

From these consideration, it appears clear how the green group shows phenotypic similarities and dissimilarities with respect to both blue group and orange group, presenting an in-between behavior. However, the detach of green patients from the rest of the population is mostly driven by the different distribution of GS levels. In fact, the blue and orange groups do not show peculiar differences ($p_{m/d} = 0.2967$), although both differ from the green group, compared to which they have a higher GS ($p_{m/d} = 0.0419$; $p_{m/d} = 0.0601$). As it will be further discussed in discussion, prognostic power of GS values should be taken with the grain of salt due to their qualitative and aggregated nature.

As for the clinical assessment of patients, the blue and green groups present similar to each other yet opposite characterizations with respect to the orange group. They display a lower percentage of patient with bone disease ($p_{ind} = 0.0769$; $p_{ind} = 0.1729$), therefore fewer people who have undergone an invasive combination of therapies ($p_{ind} = 0.0517$; $p_{ind} = 0.0863$). Moreover, although the results on the response to therapy are not significant due to the limited data available, they reveal a certain trend. In fact, both blue and green groups of patients are administered a milder therapy with respect to orange group. On one hand, such treatment results effective for the blue group, which shows the highest percentage of responders; while, on the other hand, this is not the case for the green group, which manifests the highest percentage of non-responders. Group 2 thus exhibit a clinical characterization comparable to group 0, whereas tree conformation analysis and prognostic assessment, i.e., response to therapy, agree in granting it a higher score of risk. Finally, the orange group presents the highest number of multi-metastatic patients, followed by the blue group and finally the green group, which hosts mostly oligo-metastatic patients.

From Figure 5.2.4, some extent of stratification is appreciable, although the groups' survival curves separation is not neat and statistically significant ($p = 0.12$). All patients of group 0 gradually respond since they feature mild disease, both from a structural, i.e., tree conformation, and clinical point of view. The green group host patients who the clinic would treat as not severe (in terms of number of lesions, GS and PSA baseline information), but our radiomics investigation has put in an at risk group, to be properly monitored, in terms of tree structure and tumor extension. In line with the results of our policy, these patients do not respond to therapy during the study period. Finally, the orange group carries severe patients from both a structural and a clinical point of view.

Since unsupervised approaches are thoroughly dataset dependent, hierarchical clustering grouped in the same clusters very heterogeneous patients, due to the limited data available. In fact, clinical variable variance of orange patients was consistently larger than other groups - despite not being the largest cluster. Interestingly, we fit a DBSCAN (Density Based Spatial Clustering of Applications with Noise) algorithm Khan et al. (2014) on the pruned-edit distance matrix which lead to the same clustering policy of patients. In this setting, while blue and green groups were confirmed to be clusters with similar density, the orange group was classified as noise, i.e., observations that display inconsistent density characterization. Accordingly, a couple of patients responded to therapy while the



a) Group-wise mean curves of tree heights **b) Interval-wise comparison between groups**

Figure 5.2.3: a) Curves displaying the *filtered* heights of the trees' vertices for the three groups. Operationally, curves were built as follows: for any fixed height (x-axis), for any tree in the selected group, we count the number of nodes whose height value is greater than the fixed one (y-axis). The curves in the plot represent the pointwise within-group means of such counts, and the shaded regions cover an area of 1 standard deviation around the means. The values of such counting process result in a monotonically non-increasing function detecting information about trees' heterogeneity. In fact, higher values of such function, especially as the height threshold becomes bigger and bigger, correspond to a greater number of heterogeneous lesions in the patients. Patients of group 0 (blue line) are characterized by a very homogeneous disease where trees branches are on average less and very short compared to the other groups; patients of group 1 (orange line) tend to exhibit more lesions than patients belonging to group 0, lesions which are intermediately heterogeneous, as their representation trees display both short branches and longer branches than group 0; patients in group 2 (green line) are associated to very heterogeneous diseases, displaying a similar number of lesions to group 0, but with the associated branches being much longer. A synthetic example of tree per each group is displayed in Figure 5.4.3, elucidating the differences with a graphical support. b) Functional comparison between curves: in order to test the hypothesis that curves belonging to different groups are different, we use the ANOVA procedure proposed in Pini and Vantini (2017). It outputs an interval-wise adjusted p-value function. Depending on the sort and level α of Type-I error control, significant intervals can be selected. Here, we highlighted in grey the region of significance. Of note, the curves appear different for what homogeneity-heterogeneity balance is concerned; they loose significance as they approach very big height values.

Variable	Test on	0 vs 1 (p-values)	0 vs 2 (p-values)	1 vs 2 (p-values)
GS	Mean	0.2967	**0.0419**	**0.0601**
	Variance	0.8368	0.5433	0.7093
Gleason Category	Independence	0.5129	0.5056	0.3077
Oligo or Multi (> 3)	Independence	**0.0601**	0.9260	0.1729
Oligo or Multi (> 5)	Independence	**0.0848**	0.6868	0.3339
$3 <$Lesions≤ 5	Independence	0.1969	0.9022	0.3950
N lesions	Mean	**0.0081**	0.4162	**0.0722**
	Variance	0.3871	0.4357	0.1469
Skeleton	Independence	**0.0769**	0.9622	0.1729
Total Volume (ml)	Mean	**0.0002**	0.4917	**0.0306**
	Variance	**0.0000**	**0.0047**	0.2009
PSA	Mean	**0.0116**	0.3787	0.3089
	Variance	**0.0013**	0.4714	0.1845
$\Delta PSA_{1,0}$	Mean	**0.0019**	0.3477	0.1810
	Variance	**0.0003**	0.4533	**0.0995**
$\Delta PSA_{2,0}$	Mean	0.3689	**0.0591**	0.1855
	Variance	**0.0066**	0.2159	**0.1085**
Ongoing Therapy	Independence	**0.0601**	0.5875	0.3339
Combined Therapy	Independence	**0.0517**	0.6091	**0.0863**
Therapy Response	Independence	0.6856	0.127	0.2907

Table 5.2.1: Significance in terms of p-values of the statistical tests between cluster 0 and cluster 1, cluster 0 and cluster 2, cluster 1 and cluster 2 in the proposed pipeline: non-parametric/parametric tests on difference of averages and variances were performed for (non-normal/normal) numerical variables while tests on category independence were performed for categorical variables.

majority did not respond and entered more invasive treatments. For these reasons, the orange survival curve is hardly interpretable and is left out the discussion. For sure, the high variability of this group testifies that a larger testing cohort would allow to identify further separations within this group, leading to clearer prognostic results.

5.2.2.1 Comparison with State-of-the-Art methods

The established radiomics frameworks contemplate the extraction of texture features from a single lesion, often located on the prostate where the bigger lesion or the primary tumor are found. Such features are usually fed into a classification or stratification model as to predict cancer diagnosis, staging and prognosis.

As a comparison with the state of the art, we investigated the stratification resulting from the analysis of the biggest lesions' textural description. We selected the bigger lesion of each patient, we reduced the texture vector dimensionality according to view-aware PCA dimensionality reduction procedures and we performed hierarchical clustering on the patient-to-patient Euclidean distance matrix with *ward* linkage. The clustering procedure lead to the stratification of patients into two groups, namely group 0 and group 1. It is worth noting that this clustering approach - based only on the bigger lesion and/or primary tumor - share some extent of the stratification underpinnings of the tree-based clustering. For the sake of clarity, we refer to one-lesion clustering as *tumor clustering* and to tree-based clustering as *heterogeneity clustering*. In particular, tumor clustering resulted to have a mild concordance with heterogeneity clustering (Rand Index = 0.43 Chacón (2021)).

Coherently, the tumor-based stratification leads to clinical significance. Tumor clustering pipeline discriminated between patients with different GS ($p_{m/d} = 0.0259$), number of lesions ($p_{m/d} = 0.0001$), oligo/multi-metastatic disease proportions ($p_{ind} = 0.0191$), PSA ($p_{m/d} = 0.0339$), ongoing therapy ($p_{ind} = 0.0847$) and total volume ($p_{m/d} < 0.0001$). However, $\Delta PSA_{1,0}$ ($p_{m/d} = 0.2942$), $\Delta PSA_{2,0}$ ($p_{m/d} = 0.2920$), proportion of patients exhibiting bone disease ($p_{m/d} = 0.5220$), combination of therapy ($p_{ind} = 0.3698$) and response to therapy ($p_{ind} = 0.2170$) did not result significant in tumor clustering pipeline. These findings were somehow expected. In fact, therapeutic guidelines are mainly taken on the basis of the characterization of the primary tumor. Accordingly, these results confirm the role of the primary tumor in acting as a driver for tumor heterogeneity and enforce radiomics role in the clinical treatment planning. Nevertheless, despite the coherence with qualitative clinical investigation, tumor-based stratification does not translate into a risk assessment and prediction. In fact, the Kaplan Meier curve, describing the probability of response to treatment of the two groups, appear almost superimposed ($p = 0.85$) and do not reveal any prognostic mechanism of the clustering.

As a step forward from one-lesion strategy, radiomics literature suggests to average radiomic descriptions of peer lesions belonging to a patient, as to obtain one single vector. Such vector-based representation plays for the mean imaging phenotype of all lesions expressed by a patient, taking into account the variability of the imaging profiles. Such method provide an information-complexity trade-off between one-lesion strategy and the tree-based patient representation we propose. Under these considerations, we performed patient-wise weighting of lesions' vectors, implemented the view-aware PCA dimensionality reduction methods and computed vector-based representation of each patient. The pipeline grouped all the patients in one cluster, although one patient with higher PSA was clustered separately from the rest of the cohort population as to meet hyperparameter criteria (e.g. minimum number of clusters at least equal to 2). Clear stratification was indeed not achieved in this setting, however a particularly bad-prognosis patient detached from the main group. From these findings, it follows that vector-based representation model did not lead to clear and solid results in our dataset, suggesting the non robustness of the lesions' weighting procedures.

5.3 Discussion

Current radiomic framework presents some limitations, including the inter-operator variability in imaging acquisition settings, the relatively small sample sizes bounding the performance of supervised approaches, the lack of standardization, the high dimensionality and the collinearity of radiomics variables as well as the absence of a clinical interpretation for features Smith et al. (2019). For these reasons, intra-patient tumor heterogeneity quantification has long been attempted with poorer results, hampering its embedding into daily practice. In this work, we propose a patient representation for agnostic multi-lesion cancer description, able to overcome intrisinc limitations of radiomics. The method exploits the texture analysis of lesions' imaging according to the radiomic workflow, overcoming features redundancy with PCA-based dimensionality reduction strategies. The proposed dendrogram representation results *agnostic* with respect to acquisition settings and operator variability as it is built upon evolutionary and statistical relationship within peer lesions' descriptions. Moreover, the small sample size issue is tackled by the employment of unsupervised methods. As to leverage the complex representation for stratification purposes, a suitable distance between dendrograms was required. Indeed, the pruned tree edit distance was specifically designed for heterogeneity-based hierarchical dendrograms and was the keystone to deliver a stratification policy based on agnostic disease conformations.

For what dimensionality reduction is concerned, view-aware PCA was hereby proposed as the trade-off between complexity of methods and performance of results: PCA is a simple and well-established technique for analyzing high dimensional datasets, increasing the interpretability of data while preserving the maximum amount of information. In other words, PCA is effective but simple enough to not add unnecessary complexity to the whole pipeline, overshadowing the main methodological novelties of our approach: the tree-structured patient reprsentation and the heteorgeneity-driven metric for trees. Other advanced methods have been proposed for multi-view dimensionality reduction Li et al. (2018), but interpretability and robustness are the advantages of using such a simple approach. Nevertheless, further studies could investigate the sensitivity of the representation model and the performance of the clustering policy when employing feature transformation methods that capture non-linear dependencies of across- and within-view features.

Compared to state-of-the-art disease representation, our approach shapes an exhaustive representation of intra-patient heterogeneity and devises an informed patient stratification. In fact, it leads to a more complex yet low-processed modelling of cancer disease, underlining interactions and relationships between lesions of individuals from which to infer prognostic knowledge. Clearly, one-lesion strategy did not provide a quantification of lesions' diverse phenotypes within a patient, as it only relies on the primary tumor. Nevertheless, tumor clustering lead to a coherent stratification with respect to the current clinical biomarkers, i.e., PSA, GS and oligo/multi-metastatic status. However, such clinically-informed stratification did not reach a significance in terms of prognostic power, bringing out the limitation of current clinical and radiomic-based biomarkers for treatment and prognosis. Interestingly, the proposed representation brings out a comprehensive way to capture tumor biology and heterogeneity, revealing a deeper appreciation of the disease than a single lesion or the primary tumor alone. On the other hand, the vector-based representation was confirmed insufficient to properly embed the patient's complexity of information. In fact, mean radiomic profile seems not to properly capture intra-tumor variability while it overlooks the primary tumor information entailing clinical information. In both cases - when only the primary tumor is considered and when the mean radiomic profile of lesions is computed - state of the art methods failed in perspectively stratifying patients.

Beside descriptive and prognostic purposes, the proposed tree-based representation and stratification of tumor heterogeneity permits an exhaustive comparison between the role played by the primary lesion and its involvement into phenotypic selection mechanism. This is worth to be drawn and further investigated from a tumor heterogeneity and prognostic point of view. In fact, tumor clustering showed a latent agreement with heterogeneity clustering, suggesting the reliability of the current clinical practice in assessing intra-tumor characterization from primary lesions. Accordingly, primary tumor information seems to be more informative than intra-patient mean lesions' profiles. If used in combination with dissemination indexes - such as number of metastases, dispersion of intra-patient lesions' radiomic profiles and number of involved organs -, primary tumor characterization could provide enough information to support therapeutic decisions when an exhaustive assessment of tumor metastases results too expensive.

On note, heterogeneity clustering highlighted milder significance for what GS biomarkers is concerned with respect to tumor clustering. Pertinently, although GS is a solid clinical prognostic factor driving therapy planning, it represents the histo-pathological analysis for characterizing primary and secondary tumor biology at molecular level. Accordingly, the aggregated value, that is the sum of primary differentiation pattern and secondary differentiation pattern, do not entail heterogeneity information. For instance, studies using surrogate PCa end points have suggested that outcomes for GS 7 cancers vary according to the predominance of pattern 4. PCa mortality, biochemical progression

and development of metastases differ for 3 + 4 and 4 + 3 tumors Stark et al. (2009). This means that, according to tree-based representation, patients tagged with a GS 7 may still be clustered in different prognostic groups and alter the tests on averages. For these reasons, GS should not be considered as a solid ground truth for a perspective model, rather it conveys only a association between radiomic-based heterogeneity assessment and its biological counterpart, that is tumor microscopic appearance. On the other hand, PSA and ΔPSA values significantly supported the predictive power of imaging-based representation in terms of cancer progression and disease free survival. Consistently, a decrease in PSA levels after treatment regimens was associated to therapy response. In this sense, exhaustive lesions' texture assessment and imaging-based heterogeneity quantification devise cancer subtypes that correlates with prognosis beyond clinical surrogates, eventually supporting treatment planning.

Basing on our and literature findings, the systematic digital tissue collection and its analysis should be enforced in the translational research of tumor disease and in the developing of targeted therapies. The debate around the therapeutic exploitation of imaging biomarkers for intra-tumor heterogeneity is nowadays on the cutting edge of medicine literature and it interlaces with other science field such as mathematics and geometry. This dynamic interplay between disciplines may provide a propitious route to ultimately attempt to limit tumor progression and treatment resistance. Stemming from this work, future research could consider longitudinal evolution of heterogeneity-based representation objects and, accordingly, investigate the course of the disease over time in a non invasive way.

5.4 Methods

In this section we outline the steps involved in the proposed methodological pipeline. In particular, methods for radiomics-based representation of patients' heterogeneity and its stratification are discussed. We present the challenges of analyzing a general radiomic dataset proposing an insightful dimensionality reduction approach (M1). Representation strategy is then deduced and described (M2). We then introduce an existing edit distance for comparing tree objects, on which we build the proposed metrics. It follows the derivation of an *ad hoc* metric (M3) for capturing intra-tumor heterogeneity variability and computing the similarity matrix between patients on which to perform the stratification according to hierarchical clustering.

5.4.1 M1: Dimensionality reduction

As previously introduced, radiomic features are regarded as a high dimensional vector embedding of the VOI, providing a non-invasive assessment of tumor appearance from routinely acquired imaging studies. Several softwares, e.g. LifeX software, allow to extract several texture indexes from VOIs, according to the formulas provided by the software documentation (www.lifexsoft.org). Considerable efforts have been devoted to link biological meaning with texture descriptors. So far, little evidence of tight correlation between the two has been found, preventing from univocally define tumor inherent heterogeneity of lesions. However, different textural features have been proposed and reviewed by Castellano et al. (2004) as measures of tumor-specific intra-lesion heterogeneity. Indeed, radiomics analysis is widely assumed to entail all the information needed for a definition of lesion heterogeneity Eary et al. (2008); Chaddad et al. (2018).

When managing a radiomic dataset, several challenges come across, above all high dimensionality and collinearity between features. Thus, prior to pairwise distance computation, lesions' radiomic vectors need to be properly reduced as to selectively bring out relevant information.

According to Nioche et al.Nioche et al. (2018), radiomic features divide into six semantic groups of different methodological levels of texture analysis. *First order statistics* are the statistical moments of the grey level distribution extracted from the VOI under analysis. *Shape features* describe morphological characteristics of the tumor. The *Grey Level Co-occurrence matrix* (GLCM) describes the co-occurrence of pairs of grey values in the VOI at a given distance δ (offset), usually set to 1, towards thirteen different directions. The *Grey Level Run Length matrix* (GLRLM) describes the length of homogeneous *runs* for each grey level, averaged across thirteen directions. Similarly, the *Grey Level Zone Length matrix* (GLZLM) provides information on the size of homogeneous *zones* for each grey level, averaged across three dimensions. Finally, the *Neighbour Grey Level Difference matrix* (NGLDM) corresponds to the difference of grey levels between one voxel and its twenty-six neighbors in three dimensions. From each of these groups, several indices are extracted, exhibiting a multi-view intrinsic structure that induces intra- and inter-group correlation patterns. Accordingly, such vectors disclose high collinearity between their elements that needs to be properly managed. To overcome this, we leverage the very basic idea of multi-view learning and dimensionality reduction approaches: the view-wise linear combination of features Kettenring (1971). We propose to separately apply the PCA to each of the radiomic groups, as to exploit the multi-view nature of the radiomic vectors. In this way, we may keep the information carried by each group well discerned, as it is methodologically extracted in different ways.

Upon pre-processing, namely missing values imputation and Z-transform normalization of radiomic variables, we thus perform this novel dimensionality reduction, namely *view-aware PCA*.

As depicted in Figure 5.2.1, features are grouped according to the six semantic group - *view* - as described above. Within each group, PCA is performed and two principal components are retained from the scores of each PCA, resulting in different percentages of explained variability. The process yields a total of twelve principal components, including six orthogonal pairs of linear combinations of original features. It follows that each lesion is described by a twelve-dimensional vector entailing view-wise texture information.

Further, we build the patient representation upon the such reduced radiomic vectors of peer lesions.

5.4.2 M2: Tree-based patient representation

To exhaustively represent patients' disease in terms of tumor heterogeneity, relationships between lesions needs to be learnt from data. Distance between texture descriptors could be an appropriate surrogate. Specifically, radiomic variables of a lesion - possibly after dimensionality reduction as in M1 - define a lesion-specific point in an Euclidean space. All lesions belonging to the same patient form a point cloud in $\mathbb{R}^p$, with a number of points n_i equal to the number of patient's tumor lesions and p being the number of radiomic variables.

Although some frameworks are available to compare point clouds via discrete transport (Mémoli, 2008; Nguyen et al., 2021), interpretability is often limited by the high dimensionality of the embedding space. Also, model based approaches, which capture the variability of cloud-generating processes by means of interpretable parameters, require a high number of observations in each point cloud to produce reliable estimations (Ghosal and van der Vaart, 2017).

A more insightful approach would be to transform the point cloud into a proper summary, i.e., a representation, equally informative and easily readable. Pertinently, hierarchical clustering dendrograms have been extensively studied in the last decades as they unveil the intrinsic relationship among points of a point cloud (for a review on hierarchical clustering dendrograms see Murtagh and Contreras (2017)). In our setting, the rationale

behind hierarchical clustering stems from the need to quantify to which extent lesions, i.e., their radiomic vectors, are similar within patients and how they get agglomerated, hierarchically, one to each other. A dendrogram is obtained in such a way that lesions are linked in terms of evolutionary relationship, based on similarities in their imaging characteristics. Figure 5.4.1 graphically describes the process whileSection 5.D formalizes the mathematical steps involved. Dendrograms' structure reflects the homogeneity between points of the point cloud. For instance, Figure 5.4.3 presents three dendrograms: the blue one describes a condensed point cloud, the green one presents a scattered point cloud while the orange tree denotes a hybrid situation.

To build hierarchical clustering dendrograms, a similarity measure is needed together with an agglomerative criterion - also known as *linkage* - that best suit the structure of the data and the aim of the analysis. In our setting, an appropriate similarity measure is the Euclidean distance between lesions' radiomic vectors, as suggested by Cavinato et al. Cavinato et al. (2020).

5.4.3 M3: A novel Heterogeneity-based distance

After having obtained patient representation, we proceed to defining a distance between dendrograms, which can properly reflect the affinity between patients in terms of tree conformations as manifestation of intra-tumor heterogeneity. A suitable metric should meet some requirements in order to produce effective results: (1) the comparison between dendrograms should reflect the properties of the point cloud they stem from: if two point clouds are close in terms of sparsity and conformation, we require the associated dendrograms to be close as well. In other words, any metric between dendrograms must hold some *continuity* properties with respect to the original point clouds comparison; (2) the metrics should weight differently the homogeneous part of the tree structures and the heterogeneous ones. This means that distance has to be evaluated as a trade-off between the extents of homogeneity and heterogeneity exhibited by the lesions of different patients.

5.4.3.1 Edit distance

Dendrograms are *unlabelled* object which, in our context, may have a different number of leaves and do not hold any a-priori correspondence between the leaves in different objects.

The literature dealing with the comparison of dendrograms is reviewed in Section 5.B, where we detail the limitations that prevent us from employing existing distances in our context. Chapter 2, Chapter 3 propose a novel distance for merge trees. Following the authors, we call this metric *edit distance* for merge trees and indicate it with d_E. The metric d_E is defined for weighted, rooted, unlabelled trees. As most of the metrics for unlabelled trees, its computational complexity has been shown to scale poorly with the number of leaves in the trees. However, it is particularly efficient for small-scale trees with respect to other metrics. In our setting, trees present a number of leaves less or equal to the number of tumor lesions in a patient, that is a few dozens at most. Thus, we can run the comptuation of d_E on general purpose machines, like personal computers. Unlike other metrics, continuity properties are easily proven. Moreover, d_E is interpretable, easy to understand and to communicate.

As depicted in Figure 5.4.2b), one tree T can be modified and transformed into a different tree T' by performing different sets of allowed modifications, each coming with its own cost (for details see Chapter 2, Chapter 3). The set of consequent edit operations which comes at the minimum cost is named the *optimal edit path* and represents the core of the edit distance between the two trees. The distance d_E is thus the total cost of the optimal edit path and is defined as:

$$d_E(T, T') = \inf_{\gamma \in \Gamma(T,T')} cost(\gamma) \tag{5.1}$$

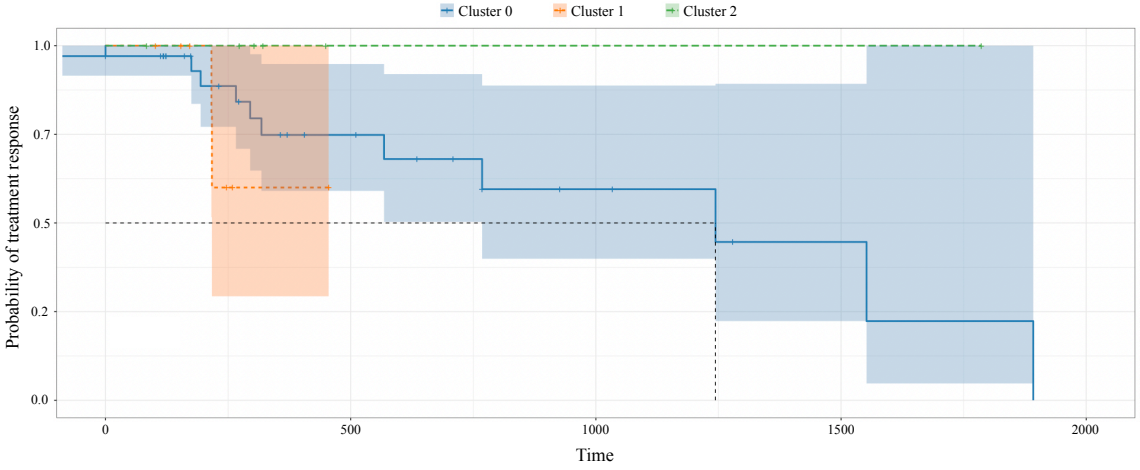


Figure 5.2.4: Group-wise Kaplan Meier curves of time to therapy response: it visually shows the probability of the response to treatment in a certain time interval. The blue line, the orange line and the green line correspond to group 0, 1 and 2 arising from clustering performed on patients' dendrograms. Groups have a different time to response. In particular, green group does not respond to therapy along the study period. Orange group shows indeterminate results due to the lack of and heterogeneity of clinical data. Blue group gradually responds throughout the study period.

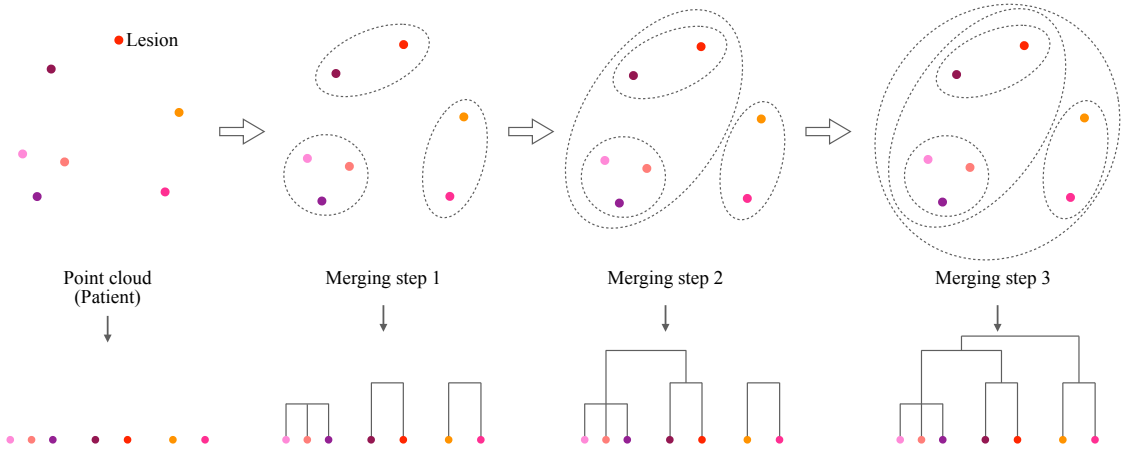


Figure 5.4.1: Tree-based patient representation via agglomerative hierarchical clustering: from the bottom up to the root, leaves get agglomerated and merged into bigger and bigger clusters, to finally converge in a single set. As a consequence, tree branches reflect pairwise similarity between lesions and the tree structure surrogates the overall dispersion among peer lesions. In the final dendrogram representation, leaves are the lesions of the patient and edges illustrate the similarity-connection between them. Leaves that are close to each other are intended by construction to be similar and exhibit a comparable radiomic profile (homogeneous) while distant leaves can be thought as lesions expressing different imaging phenotypes (heterogeneous). In this sense, dendrogram structure entails the heterogeneity quantification within the tumor, which needs to be exploited for heterogeneity-based stratification of patients. For mathematical formulation see Section 5.D.

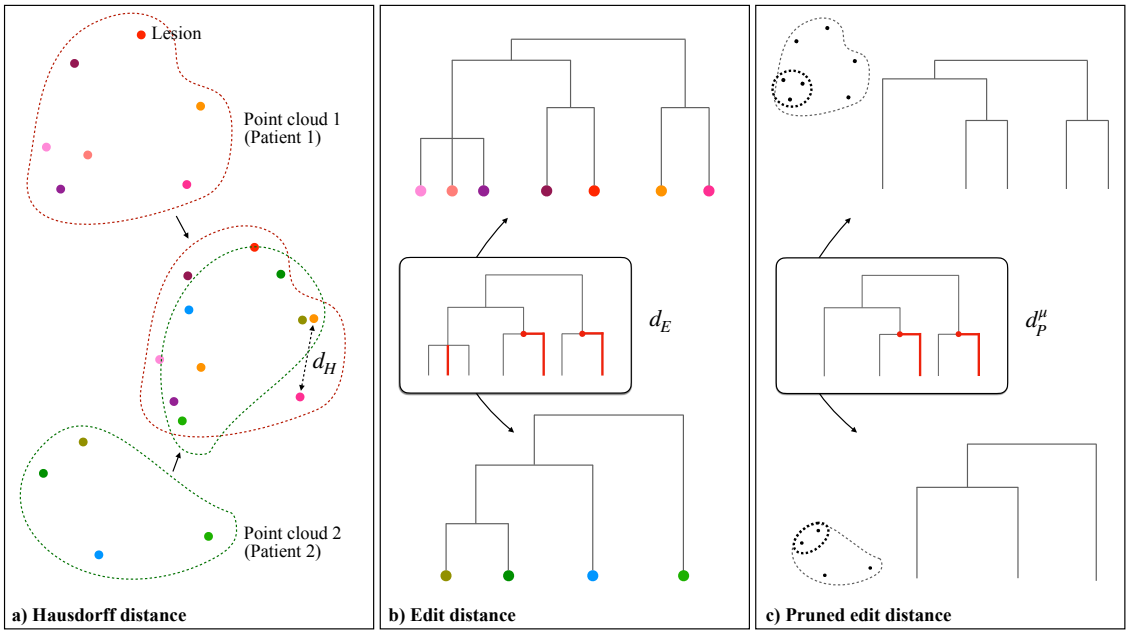


Figure 5.4.2: Continuity among metrics. a) qualitative approximation of the Gromov-Housdorff distance between two point clouds: the point clouds get overlapped and d_{G-H} is defined as the maximum distance between the two maximally distant points; Gromov-Hausdorff-closeness reflects the similarity in the spreading of points of two point clouds throughout the space. Specifically in the radiomic space, such spreading entails the quantification of inter-patient heterogeneity. This means that Hausdorff-close point clouds, i.e., patients' sets of lesions, have similar intra-patient heterogeneity characterization and thus should be regarded as similar by the metric we employ for dendrograms; b) Tree edit distance between hierarchical clustering dendrograms: the distance is given by the sum of the costs of the minimum number of modifications needed for transforming a tree into the other. Modifications include positive/negative shrinking, deletion/insertion and ghosting/splitting. The *shrinking* edit multiplies the weight value of an edge with a positive factor, which can either lengthen (positive shrinking) of shorten (negative shrinking) the original edge weight. The cost of shrinking an edge is equal to the absolute value of the difference between the initial and the final weights. *Deleting* or *inserting* an edge (v_1, v_2) removes or introduces a branch at a given height, altering the children-father structure of the tree. For any deletion/insertion, the cost is equal to the weight of the edge deleted/inserted. Finally, the *ghosting* edit eliminates a vertex v that connects only two adjacent edges (order 2 vertex) such as one new edge results from the sum of the two former edges. The opposite edit is the *splitting*. Ghosting and splitting have no cost, therefore order 2 vertices are *de facto* irrelevant when computing the cost of an edit path; c) Pruned tree edit distance between pruned dendrograms: pruning removes leaves with weights $\leq \varepsilon$, eventually aggregating homogeneous phenotypes. The operator P_ε thus gradually discard intra-patient homogeneity, disclosing only the heterogeneous - independent - tumor phenotypes. Of note, d_P^μ is different from d_E since the pruning modulates the effect of cardinality on the distance computation by removing redundant edges of the tree and compressing tree dimensionality.

where $\Gamma(T, T')$ indicates all the possible edit paths which start in T and ends in T'. The algorithm for d_E computation is exhaustively detailed in Chapter 2. Through combinatorial objects called *mappings*, it is shown that d_E is a metric in the space of merge trees and that it can be computed with a Linear Integer Programming approach Chapter 2.

Upon these premises, we proceed to verify the two aforementioned conditions. Specifically, we prove the continuity property of d_E (1) and propose a modification of d_E as to meet the homogeneity-heterogeneity requirement (2).

5.4.3.2 Continuity property of d_E

As previously stated, a continuity result with respect to the original point clouds comparison would guarantee interpretability properties for the distance between dendrograms: under certain hypotheses, if two clouds are pointwise close, also their merge trees should be close with respect to d_E. In Figure 5.4.2a), we qualitatively introduce the Gromov-Hausdorff metric between point clouds (see Section 5.E for a formal definition). It can be interpreted as a measure of the pointwise proximity between two point clouds and provide a comparison between the heterogeneity of two patients' diseases. In Section 5.E we prove that Gromov-Hausdorff-closeness (or distance) for point clouds implies Edit-closeness (or distance) for the associated dendrogram objects, i.e., multi-lesion patients representation.

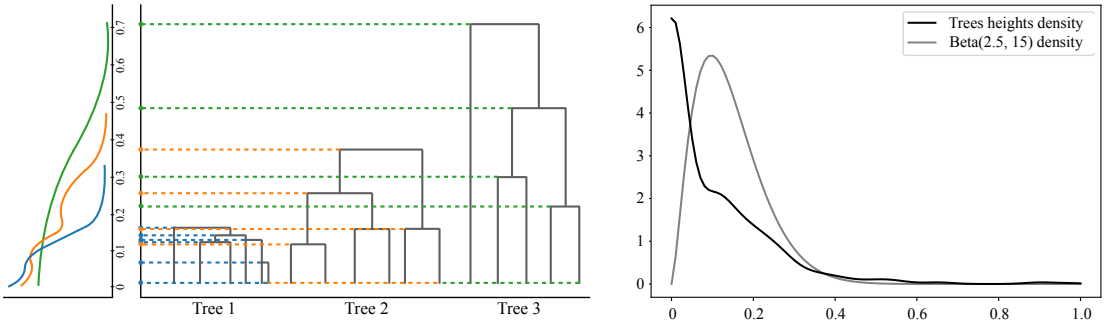
5.4.3.3 Homogeneity-heterogeneity trade-off

In the edit distance d_E, the distance values are strongly dependent on the clouds cardinalities, meaning that pairs of point clouds with higher cardinalities tend to be farther apart from pairs of point clouds with smaller cardinalities. At first sight, such assumption sounds reasonable for stratification purposes. In fact, patients with multiple lesions are known to exhibit a more severe disease than patients with fewer lesions, as the spreading of the tumor entails prognostic power. Still, the mere counting of lesions lacks of robustness in perspective studies and, in this context, may overshadow the variability between hierarchical dendrograms induced by intra-patient heterogeneity. For this reason, we propose a modification of the metric d_E as to mitigate cardinality issue.

5.4.3.4 Pruned edit distance

The kind of variability we are interested in is the one induced by patient-wise heterogeneity between lesions. By construction of the dendrogram representation, two lesions of a patient are heterogeneous - in terms of radiomic/imaging description - according to the length of the dendrogram branches connecting them. The longer the branches, the higher the inter-lesions heterogeneity and, viceversa, the shorter the branches the more homogeneous the patient's disease phenotypes. Accordingly, we may want to modulate the extent to which we consider edit costs according to branch length. In particular, we may want to induce edits applied on small edges to contribute less to the final distance than bigger edges, which we deem more relevant for stratification purposes.

We introduce the pruning operator P_ε as regularization strategy, which deletes leaves associated with edges whose weights are so small that one may want to neglect them in the analysis of heterogeneity. Given a threshold ε, we consider for deletion all leaves whose father-child edge has weight $\leq \varepsilon$. However, when two or more of candidate leaves share the same father, i.e. they are *siblings*, we delete all the leaves but the one with the bigger weight. Moreover, if the weights of the siblings are equal, as it is often the case in clustering dendrograms, we randomly choose to keep one of them, delete the other(s) and, eventually, *ghost* their father (see Figure 5.4.2 for meaning of ghosting). This pruning operation is recursively iterated until no leaves with small edges can be found. To note, removing only one leaf in case of two small-weight siblings is equivalent to considering



a) Synthetic patient dendrograms and vertices height densities **b) All patient vertices height density**

Figure 5.4.3: Choice of μ: a) costruction of qualitative densities of the vertices heights in three example dendrograms: the velocity with which leaves get merged in a dendrogram, i.e., edges length variability, reflects the heterogeneity characterization of lesions. Per every dendrogram, branches heights (rescaled on $[0, 1]$ dividing by the highest value) are annoted on the left and their associated density is inspected. The vertices heights of a patient exhibiting homogeneous lesions concentrates in a small real interval $[0, a]$ - with $a > 0$ (blue tree); the vertices heights of a patient exhibiting heterogeneous lesions spread in a range of values far from zero $[a, b]$, with $a, b > 0$ (green tree); a patient showing groups of homogeneous lesions, the one heterogeneous to the others, is associated to a dendrogram with an explicit clustering structure with clusters with multiple close leaves (orange tree). The vertices heights distribution displays two components, reflecting both the homogeneity of similar lesions - with values close to 0 - and the heterogeneity of dissimilar clusters - with values far from 0; b) μ provides the coefficients with which to weight the different pruning cutoffs ε, to neglect the homogeneity within clusters of similar lesions' phenotypes and bring out the informative heterogeneity between different phenotypes. To efficient the computation, a parametric shape of μ is used and empirical heights distributions of all patients (black line) is exploited to model the distribution. In the population heights distribution, we discern both homogeneous and heterogeneous phenotypes. The two components are demarked with a saddle point on 0.15. Accordingly, low weights of μ should be associated to $\varepsilon \ll 0.15$ and $\varepsilon \gg 0.15$ and high weights to $\varepsilon \simeq 0.15$. In fact, low ε values entail pure homogeneity information while high ε values would lead to discarding useful heterogeneity information. We thus infer to model μ as an asymmetric bell-shaped density function with one peak centered in the saddle point of the heights distribution. The Beta family of distributions, supported in $[0, 1]$, well meets the requirements; it simplifies both the numeric integration procedure and the results' interpretation. The Beta-shaped μ is centered on 0.15 (grey line), properly tuning α and β shape parameters ($\alpha = 2.5, \beta = 15$).

the two leaves as clustered together from the "beginning" in the hierarchical clustering procedure. Accordingly, siblings leaves (lesions) entail phenotype expressions so similar to be considered as one single imaging phenotype. In this way, the pruned tree displays the number of *different* phenotypes coexisting in the patient instead of the mere number of lesions. Figure 5.4.2c) displays the edits needed for transforming a pruned tree into another, whose costs determine the pruned edit distance.

Operationally speaking, the "correct" value of ε is a-priori unknown and needs to be tuned with a complexity-information trade-off. To enhance the robustness of this parameter choice, we take the weighted average of the distances between two trees pruned with all the possible values of ε. Accordingly, the definition of *pruned edit distance* for general merge trees develops as follows. Given two merge trees T and T', the pruned edit distance is:

$$d_P^{\mu}(T, T') := \int_{\mathbb{R}} d_E(P_\varepsilon(T), P_\varepsilon(T'))d\mu(\varepsilon) = \mathbb{E}_{\varepsilon\sim\mu}[d_E(P_\varepsilon(T), P_\varepsilon(T')] \tag{5.2}$$

where μ is a finite measure on $\mathbb{R}$ which provides the weighting strategy across different values of ε in order to compute a weighted average among trees distances. The higher the mass μ associated to an interval $[a, b]$, the bigger the contribution to the final result of the tree distance according to $\varepsilon \in [a, b]$. In other words, the measure μ allows to control the contribution to the final distance of branches with weight below ε, which are indeed homogeneous enough to be removed. Figure 5.4.3 elucidates the choice of μ tuned on case study data. Note that if we have a sequence of weakly converging probability measures $\mu_n \rightharpoonup \mu$, then $d_P^{\mu_n}(T, T') \to d_P^{\mu}(T, T')$. This implies that the proposed distance is robust with respect to the choice of μ: similar measures μ (in the sense of weak convergence) would give similar distances.

To assess the different behaviours between d_E and d_P^{μ} and the extent to which d_P^{μ} is suitable for our purposes, in Section 5.G we present a detailed simulation study. Moreover, we can prove that, under general conditions on μ, d_P^{μ} is still a metric (for proof see Section 5.F).

Appendix

5.A Patients' personal information summary

Table 5.A.1 and Table 5.A.2 summarize the patients' population.

Variable	Mean	Std. dev.	Median	Range
Age	72.09	7.03	71.68	54.88 – 85.24
Total volume	16.41	34.72	3.16	0.22 – 207.70
Gleason Score	7.73	1.03	7.00	5.00 – 9.00
PSA	18.16	70.96	2.66	0.09 – 591.00

Table 5.A.1: Statistical summary of patients' personal information (continuous variables).

Variable		Number of patients (%)
Number of metastases	Oligo (<3)	38 (41.3%)
	Multi (≥3)	54 (58.7%)
	Oligo (<5)	60 (65.22%)
	Multi (≥5)	32 (34.78%)
	Intermediate (3≤n<5)	22 (23.92%)
Gleason Score (dichotomous)	<7	8 (8.7%)
	=7	45 (48.91%)
	>7	31 (33.69%)
	missing	8 (8.7%)
Ongoing therapy	Y	33 (35.87%)
	N	59 (64.13%)
Initial therapy	RP	23 (25%)
	RP+RT	52 (56.52%)
	RT	9 (9.78%)
	missing	8 (8.7%)
PSA (dichotomous)	≤ 1.93	33 (35.87%)
	>1.93	48 (52.17%)
	missing	11 (11.96%)

Table 5.A.2: Statistical summary of patients' personal information (categorical variables).

5.B Distance metrics for trees: brief literature review

In this section we present a high level overview of the literature concerning the comparison between different trees.

Dendrograms are *unlabelled* object which, in our context, may have a different number of leaves and do not hold any a-priori correspondence between the leaves in different objects. The literature dealing with the comparison of dendrograms divide in two macro-areas, including (1) metrics defined for clustering dendrograms and (2) metrics designed for *merge trees*. The first family of metrics mainly deals with *labeled* trees (Robinson and Foulds, 1979; Billera et al., 2001) as byproducts of a hierarchical clustering algorithm. We refer to Flesia et al. Flesia (2009) for an exhaustive review of distance definitions. This kind of metrics are known to be heavily dependent on the graph structure of the dendrograms, like many others Poon et al. (2013); Colijn and Plazzotta (2018); Lewitus and Morlon (2016); Kim et al. (2020), leading to limitations when comparing dendrograms with a different number of leaves and lacking theoretical continuity results with respect to dendrogram-associated point clouds Smith (2019, 2020). On the other hand, within topological data analysis (Edelsbrunner and Harer, 2008), dendrograms are often referred as a particular case of merge trees, obtained when all the leaves of a merge tree lie at height 0. This allows to transfer merge trees literature to dendrogram analysis. Most of the metrics belonging to this second family, however, share one main drawback, namely the out of reach computational cost, which makes them unsuitable for our application Beketayev et al. (2014); Bauer et al. (2020); Cardona et al. (2021). Besides, the metrics with more performing algorithms Pont et al. (2022); Sridharamurthy et al. (2020) still lack the theoretical investigation to assess some practical properties, making them less worthy than others.

5.C Building the vertices heights curve

In Figure 5.C.1, we explain in details how to build the curve displaying the filtered heights of one tree' vertices. In the plot there are three detailed examples showing how to obtain the curves in Figure 5.2.3. The colors of the curves matches the colors of the trees. For each tree and for every $h \in \mathbb{R}$, the value $f(h)$ is equal to the number of vertices (highlighted with dots) which lie above h. For instance consider the blue dendrogram: we obtain a value of 3 until we reach the height where two leaves merge, and then 1 for a small interval of values, and lastly 0 from the height of the root of the blue tree onwards.

5.D Dendrograms construction

In this section we present few technical definition that we need in order to describe the dendrogram representation we employ. We describe the procedure in the general case of having a finite metric space (X, d) i.e. a finite set $\{x_1, \dots, x_n\}$ with a metric $d : X \times X \to \mathbb{R}_{\geq 0}$ which is reflexive, symmetric and satisfies the triangular inequality. In our case we work with $\{x_1, \dots, x_n\} \subset \mathbb{R}^n$ and the Euclidean norm.

Definition 5.1. *A tree structure T is given by a finite set of vertices V_T and set of edges $E_T \subset V_T \times V_T$ which form a connected rooted acyclic graph. The order of a vertex is the number of edges which have that vertex as one of the extremes. Any vertex with an edge connecting it to the root is its child and the root is its father. In this way we recursively define father and children (possibly none) relationships for any vertex on the tree. The vertices with no children are called leaves and are collected in the set L_T, while the set of children of a vertex $x \in V_T$ is called $child(x)$. Similarly, the vertex $father(x)$ is the father of the vertex x.*

The relationship $father > child$ induces a partial order on V_T. The edges E_T are given in the form of ordered couples (a, b) with $a < b$. For any vertex $v \in V_T$, $sub_T(v)$ is the subtree of T rooted in v, that is the tree structure given by the set of vertices $v' \leq v$. If clear from the context we might omit the subscript T.

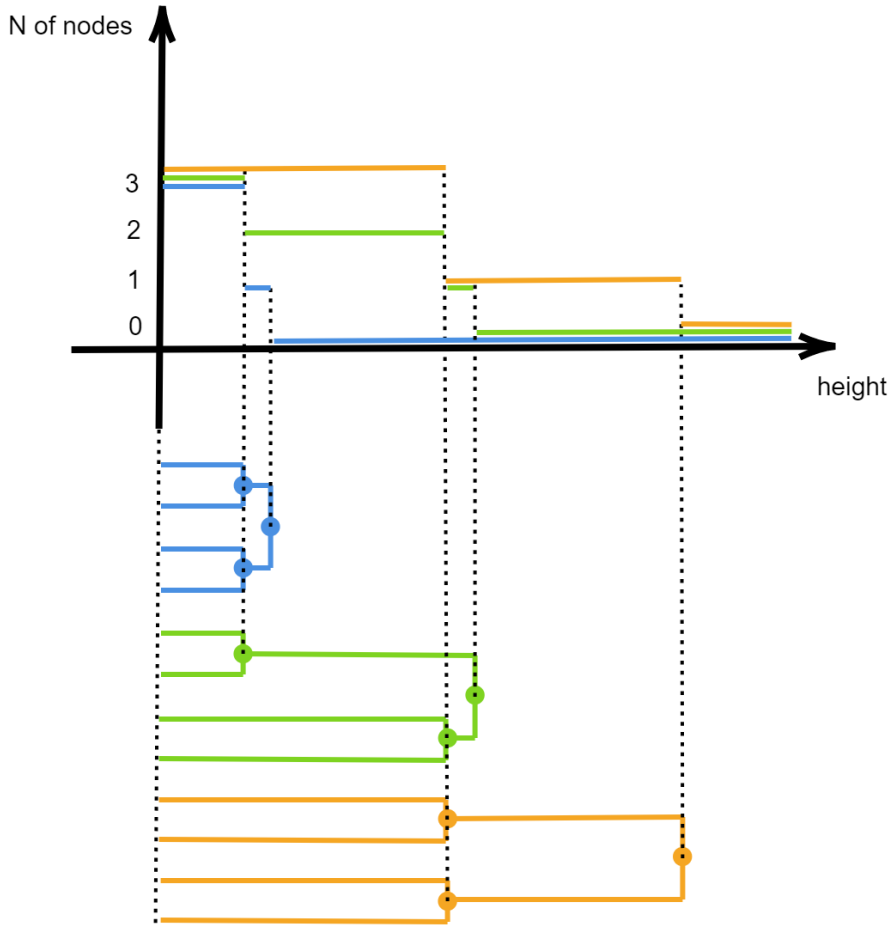


Figure 5.C.1: Procedure for building the vertices heights curve of three example patient-tree.

Now, to obtain a dendrogram we need to add some kind of length measure to a tree structure.

Definition 5.2. *A merge tree (T, f) is a finite tree structure T coupled with a monotone increasing function (with respect to partial ordering on V_T) $f : V_T \to \mathbb{R}$. If $f(l) = 0$ for all $l \in L_T$, then we say that the merge tree is a dendrogram. The function f also defines a weight value for every edge $e = (v, father(v))$: $w_T(e) = f(father(v)) - f(v)$.*

To build a hierarchical clustering dendrogram T_C from a finite metric space (C, d_C) we proceed as follows. With K we indicate the set of clusters we are considering:

(S0) at the beginning $K = \{\{c\} \,|\, c \in C\}$, and every $c \in C$ is associated to a leaf $v_c \in V_{T_C}$ with $f(v_c) = 0$;

(S1) consider all the couples of clusters $k_1, k_2 \in K$ and we measure the distance $d(k_1, k_2)$ according so some *linkage*;

(S2) pick $k, k' \in K$ such that $d(k, k') = \min_{k_i \in K; k_1 \neq k_2} d(k_1, k_2)$ and add the vertex $v_{kk'}$ to V_{T_C} with $f(v_{kk'}) = d(k, k')$. Then remove k and k' from K and add $k \cup k'$ to K;

(S3) start again from (S1) unless $K = C$.

The linkage determines the distance $d(k_1, k_2)$ between $k_1, k_2 \subset C$ and the most common examples are:

- single linkage: $d(k_1, k_2) = \min_{c_i \in k_i} d_C(c_1, c_2)$
- complete linkage: $d(k_1, k_2) = \max_{c_i \in k_i} d_C(c_1, c_2)$
- average linkage: $d(k_1, k_2) = (\#k_1 \cdot \#k_2)^{-1} \cdot \sum_{c_i \in k_i} d_C(c_1, c_2)$, where $\#k_i$ is the cardinality of the finite set k_i.
- ward linkage: see Jr. (1963)

It is well known that single linkage is very sensitive to outliers, while complete linkage is the most conservative choice in term of clustering points together. Average linkage displays a kind of in-between behaviour. For this reason we resorted to average linkage.

5.E Continuity Proposition

Having a continuity result of the distance between dendrograms with respect to some metric between point clouds would surely benefit the consistency and the interpretability of the framework: whenever a representation of a datum is employed, looking at how changes in the representation reflect on changes in the initial datum can help both assessing the consistency of the pipeline, and familiarizing with the representation. In our case we are particularly interested in looking at what happens at the distance between dendrograms as two sequences of point clouds get closer and closer. To do so, we introduce the definitions of the Hausdorff and Gromov-Hausdorff metrics between point clouds. We then prove the continuity proposition between Hausdorff distance and the Edit distance, from which it follows the continuity proposition between Gromov-Hausdorff distance and the Edit distance.

Given $C = \{x_1, \dots, x_n\}$ and $C' = \{y_1, \dots, y_m\}$ two point clouds in a metric space (X, d), we can build at least a function $\gamma : C \to C'$ such that $\gamma(x_i)$ is (one of) the closest point(s) to x_i, belonging to the cloud C'. Similarly, we can build $\varphi : C' \to C$ so that $\varphi(y_j)$ is (one of) the closest point(s) to y_j, belonging to the cloud C. The Hausdorff distance between C and C' is given by:

$$d_H(C, C') = \max\{\max_{x \in C} d(x, \gamma(x)),\ \max_{y \in C'} d(y, \varphi(y))\} \tag{5.3}$$

The distance d_H has been proven to be a metric for the space of all compact subsets of X Rockafellar and Wets (2009).

Leveraging on the Hausdorff distance, given two compact metric spaces X and Y we define the Gromov-Hausdorff metric as $d_{G-H}(X, Y) := \inf d_H(\gamma(X), \varphi(Y))$ where γ and φ vary over all possible isometries of (respectively) X and Y into another (common) metric space Z Burago et al. (2022).

Consider two point clouds in the metric space (X, d), $C = \{x_1, \dots, x_n\}$ and $C' = \{y_1, \dots, y_m\}$ and we consider T_C and $T_{C'}$ the single linkage hierarchical clustering dendrograms obtained from C and C' respectively. In the following, we prove the following result:

Proposition 5.3. *Given $C = \{x_1, \dots, x_n\}$ and $C' = \{y_1, \dots, y_m\}$ point clouds in a metric space (X, d) and given T_C and $T_{C'}$ single linkage hierarchical clustering dendrograms obtained from C and C' respectively, there is a simplicial complex S and two functions $f : S \to \mathbb{R}$ and $g : S \to \mathbb{R}$ such that the merge tree associated to f (via sublevel set filtration) is isomorphic to T_C, the merge tree associated to g is isomorphic to $T_{C'}$, and $\| f - g \|_\infty \leq 2 d_H(C, C')$.*

Proof. Let $\gamma : C \to C'$ and $\varphi : C' \to C$ be the two operators which map a point of a point cloud C' to (one of) the closest point(s) of the other cloud C and viceversa.

Consider the following simplicial complex S. Its 0 simplices are $x_1, \ldots, x_n, y_1, \ldots, y_m$ and its 1 simplices are all possible edges between 0 simplices, forming a complete graph.

Now we define two functions $f : S \to \mathbb{R}$ and $g : S \to \mathbb{R}$ such that the merge trees T_f and T_g obtained with the lower star filtration from f and g (see Chapter 4) are isomorphic to T_C and $T_{C'}$.

Define: $f(s) = 0$ for every 0 simplex s. Then for a 1 simplex of the form $e_{ij} = (x_i, x_j)$, we have $f(e_{ij}) = d(x_i, x_j)$. For 1 simplices of the form $e'_{ij} = (y_i, x_j)$ we have $f(e'_{ij}) = d(\varphi(y_i), x_j)$. Lastly, for 1 simplices of the form $e''_{ij} = (y_i, y_j)$ we have $f(e''_{ij}) = d(\varphi(y_i), \varphi(y_j))$. Note that $t \in Im(f)$ iff $t = d(x_i, x_j)$ for some i and j. Clearly, f is a finite set and we can order it: $t_0 = 0 < t_1 < \ldots$.

Similarly we define $g(e''_{ij}) = d(y_i, y_j)$, $g(e'_{ij}) = (y_i, \gamma(x_j))$ and $f(e''_{ij}) = (y_i, y_j)$.

Consider now the connected components of the graph $S^f_t := \{s \in S | f(s) \leq t\}$ for $t \in \mathbb{R}$. If $t < 0$, S^f_t is empty. If $t = t_0 = 0$, then all 0 simplices are in S^f_0, plus the 1 simplices of the form (x_i, y_j) such that $\varphi(y_j) = x_i$ and (y_i, y_j) such that $\varphi(y_i) = \varphi(y_j)$. This means that every vertex y_i is connected with exactly one point x_j and with all other y_k such that $\varphi(y_k) = x_j$. That is, there are n path connected components, one for each x_i. Call such components $[x_i]$.

Consider the value $t = t_1 = d(x_i, x_j)$. For every $y \in \varphi^{-1}(x_i)$ and $y' \in \varphi^{-1}(x_j)$, we have $f((y, y')) = f((x_i, y')) = f((y, x_j)) = f((x_i, x_j)) = d(x_i, x_j)$ and so all these 1 simplices get added, when passing from S^f_0 to $S^f_{t_1}$. Moreover, these are the only ones which get added. Which means that we get all possible edges between $[x_i]$ and $[x_j]$ but all others components are left unchanged. And this happens whenever we hit a level $t_k = d(x_i, x_j)$: we add to the simplicial complexes $S^f_{t_k}$ all possible edges between $[x_i]$ and $[x_j]$.

Now, we build the single linkage hierarchical dendogram T_C associated to C, with labels given by $\{\{x_1\}, \ldots, \{x_n\}\}$, and the merge tree T_f associated to $f : S \to \mathbb{R}$ with labels $\{[x_1], \ldots, [x_n]\}$. An internal vertices of T_C indicating the merging of two leaves $\{x_i\}$ and $\{x_j\}$ will be called $\{x_i, x_j\}$, and similarly a vertex called $\{x_i, x_j, x_k\}$ indicates that the leaves of the subtree rooted in that vertex are $\{x_i\}$, $\{x_j\}$ and $\{x_k\}$. In the same fashion, an internal vertex of T_f where to components $[x_i]$ and $[x_j]$ merge is named $[x_i] \bigcup [x_j]$. A vertex called $[x_i] \bigcup [x_j] \bigcup [x_k]$ is associated to the origin of the connected component $[x_i] \bigcup [x_j] \bigcup [x_k]$. Thus, we can define a map $\eta : V_{T_C} \to V_{T_f}$ induced by $\eta(x_i) = [x_i]$ and $\eta(\{x_i, x_j, x_k\}) = [x_i] \bigcup [x_j] \bigcup [x_k]$ which is an isomorphism of merge trees. An analogous proof yields the isomorphism between $T_{C'}$ and T_g.

To conclude the proof it is enough to notice that: $||f - g||_\infty \leq 2\varepsilon$ with $\varepsilon = d_H(C, C')$. In fact, for vertices s: $f(s) = g(s) = 0$. For an edge e, we have the following possibilities:

- $e = (x_i, x_j)$: $|f(e) - g(e)| = |d(x_i, x_j) - d(\gamma(x_i), \gamma(x_j))|$. We have $d(x_i, \gamma(x_i)) \leq \varepsilon$, $d(x_i, x_j) \leq d(\gamma(x_i), \gamma(x_j)) + 2\varepsilon$ and $d(\gamma(x_i), \gamma(x_j)) \leq d(x_i, x_j) + 2\varepsilon$; which, together, give $|f(e) - g(e)| \leq 2\varepsilon$.

- $e = (y_i, y_j)$: $|f(e) - g(e)| = |d(\varphi(y_i), \varphi(y_j)) - d(y_i, y_j)|$; reasoning as above we obtain $|f(e) - g(e)| \leq 2\varepsilon$

- $e = (x_i, y_j)$: $|f(e) - g(e)| = |d(x_i, \varphi(y_j)) - d(\gamma(x_i), x_j)|$. Again in the same fashion we have: $d(x_i, \varphi(y_j)) \leq d(x_i, y_j) + d(y_j, \varphi(y_j)) \leq d(x_i, \gamma(x_i)) + d(\gamma(x_i), x_j) + d(y_j, \varphi(y_j)) \leq d(\gamma(x_i), x_j) + 2\varepsilon$. Which entails $|f(e) - g(e)| \leq 2\varepsilon$

□

Corollary 5.4. *Given $C = \{x_1, \dots, x_n\}$ and $C' = \{y_1, \dots, y_m\}$ point clouds in (X, d) metric space, and given T_C and $T_{C'}$ the single linkage hierarchical clustering dendrograms obtained from C and C' respectively, we have $d_E(T_C, T_{C'}) \leq 6(n+m)d_H(C, C')$.*

Proof. We apply Proposition 5.3 and then we are in the position to use Theorem 1 in Chapter 4 to obtain that $d_E(T_f, T_g) \leq 2(2d_H(C, C')) \cdot (n+m)$. □

With the above results, we can prove a last corollary involving the Gromov-Hausdorff distance between compact metric spaces.

Corollary 5.5. *Given two finite metric spaces $C = \{x_1, \dots, x_n\}$ and $C' = \{y_1, \dots, y_m\}$ and given T_C and $T_{C'}$ the single linkage hierarchical clustering dendrograms obtained from C and C' respectively, we have $d_E(T_C, T_{C'}) \leq 4(n+m)d_{G-H}(C, C')$.*

Proof. We apply Proposition 5.3 and Corollary 5.4 on the images $\gamma(X)$ and $\varphi(Y)$ for every $\gamma : X \to Z$, $\varphi : Y \to Z$ isometries, and for every Z metric space. □

5.F Proof about d_P^μ being a metric

We prove the following proposition.

Proposition 5.6. *If there is $M > 0$ such that for every $m \leq M$, $\mu([0, m]) > 0$ the d_μ^P is a metric.*

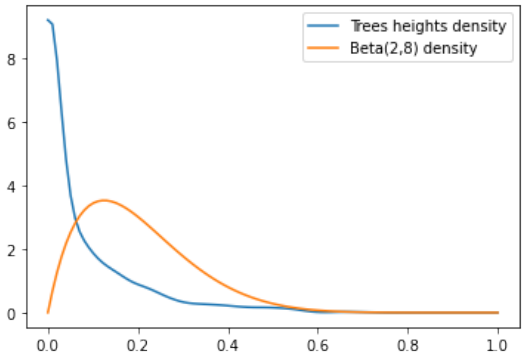
Proof. □

- suppose $d_P^\mu(T, T') = 0$. Let $m = \min\{\min_{e \in E_T} w_T(e),\ \min_{e' \in E_{T'}} w_{T'}(e')\}$; then for any $\varepsilon \in [0, m)$, $P_\varepsilon(T) = T$ and $P_\varepsilon(T') = T'$. If $d_E(T, T') > 0$, since $\mu([0, m)) > 0$, then:
$$0 < \int_{[0,m)} d_E(P_\varepsilon(T), P_\varepsilon(T'))d\mu(\varepsilon) \leq d_P^\mu(T, T') = 0$$
which is absurd. But then $d_E(T, T') = 0$ and so $T = T'$.
- symmetry is obvious
- the triangle inequality holds for d_E and so
$$d_E(P_\varepsilon(T), P_\varepsilon(T')) \leq d_E(P_\varepsilon(T), P_\varepsilon(T'')) + d_E(P_\varepsilon(T''), P_\varepsilon(T'))$$
The linearity of the integral then entails $d_P^\mu(T, T') \leq d_P^\mu(T, T'') + d_P^\mu(''T, T')$.

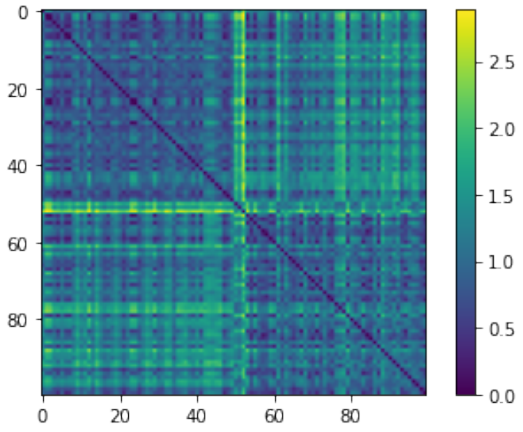
5.G Heterogeneity-based Simulation for d_μ^P

In this section, we test the metric d_μ^P and the whole pipeline employed in the case study in a supervised - in a broad sense - and easier setting. In particular, the aim of this simulation is to showcase the differences between d_E and d_μ^P and to which extent d_μ^P captures heterogeneity in a point cloud.

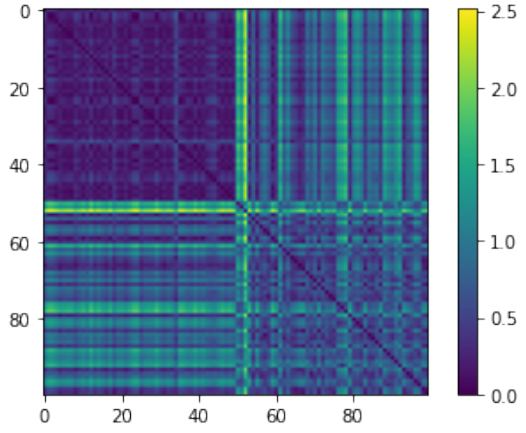
We generate point clouds in $\mathbb{R}^2$ according to two generating processes. The size n_1^i of the i-th point cloud of the first group is sampled uniformly from $[2, 20] \bigcap \mathbb{Z}$ and then a sample of size $(n_1^i, 2)$ is taken from a normal distribution $\mathcal{N}(0, \sigma_1)$, with $\sigma_1 = 1$. Similarly, the j-th point cloud of the second group has cardinality n_2^j sampled uniformly from $[2, 10] \bigcap \mathbb{Z}$, and the cloud itself is taken as a sample of size $(n_2^j, 2)$ distributed according to $\mathcal{N}(0, \sigma_2)$, with $\sigma_2 = 2$. The data set of point clouds contains 50 clouds of the first group and 50 of the second group.



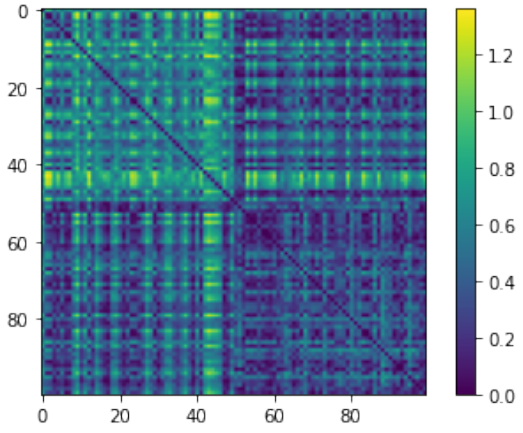
(a) **Density of vertices heights from trees in the simulation data, along with the chosen Beta distribution, which has parameters $a = 2$ and $b = 8$.**



(b) **Matrix of pairwise distances obtained with d_E.**



(c) **Matrix of pairwise distances obtained with d_μ^P.**

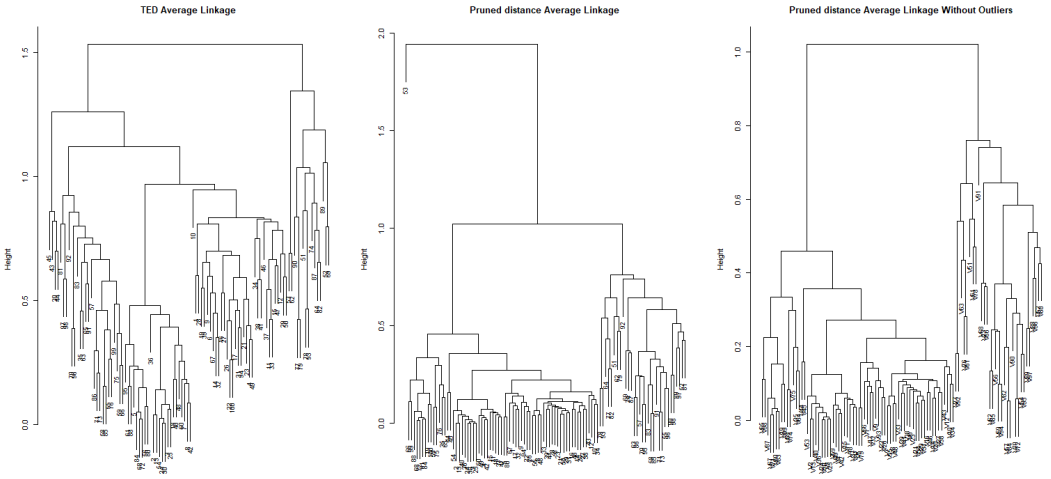


(d) **Absolute differences between the matrix obtained with d_E and d_μ^P.**

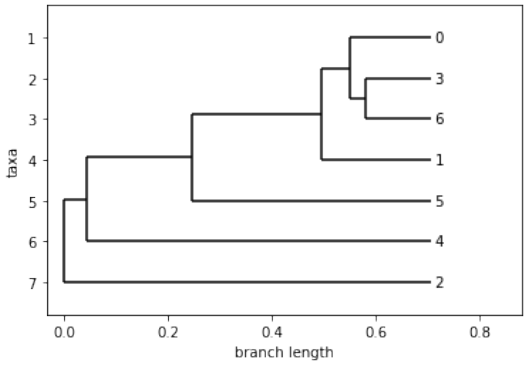
Figure 5.G.1: The plot in the left upper corner is used to fix μ in the case study of Section 5.G, according to the procedure detailed in Figure 5.4.3; the other figures show the pairwise distance matrices obtained in the case study of Section 5.G.

From the data-generating processes it is clear that the sources of variability between the two groups arise potentially from the different cardinalities of the point clouds and variance within each cloud. We want to show that, while the metric d_E is susceptible to both kind of variability, d_μ^P, with an appropriately chosen measure μ, can mitigate the variability coming from higher cardinalities in the clouds sampled according to the first process. In particular, group 1 is expected to display a lower level of heterogeneity within each point cloud and thus those trees, for our purposes, should be regarded as more similar between each other compared to the other trees. The second group instead may not display a clear clustering structure, in fact, despite exhibiting a common level of heterogeneity, the different number of leaves and the different merging structure at the level of very heterogeneous leaves could prevent all such dendrograms to form a recognizable cluster - or, equivalently, could give birth to a cluster with higher dispersion.

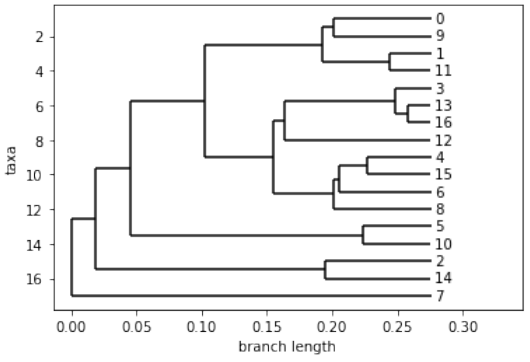
Following the pipeline presented in the main manuscript, we extract average linkage hierarchical clustering dendrograms from the set of point clouds and take pairwise distances both with d_E and d_μ^P. Examples of dendrograms belonging to the first and second groups can be found, respectively, in Figure 5.G.2b and Figure 5.G.2c. We select μ as in the main



(a) **Hierarchical Clustering with average linkage of the pairwise distance matrices respectively obtained from d_E, d_μ^P and d_μ^P but without the outlier represented by vertex 53 in the central dendrogram.**

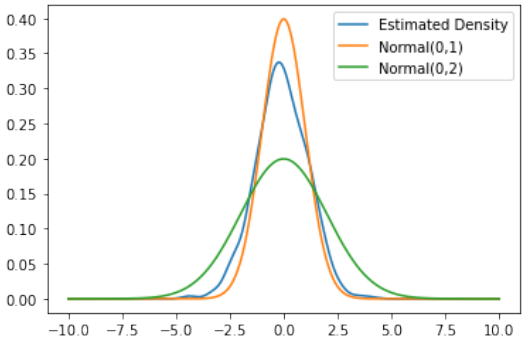


(b) **The outlier identified by the hierarchical clustering with average linkage of the matrix induced by d_μ^P.**

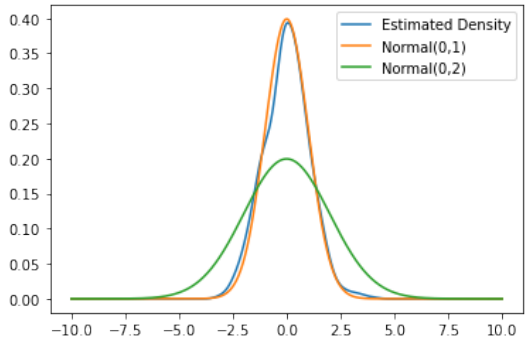


(c) **A randomly chosen dendrogram belonging to group 2. The difference in terms of heteogeneity between leaves and number of leaves, with the dendrogram in Figure 5.G.2b is evident.**

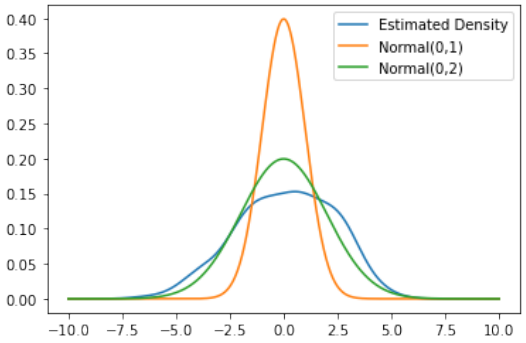
Figure 5.G.2: Cluster analysis of pairwise distance matrices obtained in the case study of Section 5.G.



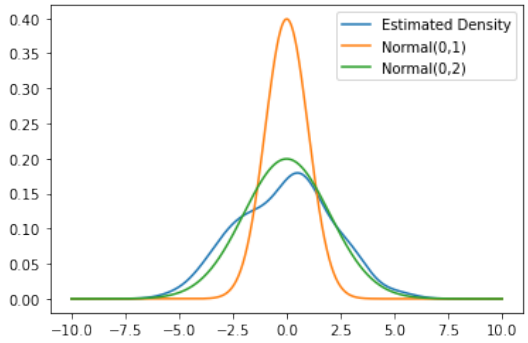
(a) **Estimated density of the first component of the data in the first cluster identified by d_μ^P, versus the densities generating the samples two groups.**



(b) **Estimated density of the second component of the data in the first cluster identified by d_μ^P, versus the densities generating the samples two groups.**



(c) **Estimated density of the first component of the data in the second cluster identified by d_μ^P, versus the densities generating the samples two groups.**



(d) **Estimated density of the second component of the data in the second cluster identified by d_μ^P, versus the densities generating the samples two groups.**

Figure 5.G.3: Densities estimated through the aggregation of the data collected in the two clusters identified by d_μ^P.

manuscript, Section 4.3.2, with the final choice being a Beta distribution with parameters $a = 2$, $b = 8$, as shown in Figure 5.G.1a. The two matrices are reported in Figure 5.G.1, with data being ordered according to the two groups: the first 50 point clouds belong to the first group, and the following 50 to the second. By visual inspection of Figure 5.G.1b and Figure 5.G.1c we can clearly see that d_E sees very little structure in the data, because of the two sources of variability (cardinality and variance) mixing up and preventing d_E to discriminate between group 1 and 2. Instead d_μ^P recognizes a clear and pronounced cluster made by point clouds from group 1 plus, potentially, some other point clouds belonging to group 2. The rest of the point clouds of group 2 still show some agglomerative structure, but less evident. The matrix in Figure 5.G.1d shows the pointwise differences between the values obtained with d_E and d_μ^P, highlighting how the different behaviour of the two metrics concentrates on the data belonging to the first group.

To get more insights into the clustering structures expressed by d_E and d_μ^P we extract the hierarchical clustering dendrograms with average linkage from the two matrices. These dendrograms are reported in Figure 5.G.2a. The leftmost tree is obtained from d_E and the central from d_μ^P. To better compare the clustering structures we remove from this last dendrogram the outlier ($v53$), obtaining the rightmost tree.

Visual inspection of the dendrograms in Figure 5.G.2a reveals a two-clusters structure

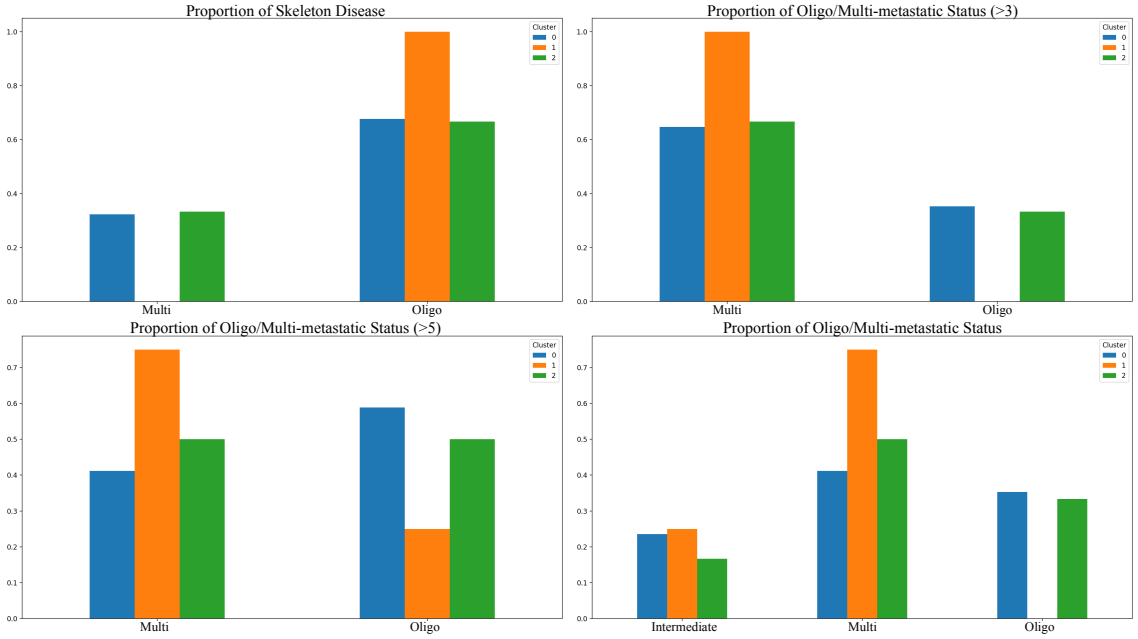


Figure 5.H.1: Results of clustering characterization: the proportion of skeleton disease and of the oligo/multi-metastatic status as devised by the two clinical cut-offs (3 and 5 lesions) are plotter per each of the three group.

in both metric spaces, with this structure being much more recognizable in the metric space induced by d_μ^P. In particular, the rightmost dendrogram shows a very cohesive and compact cluster, with very low internal variability, which is absent in the leftmost tree. The other cluster of the same tree, instead, displays a much higher level of variability.

Now we show that this clustering structure reflects the group structure that generated our data. We cut the rightmost tree to obtain two clusters. Then, for each cluster, we aggregate the points contained in the data of such cluster and we estimate the marginal densities from the obtained samples. The results of this estimation pipeline is showcased in Figure 5.G.3. We see that we retrieve the two distributions which we used to generate the components of the point clouds of the two groups.

This is precisely the behaviour we aimed to achieve: being insensitive to the cardinality of small homogeneous features, while still being sensitive to cardinalities and merging structures characterized by high heterogeneity.

5.H Additional plots for clustering interpretation

Figure 5.H.1 integrates the results, in terms of cluster characterization.

Data availability

The data supporting the findings of this study are available from Azienda Ospedaliero-Universitaria Pisana but restrictions apply to the availability of these data, which were used under license for the current study, and so are not publicly available.

6. A Graph-Matching Formulation of the Interleaving Distance between Merge Trees

Abstract

In this chapter we study the interleaving distance between merge trees from a combinatorial point of view. We use a particular type of matching between trees to obtain a novel formulation of the distance. With such formulation, we tackle the problem of approximating the interleaving distance by solving linear integer optimization problems in a recursive and dynamical fashion, obtaining lower and upper bounds. We implement the algorithms to obtain these bounds and compare the outputs with another approximation procedure presented by other authors. We believe that further research in this direction could lead to faster algorithms to compute the distance and novel theoretical developments on the topic.

6.1 Introduction

Topological data analysis (TDA) is a scientific field lying at the crossroads of topology and data analysis: objects like functions and point clouds are usually studied by means of - possibly multidimensional - complexes of homology groups (Hatcher, 2000) obtained with different pipelines. Homology groups - considered with coefficients in a field $\mathbb{K}$ - are vector spaces, whose dimension is determined by different kind of "holes" which can be found in a space and thus provide a rich characterization of the shape of the space the analyst is considering. Each statistical unit usually induces a whole family of topological spaces indexed on $\mathbb{R}^n$ which then, via homology, produces a family of vector spaces indexed on the same set - more precisely, a *functor* (Mac Lane, 1998) $(\mathbb{R}^n, \leq) \to \mathrm{Vect}_{\mathbb{K}}$. Such collections of vector spaces are called *persistence modules* (Chazal et al., 2008) - with *multidimensional* persistence modules being a special reference to the cases in which $n > 1$ - and describe the shape of the data by considering interpretable topological information at different resolutions. A topological summary or an invariant of a persistence module is a representation which usually maps persistence modules into a metric space - or even a vector space, so that some kind of analysis can be carried out. The most used topological summaries for 1-D persistence (i.e. $n = 1$) include persistence diagrams (Edelsbrunner et al., 2002), persistence landscapes (Bubenik, 2015), persistence images (Adams et al., 2017) and persistence silhouettes (Chazal et al., 2015). When dealing with 0-dimensional homology and 1-D persistence and with a space X which is path connected, however, another topological summary called merge tree (Morozov et al., 2013) can be employed. A merge tree is a tree-shaped topological summary which captures the evolution of the path connected components $\pi_0(X_t)$ of a filtration of topological spaces $\{X_t\}_{t\in\mathbb{R}}$, with $X_t \hookrightarrow X_{t'}$ if $t \leq t'$.

The main streams of research in TDA split into four different directions: 1) building new and interpretable pipelines to build persistence modules, tailored for different applications 2) obtaining - computationally accessible - invariants which can represent and summarize the information contained in the persistence modules 3) study the structure of

different spaces of invariants, with possible embeddings into metric or even linear spaces 4) study the *stability* of such embeddings with respect to noise/perturbations of the initial datum. One of the central ideas in TDA when studying stability properties, is the one of objects being ε-*interleaved*: when adding some kind of ε-noise to the data, the associated embedded summary "moves" by at most ε. Starting from the *bottleneck* distance for *persistence diagrams* (Edelsbrunner and Harer, 2008), this idea has been applied to in 1-D persistence modules (Chazal et al., 2008), multidimensional persistence modules (Lesnick, 2015), *merge trees* (Morozov et al., 2013) and more general situations (de Silva et al., 2017; Berkouk, 2021). In this chapter we focus on the problem of studying interleavings for merge trees.

Merge trees and persistence diagrams are not equivalent as topological summaries; that is, they do not represent the same information with merge trees being able to distinguish between situations which persistence diagrams - and all the other equivalent summaries - cannot (Kanari et al., 2020; Curry et al., 2022; Elkin and Kurlin, 2020; Smith and Kurlin, 2022). For this reason, in recent years, a lot of research sparkled on merge trees, mainly driven by the need of having a *stable* metric structure to compare such objects. Edit distances between Reeb graphs or merge trees have been proposed by Sridharamurthy et al. (2020); Di Fabio and Landi (2016); Bauer et al. (2020) and in Chapter 3, while other works focus on Wasserstein Distances (Pont et al., 2022), l_p distances (Cardona et al., 2021) and interleaving distances between merge trees or Reeb graphs (Morozov et al., 2013; Beketayev et al., 2014; De Silva et al., 2016; Gasparovic et al., 2019; Cardona et al., 2021).

Related Works

The problem of computing the interleaving distance between merge trees has already been approached by Agarwal et al. (2018) and Touli and Wang (2018) in an attempt to approximate the Gromov-Hausdorff (GH) distance when both metric spaces are metric trees. In this situation, in fact, up to carefully choosing the root for the metric trees, the two metrics are equivalent. Both problems - approximating the GH distance and computing the interleaving distance - have been shown to be NP-hard (Agarwal et al., 2018; Bjerkevik et al., 2020) and thus even obtaining feasible approximation algorithms for small trees is a daunting task.

In Agarwal et al. (2018) the authors provide an algorithm to approximate the interleaving distance between merge trees via binary search. They build a set which contains the solution of the optimization problem and then obtain a criterion to assess if certain values of the aforementioned set can be excluded from the set of solutions. If the length of the branches of both trees is big enough, one can carry out the binary search over all possible couples of vertices, with one vertex from the first tree and one from the second. If the trees instead posses branches which are too small, the decision procedure is coupled with a *trimming* step, to deal with such smaller edges. The algorithm returns an approximation of the interleaving distance, with the approximation factor depending on the ratio between the longest and the shortest edge in the tree and the number of vertices.

Touli and Wang (2018), instead, propose the first algorithm for the exact computation of the interleaving distance. Starting from a novel definition of the interleaving distance, they develop a faster alternative to exclude values from the same set of solutions considered by Agarwal et al. (2018). By filtering the points on the metric trees - which may not be vertices of the "combinatorial" trees - according to a) their heights b) a candidate optimal value δ, in an up-bottom fashion, points of the second tree are matched to sets of points of the first tree. Such matching means that all points of the first tree inside in the chosen collection can be potentially mapped in the point of the second tree via some function - viable for the interleaving distance. Each such matching (S, w) is then used to establish similar couples -

(set, point) - between the children of the involved points: the set of points of the first tree is to be chosen among the subsets of the children of S, and the point on the second tree among the children of w. If no such couples are found δ is discarded. The algorithm they propose to compute the interleaving distance has complexity $O(n^2 \log^3(n) 2^{2\tau} \tau^{\tau+2})$ where n is the sum of the vertices in the two merge trees, and τ is a parameter - depending on the input trees - which is defined by the authors. The key issue is that τ can be very big also for small-sized trees: in Touli and Wang (2018), Figure 2, the authors showcase a tree with 8 leaves and $\tau = 13$ ($2^{26}13^{15} \sim 10^{24}$).

Due to its computational complexity the algorithm by Touli and Wang (2018) has not been implemented by other authors working with merge trees (Curry et al., 2022), which instead have exploited another formulation of the same metric, provided by Gasparovic et al. (2019). This last formulation relies on a definition of the interleaving distance between merge trees which are endowed with a fixed set of labels on their vertices. The usual interleaving distance is then shown to be equivalent to choosing an appropriate set of labels - potentially upon adding some vertices - for the two given merge trees. This formulation, per se, does not provide computational advantages over the classical one, since evaluating all the possible labelings of a tree would still be unfeasible. However, (Curry et al., 2022) propose a labeling strategy which should provide *good* labels. Despite being computationally accessible even for very big trees, this approach has the downside of not providing methods for assessing the goodness of the approximation.

Contributions

As already mentioned, in the present chapter we take a perspective which differs from all the aforementioned works: instead of obtaining progressively sharper bounds for the distance by locally looking at differences between trees or instead of looking for optimal labelings, we aim at describing combinatorially a global matching between two merge trees whose *cost* returns the exact interleaving distance between them. Describing matchings between trees is often useful to formulate optimization problems via integer programming algorithms, which have proven to be a very useful tool to deal with the computational burden introduced by unlabeled or unordered trees (Hong et al., 2017). They can also provide very useful parametrizations, allowing for topological and geometric investigation of complex tree spaces - see Chapter 3. Indeed, we are able to produce a formulation of the interleaving distance originating a linear programming approach which gives lower and upper bounds for the interleaving distance between two trees. We use the upper bound to assess the approximation method proposed by Curry et al. (2022). The formulation we obtain in the present chapter is used by Chapter 3 to obtain some inequalities between the edit distance therein defined and the interleaving distance.

As discussed in Section 6.11, we also think that the results in the chapter can lead to yet another formulation of the distance, potentially opening up the door to novel theoretical developments.

Outline

The chapter is organized as follows. In Section 6.2 we introduce merge trees first in a combinatorial fashion and then as "continuous" metric spaces, recalling the definition of interleaving distance. In Section 6.3 we introduce a partial matching between trees called *coupling*. Relationships between these matchings and maps between trees are then presented in Section 6.4. Section 6.5 and Section 6.6 prove the equivalence between the usual definition of interleaving distance and the problem of finding optimal couplings. In the last part of the chapter we face the problem of approximating an optimal matching between trees: in Section 6.7 we prove two properties of the interleaving distance, which allow us to

write down the dynamical programming algorithm of Section 6.8, while Section 6.9 points out few facts about the error propagation in the previously presented algorithm. In Section 6.10 we test our approximation with the one proposed by other authors. Section 6.11 concludes the main body of the chapter with a brief discussion. Section 6.A contains some of the proofs of the results in the chapter, while the other ones are reported right after the statement they relate to, as they can help the reader in following the discussion.

6.2 Merge Trees and Interleaving Distance

We start by introducing merge trees as combinatorial objects along with their "continuous" counterpart.

6.2.1 Merge Trees as finite graphs

In accordance with other authors, we call *merge tree* a rooted tree with a suitable height function defined on its vertices. We now state the formal definition, introducing some pieces of notation, introduced in Chapter 2, which we will use to work with those objects throughout the chapter.

Definition 6.1. *A tree structure T is given by a set of vertices V_T and a set of edges $E_T \subset V_T \times V_T$ which form a connected rooted acyclic graph. We indicate the root of the tree with r_T. We say that T is finite if V_T is finite. The order of a vertex $v \in V_T$ is the number of edges which have that vertex as one of the extremes, and is called $ord_T(v)$. Any vertex with an edge connecting it to the root is its child and the root is its father: this is the first step of a recursion which defines the father and children relationship for all vertices in V_T. The vertices with no children are called leaves or taxa and are collected in the set L_T. The relation $child < father$ generates a partial order on V_T. The edges in E_T are identified in the form of ordered couples (a, b) with $a < b$. A subtree of a vertex v, called $sub_T(v)$, is the tree structure whose set of vertices is $\{x \in V_T | x \leq v\}$.*

Note that, identifying an edge (v, v') with its lower vertex v, gives a bijection between $V_T - \{r_T\}$ and E_T, that is $E_T \simeq V_T$ as sets. Given this bijection, we may use E_T to indicate the vertices $v \in V_T - \{r_T\}$, to simplify the notation.

Now we give the notion of *least common ancestor* between vertices.

Definition 6.2. *Given a set of vertices $A = \{a_1, \dots, a_n\} \subset V_T$, we define* $\mathrm{LCA}(a_1, \dots, a_n) = \min \bigcap_{i=1}^{n} \{v \in V_T | v \geq a_i\}$.

Now, to obtain a merge tree we add an increasing height function to a tree structure.

Definition 6.3. *A merge tree (T, f) is a finite tree structure T such that the root is of order > 1, coupled with a monotone increasing function $f : V_T \to \mathbb{R}$.*

Remark 6.4. *Definition 6.3 is slightly different from the definition of merge trees found in Gasparovic et al. (2019), Chapter 2 and other works. In particular, in the present work, we do not need to have a root at infinity and thus we remove it to avoid unpleasant technicalities. Similarly, the function coupled with the tree structure is usually referred to as h_T being an "height" function. To avoid ovearloading the notation, since we need to introduce many subscript and superscripts, we call these functions with more usual functional notations like f or g.*

Remark 6.5. *To avoid formal complications we make the following genericity assumption for any merge tree (T, f):*

(G) $f : V_T \to \mathbb{R}$ is injective.

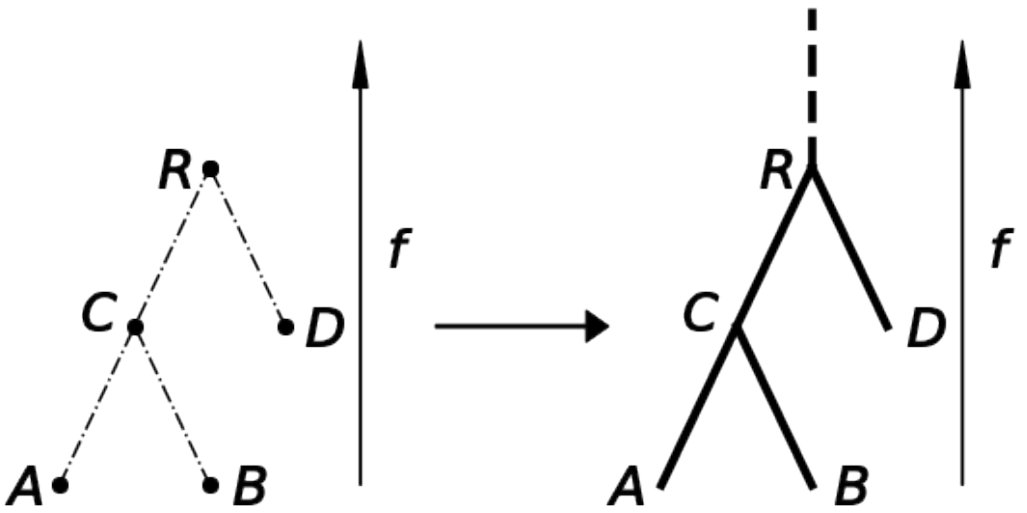


Figure 6.2.1: A merge tree (left) with its associated metric merge tree (right).

Before proceeding we need one last graph-related definition which we use to denote some particular kinds of paths on the tree structure of a merge tree.

Definition 6.6. *Given a merge tree T, a sequence of edges is an ordered sequence of adjacent edges $\{e_1, \dots, e_n\}$. Which means that we have $e_1 < \dots < e_n$, according to the order induced by the bijection $E_T \leftrightarrow V_T - \{r_T\}$ and that e_i and e_{i+1} share a vertex. We will use the notation $[v, v']$ to indicate a sequence of edges which starts in the vertex v and ends with the vertex v', with $v < v'$. Note that, if we write down $[v, v']$ using $E_T \simeq V_T - \{r_T\}$, v is included in the sequence, while v' is the first excluded vertex.*

In the left column of Figure 6.2.1 the reader can find an example of a merge tree, which can be used to get familiar with the definitions just given. For instance one can check that, using the notation of the figure, $R = \mathrm{LCA}(A, B, D)$ and $[A, C, R]$ is an ordered sequence of edges, while $[C, R, D]$ is not.

6.2.2 Metric Merge Trees

Now we consider the continuous version of a merge tree, intuitively obtained by considering all the points in the edges of a merge tree, as points of the tree itself. For a visual intuition of the following ideas, the reader may refer to Figure 6.2.1.

Definition 6.7. *Given a merge tree T, we obtain the associated metric merge tree as follows:*

$$\mathbf{T} = [f(r_T), +\infty) \coprod_{(x,x') \in E_T} [f(x), f(x')] / \sim$$

where, for every $v \in V_T$, $f(v) \sim f(v')$ if $v = v'$. We refer to the points in $\mathbf{T}$ with the same notation we use for vertices in v_T: $x \in \mathbf{T}$; with $f(x)$ we indicate the height values of x.

For every point $x \in V_T$, we can identify x with the point $f(x) \in [f(x), f(x')]$, with $(x, x') \in E_T$, or, equivalently, with $f(x) \in [f(x'), f(x)]$, if $(x', x) \in E_T$. This induces a well defined map $V_T \hookrightarrow \mathbf{T}$. Thus, given $v \in V_T$, with an abuse of notation, we can consider $v \in \mathbf{T}$. Every point in $\mathbf{T} - V_T$ belongs to one and only one interval of the form $[f(x), f(x')]$. Thus we can induce a partial order relationship on $\mathbf{T}$ by ordering points first with the partial order in E_T and then with the increasing internal order of

$[f(x), f(x')]$. *Thus we can explicitly write down the shortest path metric in* $\mathbf{T}$*:* $d(x, x') = f(\mathrm{LCA}(x, x')) - f(x) + f(\mathrm{LCA}(x, x')) - f(x')$.

Remark 6.8. *What we call "metric merge tree" is closely related to* display posets *of persistent sets, as defined in Curry et al. (2022) but to introduce that notion we need to give other technical definitions which we do not need in this work.*

Lastly, for each metric tree $\mathbf{T}$, we have a family of continuous maps, $s_T^k : \mathbf{T} \to \mathbf{T}$, with $k \geq 0$, called *structural maps*, defined as follows: $s_T^k(x) = x'$ with x' being the only point in $\mathbf{T}$ such that $x' \geq x$ and $f(x') = f(x) + k$.

6.2.3 Interleaving Distance between Merge Trees

Now we recall the main facts about the interleaving distance between merge trees (Beketayev et al., 2014).

Definition 6.9 (Morozov et al. (2013))**.** *Two continuous maps* $\alpha : \mathbf{T} \to \mathbf{G}$ *and* $\beta : \mathbf{G} \to \mathbf{T}$ *between two metric merge trees* $\mathbf{T}$ *and* $\mathbf{G}$*, are* ε*-compatible, with* $\varepsilon \geq 0$*, if:*

(I1) $g(\alpha(x)) = f(x) + \varepsilon$ *for all* $x \in \mathbf{T}$ *and* $f(\beta(y)) = g(y) + \varepsilon$ *for all* $y \in \mathbf{G}$*;*

(I2) $\alpha\beta = s_G^{2\varepsilon}$ *and* $\beta\alpha = s_T^{2\varepsilon}$*.*

The interleaving distance between $\mathbf{T}$ *and* $\mathbf{G}$ *is then:* $d_I(\mathbf{T}, \mathbf{G}) = \inf\{\varepsilon \mid$ *there* ε*-compatible maps*$\}$.

For an example of α and β continuous map satisfying *(I1)* see Figure 6.2.2. Those maps satisfy also *(I2)* as shown by Figure 6.2.3.

The work of Touli and Wang (2018) shows that the existence of α and β is in fact equivalent to the existence of a single map α with some additional properties which are stated in the next definition.

Definition 6.10 (Touli and Wang (2018))**.** *Given two metric merge trees* $\mathbf{T}$ *and* $\mathbf{G}$*, a* ε*-good map* $\alpha : \mathbf{T} \to \mathbf{G}$ *is a continuous map such that:*

(P1) $g(\alpha(x)) = f(x) + \varepsilon$ *for all* $x \in \mathbf{T}$

(P2) if $\alpha(x) < \alpha(x')$ *then* $s_T^{2\varepsilon}(x) < s_T^{2\varepsilon}(x')$

(P3) if $y \in \mathbf{G} - \alpha(\mathbf{T})$*, then, given* $w = \min\{y' > y \big| y' \in \alpha(\mathbf{T})\}$*, we have* $g(w) - g(y) \leq 2\varepsilon$*.*

As anticipated, Touli and Wang (2018) prove that two merge trees are ε-interleaved if and only if there is a ε-good map between them.

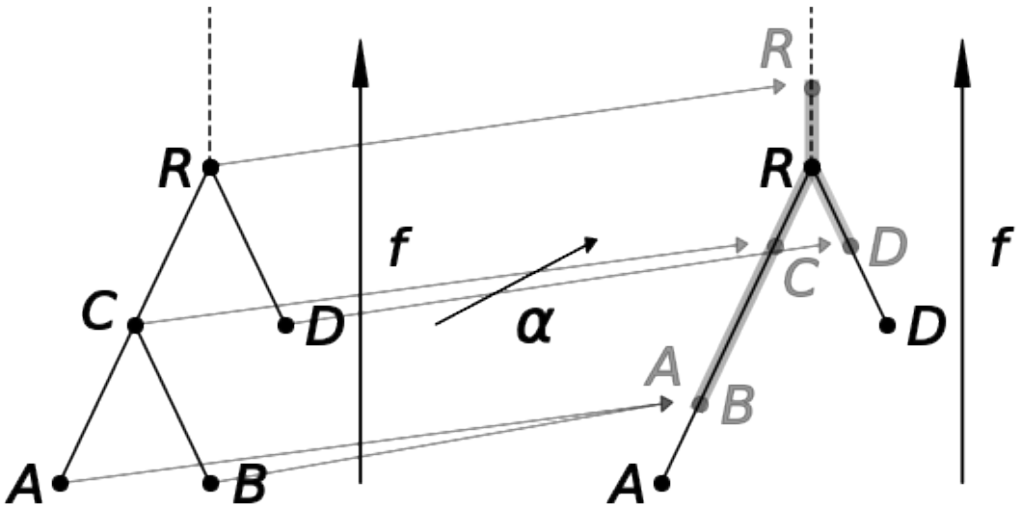
6.3 Couplings

Now we start our combinatorial investigation. Given two merge trees T and G, we would like to match their vertices and compute the interleaving distance relying only on such matches. To match the two graphs we will use a set $C \subset V_T \times V_G$ which will tell us which vertex is coupled with which. Clearly this set C must satisfy some constraints which we now introduce.

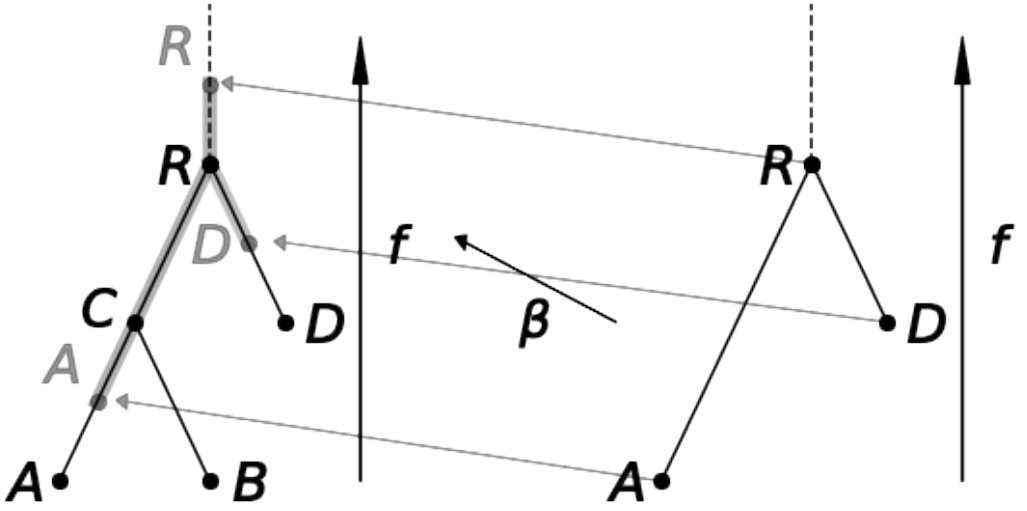
Given a finite set A, we indicate with $\#A$ its cardinality; consider now $C \subset V_T \times V_G$ and the projection $\pi_T : V_T \times V_G \to V_T$, we define the multivalued map $\Lambda_C^T : V_T \to V_T$ as follows:

$$\Lambda_C^T(v) = \begin{cases} \max_{v' < v} \pi_T(C) & \text{if } \#\{v' \in V_T \mid v' < v \text{ and } v' \in \pi_T(C)\} > 0 \\ \emptyset & \text{otherwise.} \end{cases}$$

Since V_T and V_G are posets, we can introduce a partial order relationship on any $C \subset V_T \times V_G$, having $(a, b) < (c, d)$ if and only if $a < c$ and $b < d$.



(a) A graphical representation of a continuous function $\alpha : \mathbf{T} \to \mathbf{G}$ between metric merge trees satisfying condition *(I1)* in Definition 6.9.



(b) A graphical representation of a continuous function $\beta : \mathbf{G} \to \mathbf{T}$ between metric merge trees satisfying condition *(I1)* in Definition 6.9.

Figure 6.2.2: An example of maps α and β giving the ε-interleaving of two metric merge trees.

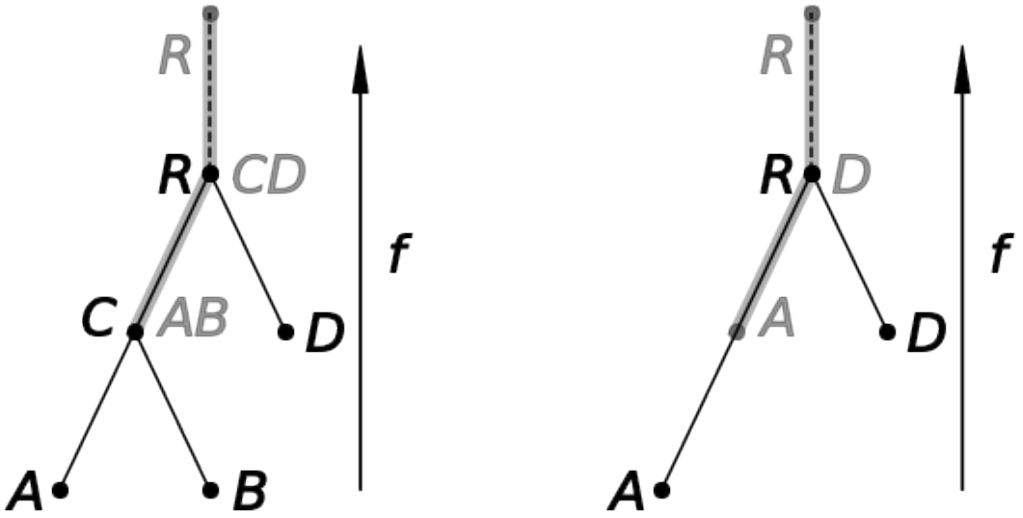


Figure 6.2.3: A representation of the images of the maps $\beta\alpha$ (left) and $\alpha\beta$ (right), with α and β being the maps in Figure 6.2.2.

Definition 6.11. *A coupling between two merge trees (T, f) and (G, g) is a set $C \subset V_T \times V_G$ such that:*

(C1) $\#\max C = 1$ *or, equivalently,* $\#\max \pi_T(C) = \#\max \pi_G(C) = 1$

(C2) the projection $\pi_T : C \to V_T$ is injective (the same for G)

(C3) given (a, b) and (c, d) in C, then $a < c$ if and only if $b < d$

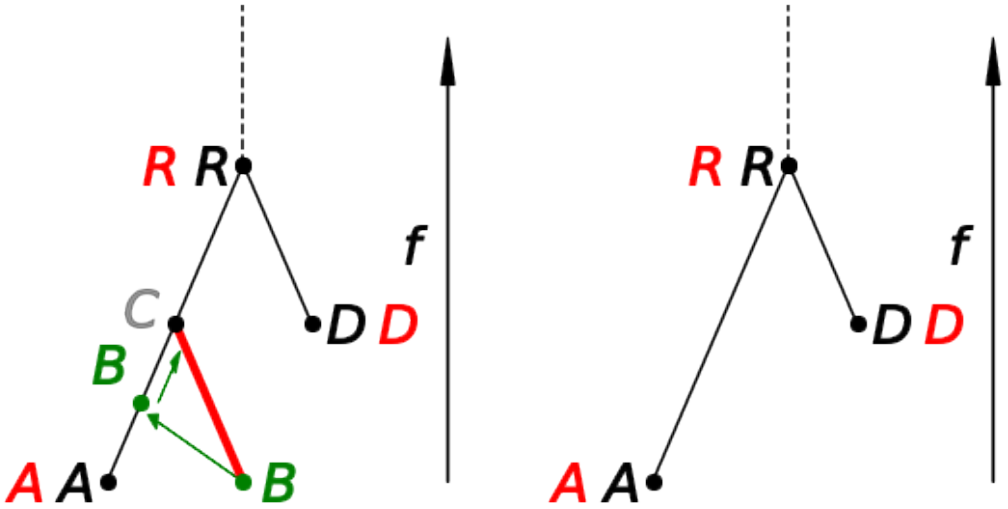
(C4) $a \in \pi_T(C)$ *implies* $\#\Lambda_C^T(a) \neq 1$ *(the same for G).*

The set of couplings between T and G is called $\mathcal{C}(T, G)$.

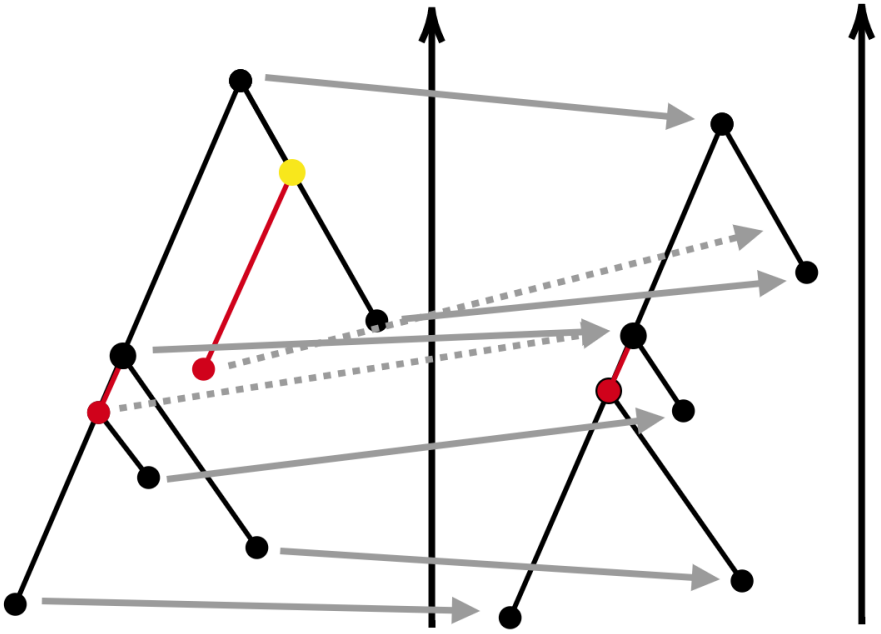
We invite the reader to follow the remaining of the section looking at Figure 6.3.1 to help understanding the comments we present on properties (C1)-(C4) and to visualize the pieces of notation we need to establish.

We collect the set of points $v \in E_T$ such that $\#\Lambda_C^T(v) = 1$ in a set we call U_C^T. Instead, the points such that $v \notin \pi_T(C)$ and $\#\Lambda_C^T(v) \neq 1$ are collected in a set D_C^T. Note that the sets $\pi_T(C), U_C^T$ and D_C^T are a partition of V_T. A couple $(v, w) \in C$ means that we match a vertex $v \in V_T$ with a vertex $w \in V_G$. It is clear from the definition that there can be vertices left out from this matching. We regard this vertices as unnecessary for the coupling C: when we will induce maps between metric merge trees starting from couplings, the position of these vertices inside the metric merge tree $\mathbf{G}$ will be completely induced by other vertices in V_T. Among the vertices not appearing in the couples of C we distinguish between two situations: $v \in D_C^T$ can be informally thought of $(v, father(v))$ needing to be removed the image of $\beta(\alpha(\mathbf{T}))$, with α and β as in Definition 6.9. For this reason we say that the vertices in D_C^T and D_C^G are *deleted* by the coupling C. Instead, the vertices $v \in U_C^T$ are vertices which are *unused*, ignored by the coupling, and will be of no importance in the computation of the distance.

In this context, property (C1) is asking that the vertices of T and of G coupled by C form a tree, while (C2) is asking that each vertex is paired with at most one other vertex of the other merge tree. Note the maps $\alpha : \mathbf{T} \to \mathbf{G}$ in Definition 6.9 are not forced to be injective, but, as we will prove in later sections, for our purposes it is enough



(a) Call T the merge tree on the left and G the one on the right. The couples of adjacent letters - red and black - indicate a coupling. The vertex C in grey represent the only vertex with $\#\Lambda(C) = 1$ and belongs to U^T; instead the leaf B belongs to D^T and clearly $\#\Lambda(B) = 0$. The green arrows suggest the path of the two-step deletion of B - note that $\varphi(B) = C$ and so $\eta(B) = A$.



(b) The grey arrows between the two merge trees indicate a mapping C. The dashed gray arrow, instead, indicate where the lower vertex of the deleted edges - the red ones - are sent by the associated function α_C - introduced in Section 6.4. The lower deleted edge - of the leftmost tree - is an internal edge and its lower vertex v is deleted with $\#\Lambda(v) > 1$ and thus it is sent to $\chi(v)$. The higher deleted edge of the leftmost tree is instead deleted with a two step deletion as in Figure 6.3.1a. The yellow vertex is unused.

Figure 6.3.1: Two examples of couplings, displaying all possible cases of unused vertices and deletions.

to couple points just one time. Condition (C3) is asking that the coupling respects the tree structures of T and G; in particular this implies that $(\max \pi_T(C), \max \pi_G(C)) \in C$. Lastly, due to condition (C4), we avoid coupling vertices which have only one element below them which is coupled.

Given a coupling, we want to associate to a vertex a cost which indicates how much the coupling moves that vertex. In order to do so, we define the following functions:

- define $\varphi_T^C : V_T \to V_T$ so that $\varphi_T^C(x) = \min\{v \in V_T \,|\, v > x \text{ and } \#\Lambda(v) \neq 0\}$. Note that since the set $\{v \in V_T | v > x\}$ is totally ordered, $\varphi_T^C(x)$ is well defined;
- similarly, define $\delta_T^C : V_T \to V_T$, defined as $\delta_T^C(x) = \min\{v \in V_T \big| v \geq x \text{ and } v \in \pi_T(C)\}$;
- define $\chi_T^C : V_T \to V_G$, defined as $\chi_T^C(x) = \mathrm{LCA}(\{\pi_G((v,w)) \,|\, v \in \Lambda_T(x)\})$;
- set $\gamma_T^C : V_T - D_C^T \to V_G$ to be:

$$\gamma_T^C(x) = \begin{cases} \arg\min\{g(w) \,|\, (v,w) \in C, v < x\} & \text{if } \#\{g(w) \,|\, (v,w) \in C, v < x\} > 0 \\ \emptyset & \text{otherwise.} \end{cases}$$

  Note that if $\#\{g(w) \,|\, (v,w) \in C, v < x\} > 0$, by *(G)*, $\gamma_T^C(x)$ is uniquely defined. Note that $\gamma_T^C(\varphi_T^C(x))$ is well defined for any $v \in V_T$.

Remark 6.12. *When clear from the context, to lighten the notation, we might omit the subscripts and superscripts. For instance if we fix $C \in \mathcal{C}(T,G)$ and $x \in V_T$, we refer to $\gamma_T^C(\varphi_T^C(x))$ as $\gamma(\varphi(x))$. And similarly we use $x \in U$, $x \in D$ respectively for $x \in U_C^T$ and $x \in D_C^T$.*

Given $x \in V_T$, in what follows we will encounter the following objects:

- $\eta(x) := \gamma(\varphi(x))$ is the vertex in V_G obtained in this way: starting from x we go towards r_T until we meet the first point of V_T which has at least one vertex in its subtree which is not deleted; we consider the subtree of such vertex and take all the vertices of V_G which are coupled with elements in this subtree (such set is not empty); lastly we take the lowest vertex among this set;
- given x such that $x \in D$ and $\#\Lambda(x) > 1$, we often consider $\delta(x)$; note that for every v such that $x \leq v < \delta(x)$ then $v \in D$ and $\#\Lambda(v) > 1$. Thus $[x, \delta(x)]$ contains only deleted vertices, up to $\delta(x)$;
- similarly, for x such that $x \in D$ and $\#\Lambda(x) > 1$, we take $\chi(x)$: that is the the lowest point in G where x can be sent compatible with C and tree structures of T and G. Thus, if $(x,y) \in C$, $y \geq \chi(x)$.

6.4 Couplings, Maps and Costs

In this section we establish some correspondence between couplings and maps $\alpha : \mathbf{T} \to \mathbf{G}$, giving also the definition of the cost of a coupling. As a first step, given a coupling C we induce two maps $\alpha_C : V_T - U_C^T \to \mathbf{G}$ and $\beta_C : V_G - U_C^G \to \mathbf{T}$, following these rules:

1. if $(x,y) \in C$, then $\alpha_C(x) := y$;
2. $x \in D_C^T$ with $\#\Lambda(x) = 0$, then $\alpha_C(x) := s_T^k(\eta(x))$ with k being:

- $k = f(x) + \frac{1}{2}(f(\varphi(x)) - f(x)) - g(\eta(x))$ if $g(\eta(x)) \leq f(x) + \frac{1}{2}(f(\varphi(x)) - f(x))$;
- $k = 0$ otherwise.

Where the idea is that we want to send x above some coupled vertex changing its height "as little as possible". Bear in mind that the sequence $[x, \varphi(x)]$ should not appear in the image of $\beta\alpha$;

3. if instead $x \in D_C^T$ with $\#\Lambda(x) > 1$, then $\alpha_C(x) := \chi(x)$.

The map β_C is obtained in the same way, exchanging the roles of (T, f) and (G, g). Figure 6.5.1b shows an example in which β_C is obtained starting from a coupling.

As a first result we obtain that the maps we induce respect the trees structures of the respective merge trees.

Proposition 6.13. *Consider $x, x' \in V_T - U_T$: if $x < x'$ then $\alpha_C(x) \leq \alpha_C(x')$.*

Now we need to extend α_C to a continuous function between $\mathbf{T}$ and $\mathbf{G}$. To formally extend the map, we need the following lemma.

Lemma 6.14. *Any $x \in \mathbf{T}$, $x \notin \pi_T(C) \bigcup D$, is contained in a sequence of edges $[l_C(x), u_C(x)]$ such that $l_C(x) \in \max\{v \leq x \,|\, v \in \pi_T(C) \cup D\}$ and $u_C(x)$ is either $u_C(x) = +\infty$ or $u_C(x) = \min\{v \geq x \,|\, v \in \pi_T(C) \cup D\}$.*

Moreover, $l_C(x)$ is unique if $\{v \in U \,|\, l_C(x) \leq v \leq x\} = \emptyset$ and, if $\{v \in U \,|\, l_C(x) \leq v \leq x\} \neq \emptyset$, there exist one and only one $l_C(x)$ such that $l_C(x) \notin D$.

Now we can obtain a continuous map $\alpha_C : \mathbf{T} \to \mathbf{G}$.

Proposition 6.15. *We can extend α_C to a continuous map between $\mathbf{T}$ and $\mathbf{G}$. This map, with an abuse of notation, is still called α_C.*

Proof. We define $\alpha_C(x)$ for $x \notin \pi_T(C) \bigcup D_C^T$. By Lemma 6.14 we have $x \in [l(x), u(x)]$, with α_C being defined for $l(x)$ and $u(x)$. Whenever $l(x)$ is not unique, we take the unique $l(x) \in \pi_T(C)$. Clearly we have $f(l(x)) < f(x) < f(u(x))$. Thus, if $u(x) < +\infty$, there is a unique $\lambda \in [0, 1]$ such that $f(x) = \lambda f(l(x)) + (1 - \lambda) f(u(x))$. Having fixed such λ, we define $\alpha_C(x)$ to be the unique point in $[\alpha_C(l(x)), \alpha_C(u(x))]$ - which is a sequence of edges thanks to Proposition 6.13 - with height $\lambda g(\alpha_C(l(x))) + (1-\lambda) g(\alpha_C(u(x)))$. If $u(x) = +\infty$ then $x > v$ with $v = \max \pi_T(C)$. Then we set $\alpha_C(x) = y$ such that $y \geq \alpha_C(v)$ and $g(y) = g(\alpha_C(v)) + f(x) - f(v)$. Note that, for x, x' such that $u(x), u(x') = +\infty$, α_C always preserves distances and $g(\alpha_C(x)) - f(x) = g(\alpha_C(v)) - f(v)$. Thus, on such points it is a continuous function.

Consider now a converging sequence $x_n \to x$ in $\mathbf{T}$. We know that definitively $\{x_n\}_{n \in \mathbb{N}}$ is contained in one or more edges containing x. Thus we can obtain a finite set of converging subsequences by intersecting $\{x_n\}$ with such edges. With an abuse of notation from now on we use $\{x_n\}_{n \in \mathbb{N}}$ to indicate any such sequence. Each of those edges is contained in a unique sequence of edges of the form $[l(x'), u(x')]$, for some x' - up to, eventually, taking $l(x') \notin D$. Thus $\{x_n\} \subset [l(x'), u(x')]$ induces a unique sequence $\{\lambda_n\} \subset [0, 1]$ such that $f(x_n) = \lambda_n f(l(x')) + (1 - \lambda_n) f(u(x'))$ and $\lambda_n \to \lambda_x$, with $f(x) = \lambda_x f(l(x')) + (1 - \lambda_x) f(u(x'))$. By Lemma 6.14 $\alpha_C(x) \in [\alpha_C(l(x')), \alpha_C(u(x'))]$. Moreover, by construction, $g(\alpha_C(x_n)) \to \lambda_x g(\alpha_C(l(x'))) + (1 - \lambda_x) g(\alpha_C(u(x'))) = g(\alpha_C(x))$. Thus α_C is continuous. □

In order to start relating maps induced by couplings and ε-good maps, we define the *cost* of the coupling C, which is given in terms of how much α_C moves the points of $\mathbf{T}$.

Definition 6.16. *Given $C \in \mathcal{C}(T,G)$ and $x \in \mathbf{T}$, we define $cost_C(x) = |\, g(\alpha_C(x)) - f(x) \,|$. Coherently we define $\| C \|_\infty = \max\{\| g \circ \alpha_C - f \circ Id_{\mathbf{T}} \|_\infty, \| f \circ \beta_C - g \circ Id_{\mathbf{G}} \|_\infty\}$.*

Note that, given $C \in \mathcal{C}(T,G)$ and $x \in \mathbf{T}$, we have one of the following possibilities:

- if $(x,y) \in C$, $cost_C(x) = |f(x) - g(y)|$;
- if $x \notin \pi_T(C) \bigcup D$, $cost_C(x) \leq \max\{cost_C(l(x)), cost_C(u(x))\}$; in fact:

$$
\begin{aligned}
&|\, g(\alpha_C(x)) - f(x) \,| = \\
&|\, \lambda g(\alpha_C(l(x))) - \lambda f(l(x)) + (1-\lambda) g(\alpha_C(u(x))) - (1-\lambda) f(u(x)) \leq \\
&\lambda cost_C(l(x)) + (1-\lambda) cost_C(u(x))
\end{aligned}
$$

- we are left with the case $x \in D$. We have two different scenarios:
  - if $\#\Lambda(x) = 0$, then $cost_C(x) = \max\{(f(\varphi(x)) - f(x))/2, g(\eta(x)) - f(x)\}$. We point out that $(f(\varphi(x)) - f(x))/2$ is the cost of deleting the path $[x, \varphi(x)]$ in two steps: we halve the distance between x and $\varphi(x)$ with α and then with β we have $f(\beta\alpha(x)) = f(\varphi(x))$. This can happen if below w (with $(\delta(x), w) \in C$) there is "room" to send x at height $f(x) + (f(\varphi(x)) - f(x))/2$; if instead $\eta(x)$ is higher than $f(x) + (f(\varphi(x)) - f(x))/2$ we simply send x to $\eta(x)$;
  - if $\#\Lambda(x) > 1$, we have $cost_C(x) = |\, f(x) - g(\chi(x)) \,|$.

As a consequence we point out two facts:

1. for every $\alpha : \mathbf{T} \to \mathbf{G}$ such that, for every $x \in \pi_T(C) \bigcup D$, $\alpha(x) = \alpha_C(x)$ we have:
$$\| g \circ \alpha_C - f \circ \mathrm{Id}_{\mathbf{T}} \|_\infty \leq \| g \circ \alpha - f \circ \mathrm{Id}_{\mathbf{T}} \|_\infty$$
2. if we fix some total ordering of the elements in $V_T \coprod V_G$, so that we can write $V_T \coprod V_{T'} = (a_1, \ldots, a_n)$, then $\| C \|_\infty = \max(cost(a_1), \ldots, cost(a_n))$.

6.5 From Couplings to ε-good Maps

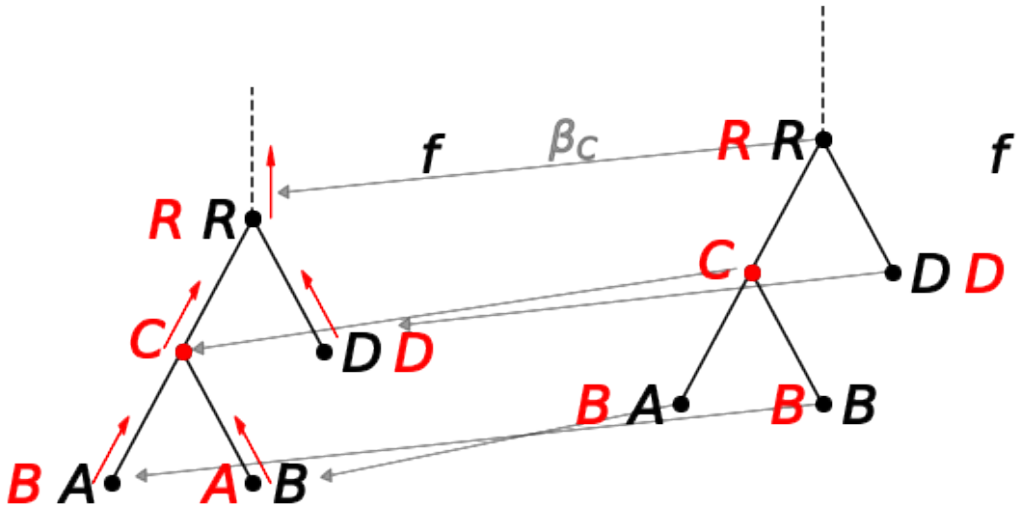
In this section we prove the first fundamental interaction between couplings and interleaving distance between trees, with our final goal being to replace the problem of finding optimal maps with the combinatorial problem of obtaining optimal couplings. Figure 6.5.1 summarizes all the steps of the procedure which is formally addressed in this section.

Given a coupling C, for any $\varepsilon \geq ||C||_\infty$ we build a map $\alpha_C^\varepsilon : \mathbf{T} \to \mathbf{G}$: $\alpha_C^\varepsilon(x) = s_G^{k_x}(\alpha_C(x))$ with $k_x = f(x) + \varepsilon - g(\alpha_C(x))$ depending on x. Now we check that, since $\varepsilon \geq cost(C)$, we always have $k_x \geq 0$. In fact it is enough to check this property on V_T:

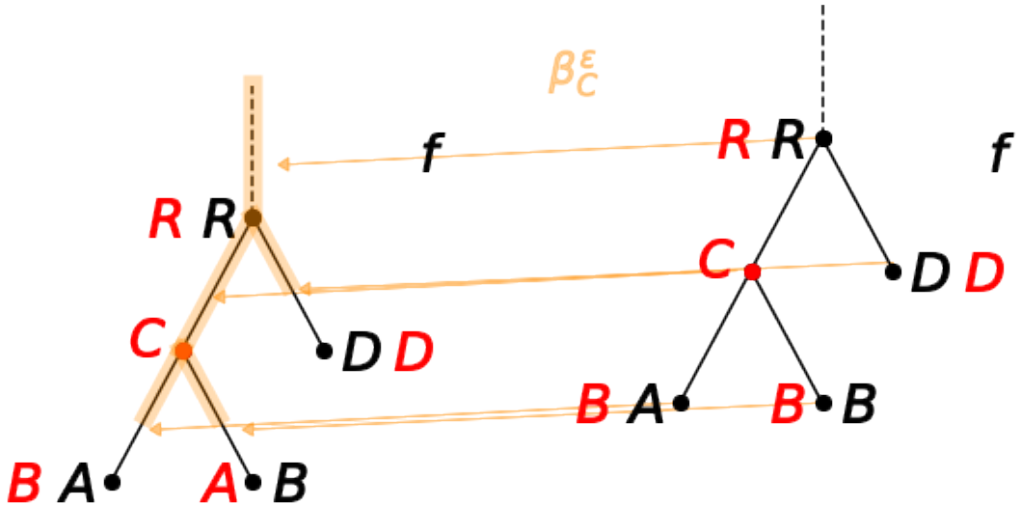
1. for $(x,y) \in C$ we have $\varepsilon \geq |f(x) - g(y)|$ and thus $f(x) + \varepsilon - g(y) \geq 0$;
2. for $x \in D_C^T$ and $\#\Lambda(x) = 0$, we have $\varepsilon \geq \max\{(f(\varphi(x)) - f(x))/2, g(\eta(x)) - f(x)\}$. Thus $f(x) + \varepsilon \geq g(\eta(x))$ and so $k \geq 0$. Similarly if $\#\Lambda(x) > 1$, we have $w = \chi(x)$ and $k = f(x) + \varepsilon - g(w) \geq 0$.

We want to prove that α_C^ε is a ε-good map. Note that, by construction $g(\alpha_C^\varepsilon(x)) = g(\alpha_C(x)) + f(x) + \varepsilon - g(\alpha_C(x)) = f(x) + \varepsilon$.

Remark 6.17. *The map α_C^ε is such that, given $x, x' \in \mathbf{T}$ with $x < x'$, we have $\alpha_C^\varepsilon(x) < \alpha_C^\varepsilon(x')$.*



(a)



(b)

Figure 6.5.1: With the same notation/colors used in Figure 6.3.1a and display the machinery defined in Section 6.5. Start from the subfigure (a). The gray arrows represent the map $\beta_C : V_G \to V_T$, which then is naturally extended to the metric trees. The image of such map is then shifted upwards using the structure map s^k_T as in Section 6.5, to obtain β^{ε}_C, with the shift being indicated by the upwards red arrows. In subfigure (b), the orange arrows - and the shaded orange portion of $\mathbf{T}$ - represent the ε-good map β^{ε}_C obtained with the composition of β_C with the upward shift.

As a first step we obtain that α_C^ε satisfies some necessary conditions in order to be ε-good: it is in fact continuous and moves point "upwards" by ε.

Proposition 6.18. *The map α_C^ε is a continuous map between $\mathbf{T}$ and $\mathbf{G}$ such that for every $x \in \mathbf{T}$ we have $g(\alpha_C^\varepsilon(x)) = f(x) + \varepsilon$.*

Before proving the main result of this section we need a short lemma.

Lemma 6.19. *Let $(v, w), (v', w') \in C$ and let $x = \mathrm{LCA}(v, v')$, $y = \mathrm{LCA}(w, w')$. Then $| f(x) - g(y) | \leq \varepsilon$.*

At this point we are ready to prove the remaining properties which make α_C^ε an ε-good map.

Theorem 6.20. *The map α_C^ε is an ε-good map.*

6.6 From ε-Good Maps to Couplings

This time we start from an ε-good map $\alpha : \mathbf{T} \to \mathbf{G}$ and our aim is to induce a C with $\| C \|_\infty \leq \varepsilon$.

Given a metric merge tree $\mathbf{T}$ we define the following map: $L : \mathbf{T} \to V_T$ given by $L(x) = \max\{v \in V_T \,|\, v \leq x\}$. Note that if $x \in V_T$ then $L(x) = x$, otherwise x is an internal vertex of one edge (a, b), with possibly $b = +\infty$, and $L(x) = a$. With this notation we introduce the following maps, which are analogous to $\alpha^\downarrow$ and $\beta^\downarrow$ defined in Agarwal et al. (2018) and to other maps defined in the proof of Theorem 4.22 in Chapter 4:

$$\begin{aligned} \phi :& V_T \to V_G && (6.1)\\ & v \mapsto L(\alpha(v)) && (6.2) \end{aligned}$$

$$\begin{aligned} \psi :& V_G \to V_T && (6.3)\\ & w \mapsto L(\beta(v)) && (6.4) \end{aligned}$$

Note that we always have $g(\phi(v)) \leq g(\alpha(v)) \leq f(v) + \varepsilon$. These maps will be keys in the proof of the upcoming theorem since they will help us in closing the gap between the continuous formulation of the interleaving distance and the discrete matching of merge trees via couplings. The reader should refer to Figure 6.6.1 for a visual example of the above definitions.

We prove a corollary which characterizes the maps we just defined, and will be used in what follows.

Corollary 6.21. *Let $v, v' \in V_T$; if $v < v'$ then $\phi(v) \leq \phi(v')$.*

Clearly this result implies that, in the setting of the corollary, $\psi(\phi(v)) \leq \psi(\phi(v'))$.

Now we prove the main result of this section.

Theorem 6.22. *Given α (and β) ε-good maps between $\mathbf{T}$ and $\mathbf{G}$, then there is a coupling C between T and G such that $\| C \|_\infty \leq \varepsilon$.*

Putting together Theorem 6.20 and Theorem 6.22, we see that two merge trees are ε-interleaved if and only if there is a coupling C between the tree such that $\| C \|_\infty = \varepsilon$. Thus, computing the interleaving distance amounts to finding a minimal-cost coupling between two merge trees. As a byproduct of this result, we obtain another proof that the interleaving distance between metric merge tree can be found as a minimum and not just as an infimum as in Definition 6.9, for the set of available couplings is finite.

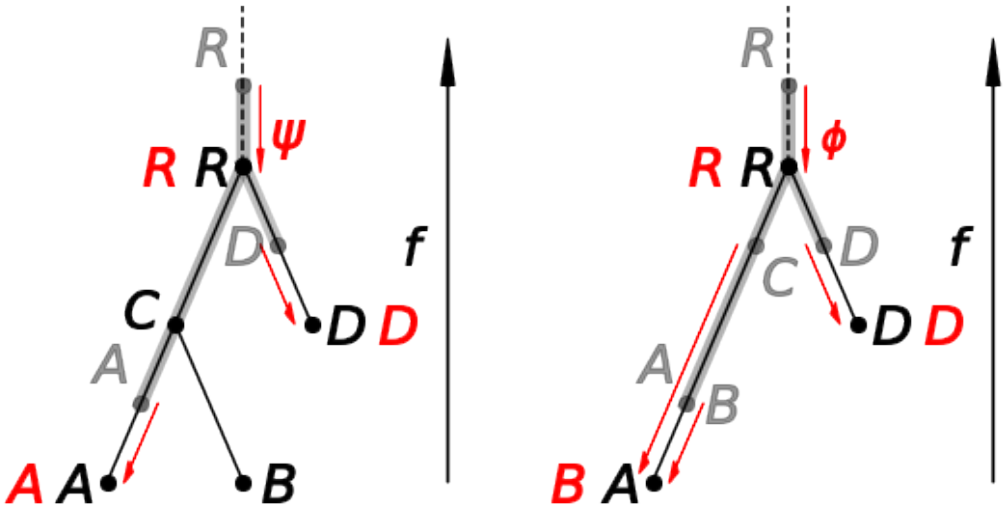


Figure 6.6.1: Given the two maps of Figure 6.2.2, the shaded gray represent the image of those maps, the red arrows give the maps ψ (left) and ϕ (right), and the red letters next to the black ones indicate two possible couplings (one in leftmost tree and one in the rightmost tree) satisfying Theorem 6.22. These couplings are compatible with the procedure outlined in the proof of Theorem 6.22 upon perturbing the leftmost tree to meet the generality condition (G).

6.7 Properties of Couplings

In this section we present two different properties of the interleaving distance which are useful for the approximation scheme we propose in Section 6.8.

6.7.1 Decomposition Properties

In the following we exploit the equivalent formulation of the interleaving distance via couplings to prove that it can be approximated via the solution of in smaller independent subproblems which are then aggregated to solve the global one.

The main result of the section involves couplings with some strong properties, which we call *special* and collect under the following notation:

$$\mathcal{C}^o(T,G) := \{C \in \mathcal{C}(T,G) \,|\, \arg\min_{v<\max \pi_T(C)} f(v) \in \pi_T(C) \text{ and } \arg\min_{w<\max \pi_G(C)} g(w) \in \pi_G(C)\}.$$

Before proceeding, we need few pieces of notation used to lighten the dissertation and one last technical definition. Given $x \in V_T$ and $y \in V_G$ we define $T_x = sub_T(x)$ and $G_y = sub_G(y)$. Moreover $\mathcal{C}_R(T,G) := \{C \in \mathcal{C}(T,G) \,|\, (r_T, r_G) \in C\}$. Similarly, we have $\mathcal{C}^o_R(T,G) := \mathcal{C}_R(T,G) \bigcap \mathcal{C}^o(T,G)$.

Definition 6.23. *Given a partially ordered set $(A,<)$, A is antichain if for any $a,b \in A$, $a \neq b$, there is not an element c such that $c > a$ and $c > b$.*

Now we introduce the combinatorial objects we will use to decompose the following optimization problem: $\min_{C \in \mathcal{C}(T,G)} ||C||_\infty$. The reader may look at Figure 6.7.1 to find examples involving the following definition.

Definition 6.24. *We define $\mathcal{C}^*(T,G)$ as the set of $C^* \subset V_T \times V_G$ such that:*

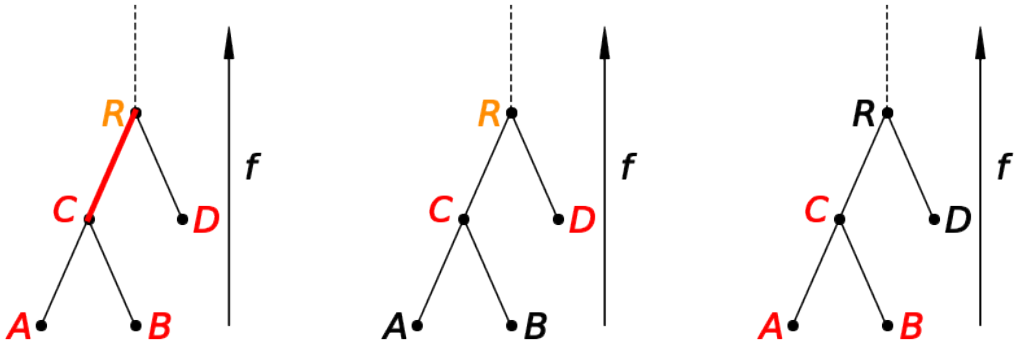


Figure 6.7.1: The letters in red show three examples of $C^* \subset V_T \times V_G$ satisfying property $(A0)$. However the rightmost example does not satisfy the antichain condition $(A1)$, for there is $C > A$ and $C > B$. In orange we highlight the vertices giving $\mathrm{LCA}(\pi_T(C^*))$. The green branch in the leftmost tree signifies that, since $\#\Lambda_{C^*}(C) = 2$, C is deleted.

(A0) $C^* \bigcup \{ (\mathrm{LCA}(\pi_T(C^*)), \mathrm{LCA}(\pi_G(C^*))) \} \in \mathcal{C}(T, G)$;

(A1) C^* *is an antichain.*

The idea behind the definition of $\mathcal{C}^*(T, G)$ is that we assume that we already know $\arg\min\{\| C' \|_\infty \mid C' \in \mathcal{C}_R(T_x, G_y)\}$ for every $x \in E_T$ and $y \in E_G$, and we use $C^* \in \mathcal{C}^*(T, G)$ to optimally aggregate these results to find the optimal global coupling between T and G. We formalize such concepts in the following definition.

Definition 6.25. *Let $C_{x,y} \in \mathcal{C}_R(T_x, G_y)$ for every $x, y \in V_T \times V_G$. Given $C^* \in \mathcal{C}^*(T, G)$, with $r = \mathrm{LCA}(\pi_T(C^*))$ and $r' = \mathrm{LCA}(\pi_G(C^*))$, we define the* extension *of C^* by means of $\{C_{x,y}\}_{(x,y)\in C^*}$ as $E = \{(r, r')\} \bigcup_{(x,y)\in C^*} C_{x,y}$. The set of all possible extensions of C^* is called $E(C^*)$. If $C_{x,y} \in \arg\min\{\| C' \|_\infty \mid C' \in \mathcal{C}_R(T_x, G_y)\}$ for all x, y then we call the extension minimal. We collect all minimal extensions of C^* in the set $E_m(C^*)$ - which is never empty. If all $C_{x,y}$ and E are special couplings then we call the extension special. We collect all special extensions of C^* in the set $E^o(C^*)$ - which is never empty. We name $E^o_m(C^*) = E_m(C^*) \cap E^o(C^*)$ the set of minimal special extensions of C^* - this set could be empty.*

Since extensions are couplings, it is obvious that:

$$d_I(T, G) \leq \min_{C^* \in \mathcal{C}^*(T,G)} \min_{C \in E_m(C^*)} \| C \|_\infty . \tag{6.5}$$

$$d_I(T, G) \leq \min_{C^* \in \mathcal{C}^*(T,G)} \min_{C \in E^o(C^*)} \| C \|_\infty . \tag{6.6}$$

See also Figure 6.7.2a. Moreover it is also clear that any coupling is an extension of some $C^* \in \mathcal{C}^*(T, G)$ and thus:

$$d_I(T, G) = \min_{C^* \in \mathcal{C}^*(T,G)} \min_{C \in E(C^*)} \| C \|_\infty .$$

The upcoming theorem states that there strong relationships between d_I and extensions obtained via a fixed family of $C_{x,y} \in \mathcal{C}_R(T_x, G_y)$.

Theorem 6.26 (Decomposition)**.** *Consider two merge trees T and G and take a collection of $C_{x,y} \in \arg\min\{\| C' \|_\infty \mid C' \in \mathcal{C}_R(T_x, G_y)\}$. Given $C^* \in \mathcal{C}^*(T, G)$ we name $e(C^*)$ the extension obtained by means of $C_{x,y}$. We have:*

$$\min_{C^* \in \mathcal{C}^*(T,G)} \max\{cost_{e(C^*)}(v) \mid v \in \pi_T(e(C^*)) \text{ or } \#\Lambda_{e(C^*)}(v) > 0\} \leq d_I(T, G) \tag{6.7}$$

Moreover, if $C_{x,y}$ for every x, y is also a special coupling we have:

$$d_I(T,G) = \min_{C^* \in \mathcal{C}^*(T,G)} \| e(C^*) \|_\infty . \tag{6.8}$$

We remark that Theorem 6.26 is in some sense unexpected: if Equation (6.5) and Equation (6.6) are in some sense trivial, Equation (6.7) and Equation (6.8) certainly are not. Mainly because the couplings $C_{x,y}$ are fixed at the beginning and not chosen with some optimization strategy. Moreover, the proof of the latter one depends strongly on the possibility to find optimal couplings of the form $\mathcal{C}^o(T,G)$. Note that given $C^* \in \mathcal{C}^*(T,G)$ and $C_{x,y} \in \mathcal{C}^o_R(T_x, G_y)$ special couplings for all $(x,y) \in C^*$, then the extension

$$\{(\mathrm{LCA}(\pi_T(C^*)), \mathrm{LCA}(\pi_G(C^*)))\} \bigcup_{(x,y)\in C^*} C_{x,y}$$

is not a special coupling in general - see Figure 6.7.2b. Similarly, minimal extensions are not optimal couplings in general, see again Figure 6.7.2b.

6.7.2 Approximation by Pruning Operators

Now we prove a second property of the interleaving distance which is very useful when looking for approximations of d_I. In particular, we find a way to approximate the distance sensibly reducing the computational complexity of the problem by removing leaves and computing distances between smaller trees.

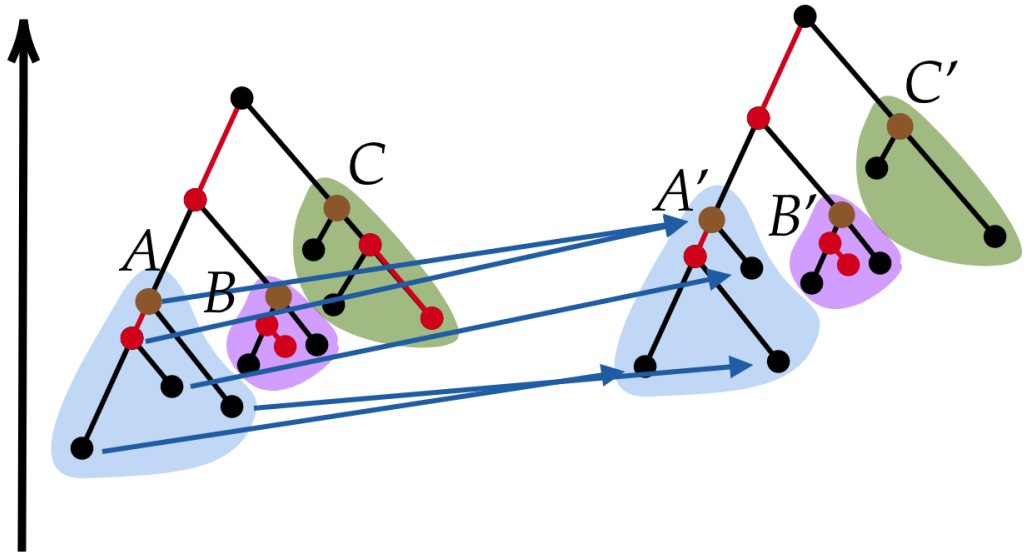
We briefly introduce the pruning operator P_ε as described in Chapter 2 and Chapter 4. Given a merge tree T and $\varepsilon > 0$, the merge tree $P_\varepsilon(T)$ is obtained through a recursive procedure: given a leaf x and its father x', if $f(x') - f(x) < \varepsilon$ we say that x is a small-weight leaf; we want to remove all small-weight leaves - and their fathers if they become order 2 vertices - from T unless two or more of them are siblings, i.e. children of the same father. In this case we want to remove all leaves but the one being the lowest leaf. To make this procedure well defined and to make sure that, in the end, no small-weight leaves are left in the tree, we need to choose some ordering of the leaves and to resort to recursion.

(P) Take a leaf l such that $f(father(l)) - f(l)$ is minimal; if $f(father(l)) - f(l) < \varepsilon$, remove l and its father if it becomes an order 2 vertex after removing l.

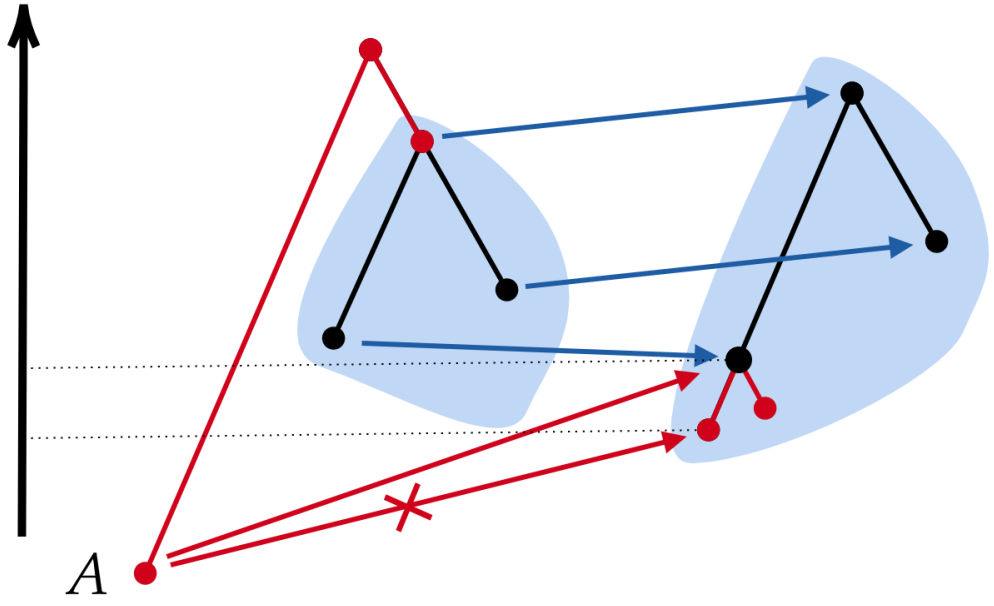
We take $T_0 = T$ and apply operation (P) to obtain T_1. On the result we apply again (P) obtaining T_2 and we go on until we reach a fixed point $T_n = T_{n+1} = P_\varepsilon(T)$ of such sequence. To shed some light on this definition we prove the following lemma; Figure 6.7.3 can be helpful in following the statements.

Lemma 6.27. *Given T, $\varepsilon > 0$ and $P_\varepsilon(T)$, we have a natural injective map $V_{\mathcal{P}_\varepsilon(T)} \hookrightarrow V_T$ which identifies vertices in $\mathcal{P}_\varepsilon(T)$ with vertices in T. Call $C_\varepsilon \subset V_{\mathcal{P}_\varepsilon(T)} \times V_T$ the set of couples induced by $V_{\mathcal{P}_\varepsilon(T)} \hookrightarrow V_T$. The following hold:*

1. *C_ε is a coupling;*
2. *$L_{P_\varepsilon(T)} \subset L_T$ and for every v and v' such that $v' < v$ and $f(v) - f(v') \geq \varepsilon$, there is $l \in L_{P_\varepsilon(T)}$ such that $\mathrm{LCA}(l, v') < v$; in particular $\arg\min f \in L_{P_\varepsilon(T)}$;*
3. *for every $v \in V_T - V_{P_\varepsilon(T)}$ we have $\#\Lambda_{C_\varepsilon}(v) \leq 1$; in particular if $v \in D$, we have $\#\Lambda(v) = 0$ and $f(\varphi_{C_\varepsilon}(v)) - f(v) < \varepsilon$;*
4. *the map: $\eta_{C_\varepsilon} : D^T_{C_\varepsilon} \to V_T$ such that $f(\eta_{C_\varepsilon}(x)) < f(x)$ for all $x \in D^T_{C_\varepsilon}$;*



(a) The couples $\{(A, A'), (B, B'), (C, C')\}$ form antichains via π_T and π_G. The shaded regions indicate how the subtrees rooted in the vertices of the antichain are matched. Minimal couplings are displayed with deletions in red. The remaining vertices can be coupled by visual inspection. We have highlighted the couples of the blue subtree just as an example. Putting together all the couples we obtain an extension of the antichain.



(b) In this figure we can see a non special extension as the lower vertex of the tree on the right is not matched. Even though the coupling displayed by the blue arrows is special for the blue subtrees, it does not extend to a special coupling between the two trees: the deletion of A should go through the lowest vertex of the right tree, while it is forced to go though its father. As signified by the red arrows. We can also appreciate that minimal extensions are not optimal couplings in general.

Figure 6.7.2: Two figures related to decomposition and extensions of couplings.

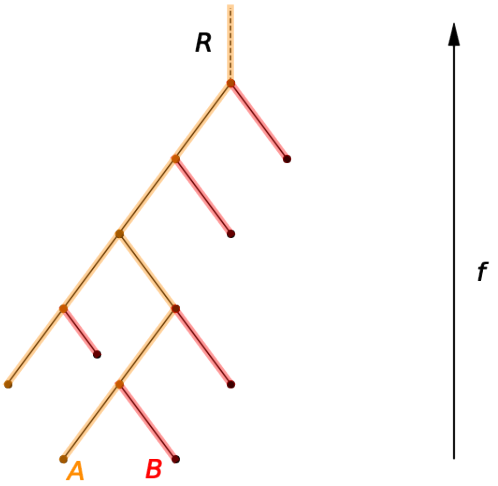


Figure 6.7.3: An example of a pruning operator applied on a merge tree T. The red branches are removed from the tree, while the orange ones are kept and represent the metric merge tree $P_\varepsilon(T)$. We highlight that the root of the tree changes. The superimposition of the orange shaded tree on the black one gives the injection between them. Lastly, note that deleting A instead of B (these are small-weight siblings with same weights - violating (G)) would give isomorphic merge trees.

5. $\| C_\varepsilon \|_\infty \leq \varepsilon/2$.

Having characterized the pruned trees with the above lemma, we can obtain the following proposition.

Proposition 6.28. *Given two merge trees T and G, we have:*

$$d_I(T, G) \leq \max\{d_I(P_\varepsilon(T), P_\varepsilon(G)), \varepsilon/2\}$$

As a result, if the number of leaves of T and G is too high, we know that we can prune them, reducing the computational complexity of d_I, to obtain an estimate from above of $d_I(T, G)$.

We close this section with a claim we would like to investigate in the future.

Claim 6.29. *The coupling C_ε is a minimizing coupling.*

6.8 Approximating the Interleaving Distance: Linear Integer Optimization

In this section we exploit Theorem 6.26 to obtain a dynamical approach to approximate the optimal coupling between two merge trees, by solving recursively linear integer programming problems.

6.8.1 Computing the Cost of a Coupling Extension $e(C^*)$

As a first step we separate the problem of finding $C^* \in \mathcal{C}^*(T, G)$ with a norm-minimizing extension into two separated problems: the first one is to find a minimal-fixed root coupling, the second one is to compute the cost of deleting the vertices which are not below the roots. To make the upcoming lemma more clear we establish the following notation:

if $r = \mathrm{LCA}(\pi_T(C^*))$ and $r' = \mathrm{LCA}(\pi_G(C^*))$, then for $v \not\leq r$, $v_r = \min\{v' \geq r \text{ and } v' \geq v\}$ and $f_r = \min_{v \leq r} f(v)$. Lastly:

$$H_{r,r'} = \max\{\max_{v \not\leq r} 0.5 \cdot (f(v_r) - f(v)), \max_{v \not\leq r} g_{r'} - f(v), \max_{w \not\leq r'} 0.5 \cdot (g(w_{r'}) - g(w)), \max_{w \not\leq r'} f_r - g(w)\}.$$

Note that $H_{r,r'}$ does not depend on the chosen extension of C^*, and, in fact, it depends only on r, r'. Lemma 6.30 states that $H_{r,r'}$ accounts for the deletions of all the vertices which are not below r or r'.

Lemma 6.30. *Given T, G merge trees and $C^* \in \mathcal{C}^*(T, G)$, with $r = \mathrm{LCA}(\pi_T(C^*))$ and $r' = \mathrm{LCA}(\pi_G(C^*))$ then:*

$$\| C_{r,r'} \|_\infty = \max\{\| C'_{r,r'} \|_\infty, H_{r,r'}\} \tag{6.9}$$

where $C_{r,r'}$ is any extension of C^* and $C'_{r,r'}$ is equal as a set to $C_{r,r'}$ but is considered as a coupling in $\mathcal{C}(T_r, G_{r'})$.

Since the computation of $H_{r,r'}$ can be easily accessed in a greedy fashion, from now on, we focus on find an approximate solution for $C \in \arg\min\{\| C' \|_\infty \mid C' \in \mathcal{C}_R(T, G)\}$.

6.8.2 Iterative Approach

Now we start working out a procedure to approximate $C \in \arg\min\{\| C' \|_\infty \mid C' \in \mathcal{C}_R(T, G)\}$. The assumption of our approach is that we already have computed $C_{x,y} \in \mathcal{C}_R(T_x, G_y)$ that we want to use to extend any $C^* \in \mathcal{C}^*(T, G)$, as in Theorem 6.26. So for instance, if we want to work with special extensions, we assume that we have $C_{x,y} \in \arg\min\{||C'||_\infty \big| C' \in \mathcal{C}^o_R(T_x, G_y)\}$ for all $x \in E_T$ and $y \in E_G$ and we exploit them to obtain a special extension of some $C^* \in \mathcal{C}^*(T, G)$. If we want to work with minimal extensions, instead, we would assume to have $C_{x,y} \in \arg\min\{||C'||_\infty \big| C' \in \mathcal{C}_R(T_x, G_y)\}$. We anticipate that both kinds of extensions are important for our purposes as special extension will be used in the following to produce upper bounds for the interleaving distance, while minimal extension will be used accordingly to Theorem 6.26 to obtain a lower bound.

We fix the following notation: $f_x = \min_{v \leq x} f(v)$ and $g_y = \min_{w \leq y} g(w)$. Lastly, we fix a constant $K > 0$ such that $K > \max_{x \in V_T, y \in V_G} |f(x) - g(y)|$. In Section 6.8.5 we point out which steps of the upcoming procedure may produce errors.

6.8.3 Variables and Constraints

We consider the following set of binary variables: $a_{x,y}$ for $x \in E_T$ and $y \in E_G$; u_x for $x \in E_T$ and u_y for $y \in E_G$. The vector of all variables (upon choosing some ordering) will be referred to as $\mathcal{V}$.

We also introduce the following auxiliary variables:

- $c_x = \sum_y a_{x,y}$ and $c_y = \sum_x a_{x,y}$
- $\Lambda_x = \sum_{v \leq x} c_v$ and $\Lambda_y = \sum_{w \leq y} c_w$
- $d_x = \sum_{v \leq x} c_v + \sum_{v > x} c_v$ and $d_y = \sum_{w \leq y} c_w + \sum_{w > y} c_w$

and the following constraints:

(1) for every $l \in L_T$: $\sum_{l \leq x < r_T} c_x \leq 1$ and for every $l' \in L_G$: $\sum_{l' \leq y < r_G} c_y \leq 1$

(2) $u_x \leq 0.5\Lambda_x$ and $u_y \leq 0.5\Lambda_y$ for every x and y;

(3) $u_x \geq m_x\Lambda_x + q_x$ and $u_y \geq m_y\Lambda_y + q_y$ for every x and y. With m_x, q_x being any pair of (m, q) such that the following are satisfied: $q < 0$, $m + q < 0$, $2m + q > 0$, $m\#L_T < 1$ (analogously for m_y, q_y);

(4) (only for special extensions) let $\widetilde{x} = \arg\min_{v \in V_T} f(v)$ and $\widetilde{y} = \arg\min_{w \in V_G} g(w)$; then we ask $\sum_{\widetilde{x} \leq x < r_T} c_x \geq 1$ and $\sum_{\widetilde{y} \leq y < r_G} c_y \geq 1$.

The set of vectors of variables which satisfy these constraints is called $\mathcal{K}_m$ or $\mathcal{K}^o$ depending on whether (4) are, respectively, included or not. Note that $\mathcal{K}_m \supset \mathcal{K}^o$. To lighten the notation, when we do not wish to distinguish between $\mathcal{K}_m$ and $\mathcal{K}^o$, we simply write $\mathcal{K}$. Now we briefly comment on the variables and constraints to try to give some intuition about their roles:

- the variables $a_{x,y}$ are used to match x with y, i.e. add the couple (x, y) to the coupling. In particular constraint (1) ensures that if $\mathcal{V} \in \mathcal{K}$, the set $C(\mathcal{V}) = \{(x, y) \big| v_{x,y} = 1\}$ belongs to $\mathcal{C}^*(T, G)$;

- the variables d_x and u_x are used to determine if x must be deleted. Start with d_x. The main idea which our optimization procedure is based on, is that if $a_{v,w} = 1$ with $v > x$, it means that x is taken care of by the coupling $C_{v,w}$ and thus we want to "ignore" such x. In other words, having $d_x = 0$, implies that x is deleted with $\#\Lambda(x) = 0$;

- now we turn to u_x. Observe that, if $\Lambda_x = 0, 1$, then $u_x = 0$ while if $\Lambda_x \geq 2$, $u_x = 1$. Note that $\Lambda_x \leq \#L_T$ and $\Lambda_y \leq \#L_G$. Constraints (2) and (3) are fundamental to fix the value of u_x, depending linearly on Λ_x: $u_x = 1$ if and only if $x \geq \mathrm{LCA}(v, v')$, with $a_{v,w} = 1$ and $a_{v',w'} = 1$, for some $w, w' \in V_G$. Thus having $u_x = 1$ means that x is deleted with $\#\Lambda(x) > 1$;

- lastly, we comment on constraints (4). These constraints are asking that there is one point above the lowest vertex in each tree which is coupled by $C(\mathcal{V})$; this in particular implies that, if we assume $C_{x,y} \in \mathcal{C}^o(T_x, G_y)$, then $\widetilde{x}$ and $\widetilde{y}$ are never deleted. Thus, $\{(r_T, r_G)\} \bigcup_{(x,y) \in C(\mathcal{V})} C_{x,y}$ is a special extension, i.e. a coupling in $\mathcal{C}^o_R(T, G)$.

As a consequence of these observations, any $C \in \mathcal{C}^*(T, G)$ induces a unique vector of variables $\mathcal{V} \in \mathcal{K}$, such that $C(\mathcal{V}) = C$. Moreover if $C_{x,y} \in \mathcal{C}^o(T_x, G_y)$ and $\mathcal{V} \in \mathcal{K}^o$, then $\{(r_T, r_G)\} \bigcup_{(x,y) \in C(\mathcal{V})} C_{x,y}$ is a special extension of $C(\mathcal{V})$.

6.8.4 Objective function

A key fact that we highlight is that, by property (A1), if $x = \mathrm{LCA}(v, v')$, with $a_{v,w} = 1$, $a_{v',w'} = 1$ and $x < r_T$, then $\delta(x) = r_T$ due to the antichain condition. So we know that any vertex x' such that $x \leq x' < r_T$, is in $D^T_{C(\mathcal{V})}$. Thus, given $x \in V_T$, and with x_f being its father, the "local" objective function, depends on the following quantities:

- $\sum_y a_{x,y} \parallel C_{x,y} \parallel_\infty$: it is the cost of matching T_x and G_y, if (x, y) is added to C. If $C_{x,y}$ is not a minimal coupling, this is an upper bound;

- $|\, g(r_G) - f(x) \,|\, u_x$: it is an upper bound to the cost of deleting x with $\#\Lambda(x) > 1$;

- $A_x = 0.5(f(x_f) - f_x)(1 - d_x)$: in case x is deleted with $\#\Lambda(x) = 0$, it gives a lower bound to the deletion of T_x, as it is equal to deleting the lowest point below x to the height of the father of x, in two step. It is an exact value if $x_f = \varphi(x)$ and $g(\eta(x)) - f_x < A_x$;

- $B_x = \{\sum_y a_{v,y}\,(g_y - f_x) - Kd_x \,|\, v < x_f\}$: in case x is deleted with $\#\Lambda(x) = 0$ it helps giving an upper bound to the deletion of T_x, depending on what happens below x_f. If there is $v < x_f$ such that v is coupled with y and $C_{x,y}$ is a special coupling, then we know that $g_y \geq g(\eta(x))$, and so $g_y - f_x \geq g(\eta(x)) - f_x$. Thus, if $C_{x,y}$ is a special coupling, $\max\{\max B_x, A_x\} \geq cost(x)$.

For any $x \in E_T$ we define:

$$\Gamma^{\uparrow}(x) = \max\left(\sum_y a_{x,y} \parallel C_{x,y} \parallel_\infty, (g(r_G) - f(x))u_x, A_x, \max B_x\right)$$

and:

$$\Gamma^{\uparrow}(y) = \max\left((f(r_T) - g(y))u_y, A_y, \max B_y\right)$$

for any $y \in E_G$.

In full analogy we set:

$$\Gamma^{\downarrow}(x) = \max\left(\sum_y a_{x,y} \parallel C_{x,y} \parallel_\infty, A_x\right)$$

and:

$$\Gamma^{\downarrow}(y) = A_y$$

for any $y \in E_G$.

Lastly,: $\Gamma^{\uparrow}(r_T) = \Gamma^{\downarrow}(r_T) = |r_T - r_G|$.

With an abuse of notation we write:

$$\Gamma^{\uparrow}(\mathcal{V}) = \max_{x \in V_T \coprod V_G} \Gamma^{\uparrow}(x) \tag{6.10}$$

$$\Gamma^{\downarrow}(\mathcal{V}) = \max_{x \in V_T \coprod V_G} \Gamma^{\downarrow}(x). \tag{6.11}$$

We sum up the main properties of the definitions we have just stated with the following proposition.

Proposition 6.31. *Given $C \in \mathcal{C}^*(T,G)$ and $C_{x,y} \in \mathcal{C}^o_R(T_x, G_y)$ for all $(x,y) \in C$, we call*

$$C^o := \{(r_T, r_G)\} \bigcup_{(x,y)\in C} C_{x,y}.$$

If C^o is a special extension then:

$$||C^o||_\infty \leq \Gamma^{\uparrow}(\mathcal{V}) \tag{6.12}$$

with $\mathcal{V}$ being the unique set of variables in $\mathcal{K}^o$ such that $C(\mathcal{V}) = C^o$.

Viceversa, given $C \in \mathcal{C}^*(T,G)$, and $C_{x,y} \in \mathcal{C}_R(T_x, G_y)$ with minimal norm for all $(x,y) \in C$, we call

$$C_m := \{(r_T, r_G)\} \bigcup_{(x,y)\in C} C_{x,y}.$$

Then:

$$\Gamma^{\downarrow}(\mathcal{V}) \leq \max\{cost_{C_m}(v) \,|\, v \in \pi_T(C_m) \text{ or } \#\Lambda_{C_m}(v) > 0\} \tag{6.13}$$

with $\mathcal{V}$ being the unique set of variables in $\mathcal{K}_m$ such that $C(\mathcal{V}) = C_m$.

Putting together Theorem 6.26 and Proposition 6.31 we obtain as a corollary:

Corollary 6.32. *Consider T, G merge trees, and take:*

1. *a collection of $C_{x,y} \in \arg\min\{\| C' \|_\infty \mid C' \in \mathcal{C}_R(T_x, G_y)\}$.*
2. *a collection of $C'_{x,y} \in \arg\min\{\| C' \|_\infty \mid C' \in \mathcal{C}^o_R(T_x, G_y)\}$.*

Then:

$$\min_{\mathcal{V}\in\mathcal{K}_m} \Gamma^{\downarrow}(\mathcal{V}) \leq \min\{\| C \|_\infty \mid C \in \mathcal{C}_R(T, G)\} \leq \min_{\mathcal{V}\in\mathcal{K}^o} \Gamma^{\uparrow}(\mathcal{V})$$

where $\Gamma^{\downarrow}$ is computed with $\{C_{x,y}\}$ and $\Gamma^{\uparrow}$ is computed with $\{C'_{x,y}\}$.

Now we get rid of the fixed roots, obtaining an approximation for $d_I(T, G)$ by putting together Theorem 6.26, Lemma 6.30 and Proposition 6.31.

Corollary 6.33. *In the same setting as Corollary 6.32, for each $(x, y) \in V_T \times V_G$ we have:*

$$W^{\downarrow}_{x,y} := \min_{\mathcal{V}\in\mathcal{K}_m} \Gamma^{\downarrow}(\mathcal{V}) \leq \min\{\| C \|_\infty \mid C \in \mathcal{C}_R(T_x, G_y)\} \leq W^{\uparrow}_{x,y} := \min_{\mathcal{V}\in\mathcal{K}^o} \Gamma^{\uparrow}(\mathcal{V}).$$

where the constraints defining $\mathcal{K}$ clearly depend on the vertices used to generate the subtrees, $\Gamma^{\downarrow}$ is computed with $\{C_{x,y}\}$ and $\Gamma^{\uparrow}$ is computed with $\{C'_{x,y}\}$.

Consequently:

$$\min_{(x,y)\in V_T\times V_G} \max\{H_{x,y}, W^{\downarrow}_{x,y}\} \leq d_I(T, G) \leq \min_{(x,y)\in V_T\times V_G} \max\{H_{x,y}, W^{\uparrow}_{x,y}\}. \tag{6.14}$$

6.8.5 Approximations

We take this section to briefly isolate which are the situations in which our procedure may produce errors w.r.t. the true interleaving distance.

1. Clearly r_G (r_T) is an upper bound for $\chi(x)$ ($\chi(y)$) with x (y) being deleted with $\#\Lambda(x) > 1$ ($\#\Lambda(x) > 1$) and so $|\, g(r_G) - f(x) \,|\, u_x$ is an upper bound to the cost of the corresponding deletion.
2. Computing $\eta(v)$ for $v \in D_T$ with $\#\Lambda(v) = 0$: we may have $|\, f(v) - g(\eta(v)) \,| \leq \max B_{x'}$ for $x = \varphi(v)$ and $x = father(x')$.

6.8.6 Linearization

At this point we have introduced a set of linear constraints, needed to optimize a non linear function (either $\Gamma^{\uparrow}$ or $\Gamma^{\downarrow}$) of the form $\min_{\mathcal{V}\in\mathcal{K}} \max_i F_i(\mathcal{V})$ for some real-valued functions F_j which are linear in $\mathcal{V}$. We can turn this into a linear optimization problem by exploiting a standard trick, introducing auxiliary variables and with additional constraints.

Suppose we need to find $\min_s \max(f(s), g(s))$, with f, g real valued functions; we then introduce the variable u, with the constraints $u \geq f(s)$ and $u \geq g(s)$ and solve the problem $\min_{s,u\geq f(s),u\geq g(s)} u$. We want to use this procedure to compute $\min_{\mathcal{V}\in\mathcal{K}} \Gamma^{\uparrow}(\mathcal{V})$ and $\min_{\mathcal{V}\in\mathcal{K}} \Gamma^{\downarrow}(\mathcal{V})$. We write down the details only for $\min_{\mathcal{V}\in\mathcal{K}} \Gamma^{\uparrow}(\mathcal{V})$, the case $\Gamma^{\downarrow}(\mathcal{V})$ follows easily.

Given $x \in E_T$ we define $F^1_x = \sum_y a_{x,y} \| C_{x,y} \|_\infty$, $F^2_x = (g(r_G) - f(x))u_x$, and $F^3_x = A_x$. Given $y \in E_G$, instead, we set $F^1_y = (f(r_T) - g(y))u_y$, and $F^2_y = A_y$. Analogously, we have $F^1_{r_T} = |\, f(r_T) - g(r_G) \,|$. Having fixed a total ordering on $V_T = \{a_0, \ldots, a_n\}$ and $V_G = \{b_0, \ldots, b_m\}$, respectively, we call $\mathcal{F}$ the vector containing all the components F^i_x and F^j_y of all the points taken in the chosen order:

$$\mathcal{F} := \left(F^1_{a_1}, F^2_{a_1}, F^3_{a_1}, \ldots, F^1_{a_i}, F^2_{a_i}, F^3_{a_i}, \ldots, F^1_{b_1}, F^2_{b_1}, \ldots, F^1_{b_m}, F^2_{b_m}\right) = (F_i) \tag{6.15}$$

Similarly, for every x, we order the elements of the set B_x, so that $B_x = \big(\dots, B_x^h, \dots\big)$; then we set

$$\mathcal{B} := \Big(B_{a_0}^1, \dots, B_{a_0}^{h_0}, \dots, B_{a_i}^1, \dots, B_{a_i}^{h_i}, \dots, B_{b_0}^1, \dots, B_{b_0}^{t_0}, \dots, B_{b_j}^1, \dots, B_{b_j}^{t_j}\Big) = (B_i) \quad (6.16)$$

So we introduce the auxiliary variable z and add the following constraints to the ones presented in Section 6.8.3:

(5) $z \geq F_i$ for all i;

(6) $z \geq B_i$ for every i;

and then solve $\min_{z,\mathcal{V}} z$ with all these constraints. We stress again that F_i and B_i are linear in $\mathcal{V}$; so the final problem is linear with integer valued variables. In case of $\Gamma^{\downarrow}$ we repeat the same operations, omitting the constraints in (6).

6.8.7 Bottom-Up Algorithm

In this section the results obtained in Section 6.7.1 and the formulation established in the previous parts of Section 6.8 are used to obtain the algorithm implemented to approximate the metric d_I between merge trees. We need to introduce some last pieces of notation in order to describe the "bottom-up" nature of the procedure.

Given $x \in V_T$, define $len(x)$ to be the cardinality of $\{v \in V_T \,|\, x \leq v \leq r_T\}$ and $len(T) = \max_{v \in V_T} len(v)$; similarly $lvl(x) = len(T) - len(x)$ and $lvl_T(n) = \{v \in V_T \,|\, lvl(v) = n\}$

The key property is that: $lvl(x) > lvl(v)$ for any $v \in sub_T(x)$. Thus, for instance, if $W^{\uparrow} x, y$ is known for any $x \in lvl_T(n)$ and $y \in lvl_G(m)$, then for any v, w in V_T, V_G such that $lvl(v) < n$ and $lvl(w) < m$, $W^{\uparrow}_{v,w}$ is known as well. We write down Algorithm 2, which refers to the computation of $\min \Gamma^{\uparrow}$. Note that thanks to constraints (4) this recursive procedure is always guaranteed to provide a special coupling.

In case of $\Gamma^{\downarrow}$ we repeat the same algorithm, omitting the constraints in (4) and (6) and with $W^{\downarrow}_{x,y} = \min_{z,\mathcal{V}} z$ being a lower bound for $d_I(T_x, G_y)$.

Algorithm 2. Bottom-Up Algorithm.

Result: Upper bound for $d_I(T, G)$
initialization: $N = len(T)$, $M = len(G)$, $n = m = 0$;
while $n \leq N$ *or* $m \leq M$ **do**
  for $(x, y) \in V_T \times V_{T'}$ *such that* $lvl(x) \leq n$ *and* $lvl(y) \leq m$ **do**
    Calculate $H_{x,y}$;
    Calculate $W^{\uparrow}_{x,y} = \min_{z,\mathcal{V}} z$ subject to constraints (1)-(6), using $W^{\uparrow}_{x',y'}$ as upper bound for $\| C^o_{x',y'} \|_{\infty}$, for all couples (x', y') already considered;
  end
  $n = n + 1$; $m = m + 1$;
end
return $\min_{(x,y) \in V_T \times V_G} \max\{H_{x,y}, W^{\uparrow}_{x,y}\}$

6.9 Error Propagation

We make a brief observation to take care of the interactions arising between the approximations of the interleaving distance defined in Section 6.8.4 and the bottom-up procedure proposed in Section 6.8.7.

Consider the setting of Section 6.8.4 and suppose that, instead of having computed the optimal couplings $C_{x,y}$, we have some approximations $C'_{x,y}$ - as it is the case for $W^{\uparrow}_{x,y}$ and $W^{\downarrow}_{x,y}$ - with an error term $e_{x,y}$ such that $\mid \parallel C_{x,y} \parallel_\infty - \parallel C'_{x,y} \parallel_\infty \mid < e_{x,y}$. We see that the errors $e_{x,y}$ affect only the components of the form $\sum_y a_{x,y} \parallel C_{x,y} \parallel_\infty$. Moreover, the potential errors occurring in the estimates $\Gamma^{\uparrow}$ and $\Gamma^{\downarrow}$, appear at the level of B_x and u_x. But $e_{x,y}$, u_x and B_x do not interact or aggregate at any time in the objective functions. Thus, the eventual error at the level of B_x and u_x does not depend on any $e_{x,y}$: they are always independent and do not interfere with each other. Which means that errors do not propagate exploding in size: the final error in the algorithm presented in Section 6.8.7 is the maximum of all the independent errors occurring at every iteration.

Remark 6.34. *In the definition of $\Gamma^{\uparrow}$ the biggest potential source of error are the terms $\mid g(r_G) - f(x) \mid u_x$ which make very costly the deletion of internal vertices in order to swap father children relationships. This is an issue which brings together our approximation scheme for the interleaving distance and other distances for merge trees which have been defined in literature, such as Sridharamurthy et al. (2020); Wetzels et al. (2022); Pont et al. (2022). A very detailed explanation of the problems arising from this fact can be found in Chapter 3, along with the solutions that the authors of the previously mentioned works propose to mitigate them. All these solutions apply also to our case, but, on top of them, in the definition of $\Gamma^{\uparrow}$ we could also replace $\mid g(r_G) - f(x) \mid u_x$ with $\mid f(father(x)) - f(x) \mid u_x$. We believe that this option, on average, should produce lower errors (w.r.t. $\Gamma^{\uparrow}$) and more stable behaviours when compared to the metrics Sridharamurthy et al. (2020); Wetzels et al. (2022); Pont et al. (2022). However, we leave to future works the assessment of the properties of this third approximation scheme. We point out that $\mid f(father(x)) - f(x) \mid u_x$ in general in neither a lower or an upper bound to the cost of deleting x and that is why we are not considering it in our theoretical investigation. Lastly, replacing $\mid g(r_G) - f(x) \mid u_x$ with the exact the deletion cost, would turn the linear optimization problem into a polynomial one. Such polynomial problem could then be turned into a linear one with auxiliary variables, but we believe that would make its computational cost too high.*

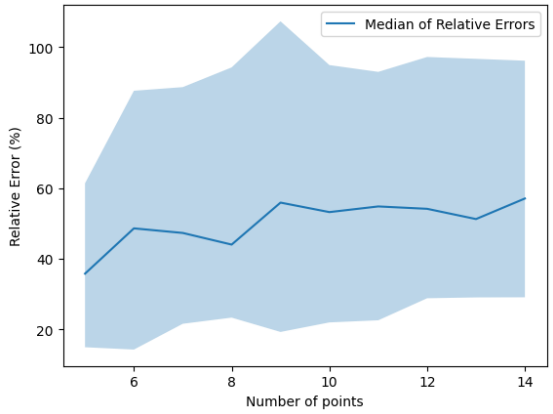
6.10 Simulation Study

In this section we test our approximations versus another method to approximate d_I recently proposed by Curry et al. (2022), relying on the work of Gasparovic et al. (2019). The approximation proposed by Curry et al. (2022) turns the unlabeled problem of the interleaving distance between merge trees into a labeled interleaving problem by proposing a suitable set of labels. The optimal labeling would give the exact value of the interleaving distance, but, in general, this procedure just returns an upper bound. We call d_{lab} the approximation obtained with the labeled method proposed by Curry et al. (2022) and d_u and d_l respectively our approximation from above (i.e. with $\Gamma^{\uparrow}$) and below (i.e. with $\Gamma^{\downarrow}$).

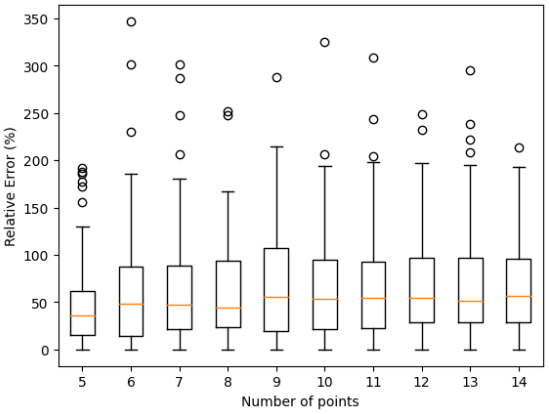
For any fixed i, we generate a couple of point clouds $C^k_i = \{(x^k_j, y^k_j) \mid j = 1, \dots, n_i\} \subset \mathbb{R}^2$, with $k = 1, 2$, according to the following process:

$$
\begin{aligned}
x^k_j &\sim^{iid} \mathcal{N}(5, \sigma_{x,k}) \; j = 1, \dots, n_i \\
y^k_j &\sim^{iid} \mathcal{N}(5, \sigma_{y,k}) \; j = 1, \dots, n_i \\
\sigma_{x,k} &\sim \mathcal{N}(3, 1) \\
\sigma_{y,k} &\sim \mathcal{N}(3, 1).
\end{aligned}
$$

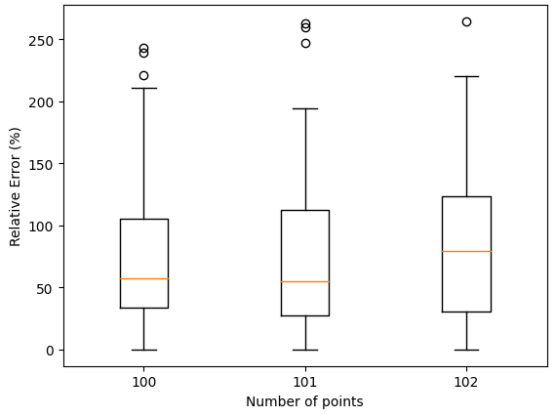
Note that n_i regulates the number of leaves in the trees (which we fix before sampling C^k_i). From C^k_i we obtain the single linkage hierarchical clustering dendrogram $T_{C^k_i}$ (that



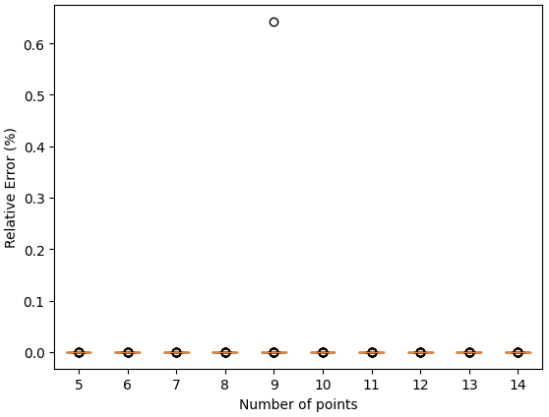
(a) Median of the error percentage as a function of the number of leaves, with the shaded region being the central quartiles.



(b) Boxplot of the error percentage as a function of the number of leaves.



(c) Boxplot of the error percentage as a function of the number of leaves, with the number of leaves being big (and so resorting to d_{opt}).



(d) Boxplot of the differences between the upper and lower bounds to the interleaving distance, as a function of the number of leaves.

Figure 6.10.1: Descriptive statistics of the error percentages obtained with the approximations d_l, d_u, d_{opt} and d_{lab}, as a function of the number of leaves, with $n_i \in \{2, \dots, 15\}$ - Figure 6.10.1a, Figure 6.10.1b, Figure 6.10.1d - and $n_i \in \{100, 01, 102\}$ - Figure 6.10.1c.

is, the merge tree representing the Vietoris Rips filtration of C_i^k). And then compute $d_u(T_{C_i^1}, T_{C_i^2})$, $d_l(T_{C_i^1}, T_{C_i^2})$ and $d_{lab}(T_{C_i^1}, T_{C_i^2})$. The distance d_{lab} is computed with the code available at https://github.com/trneedham/Decorated-Merge-Trees, while d_u and d_l are computed via the procedure described in Section 6.8.7. For each $n_i \in \{2, \dots, 15\}$ we sample 100 pairs of point clouds and then compute the relative error $(d_{lab} - d_u)/d_u$ and $(d_l - d_u)/d_u$.

We repeat the same experiment, this time with $n_i \in \{100, 101, 102\}$. Since in this case d_u requires too much time to be computed exactly, we exploit Proposition 6.28 and consider the smallest $\varepsilon > 0$ such that $P_\varepsilon(T_{C_i^1})$ and $P_\varepsilon(T_{C_i^2})$ have fewer or equal then 15 leaves. We then call $d_{opt}(T_{C_1}, T_{C_2}) = \max\{\varepsilon/2, d_u(P_\varepsilon(T_{C_1}), P_\varepsilon(T_{C_2}))\}$.

In general we have the following inequality:

$$d_I(T, G) \leq d_{opt}(T, G), d_{lab}(T, G).$$

The results of the simulations can be seen in Figure 6.10.1. Looking at Figure 6.10.1a, Figure 6.10.1b, Figure 6.10.1c we see that, in the context of our data-generating process,

the approximation given by d_{lab} is very unreliable, producing a median error which around $50 - 60\%$, but with some outliers which are completely off from the values obtained with d_u, with errors of more than 3 times the actual value. Figure 6.10.1d instead shows that, with our data-generating pipeline, the gap between d_u and d_l is almost always 0, which means that we are exactly computing the interleaving distance.

As a conclusive remark, we say that the computational advantages of the labeled approach are immense and potentially adequate even for real time applications, but from our simulation we see that the results need to be taken with care, for the approximation produced is not always good. On the other hand, with the present implementation, the computational cost of our approach becomes prohibitive quite quickly as the number of leaves in the trees increases, even if there might be situations - like the one in this simulation - in which the approximation scheme we used for $n_i \geq 100$ can produce good estimates. We think that interactions between the two approaches could lead to substantial speed-ups: being able to fix the value of some variables in our algorithm basing on the labeling scheme by Curry et al. (2022) could greatly reduce the dimensionality of the problem and thus its computational cost.

6.11 Discussion

In this chapter we propose a graph-matching approach to the interleaving distance between merge trees, trying to understand this metric by means of matchings between unlabeled combinatorial objects rather than with ε-good maps between metric trees. The relationship between this kind of matchings, which we call couplings, and continuous maps between merge trees helps us in producing another formulation of the interleaving distance. In the second part of the chapter we exploit this formulation to obtain some novel properties of the interleaving distance, along with two approximation schemes relying on a dynamical linear integer programming approach. We test this schemes in a simulation study against another approximation procedure recently published, producing for the first time some error estimates for this second procedure. The algorithms we produce have some drawbacks in terms of computational cost, which make it feasible only on small data sets, but still shows that the reliability of the other implementation available to work with this distance is often not very good.

There are many directions which we would like to investigate in the future. We would like to further test the lower and upper bounds we propose in relationship to the third approximation scheme presented in Remark 6.34, perhaps forcing situations in which the lower and upper bound are far apart. Further testing would also benefit the understanding of upper bound we obtain via pruning. Similarly we would like to compare approximation of the interleaving distance with other distances proposed for merge trees, to assess the different "unstable" behaviours. On top of that, we believe that the relationship between a coupling C and the map α^C can be used to obtain yet another definition of the interleaving distance using only continuous monotone maps between metric merge trees. In particular: given $\alpha : \mathbf{T} \to \mathbf{G}$ and $\mu : \mathbf{T} \to \mathbf{T}$ if we define $H(\alpha) := \max_{x \in \mathbf{T}} |\, g(\alpha(x)) - f(x) \,|$ and $2 \cdot D(\mu) = \max_{x \in \mathbf{T}} d(x, \mu(x))$ (where d is the shortest path metric on $\mathbf{T}$) we think that:

$$d_I(\mathbf{T}, \mathbf{G}) = \min_{\alpha,\beta} \{H(\alpha), H(\beta), D(\beta\alpha), D(\alpha\beta)\}$$

where α and β are continuous monotone maps.

On top of that, the global modeling of the matching between the trees offered by couplings, opens up the door also to optimal couplings which also satisfy some additional local optimality criterion. This could be used to obtain some useful parametrizations of particular optimal maps between merge trees, leading to a better understanding of the geodesic structure of the space of merge trees with the interleaving distance.

Appendix

6.A Proofs

Proof of Proposition 6.13.

- The thesis is obvious if $x, x' \in \pi_T(C)$;
- if $x \in \pi_T(C)$ and $x' \in D$, then $w \geq \chi(x')$, with $(x, w) \in C$, otherwise (C3) is violated since $\#\Lambda(x) > 1$, and so $\alpha_C(x) \geq \alpha_C(x')$;
- if $x' \in \pi_T(C)$ and $x \in D$, then $\chi(x) \leq w$ with $(x', w) \in C$;
- lastly, suppose $x, x' \in D$ and consider the following cases:
  - if $\#\Lambda(x) = \#\Lambda(x') = 0$ then we have $\varphi(x) = \varphi(x')$ and thus $f(x) + \frac{1}{2}(f(\varphi(x)) - f(x)) \leq f(x') + \frac{1}{2}(f(\varphi(x')) - f(x'))$. This entails $g(\alpha_C(x)) \geq g(\alpha_C(x'))$;
  - if $\#\Lambda(x), \#\Lambda(x') > 1$ then we have $\chi(x) \leq \chi(x')$ and the thesis clearly follows;
  - suppose lastly that $\#\Lambda(x) = 0$ and $\#\Lambda(x') = 1$; then $\eta(x) \leq \chi(x')$.

■

Proof of Lemma 6.14.

Given $x \in \mathbf{T}$, if $\max \pi_T(C) \in \{v \in \mathbf{T} \,|\, v \geq x\}$, we have a well defined upper extreme $u_C(x)$. Moreover, since $\{v \in \mathbf{T} \,|\, v \geq x\}$ is totally ordered, $u_C(x)$ is unique. If $\max \pi_T(C) \notin \{v \in \mathbf{T} \,|\, v \geq x\}$, i.e. $\mathrm{LCA}(x, \max \pi_T(C)) > \max \pi_T(C)$, then $u_C(x) = +\infty$.

Now consider $x \in L_T$. Since $sub_T(x) = \{x\}$ then $x \notin U$. Thus also $l_C(x)$ is well defined. On top of that, since by hypotheses we are considering only vertices such that $x \notin \pi_T(C) \bigcup D$, $x \neq l_C(x) \neq u_C(x)$. Thus $[l_C(x), u_C(x)]$ is a non degenerate sequence of edges.

Suppose $\{v \in U \,|\, l_C(x) \leq v \leq x\} = \emptyset$ and $v', v'' \in \max\{v \leq x \,|\, v \in \pi_T(C) \cup D\}$. Clearly $[v', x] \bigcap [v'', x] \bigcap V_T \neq \emptyset$; so consider $p \in [v', x] \bigcap [v'', x] \bigcap V_T$. We know $p \notin U$, thus $p \in \pi_T(C) \cup D$, which is absurd since $v', v'' < p$.

Lastly, suppose $x \in U$ (and so $\#\Lambda(x) = 1$) and $\max\{v \leq x \,|\, v \in \pi_T C \cup D\} \subset D$. Then $\#\Lambda(l_C(x)) \neq 1$ for any $l_C(x)$. Let $\{v\} = \Lambda(x)$ and consider $[v, x]$. If $v' \in U$ for all $v < v' < x$, we are done. Clearly there cannot be vertices v' with $v < v' < x$ which are in $\pi_T(C)$. So suppose there is $v' \in D$, with $v < v' < x$. Since $v \in \Lambda(v')$, we have $\#\Lambda(v') > 0$, but then $\#\Lambda(v') > 1$ which means $v, v'' \in \Lambda(v')$ for some $v'' < v'$. Clearly in $[v'', x]$ there can be no vertex contained in $\pi_T(C)$ apart from v''. Thus $\#\Lambda(x) > 1$, which is absurd.

■

Proof of Proposition 6.18.

We need to check continuity. Consider $x_n \to x$ in $\mathbf{T}$. We know that their order relationships is preserved by α_C and α_C^ε. On top of that $f(x_n) \to f(x)$ yields $f(x_n)+\varepsilon \to f(x)+\varepsilon$ and the result follows.

■

Proof of Lemma 6.19.

We know that $\Lambda(x), \Lambda(y) > 1$. Thus x, y are either deleted or coupled. Note that $\chi(x) \geq y$ and $\chi(y) \geq x$. If both of them are coupled then $(x, y) \in C$. Suppose x is coupled - with $\delta(y)$ - and y is deleted. Then $cost(y) = |\, f(\chi(y)) - g(y)\,| < \varepsilon$ and $cost(x) = |\, f(x) - g(\delta(y))\,| < \varepsilon$. Since $\delta(y) \geq y$ and $\chi(y) \geq x$ we obtain the thesis.

■

Proof of Theorem 6.20.

- α_C^ε is continuous by Proposition 6.18.
- (P1) holds by Proposition 6.18.
- Now we prove (P2). Suppose we have $\alpha_C^\varepsilon(v) < \alpha_C^\varepsilon(v')$. And let $x = \mathrm{LCA}(v, v')$.

  We can suppose $\#\Lambda(x) > 1$ and $\mathrm{LCA}(\Lambda(x)) = x$; otherwise at least one between $\varphi(v) \geq x$ and $\varphi(v') \geq x$ holds. Suppose the second one holds, then $\#\Lambda(v') = 0$ and $\varepsilon \geq (f(\varphi(v')) - f(v'))/2$ and $\varphi(v') \geq x$. Thus, (P2) holds. The same if the first one holds.

  Now we show that if $\#\Lambda(x) > 1$ we can find $a, b \in V_T$ such that:

  - $x = \mathrm{LCA}(a, b)$;
  - $(a, a'), (b, b') \in C$;
  - $\alpha_C(v) \geq \alpha_C(a)$ and $\alpha_C(v') \geq \alpha_C(b)$.

  Note that, in this case, upon calling $y = \mathrm{LCA}(a', b')$ we have: $|\, f(x) - g(y)\,| \leq \varepsilon$ by Lemma 6.19, $\alpha_C^\varepsilon(v') \geq \{\alpha_C(v), \alpha_C(a), \alpha_C(b)\}$ and so $\alpha_C(v') \geq y$. Which means that $f(v') + \varepsilon \geq g(y)$. Thus $f(x) - f(v') \leq 2\varepsilon$.

  We enumerate all the possible situations for v - clearly the same hold for v':

  - $v \in \pi_T(C)$: then $a = v$;
  - $\Lambda(v) = 0$ then $a = v''$ with $(v'', \eta(v)) \in C$; by hypothesis, $\varphi(v) \leq x$ and $v'' < x$;
  - $a \in \Lambda(v)$ if $v \notin \pi_T(C)$ and $\#\Lambda(v) > 0$.
- Now we prove (P3). If $w \notin Im(\alpha_C^\varepsilon)$ then $w \in D_C^G$. In fact if $(x, w') \in C$ with $w' < w$, then $[w', w] \subset Im(\alpha_C^\varepsilon)$, since there are $l(w) \geq w'$ and $u(w) \leq r_G$. But then we know that there is $w'' = \min\{y \in V_G | y > w \text{ and } y \notin D_C^G\}$ with $g(w'') - g(w) \leq \varepsilon$.

■

Proof of Corollary 6.21.

We know that $w := \alpha(v) \leq w' := \alpha(v')$, thus $\max\{y \in V_G \,|\, y \geq w\} \leq \max\{y \in V_G \,|\, y \geq w'\}$ and the thesis follows.

■

Proof of Theorem 6.22.

The proof is similar to the proof of Theorem 1 in Chapter 4, but we report it because we need some subtle modifications due to the differences between the *mappings*, defined by Chapter 4, and the couplings we use.

We build C by subsquently adding couples starting from an empty set. The proof is divided in sections which should help the reader in following the various steps.

6.B Leaves of T

In this section we take care of the leaves of the merge tree T.

6.B.1 Selecting the Coupled Leaves

We consider the following set of leaves:

$$\mathcal{L}_T = \{v \in L_T \,|\, \nexists v' \in L_T \text{ such that } \alpha(v) < \alpha(v')\} \tag{6.17}$$

We give a name to the condition:

(*a*) $\nexists v' \in L_T$ such that $\alpha(v) < \alpha(v')$

so that we can more easily use it during the proof. Note that we can avoid treating the case $\alpha(v) = \alpha(v')$ thanks to (G).

The set $\mathcal{L}_T$ is the set of leaves which will coupled by C: we add to C all the couples of the form $(v, \phi(v))$ with $v \in \mathcal{L}_T$. We characterize those couples with the following proposition.

Lemma 6.35. *Given $v, v' \in \mathcal{L}_T$, then $\phi(v) \geq \phi(v')$ if and only if $v = v'$. Moreover, for every $v' \in L_T$ such that (a) does not hold, there is $v \in \mathcal{L}_T$ such that $\alpha(v) < \alpha(v')$.*

Proof. The first part of the proof reduces to observing that $\phi(v) \leq \phi(v')$ if and only if $\alpha(v) \leq \alpha(v')$.

Now consider $v' \in L_T$ such that (a) does not hold. We know there is v_0 such that $\alpha(v_0) < \alpha(v')$. If $v_0 \in \mathcal{L}_T$ we are done, otherwise there is v_1 such that $\alpha(v_1) < \alpha(v_0) < \alpha(v')$. Note that $f(v_1) < f(v_0)$. Thus we can carry on this procedure until we find $v_i \in \mathcal{L}_T$. Note that $\arg\min_{V_T} f \in \mathcal{L}_T$, thus, in a finite number of step we are done. □

6.B.2 Cost Bound on Couples

Now we want to prove the following proposition which gives an upper bound for the cost of the couples added to C.

Lemma 6.36. *Given $v \in \mathcal{L}_T$, then $|\, f(v) - g(\phi(v)) \,| \leq \varepsilon$.*

Proof. Suppose the thesis does not hold. Since $g(\phi(v)) \leq f(v) + \varepsilon$, contradicting the thesis means that we have $v \in \mathcal{L}_T$ such that:

(*b*) $g(\phi(v)) + \varepsilon < f(v)$.

Let $w = \phi(v)$. If (b) holds, then $g(father(w)) - g(w) > g(\alpha(v)) - g(w) > 2\varepsilon$. Let $v' = \psi(w) \leq \beta(w)$. Note that $f(v') < f(v)$. We have $\phi(v') \leq \alpha(v') \leq \alpha(\beta(w)) = s_G^{2\varepsilon}(w)$. But since $g(father(w)) - g(w) > 2\varepsilon$, we also have $\alpha(v') \leq \alpha(v)$ with $v' \neq v$ which is absurd by Lemma 6.35. □

6.B.3 Cost Bound on Deletions

In this step we prove the following proposition which gives an analogous bound to the one of Lemma 6.36, but for the deleted leaves of T.

Lemma 6.37. *Given $v \notin \mathcal{L}_T$, then there exists $x > v$ such that:*

- *there is $v' < x$ such that $v' \in \mathcal{L}_T$;*
- $f(x) \leq f(v) + 2\varepsilon$.

Proof. Since (a) does not hold for v, we use Lemma 6.35 to obtain $v' \in \mathcal{L}_T$ such that $\alpha(v') < \alpha(v)$. But being α an ε-good map, we have $s_T^{2\varepsilon}(v') \leq s_T^{2\varepsilon}(v)$ which implies $f(\mathrm{LCA}(v, v')) \leq f(v) + 2\varepsilon$. Thus $x = \mathrm{LCA}(v, v')$ ends the proof. □

Lemma 6.37 implies that, using the notation of the proposition, $\varphi(v) \leq x$. Then $f(v) < f(v')$ implies that $g(\phi(v)) \leq f(v') + \varepsilon < f(v) + \varepsilon$. Thus $g(\eta(v)) < f(v) + \varepsilon$. Since $f(x) \leq f(v) + 2\varepsilon$ we have that the cost of deleting any $x' < x$ with $\#\Lambda(x') = 0$ is less then ε.

6.C Leaves of G

The result we need in this section is the following.

Lemma 6.38. *Given $w \in L_G$, there exist $y \geq w$ such that:*

- *there is $w' = \phi(v)$ with $v \in \mathcal{L}_T$ and $w' < y$;*
- $g(y) \leq g(w) + 2\varepsilon$.

Proof. Consider $\beta(w)$. Let $v \leq \beta(w)$ leaf. We have $\alpha(\beta(w)) \geq \mathrm{LCA}(\alpha(v), w)$. If $v \in \mathcal{L}_T$ we are done for $\phi(v) \leq \alpha(v)$. If (a) does not hold by Lemma 6.35 it means that there is $v' \in \mathcal{L}_T$ such that $\alpha(v') < \alpha(v)$. We are done since $g(\alpha(\beta(w))) = g(w) + 2\varepsilon$. □

Note that if $w < \phi(v)$ for some $v \in \mathcal{L}_T$, then, by Lemma 6.38, $g(\phi(v)) \leq g(w) + 2\varepsilon$. In fact, using the notation of Lemma 6.38, in this case we have $w' = y = \phi(v)$ by definition. As in Section 6.B.3, we have that the cost of the deletion of any $w < y$ such that $\#\Lambda(w) = 0$ is at most ε.

6.D Internal Vertices

Now we need extend the coupling C taking into account the internal vertices of T. We will do so after simplifying our merge trees in two different ways: first we remove all vertices which are deleted with $\#\Lambda(p) = 0$ and then we take out all inessential internal vertices.

6.D.1 Pruning

Let $T_0 = T$ and $G_0 = G$. We define T_1 as the merge tree obtained from T_0 deleting the following set of vertices (and the corresponding edges): x such that $\#\Lambda(x) = 0$ and $x \notin \pi_T(C)$. Note that, for any $x \in V_T$ either there is $v \leq x$ with $v \in \mathcal{L}_T$ or for any leaf below x, we can apply Lemma 6.37 and the consequential observations.

The tree G_1 is obtained from G_0 in an analogous way: any time we have $w \in V_G$ with $\#\Lambda(w) = 0$ and $w \notin \pi_G(C)$, w is deleted from G_0, along with the edge $(w, father(w))$.

Before proceeding we point out that, by construction, the leaves of T_1 are exactly $\mathcal{L}_T$.

6.D.2 Restricting α

Thanks to Corollary 6.21 we have that, anytime we delete some vertex in G_0 to obtain G_1 and that vertex is in the image of ϕ, we are sure that also its counterimage is deleted from T_0. Now, for every $v \in \mathbf{T}$, by construction we have that $\alpha(v)$ belongs to an edge removed from G_0 if and only if $\phi(v)$ is deleted from G_0. Let $v' = L(v) \in V_T$. Then $\phi(v') \leq \phi(v)$ and thus $\phi(v')$ is deleted as well. Which entails that v' is deleted as well. All of this, put together implies that we can restrict α to $\mathbf{T}_1$ (the metric tree obtained from T_1) and its image lies in $\mathbf{G}_1$ (obtained from G_1).

6.D.3 ε-Good Restriction

We define $\alpha_1 := \alpha_{|\mathbf{T}_1} : \mathbf{T}_1 \to \mathbf{G}_1$. We want to prove that α_1 is still an ε-good map. Clearly $(P1)$ and $(P2)$ still hold upon restricting the domain. We just need to show $(P3)$. Suppose there is $w \in V_{G_0}$ such that $w \notin \alpha_1(\mathbf{T}_1)$. We distinguish between two cases: (1) $w \notin \phi(\mathbf{T}_1)$ and (2) $w \in \phi(\mathbf{T}_1)$. Consider scenario (1): $w \notin \phi(\mathbf{T}_1)$ clearly implies that $\#\Lambda(w) = 0$ and so w is deleted; scenario (2) instead means that there is $\alpha(v) = \min\{\alpha(v') > w \,|\, v' \in \mathbf{T}_1\}$ with $L(\alpha(v)) = \phi(v) = L(w)$. Clearly $\{w' \in \mathbf{G}_0 \,|\, w' < w\} \bigcap \alpha(\mathbf{T}_1) = \emptyset$ and thus v is a leaf of T_1, which means $v \in \mathcal{L}_T$. This also implies that $w \notin \alpha(\mathbf{T})$ and in particular $\alpha(v) = \min\{\alpha(v') > w \,|\, v' \in \mathbf{T}_0\}$, thus condition $(P3)$ is satisfied.

6.D.4 Properties of $\phi : \mathbf{T}_1 \to \mathbf{G}_1$ and Removal of Inessential Vertices

We know that $\phi : L_{T_1} \to L_{G_1}$ is injective. On top of that we have proved that, if $w \notin \alpha_1(\mathbf{T}_1)$ then $L(w) = \phi(v)$ for some $v \in \mathcal{L}_T$. Thus $\phi : L_{T_1} \to L_{G_1}$ is a bijection.

From now on we will ignore any vertex v such that $\#child(v) = 1$. Formally, we introduce T_2 (and G_2) obtained from T_1 (G_1) removing all the vertices $v \in V_{T_1}$ such that $\#child(v) = 1$ (similarly $w \in V_{G_1}$ such that $\#child(w) = 1$): consider $\{v'\} = child(v)$. We remove v from V_{T_1} and replace the edges (v', v) and $(v, father(v))$ with the edge $(v', father(v))$. We do this operation recursively until no vertices such that $\#child(v) = 1$ can be found.

6.D.5 Coupling and Deleting the internal vertices

We start with the following lemma.

Lemma 6.39. *For every $x \in V_{T_2}$, and $y \in V_{G_2}$, we have $|\, x - \chi(x) \,| \leq \varepsilon$ and $|\, y - \chi(y) \,| \leq \varepsilon$.*

Proof. Let $x \in V_{T_2} - L_{T_2}$. Then $x = \mathrm{LCA}(v_1, \ldots, v_n)$ with $v_1, \ldots, v_n$ being the leaves of $sub_{T_2}(x)$. Then $\alpha(x) > \alpha(v_i)$ for every i and thus $\alpha(x) \leq w = \chi(x) = \mathrm{LCA}(\alpha(v_1), \ldots, \alpha(v_n))$. Clearly $x \leq \beta(w)$ for analogous reasons. Thus $|\, x - w \,| \leq \varepsilon$. For $y \in V_{G_2}$ we can make an analogous proof. □

Let $v_1, \ldots, v_n \in L_{T_2}$ and $\phi(v_1), \ldots, \phi(v_n) \in L_{G_2}$. Let $x = \mathrm{LCA}(v_1, \ldots, v_n)$ and $y = \mathrm{LCA}(\phi(v_1), \ldots, \phi(v_n))$. We know that $\alpha(x) \geq y$ and $\beta(y) \geq x$ and so $|\, f(x) - g(y) \,| \leq \varepsilon$.

Thus we proceed as follows: we order all the internal vertices of T_2 according to their height values and we start from the higher one (the root of T_2 - note that this in general is not the root of T_0), coupling it with the other root (thus $(C1)$ is verified). Consider a lower x and let $v_1, \ldots, v_n$ be the leaves of $sub_{T_2}(x)$ (and thus $x = \mathrm{LCA}(v_1, \ldots, v_n)$). Let $w = \mathrm{LCA}(\phi(v_1), \ldots, \phi(v_n))$. If the leaves of $sub_{G_2}(w)$ are $\phi(v_1), \ldots, \phi(v_n)$ we add the couple (x, w) otherwise we skip x - which will be deleted, with $\#\Lambda(x) > 1$. Note that $w = \chi(x)$. Moreover, by Lemma 6.39, $|\, f(x) - g(w) \,| \leq \varepsilon$. Going from the root downwards we repeat this procedure for every x.

We clearly have:

- that (C3) is satisfied;
- $cost((v,w)) \leq \varepsilon$;
- if $v \in V_{T_2}$ is deleted then $\#\Lambda(v) > 1$ and $cost(v) = |\, f(v) - g(\chi(v)) \,| \leq \varepsilon$;
- if $w \in V_{G_2}$ is deleted then $\#\Lambda(w) > 1$ and $cost(w) = |\, g(w) - f(\chi(w)) \,| \leq \varepsilon$ by Lemma 6.39.

6.E Coupling Properties and Costs

First we verify that C is a coupling:

($C1$) verified. See Section 6.D.5;

($C2$) verified. See Section 6.B and Section 6.D.5: we carefully designed the coupling so that no vertex is coupled two times;

($C3$) verified. It is explicitly proven in Section 6.D.5;

($C4$) In Section 6.D.4 we remove all vertices such that $\#child(x) = 1$ to obtainT_2 and G_2; all the couples in C are either leaves of vertices in T_2 and G_2. So, since all leaves of T_2 and G_2 are coupled its impossible to have a vertex p belonging to any tree such that $\#\Lambda(p) = 1$.

Lastly we verify its costs:

- if $(x,y) \in C$ then $|\, f(x) - g(y) \,| \leq \varepsilon$ as verified in Section 6.B.2 for $x \in L_T$ and in Section 6.D.5 for x being an internal vertex;
- if $x \in V_T \cap D$, with $\#\Lambda(x) = 0$, it is verified that $|\, f(x) - \phi(x) \,| \leq 2\varepsilon$ in Section 6.B.3; if instead $y \in V_G \cap D$ we verify $|\, g(y) - \phi(y) \,| \leq 2\varepsilon$ in Section 6.C; in both cases we have a vertex lower that x (y) which, by construction, it is coupled. And the cost of the couple is less then ε. Thus the deletion of x (y) costs less than or equal to ε;
- if $p \in D$, with $\#\Lambda(p) > 1$, then we verify in Section 6.D.5 that the cost of this deletion is less than or equal to ε.

This concludes the proof.

■

Proof of Theorem 6.26.

Given $C \in \mathcal{C}(T,G)$ optimal coupling such that $\{(r,r')\} = \max C$, we define $\mathcal{M}(C) := \max(C - \{(r,r')\})$. Clearly $\mathcal{M}(C) \in \mathcal{C}^*(T,G)$. For any $(x,y) \in \mathcal{M}(C)$, consider $C^o_{x,y} \in \mathcal{C}^o_R(T_x, G_y)$ such that $\| C^o_{x,y} \|_\infty \leq \| C_{|\,(x,y)} \|_\infty$ with $C_{|\,(x,y)} := \{(v,w) \in C \,|\, (v,w) < (x,y)\}$. By **??** and **??** we know that $C^o_{x,y}$ exists for every x, y. Note that $C_{|\,(x,y)} \in \mathcal{C}_R(T_x, G_y)$.

We want to prove that the extension $C' := \{(r,r')\} \bigcup_{(x,y)\in\mathcal{M}(C)} C^o_{x,y}$ satisfies $\| C' \|_\infty \leq \| C \|_\infty$. The set C' clearly is a coupling since for every $x, x' \in \pi_T(\mathcal{M}(C))$, $sub_T(x)$ and $sub_T(x')$ are disjoint. And the same for $y, y' \in \pi_G(\mathcal{M}(C))$. We need to consider the costs of different kinds of vertices separately. In particular we indicate with (a) whenever we have $v \in V_T$ such that $v \leq x$ for some $x \in \pi_T(\mathcal{M}(C))$. If this condition does not hold we say that (b) holds for v. The same definitions apply also for vertices in G. For instance (a) holds for all vertices in $\pi_T(\mathcal{M}(C))$ and $\pi_G(\mathcal{M}(C))$. Similarly, if (b) holds for v, then it holds also for all $v' > v$.

- Consider $v \in V_T$ such that (a) holds for some $(x, y) \in \mathcal{M}(C)$; we have that $\chi_{C'}(v) \leq y$, $\varphi_{C'}(v) \leq x$ and $\eta_{C'}(v) \leq y$, so $cost(v)$ depends only on $C^o_{(x,y)}$. Thus $cost_{C'}(v) = cost_{C^o_{(x,y)}}(v) \leq \| C_{|(x,y)} \|$ by construction;

- suppose now (b) holds for $v \in V_T$; in this case we have by construction $\chi_C(v) = \chi_{C'}(v)$, $\Lambda_{C'}(v) = \Lambda_C(v)$ and thus $\varphi_C(v) = \varphi_{C'}(v)$. Suppose now $\Lambda_{C'}(v) = 0$: (a) holds for $\eta_{C'}(v)$ by means of some $y \in \pi_G(\mathcal{C}(M))$ and thus $g(\eta_{C'}(v)) = \min_{w' \leq y} g(w') \leq g(\eta_C(v))$ (thanks to the properties of couplings in $\mathcal{C}^o_R(T_x, G_y)$). Since $\varphi_C(v) = \varphi_{C'}(v)$ we then have $cost_{C'}(v) \leq cost_C(v)$. Lastly consider $\#\Lambda_{C'}(v) > 1$. In this case we have $\chi_C(v) = \chi_{C'}(v) = r'$ and thus $cost_{C'}(v) = cost_C(v)$. Note that in all these situations, $cost_{C'}(v)$ does not depend on the chosen extension.

Exchanging the role of T and G we obtain the same results for the vertices of G. To conclude it is enough to observe that we can always choose $\| C^o_{x,y} \|_\infty \leq \| C_{|(x,y)} \|_\infty$ to obtain an optimal extension and, moreover, $\| E^o(C^*) \|_\infty$ is well defined for any optimal extension.

■

Proof of Lemma 6.27.

1. This is due to the fact that all vertices of $P\varepsilon(T)$ are coupled, $P\varepsilon(T)$ is still a rooted tree and (P) does not alter the order relationships of the remaining vertices;

2. since (P) removes at most one leaf from the previous tree and adds no new leaves, it is clear that $L_{P_\varepsilon(T)} \subset L_T$. Consider now $l \in L_T$, $v_- = \max\{v' \in V_T \,|\, v \geq l \text{ and } f(v') < f(l) + \varepsilon\}$ and $v_+ = \min\{v' \in V_T \,|\, v \geq l \text{ and } f(v') \geq f(l) + \varepsilon\}$. Note that $(v_+, v_-) \in E_T$. By construction $v' = \arg\min_{v'' \leq v_+} f(v'')$ is either the last point below v_+ to be deleted or is in $L_{P_\varepsilon(T)}$. Suppose it is the last point below x_+ to be removed by (P). By construction, when (P) removes v', $f(father(v')) - f(v') < \varepsilon$, but, since all other vertices below x_+ have been deletex $father(v') \geq x_-$ and so $f(father(v')) - f(v') \geq \varepsilon$, which is absurd. Exploiting this fact, also this point is proven;

3. consider $v \in V_T - V_{P_\varepsilon(T)}$; let $\Lambda_{C_\varepsilon}(v) = \{a_1, \ldots, a_n\}$. By construction $v \geq x = \text{LCA}(\Lambda_{C_\varepsilon}(v))$. We know that a vertex is removed from T if, at a certain point along the recursive application of (P), it becomes an order 2 vertex or a small-weight leaf. Thus the vertex x is not removed from T unless $n \leq 1$. Which is absurd because then $\Lambda_{C_\varepsilon}(v) = \{x\}$. Thus $\#\Lambda(v) \leq 1$. As a consequence, $v \in D$ if and only if $\#\Lambda(v) = 0$.

   The vertex $\varphi_{C_\varepsilon}(v)$ is the first vertex x above v such that there is a leaf v', $v' < x$, with $f(x) - f(v') \geq \varepsilon$. If $f(x) - f(v) \geq \varepsilon$ then by point (2) there exist $v' \in L_{P_\varepsilon(T)}$ with $\text{LCA}(v', v) < \varphi_{C_\varepsilon}(v)$ which is absurd;

4. consider two leaves $l, l' \in L_T$ with $f(l) < f(l')$, with $l \notin V_{P_\varepsilon(T)}$ and $l' \in V_{P_\varepsilon(T)}$. On top of that suppose $l' < \varphi_{C_\varepsilon}(l)$.By point (3) we have $f(\varphi_{C_\varepsilon}(l)) - f(l') < \varepsilon$ which means that l' is a small-weight leaf. Which is absurd;

5. combining points (3) and (4) we see that we only have deletions with $\#\Lambda_{C_\varepsilon}(v) = 0$ and $2 \cdot cost_{C_\varepsilon}(c) = f(\varphi_{C_\varepsilon}(v)) - f(v) \leq \varepsilon$.

■

Proof of Proposition 6.28.

Consider $C \in \mathcal{C}^o(P_\varepsilon(T), P_\varepsilon(G))$. Then C can be seen also as a coupling $C \in \mathcal{C}^o(T, G)$. We partition the vertices V_T into three sets:

- the case $v \in V_{P_\varepsilon(T)}$ is not a concern since the cost doesn't change when considering $v \in V_T$ or $v \in V_{P_\varepsilon(T)}$;
- if $v \in U^T_{C_\varepsilon}$ then $v \in U^T_C$ or D^T_C; in the first case we ignore it, in the second case there is $\{v'\} = \max\{v'' < v \,|\, v'' \in V_{P_\varepsilon(T)}\}$, such that $v' \in D^T_C$. And so $cost_C(v) < cost_C(v')$;
- if $v \in D^T_{C_\varepsilon}$, then $v \in D^T_C$ and $\#\Lambda_C(v) = 0$; by construction $\varphi_{C_\varepsilon}(v) \leq \varphi_C(v)$. If $\varphi_{C_\varepsilon}(v) < \varphi_C(v)$ then also η_{C_ε} is deleted and, since $f(\eta_{C_\varepsilon}(v)) < f(v)$, $cost_C(v) < cost_C(\eta_{C_\varepsilon}(v))$.

  So we are left with the case $\varphi_{C_\varepsilon}(v) = \varphi_C(v)$. If $g(\eta_C(v)) - f(v) > 0.5 \cdot (f(\varphi_C(v) - f(v)))$ then, again, $cost_C(v) < cost_C(\eta_{C_\varepsilon}(v))$, for $f(\eta_{C_\varepsilon}(v)) < f(v)$. Thus we always have $cost_C(v) \leq \max\{\varepsilon/2, cost_C(\eta_{C_\varepsilon}(v))\}$.

Exchanging the role of T and G and repeating the same observations, we obtain that, if we consider $C \in \mathcal{C}^o(T, G)$, we have $d_I(T, G) \leq \| C \|_\infty \leq \max\{d_I(P_\varepsilon(T), P_\varepsilon(G)), \varepsilon/2\}$.

■

Proof of Lemma 6.30.

- if $v \leq r$ then $r' \geq \chi(v), \varphi(v)$ and $r' \geq \eta(v)$, so $cost(v)$ depends only on $C_{r,r'}$;
- if $v \not\leq r$ then it is either unused, if $v > r$, or it is deleted with $\#\Lambda(v) = 0$. Then the cost of such deletion is either $0.5(f(v_r) - f(v))$ or $g(\eta(v)) - f(v)$. Since $C_{r,r'} \in \mathcal{C}^o_R(T_r, G_{r'})$, $\eta(v) = g_r$ and we finish the proof.

■

Proof of Proposition 6.31.

We just explore the different pieces of the cost function to assess the thesis:

- in the case of $(x, y) \in C(\mathcal{V})$, the cost of coupling T_x and G_y is given by $\sum_y a_{x,y} \| C_{x,y} \|_\infty$;
- if $u_x = 1$ we have $| g(r_G) - f(x) | u_x$ which is the cost of deleting x with $\#\Lambda(x) > 1$. That is, any time $x \geq x'$ with $x' = \mathrm{LCA}(v, v')$ for $v, v' \in \pi_T(C(\mathcal{V}))$. Recall that, in this case, we have $\chi(x) \leq r_G$;
- deleting x with $\#\Lambda(x) = 0$ is taken care by remaining part the cost function. For a vertex x lets indicate with x_f its father and set $A_x = 0.5(f(x_f) - f_x)(1 - d_x)$ and $B_x = \{\sum_y a_{v,y}\,(g_y - f_x) - K d_x \,|\, v < x_f\}$. Now we try to unveil the meaning of A_x and B_x. Recall that deleting x with $\#\Lambda(x) = 0$ corresponds to having $d_x = 0$. If $d_x = 1$, then $A_x = 0$ and $\max B_x < 0$; while if $d_x > 1$, both A_x and $\max B_x$ are negative. Consider now $d_x = 0$. Then $\varphi_{C^*}(x) = x'_f$, with x'_f being father of some $x' \geq x$. Clearly $d_{x'} = 0$ as well and $A_{x'} > A_x$. In particular $A_{x'} = \max_{v \leq x'} 0.5\,(f(\varphi_{C^*}(v)) - f_v)$. Now, we turn to B_x. If $x < x'$ then $x_f < \varphi_{C^*}(x) = x'_f$, and so $d_{x_f} = 0$, entailing $\max B_x = \min B_x = 0$. Instead, if $x = x'$, we have by construction $v < x'_f$ with $a_{v,y} = 1$. Then $\max\{\sum_y a_{v,y} g_y \,|\, v < x_f\} \geq g(\eta_{C^*}(x'))$ and so:

$$A_{x'} \leq \max_{v \leq x'} cost_{C^*}(v) \leq \max\{A_{x'}, \max B_{x'}\}. \tag{6.18}$$

■

Proof of Corollary 6.33.

Given $C^* \in \mathcal{C}^*(T, G)$, with $r = \text{LCA}(\pi_T(C^*)$ and $r' = \text{LCA}(\pi_G(C^*)$, thanks to Corollary 6.32 we can approximate with $\Gamma^{\uparrow}$ and $\Gamma^{\downarrow}$ the costs of the vertices in T_r and $T_{r'}$. By Lemma 6.30, $H_{r,r'}$ then takes care of the vertices $v \not\leq r$ and $w \not\leq r'$ and we can approximate $\| E^o(C^*) \|_\infty$. With Theorem 6.26 we conclude the proof.

■

7. Conclusion

Throughout the chapters of this dissertation we explore the use of a tree-shaped topological summary called merge tree. With the help of previous literature on the topic and original developments carried out in this thesis, we show that these objects have a lot of potential the case study at hand can be tackled with tools related to topological data analysis. In particular we develop a framework to work with functions defined on merge trees - which can be used to greatly enrich the information contained in such topological summaries. Then, we define a novel metric structure for merge trees, structure which posseses some stability properties and can be computed exactly with a recursive optimization procedure. This metric is put to work on a benchmark functional data analysis case study were classical techniques must resort to non-trivial alignment procedures to work with functions up to re-parametrizations. Similarly, we use (a modified version of) such distance to perform an unsupervised analysis of tree-based patient representations arising from a radiomic data set, a research field which is fundamental for non-invasive methods in cancer treatment. In both cases the metric proves to be useful, matching the performance of persistence diagrams in one situation and, in the second, capturing a meaningful clustering that can be interpreted in terms of clinical variables. Stability properties always guarantee the interpretability of the employed pipelines.

In general, TDA's approach - i.e. representing data by means of topological summaries - has proven to be very effective when data need to be considered up to coarse equivalence classes. However, a greater level of flexibility could surely help when a finer level of detail is needed, as shown by some pipelines which can indeed provide an injective operator mapping data to representations (Amezquita et al., 2020). The use of merge trees over persistence diagrams and the definition of functions to enrich these tree-shaped summaries goes exactly in this direction.

There are, however, some clear drawbacks characterizing the novelties proposed in this manuscript, which are stated multiple times throughout the chapters:

- the space in which TDA's topological summaries live is often very irregular, and the metric space of merge trees we define is no exception. Statistical methods which can be meaningfully employed in such spaces are thus limited by the mathematical pathologies which such spaces present: unbound curvature, "diffuse" non uniqueness of geodesics, non-uniqueness of Frechét means etc. make uncertainty quantification a daunting challenge;

- the computational complexity of dealing with merge trees becomes prohibitive quickly as either the size of the data set or of the trees increase. In literature some methods to overcome such computational barriers have been proposed but at the expenses of stability or reliability of the approximation scheme. Our edit distance can be exactly computed but remains suited for small data sets and with trees of small size.

Natural further developments of the present work could go in the directions of tackling both the aforementioned problems:

- despite the technical difficulties presented by badly behaved metric spaces, we believe that some statistical tools like Frechét means, principal component analysis and regression could be defined also for trees. In particular we would like to test and study the approximation scheme proposed for Frechét means and to define l_1-principal component analysis and linear regression, via the local approximation obtained in Chapter 3;

- similarly, the attempt developed in Chapter 4 to study merge trees via tree-related statistics proves to have a lot of potential and needs a dedicated study to better understand how the extracted features interact with the proposed edit distance. Note that such statistical summaries are much more computationally accessible than most of the metrics defined for merge trees.

Lastly, in this thesis we did not carry out any extensive case study relying on the use of functions defined on merge trees. We are strongly convinced that the idea of decomposing measure-related or homological information via the merging structure of path connected components can improve the range of situations that can be tackled via topological tools. Moreover, we would really like to improve our framework to make it *intrinsic*, so that, when editing the local representation of a function, at any step, we always obtain a local representation of another function.

To summarize, in this thesis we have considered a particular topological summary - which has been used and appreciated also in areas outside applied topology - and we have made a series of theoretical developments and applications aimed at building a framework which allows the analysis of populations of such objects. We have drawn comparisons with other attempts pursuing the same goal, highlighting pros and cons of the different works. In the process, we have used a number of topological, categorical, metric, combinatorial and optimization tools which show how the interplay of different fields of mathematics can contribute to the development of data analysis techniques. These, in turn, can be used to tackle challenges arising in the most diverse fields of science, especially the ones requiring a collaborative and multidisciplinary approach due to their complexity.

Bibliography

Adams, H., T. Emerson, M. Kirby, R. Neville, C. Peterson, P. Shipman, S. Chepushtanova, E. Hanson, F. Motta, and L. Ziegelmeier (2017). Persistence images: A stable vector representation of persistent homology. *J. Mach. Learn. Res. 18*, 8:1–8:35.

Agarwal, P. K., K. Fox, A. Nath, A. Sidiropoulos, and Y. Wang (2018, apr). Computing the gromov-hausdorff distance for metric trees. *ACM Trans. Algorithms 14*(2).

Amezquita, E. J., M. Quigley, T. Ophelders, J. Landis, E. Munch, D. Chitwood, and D. Koenig (2020). Quantifying barley morphology using the euler characteristic transform. In *TDA & Beyond*.

Arsigny, V., P. Fillard, X. Pennec, and N. Ayache (2006). Log-euclidean metrics for fast and simple calculus on diffusion tensors. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine 56*(2), 411–421.

Arya, S., J. Curry, and S. Mukherjee (2022). A sheaf-theoretic construction of shape space. *arXiv 2204.09020*.

Audin, M., M. Damian, and R. Erné (2014). *Morse theory and Floer homology*. Springer.

Balk, S. P., Y.-J. Ko, and G. J. Bubley (2003). Biology of prostate-specific antigen. *Journal of clinical oncology 21*(2), 383–391.

Bauer, U., C. Landi, and F. Mémoli (2020). The reeb graph edit distance is universal. *Foundations of Computational Mathematics*, 1–24.

Bauer, U., E. Munch, and Y. Wang (2014). Strong equivalence of the interleaving and functional distortion metrics for reeb graphs. *arXiv 1412.6646*.

Beketayev, K., D. Yeliussizov, D. Morozov, G. H. Weber, and B. Hamann (2014). Measuring the distance between merge trees. In *Topological Methods in Data Analysis and Visualization*.

Berkouk, N. (2021). Algebraic homotopy interleaving distance. In F. Nielsen and F. Barbaresco (Eds.), *Geometric Science of Information*, Cham, pp. 656–664. Springer International Publishing.

Bhattacharya, S., R. Ghrist, and V. Kumar (2015, 06). Persistent homology for path planning in uncertain environments. *IEEE Transactions on Robotics 31*, 1–13.

Biasotti, S., D. Giorgi, M. Spagnuolo, and B. Falcidieno (2008, 02). Reeb graphs for shape analysis and applications. *Theoretical Computer Science 392*, 5–22.

Bille, P. (2005). A survey on tree edit distance and related problems. *Theoretical computer science 337*(1-3), 217–239.

Billera, L. J., S. P. Holmes, and K. Vogtmann (2001). Geometry of the space of phylogenetic trees. *Advances in Applied Mathematics 27*(4), 733–767.

Bjerkevik, H. B., M. B. Botnan, and M. Kerber (2020). Computing the interleaving distance is np-hard. *Foundations of Computational Mathematics 20*(5), 1237–1271.

Bock, A., H. Doraiswamy, A. Summers, and C. Silva (2017, 08). Topoangler: Interactive topology-based extraction of fishes. *IEEE Transactions on Visualization and Computer Graphics PP*, 1–1.

Bône, A. (2020). *Learning adapted coordinate systems for the statistical analysis of anatomical shapes. Applications to Alzheimer's disease progression modeling.* Ph. D. thesis, Sorbonne Université.

Bridson, M. R. and A. Haefliger (2013). *Metric spaces of non-positive curvature*, Volume 319. Springer Science & Business Media.

Bronstein, M. M., J. Bruna, T. Cohen, and P. Veličković (2021). Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv 2104.13478*.

Bubenik, P. (2015). Statistical topological data analysis using persistence landscapes. *Journal of Machine Learning Research 16*(3), 77–102.

Bubenik, P. and A. Wagner (2020). Embeddings of persistence diagrams into hilbert spaces. *Journal of Applied and Computational Topology 4*(3), 339–351.

Burago, D., Y. Burago, and S. Ivanov (2022). *A course in metric geometry*, Volume 33. American Mathematical Society.

Burggraaff, C. N., F. Rahman, I. Kaßner, S. Pieplenbosch, S. F. Barrington, Y. W. Jauw, G. J. Zwezerijnen, S. Müller, O. S. Hoekstra, J. M. Zijlstra, et al. (2020). Optimizing workflows for fast and reliable metabolic tumor volume measurements in diffuse large b cell lymphoma. *Molecular imaging and biology 22*(4), 1102–1110.

Calissano, A., A. Feragen, and S. Vantini (2020). Populations of unlabeled networks: Graph space geometry and geodesic principal components. *MOX Reports*.

Cardona, R., J. Curry, T. Lam, and M. Lesnick (2021). The universal ℓ^p-metric on merge trees. *arXiv 2112.12165 [cs.CG]*.

Carlsson, G., T. Ishkhanov, V. De Silva, and A. Zomorodian (2008). On the local behavior of spaces of natural images. *International journal of computer vision 76*(1), 1–12.

Carlsson, G. and F. Mémoli (2013). Classifying clustering schemes. *Foundations of Computational Mathematics 13*.

Castellano, G., L. Bonilha, L. Li, and F. Cendes (2004). Texture analysis of medical images. *Clinical radiology 59*(12), 1061–1069.

Cavinato, L., M. Pegoraro, A. Ragni, and F. Ieva (2022). Imaging-based representation and stratification of intra-tumor heterogeneity via tree-edit distance. *arXiv 2208.04829[stat.ME]*.

Cavinato, L., M. Sollini, M. Kirienko, M. Biroli, F. Ricci, L. Calderoni, E. Tabacchi, C. Nanni, P. L. Zinzani, S. Fanti, et al. (2020). Pet radiomics-based lesions representation in hodgkin lymphoma patients. In *The 50th Scientific Meeting of the Italian Statistical Society*, pp. 474–479.

Ceriani, L., G. Gritti, L. Cascione, M. C. Pirosa, A. Polino, T. Ruberto, A. Stathis, A. Bruno, A. A. Moccia, L. Giovanella, et al. (2020). Sakk38/07 study: integration of baseline metabolic heterogeneity and metabolic tumor volume in dlbcl prognostic model. *Blood advances 4*(6), 1082–1092.

Chacón, J. E. (2021). A close-up comparison of the misclassification error distance and the adjusted rand index for external clustering evaluation. *British Journal of Mathematical and Statistical Psychology 74*(2), 203–231.

Chaddad, A., T. Niazi, S. Probst, F. Bladou, M. Anidjar, and B. Bahoric (2018). Predicting gleason score of prostate cancer patients using radiomic analysis. *Frontiers in oncology 8*, 630.

Chakraborty, R. and B. C. Vemuri (2015). Recursive Fréchet mean computation on the Grassmannian and its applications to computer vision. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4229–4237. IEEE.

Chazal, F., D. Cohen-Steiner, M. Glisse, L. Guibas, and S. Oudot (2008, 12). Proximity of persistence modules and their diagrams. *Proc. 25th ACM Sympos. Comput. Geom.*.

Chazal, F., V. De Silva, M. Glisse, and S. Oudot (2016). *The structure and stability of persistence modules*. Springer.

Chazal, F., B. T. Fasy, F. Lecci, A. Rinaldo, and L. Wasserman (2015). Stochastic convergence of persistence landscapes and silhouettes. *J. Comput. Geom. 6*, 140–161.

Chazal, F. and B. Michel (2017). An introduction to topological data analysis: fundamental and practical aspects for data scientists. *arXiv 1710.04019*.

Che, M., F. Galaz-García, L. Guijarro, and I. M. Solis (2021). Metric geometry of spaces of persistence diagrams. *arXiv 2109.14697v1 [math.MG]*.

Chicklore, S., V. Goh, M. Siddique, A. Roy, P. K. Marsden, and G. J. Cook (2013). Quantifying tumour heterogeneity in 18 f-fdg pet/ct imaging by texture analysis. *European journal of nuclear medicine and molecular imaging 40*(1), 133–140.

Chung, M. K., P. Bubenik, and P. T. Kim (2009). Persistence diagrams of cortical surface data. In J. L. Prince, D. L. Pham, and K. J. Myers (Eds.), *Information Processing in Medical Imaging*, Berlin, Heidelberg, pp. 386–397. Springer Berlin Heidelberg.

Cohen-Steiner, D., H. Edelsbrunner, and J. Harer (2007). Stability of persistence diagrams. *Discrete & Computational Geometry 37*, 103–120.

Cohen-Steiner, D., H. Edelsbrunner, J. Harer, and Y. Mileyko (2010, 02). Lipschitz functions have lp-stable persistence. *Foundations of Computational Mathematics 10*, 127–139.

Colijn, C. and G. Plazzotta (2018). A metric on phylogenetic tree shapes. *Systematic Biology 67*(1), 113–126.

Cottereau, A.-S., C. Nioche, A.-S. Dirand, J. Clerc, F. Morschhauser, O. Casasnovas, M. Meignan, and I. Buvat (2020). 18f-fdg pet dissemination features in diffuse large b-cell lymphoma are predictive of outcome. *Journal of Nuclear Medicine 61*(1), 40–45.

Culp, M. B., I. Soerjomataram, J. A. Efstathiou, F. Bray, and A. Jemal (2020). Recent global patterns in prostate cancer incidence and mortality rates. *European urology 77*(1), 38–52.

Cummings, M. C., P. T. Simpson, L. E. Reid, J. Jayanthan, J. Skerman, S. Song, A. E. McCart Reed, J. R. Kutasovic, A. L. Morey, L. Marquart, et al. (2014). Metastatic progression of breast cancer: insights from 50 years of autopsies. *The Journal of pathology 232*(1), 23–31.

Curry, J. (2018). The fiber of the persistence map for functions on the interval. *Journal of Applied and Computational Topology 2*.

Curry, J., J. DeSha, A. Garin, K. Hess, L. Kanari, and B. Mallery (2021). From trees to barcodes and back again ii: Combinatorial and probabilistic aspects of a topological inverse problem. *arXiv 2107.11212*.

Curry, J., H. Hang, W. Mio, T. Needham, and O. B. Okutan (2022). Decorated merge trees for persistent topology. *Journal of Applied and Computational Topology*.

Curry, J. M. (2014). *Sheaves, cosheaves and applications*. University of Pennsylvania.

Curto, C. (2016, 05). What can topology tell us about the neural code? *Bulletin of the American Mathematical Society 54*.

Davis, B. C. (2008). *Medical image analysis via Fréchet means of diffeomorphisms*.

De Boor, C. and J. W. Daniel (1974). Splines with nonnegative b-spline coefficients. *Mathematics of computation 28*(126), 565–568.

De Silva, V., E. Munch, and A. Patel (2016). Categorified reeb graphs. *Discrete & Computational Geometry 55*(4), 854–906.

de Silva, V., E. Munch, and A. Stefanou (2017). Theory of interleavings on categories with a flow. *arXiv 1706.04095 [math.CT]*.

Di Fabio, B. and C. Landi (2016). The edit distance for reeb graphs of surfaces. *Discrete & Computational Geometry 55*(2), 423–461.

Draisma, G., R. Postma, F. H. Schröder, T. H. van der Kwast, and H. J. de Koning (2006). Gleason score, age and screening: modeling dedifferentiation in prostate cancer. *International journal of cancer 119*(10), 2366–2371.

Dryden, I. L. and K. V. Mardia (1998). *Statistical Shape Analysis*. Chichester: Wiley.

Dupuis, P., U. Grenander, and M. I. Miller (1998). Variational problems on flows of diffeomorphisms for image matching. *Quarterly of Applied Mathematics 56*(3), 587–600.

Eary, J. F., F. O'Sullivan, J. O'Sullivan, and E. U. Conrad (2008). Spatial heterogeneity in sarcoma 18f-fdg uptake as a predictor of patient outcome. *Journal of Nuclear Medicine 49*(12), 1973–1979.

Edelsbrunner, H. and J. Harer (2008). Persistent homology-a survey. *Contemporary mathematics 453*, 257–282.

Edelsbrunner, H., D. Letscher, and A. Zomorodian (2002). Topological persistence and simplification. *Discrete & Computational Geometry 28*, 511–533.

Eertink, J. J., T. van de Brug, S. E. Wiegers, G. J. Zwezerijnen, E. A. Pfaehler, P. J. Lugtenburg, B. van der Holt, H. C. de Vet, O. S. Hoekstra, R. Boellaard, et al. (2022). 18f-fdg pet baseline radiomics features improve the prediction of treatment outcome in diffuse large b-cell lymphoma. *European journal of nuclear medicine and molecular imaging 49*(3), 932–942.

Elkin, Y. and V. Kurlin (2020). The mergegram of a dendrogram and its stability. *ArXiv 2007.11278*.

Emmett, K., B. Schweinhart, and R. Rabadan (2015). Multiscale topology of chromatin folding. In *BICT*.

Epstein, J. I., L. Egevad, M. B. Amin, B. Delahunt, J. R. Srigley, and P. A. Humphrey (2016). The 2014 international society of urological pathology (isup) consensus conference on gleason grading of prostatic carcinoma. *The American journal of surgical pathology 40*(2), 244–252.

Epstein, J. I., M. J. Zelefsky, D. D. Sjoberg, J. B. Nelson, L. Egevad, C. Magi-Galluzzi, A. J. Vickers, A. V. Parwani, V. E. Reuter, S. W. Fine, et al. (2016). A contemporary prostate cancer grading system: a validated alternative to the gleason score. *European urology 69*(3), 428–435.

Esparza-López, J., E. Escobar-Arriaga, S. Soto-Germes, and M. de Jesús Ibarra-Sánchez (2017). Breast cancer intra-tumor heterogeneity: one tumor, different entities. *Revista de investigacion clinica 69*(2), 66–76.

Fasy, B., F. Lecci, A. Rinaldo, L. Wasserman, S. Balakrishnan, and A. Singh (2014, 03). Confidence sets for persistence diagrams. *The Annals of Statistics 42*, 2301–2339.

Felsenstein, J. and J. Felenstein (2004). *Inferring phylogenies*, Volume 2. Sinauer associates Sunderland, MA.

Feragen, A., P. Lo, M. de Bruijne, M. Nielsen, and F. Lauze (2012, 12). Toward a theory of statistical tree-shape analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Ferraty, F. and P. Vieu (2006). *Nonparametric functional data analysis: theory and practice*. Springer Science & Business Media.

Fisher, R., L. Pusztai, and C. Swanton (2013). Cancer heterogeneity: implications for targeted therapeutics. *British journal of cancer 108*(3), 479–485.

Flesia, A. (2009). Unsupervised classification of tree structured objects. In *BIOMAT 2008*, pp. 280–299. World Scientific.

Fletcher, P. (2013, 11). Geodesic Regression and the Theory of Least Squares on Riemannian Manifolds. *International Journal of Computer Vision 105*.

Gameiro, M., Y. Hiraoka, S. Izumi, M. Kramár, K. Mischaikow, and V. Nanda (2014, 03). A topological measurement of protein compressibility. *Japan Journal of Industrial and Applied Mathematics 32*, 1–17.

Garba, M. K., T. M. Nye, J. Lueg, and S. F. Huckemann (2021). Information geometry for phylogenetic trees. *Journal of Mathematical Biology 82*(3), 1–39.

Gasparovic, E., E. Munch, S. Oudot, K. Turner, B. Wang, and Y. Wang (2019). Intrinsic interleaving distance for merge trees. *ArXiv 1908.00063*.

Ghosal, S. and A. van der Vaart (2017). *Fundamentals of Nonparametric Bayesian Inference*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press.

Gillies, R. J., P. E. Kinahan, and H. Hricak (2016). Radiomics: images are more than pictures, they are data. *Radiology 278*(2), 563–577.

Giovacchini, G., M. Picchio, A. Briganti, C. Cozzarini, V. Scattoni, A. Salonia, C. Landoni, L. Gianolli, N. Di Muzio, P. Rigatti, et al. (2010). [11c] choline positron emission tomography/computerized tomography to restage prostate cancer cases with biochemical failure after radical prostatectomy and no disease evidence on conventional imaging. *The Journal of urology 184*(3), 938–943.

Giusti, C., R. Ghrist, and D. Bassett (2016, 01). Two's company, three (or more) is a simplex: Algebraic-topological tools for understanding higher-order structure in neural data. *Journal of Computational Neuroscience 41*.

Hatcher, A. (2000). *Algebraic topology*. Cambridge: Cambridge Univ. Press.

Hein, J., T. Jiang, L. Wang, and K. Zhang (1995). On the complexity of comparing evolutionary trees. In Z. Galil and E. Ukkonen (Eds.), *Combinatorial Pattern Matching*, Berlin, Heidelberg, pp. 177–190. Springer Berlin Heidelberg.

Hofer, C., R. Kwitt, M. Niethammer, and A. Uhl (2017). Deep learning with topological signatures. In *NIPS*.

Hong, E., Y. Kobayashi, and A. Yamamoto (2017). Improved methods for computing distances between unordered trees using integer programming. In *International Conference on Combinatorial Optimization and Applications*, pp. 45–60. Springer.

Horváth, L. and P. Kokoszka (2012). *Inference for functional data with applications*, Volume 200. Springer Science & Business Media.

Huckemann, S., T. Hotz, and A. Munk (2010). Intrinsic shape analysis: Geodesic pca for riemannian manifolds modulo isometric lie group actions. *Statistica Sinica*, 1–58.

Huckemann, S. F. and B. Eltzner (2018). Backward nested descriptors asymptotics with inference on stem cell differentiation. *The Annals of Statistics 46*(5), 1994–2019.

James, I. M. (1976). *The topology of Stiefel manifolds*, Volume 24. Cambridge University Press.

Jr., J. H. W. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association 58*(301), 236–244.

Jung, S., I. L. Dryden, and J. S. Marron (2012). Analysis of principal nested spheres. *Biometrika 99*(3), 551–568.

Kanari, L., A. Garin, and K. Hess (2020). From trees to barcodes and back again: Theoretical and statistical perspectives. *Algorithms 13*(12).

Karcher, H. (1977). Riemannian center of mass and mollifier smoothing. *Communications on pure and applied mathematics 30*(5), 509–541.

Kendall, D. G. (1977). The diffusion of shape. *Advances in Applied Probability*.

Kendall, D. G. (1984). Shape Manifolds, Procrustean Metrics, and Complex Projective Spaces. *Bulletin of the London Mathematical Society*.

Kettenring, J. R. (1971). Canonical analysis of several sets of variables. *Biometrika 58*(3), 433–451.

Khan, K., S. U. Rehman, K. Aziz, S. Fong, and S. Sarasvady (2014). Dbscan: Past, present and future. In *The fifth international conference on the applications of digital information and web technologies (ICADIWT 2014)*, pp. 232–238. IEEE.

Kim, J., N. A. Rosenberg, and J. A. Palacios (2020). Distance metrics for ranked evolutionary trees. *Proceedings of the National Academy of Sciences 117*(46), 28876–28886.

Kovacev-Nikolic, V., P. Bubenik, D. Nikolic, and G. Heo (2016, 03). Using persistent homology and dynamical distances to analyze protein binding. *Statistical applications in genetics and molecular biology 15*, 19–38.

Kramár, M., A. Goullet, L. Kondic, and K. Mischaikow (2013, 04). Persistence of force networks in compressed granular media. *Physical review. E, Statistical, nonlinear, and soft matter physics 87*, 042207.

Lacombe, T., M. Cuturi, and S. Oudot (2018). Large scale computation of means and clusters for persistence diagrams using optimal transport. In *NeurIPS*.

Lavine, B. and J. Workman (2008). Chemometrics. *Analytical chemistry 80*(12), 4519–4531.

Lesnick, M. (2015). The theory of the interleaving distance on multidimensional persistence modules. *Foundations of Computational Mathematics 15*(3), 613–650.

Lewitus, E. and H. Morlon (2016). Characterizing and comparing phylogenies from their laplacian spectrum. *Systematic biology 65*(3), 495–507.

Leygonie, J., M. Carrière, T. Lacombe, and S. Oudot (2021). A gradient sampling algorithm for stratified maps with applications to topological data analysis. *arXiv 2109.00530*.

Leygonie, J., S. Oudot, and U. Tillmann (2021). A framework for differential calculus on persistence barcodes. *Foundations of Computational Mathematics*, 1–63.

Li, Y., F.-X. Wu, and A. Ngom (2018). A review on machine learning principles for multi-view biological data integration. *Briefings in bioinformatics 19*(2), 325–340.

Lum, P., G. Singh, A. Lehman, T. Ishkanov, M. Alagappan, J. Carlsson, G. Carlsson, and M. Vejdemo Johansson (2013, February). Extracting insights from the shape of complex data using topology. *Scientific Reports 3*.

Mac Lane, S. (1998). *Categories for the Working Mathematician*. Springer Science+Business Media New York 1978: Springer New York, NY.

MacPherson, R. and B. Schweinhart (2010, 11). Measuring shape with topology. *Journal of Mathematical Physics 53*.

Maletić, S., Y. Zhao, and M. Rajkovic (2015, 10). Persistent topological features of dynamical systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science 26*.

Marron, J. S., J. Ramsay, L. Sangalli, and A. Srivastava (2014). Statistics of time warpings and phase variations. *Electronic Journal of Statistics 8*, 1697–1702.

Marron, J. S., J. Ramsay, L. Sangalli, and A. Srivastava (2015). Functional data analysis of amplitude and phase variation. *Statistical Science 30*(4), 468–484.

Mayerhoefer, M. E., A. Materka, G. Langs, I. Häggström, P. Szczypiński, P. Gibbs, and G. Cook (2020). Introduction to radiomics. *Journal of Nuclear Medicine 61*(4), 488–495.

Mémoli, F. (2008). Gromov-hausdorff distances in euclidean spaces. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1–8. IEEE.

Mémoli, F., Z. Wan, and Y. Wang (2022). Persistent laplacians: Properties, algorithms and implications. *SIAM Journal on Mathematics of Data Science 4*(2), 858–884.

Michor, P., D. Mumford, J. Shah, and L. Younes (2007, 07). A metric on shape space with explicit geodesics. *Atti Accad. Naz. Lincei Cl. Sci. Fis. Mat. Natur. Rend. Lincei (9) Mat. Appl. 19*.

Mileyko, Y., S. Mukherjee, and J. Harer (2011, 12). Probability measures on the space of persistence diagrams. *Inverse Problems - INVERSE PROBL 27*.

Miller, E., M. Owen, and J. S. Provan (2015). Polyhedral computational geometry for averaging metric phylogenetic trees. *Advances in Applied Mathematics 68*, 51 – 91.

Milnor, J. (2016). Morse theory.(am-51), volume 51. In *Morse Theory.(AM-51), Volume 51*. Princeton university press.

Moakher, M. and M. Zéraï (2011). The riemannian geometry of the space of positive-definite matrices and its application to the regularization of positive-definite matrix-valued data. *Journal of Mathematical Imaging and Vision 40*(2), 171–187.

Morozov, D., K. Beketayev, and G. Weber (2013, 01). Interleaving distance between merge trees. *Discrete and Computational Geometry 49*, 22–45.

Morozov, D. and G. Weber (2013). Distributed merge trees. In *PPoPP '13*.

Munkres, J. R. (2018). *Elements of algebraic topology*. CRC press.

Murtagh, F. and P. Contreras (2017, 09). Algorithms for hierarchical clustering: An overview, ii. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 7*.

Naitzat, G., A. Zhitnikov, and L.-H. Lim (2020). Topology of deep neural networks. *Journal of Machine Learning Research 21*(184), 1–40.

Nguyen, T., Q.-H. Pham, T. Le, T. Pham, N. Ho, and B.-S. Hua (2021). Point-set distances for learning representations of 3d point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10478–10487.

Nioche, C., F. Orlhac, S. Boughdad, S. Reuzé, J. Goya-Outi, C. Robert, C. Pellot-Barakat, M. Soussan, F. Frouin, and I. Buvat (2018). Lifex: a freeware for radiomic feature calculation in multimodality imaging to accelerate advances in the characterization of tumor heterogeneity. *Cancer research 78*(16), 4786–4789.

Ost, P., K. Decaestecker, B. Lambert, V. Fonteyne, L. Delrue, N. Lumen, F. Ameye, and G. De Meerleer (2014). Prognostic factors influencing prostate cancer-specific survival in non-castrate patients with metastatic prostate cancer. *The Prostate 74*(3), 297–305.

Pal, S., T. J. Moore, R. Ramanathan, and A. Swami (2017). Comparative topological signatures of growing collaboration networks. In B. Gonçalves, R. Menezes, R. Sinatra, and V. Zlatic (Eds.), *Complex Networks VIII*, Cham, pp. 201–209. Springer International Publishing.

Pascucci, V. and K. Cole-McLaughlin (2003, 10). Parallel computation of the topology of level sets. *Algorithmica 38*, 249–268.

Patel, A. (2018). Generalized persistence diagrams. *Journal of Applied and Computational Topology 1*.

Patrangenaru, V. and L. Ellingson (2015). *Nonparametric Statistics on Manifolds and Their Application to Object Data Analysis*. CRC Press.

Pegoraro, M. (2021a). *Geometric data analysis: between equivalence classes and non-euclidean spaces*. Ph. D. thesis, Politecnico di Milano.

Pegoraro, M. (2021b). A graph-matching formulation of the interleaving distance between merge trees. *arXiv 2111.15531[math.co]*.

Pegoraro, M. (2021c). A locally stable edit distance for functions defined on merge trees. *arXiv 2108.13108v2 [math.CO]*.

Pegoraro, M. (2021d). A locally stable stable edit distance for merge trees. *arXiv 2111.02738 [math.GN]*.

Pegoraro, M. and M. Beraha (2022). Projected statistical methods for distributional data on the real line with the wasserstein metric. *Journal of Machine Learning Research 23*(37), 1–59.

Pegoraro, M. and P. Secchi (2021). Functional data representation with merge trees. *arXiv 2108.13147v2 [stat.ME]*.

Pennec, X. (2006). Intrinsic Statistics on Riemannian Manifolds: Basic Tools for Geometric Measurements. *Journal of Mathematical Imaging and Vision 25*, 127–154.

Pennec, X. (2018). Barycentric subspace analysis on manifolds. *The Annals of Statistics 46*(6A), 2711–2746.

Pennec, X., S. Sommer, and T. Fletcher (2019). *Riemannian Geometric Statistics in Medical Image Analysis*. Academic Press.

Perea, J., A. Deckard, S. Haase, and J. Harer (2015, 08). Sw1pers: Sliding windows and 1-persistence scoring; discovering periodicity in gene expression time series data. *BMC bioinformatics 16*, 257.

Perea, J. and J. Harer (2013, 07). Sliding windows and persistence: An application of topological methods to signal analysis. *Foundations of Computational Mathematics 15*.

Pezaro, C., H. H. Woo, and I. D. Davis (2014). Prostate cancer: measuring psa. *Internal medicine journal 44*(5), 433–440.

Pigoli, D., J. A. Aston, I. L. Dryden, and P. Secchi (2014). Distances and inference for covariance operators. *Biometrika 101*(2), 409–422.

Pini, A. and S. Vantini (2017). Interval-wise testing for functional data. *Journal of Nonparametric Statistics 29*(2), 407–424.

Pokorny, F., M. Hawasly, and S. Ramamoorthy (2015, 08). Topological trajectory classification with filtrations of simplicial complexes and persistent homology. *The International Journal of Robotics Research 35*.

Pont, M., J. Vidal, J. Delon, and J. Tierny (2022). Wasserstein distances, geodesics and barycenters of merge trees. *IEEE Transactions on Visualization and Computer Graphics 28*(1), 291–301.

Poon, A. F., L. W. Walker, H. Murray, R. M. McCloskey, P. R. Harrigan, and R. H. Liang (2013). Mapping the shapes of phylogenetic trees from human and zoonotic rna viruses. *PLoS one 8*(11), e78122.

Ramsay, J. O. and B. W. Silverman (2005). *Functional Data Analysis*. New York, NY, USA: Springer.

Ripley, B. and U. Grenander (1995, 01). General pattern theory: A mathematical theory of regular structures. *Journal of the Royal Statistical Society. Series A (Statistics in Society) 158*, 635.

Rizvi, A., P. Camara, E. Kandror, T. Roberts, I. Schieren, T. Maniatis, and R. Rabadan (2017, 05). Single-cell topological rna-seq analysis reveals insights into cellular differentiation and development. *Nature Biotechnology 35*.

Robinson, D. F. and L. R. Foulds (1979). Comparison of weighted labelled trees. In *Combinatorial mathematics VI*, pp. 119–126. Springer.

Rockafellar, R. T. and R. J.-B. Wets (2009). *Variational analysis*, Volume 317. Springer Science & Business Media.

Sangalli, L., P. Secchi, and S. Vantini (2014). Analysis of aneurisk65 data: k-mean alignment. *Electronic Journal of Statistics 8*, 1891–1904.

Sangalli, L., P. Secchi, S. Vantini, and A. Veneziani (2009a, 07). Efficient estimation of three-dimensional curves and their derivatives by free-knot regression splines, applied to the analysis of inner carotid artery centrelines. *Journal of the Royal Statistical Society Series C 58*, 285–306.

Sangalli, L., P. Secchi, S. Vantini, and V. Vitelli (2010, 05). K-mean alignment for curve clustering. *Computational Statistics & Data Analysis 54*, 1219–1233.

Sangalli, L. M., P. Secchi, S. Vantini, and A. Veneziani (2009b). A case study in exploratory functional data analysis: Geometrical features of the internal carotid artery. *Journal of the American Statistical Association 104*(485), 37–48.

Scoccola, L. and J. A. Perea (2022). Fiberwise dimensionality reduction of topologically complex data with vector bundles. *arXiv 2206.06513*.

Scolamiero, M., W. Chachólski, A. Lundman, R. Ramanujam, and S. Öberg (2017). Multidimensional persistence and noise. *Foundations of Computational Mathematics 17*(6), 1367–1406.

Shinagawa, Y., T. L. Kunii, and Y. L. Kergosien (1991a). Surface coding based on morse theory. *IEEE Computer Graphics and Applications 11*(5), 66–78.

Shinagawa, Y., T. L. Kunii, and Y. L. Kergosien (1991b). Surface coding based on morse theory. *IEEE Computer Graphics and Applications 11*(5), 66–78.

Siegel, D. A., M. E. O'Neil, T. B. Richards, N. F. Dowling, and H. K. Weir (2020). Prostate cancer incidence and survival, by stage and race/ethnicity—united states, 2001–2017. *Morbidity and Mortality Weekly Report 69*(41), 1473.

Silva, V. and R. Ghrist (2007, 04). Coverage in sensor networks via persistent homology. *Algebraic & Geometric Topology 7*.

Sizemore, A., C. Giusti, and D. Bassett (2015, 12). Classification of weighted networks through mesoscale homological features. *Journal of Complex Networks 5*.

Smith, C. P., M. Czarniecki, S. Mehralivand, R. Stoyanova, P. L. Choyke, S. Harmon, and B. Turkbey (2019). Radiomics and radiogenomics of prostate cancer. *Abdominal Radiology 44*(6), 2021–2029.

Smith, D. C., R. L. Dunn, M. S. Strawderman, and K. J. Pienta (1998). Change in serum prostate-specific antigen as a marker of response to cytotoxic therapy for hormone-refractory prostate cancer. *Journal of clinical oncology 16*(5), 1835–1843.

Smith, M. R. (2019). Bayesian and parsimony approaches reconstruct informative trees from simulated morphological datasets. *Biology letters 15*(2), 20180632.

Smith, M. R. (2020). Information theoretic generalized robinson–foulds metrics for comparing phylogenetic trees. *Bioinformatics 36*(20), 5007–5013.

Smith, P. and V. Kurlin (2022). Families of point sets with identical 1d persistence. *arXiv 2202.00577 [cs.CG]*.

Sollini, M., F. Bartoli, L. Cavinato, F. Ieva, A. Ragni, A. Marciano, R. Zanca, L. Galli, F. Paiar, F. Pasqualetti, et al. (2021). [18f] fmch pet/ct biomarkers and similarity analysis to refine the definition of oligometastatic prostate cancer. *EJNMMI research 11*(1), 1–10.

Sollini, M., M. Kirienko, L. Cavinato, F. Ricci, M. Biroli, F. Ieva, L. Calderoni, E. Tabacchi, C. Nanni, P. L. Zinzani, et al. (2020). Methodological framework for radiomics applications in hodgkin's lymphoma. *European Journal of Hybrid Imaging 4*, 1–17.

Sørensen, S. S., C. A. Biscio, M. Bauchy, L. Fajstrup, and M. M. Smedskjaer (2020). Revealing hidden medium-range order in amorphous materials using topological data analysis. *Science Advances 6*(37), eabc2320.

Sridharamurthy, R., T. B. Masood, A. Kamakshidasan, and V. Natarajan (2020). Edit distance between merge trees. *IEEE Transactions on Visualization and Computer Graphics 26*(3), 1518–1531.

Srivastava, A., I. Jermyn, and S. H. Joshi (2007). Riemannian analysis of probability density functions with applications in vision. *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 1–8.

Srivastava, A., E. Klassen, S. Joshi, and I. Jermyn (2010, 09). Shape analysis of elastic curves in euclidean spaces. *IEEE transactions on pattern analysis and machine intelligence*.

Srivastava, A., W. Wu, S. Kurtek, E. Klassen, and J. S. Marron (2011a). Registration of functional data using fisher-rao metric. *arXiv 1103.3817v2 [math.ST]*.

Srivastava, A., W. Wu, S. Kurtek, E. Klassen, and J. S. Marron (2011b). Registration of functional data using fisher-rao metric. *arXiv*.

Stanta, G. and S. Bonin (2018). Overview on clinical relevance of intra-tumor heterogeneity. *Frontiers in medicine 5*, 85.

Stark, J. R., S. Perner, M. J. Stampfer, J. A. Sinnott, S. Finn, A. S. Eisenstein, J. Ma, M. Fiorentino, T. Kurth, M. Loda, et al. (2009). Gleason score and lethal prostate cancer: does 3+ 4= 4+ 3? *Journal of Clinical Oncology 27*(21), 3459.

Stefanou, A. (2020). Tree decomposition of reeb graphs, parametrized complexity, and applications to phylogenetics. *Journal of Applied and Computational Topology 4*(2), 281–308.

Sturm, K. T. (2003). Probability measures on metric spaces of nonpositive curvature. *Heat Kernels and Analysis on Manifolds, Graphs, and Metric Spaces 338*, 357–391.

Tai, K.-C. (1979, July). The tree-to-tree correction problem. *J. ACM 26*, 422–433.

Touli, E. F. (2020). Frechet-like distances between two merge trees. *ArXiv 2004.10747*.

Touli, E. F. and Y. Wang (2018). Fpt-algorithms for computing gromov-hausdorff and interleaving distances between trees. In *ESA*.

Tralie, C. J. (2017). *Geometric Multimedia Time Series*. Ph. D. thesis, Duke University.

Trouvé, A. (1998). Diffeomorphisms groups and pattern matching in image analysis. *International Journal of Computer Vision 28*(3), 213–221.

Turaga, P., A. Veeraraghavan, A. Srivastava, and R. Chellappa (2011). Statistical computations on grassmann and stiefel manifolds for image and video-based recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence 33*(11), 2273–2286.

Turner, K., Y. Mileyko, S. Mukherjee, and J. Harer (2012, 06). Frechet means for distributions of persistence diagrams. *Discrete & Computational Geometry 52*.

Turner, K., Y. Mileyko, S. Mukherjee, and J. Harer (2014). Fréchet means for distributions of persistence diagrams. *Discrete & Computational Geometry 52*(1), 44–70.

Vantini, S. (2009, 01). On the definition of phase and amplitude variability in functional data analysis. *Test 21*, 1–21.

Wang, H. and J. S. Marron (2007, 10). Object oriented data analysis: Sets of trees. *Ann. Statist. 35*(5), 1849–1873.

Wang, Y., H. Ombao, and M. Chung (2018, 09). Topological data analysis of single-trial electroencephalographic signals. *The Annals of Applied Statistics 12*, 1506–1534.

Wetzels, F., H. Leitte, and C. Garth (2022). Branch decomposition-independent edit distances for merge trees. *Comput. Graph. Forum 41*(3), 367–378.

wu, K. and S. Zhang (2013, 01). A contour tree based visualization for exploring data with uncertainty. *International Journal for Uncertainty Quantification 3*, 203–223.

Xia, K., X. Feng, Y. Tong, and G. We (2015, 03). Persistent homology for the quantitative prediction of fullerene stability. *Journal of computational chemistry 36*.

Xia, K., Z. Li, and L. Mu (2016, 12). Multiscale persistent functions for biomolecular structure characterization. *Bulletin of Mathematical Biology 80*.

Xu, B., C. J. Tralie, A. Antia, M. Lin, and J. A. Perea (2019). Twisty takens: a geometric characterization of good observations on dense trajectories. *Journal of Applied and Computational Topology 3*(4), 285–313.

Xu, D. and Y. Tian (2015, 08). A comprehensive survey of clustering algorithms. *Annals of Data Science 2*.

y Cajal, S. R., M. Sesé, C. Capdevila, T. Aasen, L. De Mattos-Arruda, S. J. Diaz-Cano, J. Hernández-Losa, and J. Castellví (2020). Clinical implications of intratumor heterogeneity: challenges and opportunities. *Journal of Molecular Medicine 98*(2), 161–177.

Younes, L. (1998). Computable elastic distances between shapes. *SIAM Journal on Applied Mathematics 58*(2), 565–586.

Yu, Q., X. Lu, and J. S. Marron (2013). Principal nested spheres for time-warped functional data analysis. *Journal of Computational and Graphical Statistics 26*, 144 – 151.

Zomorodian, A. and G. on (2005, 02). Computing persistent homology. *Discrete and Computational Geometry 33*, 249–274.

8. Research Summary for the Year 2021-2022

As requested by Universitá di Bologna, I close my thesis with a summary of my research activities for the time-window in which I have been enrolled in the PhD program of "Data Science and Computation", that is, the academic year 2021-2022.

Ricerca Svolta

Elenco brevemente i miei principali temi di ricerca.

Introduzione

La mia ricerca si divide in due aree principali: l'utilizzo di rappresentazioni ad albero per fare analisi dati e l'utilizzo di metriche di Wasserstein per fare analisi di dataset di distribuzioni. La tesi di dottorato, in particolare, raccoglie tutti i lavori realizzati sul tema alberi.

I principali tipi di dato che possono essere rappresentati con delle strutture ad albero chiamate *merge trees* (Beketayev et al., 2014) sono due: funzioni a valori in $\mathbb{R}$ e nuvole di punti in spazi metrici. Questo tipo di approccio alla rappresentazione dei dati viene inquadrato in maniera naturale all'interno della *topological data analysis* (TDA) (Edelsbrunner and Harer, 2008). In particolare, la rappresentazione tramite merge tree di ogni dato iniziale rappresenta un'alternativa a quanto viene fatto molto spesso in TDA, in cui ogni osservazione viene rappresentata tramite un *diagramma di persistenza* (PD) (Edelsbrunner and Harer, 2008) e l'analisi viene successivamente svolta sui diagrammi ottenuti. Uno dei punti di maggior interesse di questo tipo di rappresentazioni é che risultano avere delle forti proprietá di invarianza rispetto a trasformazioni del dato iniziale. Questo tipo di proprietá é di grande interesse in statistica ed ha sempre guidato l'evoluzione di tutta quella parte di letteratura scientifica che si dedica alla *shape analysis* (Kendall, 1977).

La rappresentazione tramite merge trees presenta delle importanti differenze rispetto a quella tramite diagrammi di persistenza, differenze che comportano dei vantaggi e degli svantaggi: informalmente parlando, i merge trees sono in grado di discriminare situazioni che i diagrammi di persistenza non riescono a distinguere, anche se il costo computazionale derivante dall'utilizzo degli alberi é quasi sempre molto elevato.

Su questo tema il mio contributo si articola nei seguenti punti:

1. la definizione di un framework metrico per lavorare su funzioni definite su merge trees (Pegoraro, 2021c); queste funzioni possono essere utilizzate per lavorare con alberi arricchiti di ulteriori informazioni riguardanti il dato iniziale;

2. la definizione di una nuova metrica per merge trees e lo studio dello spazio metrico risultante, per ottenere proprietá che consentano lo sviluppo di tecniche statistiche nello spazio degli alberi (Pegoraro, 2021d).

3. lo studio dell'utilizzo della rappresentazione ad albero per dati di tipo funzionale (Pegoraro and Secchi, 2021), con tutte problematiche relative allo smoothing di dati

funzionali e alla continuitá di questa rappresentazione rispetto al dato iniziale, con la metrica definita in Pegoraro (2021c);

4. una nuova formulazione per la *interleaving distance* tra merge trees, con la quale si ottiene un nuovo algoritmo per l'approssimazione della distanza;

5. un'applicazione dei merge trees per l'analisi di un data set di dati radiomici: il dato di ogni paziente, che contiene tutte le lesioni tumorali dello stesso, viene rappresentato tramite un dendrogramma gerarchico. Il risultante data set di dendrogrammi viene poi analizzato con una versione modificata della metrica definita in Pegoraro (2021d), adattata al problema considerato.

Anno Accademico 2021-2022

Durante l'anno accademico 2021-2022 ho svolto le seguenti attivitá di ricerca:

- Realizzazione dell'articolo "A Graph-Matching Formulation of the Interleaving Distance between Merge Trees" (Pegoraro, 2021b): viene studiata la interleaving distance tra merge trees, la distanza piú utilizzata per lavorare da un punto di vista teorico sui merge trees, formulandola come un problema di matching tra i grafi dei due alberi. Si ottiene cosí un algoritmo ricorsivo basato su tecniche di ottimizzazione intera in grado di ottenere un'approssimazione per eccesso di questa distanza. Vi é al momento solo un'altra implementazione disponibile per approssimare la interleaving distance tra alberi e viene effettuato un paragone tra gli errori ottenuti con i due approcci: l'approccio perseguito in Pegoraro (2021b), seppur computazionalmente molto piú oneroso dell'altro, produce errori notevolmente minori, a volte anche oltre 1000 volte minori.

- Realizzazione dell'articolo "Imaging-based representation and stratification of intratumor Heterogeneity via tree-edit distance" (Cavinato et al., 2022): in questo articolo, realizzato in collaborazione con Lara Cavinato (PhD student del Politecnico di Milano), Francesca Ieva (Professore Associato del Politecnico di Milano) e Alessandra Ragni (PhD student del Politecnico di Milano) utilizziamo una versione modificata della distanza tra alberi (dendrogrammi gerarchici) definita in Pegoraro (2021c), per analizzare un data set contenente dati radiomici, cercando di catturare la variabilitá tra i pazienti causata dall'eterogeneitá delle lesioni tumorali. Stratificando i pazienti in base al livello di eterogeneitá delle lesioni tumorali sfruttando una rappresentazione ad albero del singolo paziente e una rivisitazione della metrica tra alberi definita in Pegoraro (2021c), otteniamo una clusterizzazione dei pazienti in cui si leggono differenze significative in termini di distribuzione di diverse variabili cliniche.

- Rivisitazione degli articoli Pegoraro (2021c) e Pegoraro (2021d): il primo articolo ora é completamente incentrato sulla definizione di un framework per analizzare funzioni definite su merge trees. Tutta la parte teorica riguardante filtrazioni, abstract merge trees e display posets, la notazione e la definizione di spazi funzionali su merge trees sono completamente nuove. Il secondo articolo ora contiene la definizione di una nuova metrica tra merge trees, una approfondita analisi comparativa tra le diverse metriche esistenti, evidenziando pro e contro di ogni approccio e un nuovo risultato di decomposizione per le geodesiche che permette si avere una approssimazione locale dello spazio degli alberi via $\mathbb{R}^n$ e uno schema numerico per l'approssimazione delle medie alla Frechét.

- Rivisitazione dell'articolo "Functional Data Representation with Merge Trees" (Pegoraro and Secchi, 2021): in seguito ad una review ricevuta da una rivista scientifica, l'articolo é stato oggetto di profondi cambiamenti che includono nuovi risultati teorici, una nuova e piú fruibile dimostrazione del teorema principale e una nuova scrittura di alcuni paragrafi che necessitavano argomentazioni piú forti o complete. Sono stati inoltre aggiunti nuovi paragrafi nel materiale supplementare.

- Parziale scrittura dell'articolo "Wasserstein Distributional Data Analysis on the Circumference": con Mario Beraha (PhD student del Corso di Dottorato Data Science and Computation, Universitá di Bologna) proponiamo un framework per fare analisi statistica (PCA e regressione "lineare") di distribuzioni su uno spazio base che é la circonferenza, utilizzando la metrica di 2-Wasserstein. L'articolo é ancora da terminarsi, ma tutti i risultati necessari per la costruzione del framework sono stati ottenuti. Non ci sono noti altri casi studio di dataset di distribuzioni (utilizzando strumenti di trasporto ottimo) in cui lo spazio base non é euclideo.

Conferenze, Scuole Estive, Seminari Anno Accademico 2021-2022

- "Data Analysis with Tree-Shaped Topological Summaries", 2022, Applied Topology Seminars, EPFL;

- "Data Analysis with Merge Trees", 2022, Mathematics for Complex Data, KTH Stockholm (poster);

- "Functional Data Representation with Merge Trees", 2022, IFCS2022 : Classification and Data Analysis in the Digital Age, Porto (invited talk);

- "Wasserstein Distributional Data Analysis on the Circumference", ECDA 2022, Napoli (invited talk).

Pubblicazioni su Riviste Anno Accademico 2021-2022

- "Projected Statistical Methods for Distributional Data on the Real Line with the Wasserstein Metric"(Pegoraro and Beraha, 2022).