ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

DOTTORATO DI RICERCA IN

MATEMATICA

Ciclo 37

Settore Concorsuale: 01/A3 - ANALISI MATEMATICA, PROBABILITÀ' E STATISTICA MATEMATICA

Settore Scientifico Disciplinare: MAT/05 - ANALISI MATEMATICA

A MACHINE LEARNING APPROACH IN COASTAL ECOLOGY AND THEORETICAL ADVANCEMENTS IN KERNEL BASED RANDOM FORESTS

Presentata da: Isidoros Iakovidis

Coordinatore Dottorato

Giovanni Mongardi

Supervisore

Nicola Arcozzi

Esame finale anno 2025

Abstract

This PhD thesis is part of a PON scholarship DOT1303154-3 *Dottorati PON - Bando 2021 - Cycle 37 (XXXVII) - Action IV.5 - Doctorates on Green topics*. In the first part of the thesis, an application is provided of machine learning algorithms in the ecological coastal coasts. In the second part we examine thoroughly and in depth the mathematical properties of some of the machinery used in the first part, providing theoretical improvements of the models.

This thesis is joint work with N.Arcozzi and F.Bozzeda in the following articles and preprints.

1. Iakovidis Isidoros, Nicola Arcozzi. Improved convergence rates for some kernel random forest algorithms[J]. Mathematics in Engineering, 2024, 6(2): 305-338. doi: 10.3934/mine.2024013 [43]
2. A simplified directional KeRF algorithm. (with N.Arcozzi) [44]
3. On the ecology modeling of coastal beaches. (with F.Bozzeda)

Contents

Introduction

This PhD thesis is developed under the PON scholarship PON (Programmi Operativi Nazionali) DOT1303154-3 *Dottorati PON - Bando 2021 - Cycle 37 (XXXVII) - Action IV.5 - Doctorates on Green topics* supported by the Italian Ministry of Education and Merit, focusing on Innovation and Green topics. The National Operational Program (PON-green) aims to provide funds for research activities regarding green transition, ecosystem preservation, and reduction of climate change impacts.

Among the most important effects of climate change is the increase in frequency and intensity of violent atmospheric events (IPCC 2023)[62]. Coastal areas are highly populated and important from both economic and ecological perspectives (for example food production), making them a key focus of modern ecological research. Coastal ecology has been defined as the study of the environment that connects the land and the sea.

The initial focus of the PhD thesis study is benthic organisms inhabiting sandy beaches as a key component of sandy beach systems. Benthos communities play an important role as bioindicators on coastal coasts and they can be categorized according to their size, their type, and their location [46].

In the past few years, the mathematical breakthrough of machine learning has opened up new opportunities and strategies for ecological research, offering new tools for discovering and explaining patterns by performing regression and classification tasks.

Machine learning algorithms are procedures that automatize decision-making processes via learning from examples. Different model constructions represent various categories of machine learning. Roughly, one can divide learning procedures into three categories.

Unsupervised learning algorithms, where the model tries to identify patterns and build structures within the data without labels. Among the important tasks that an unsupervised learning algorithm can achieve are clustering, density destination, and dimensionality reduction through various techniques such as k-means clustering, kernel density estimation, and principal component analysis [35], [33].

Reinforcement learning is another machine learning approach where a notion of an agent exists that is trying to maximize rewards by taking actions in a dynamic environment. For the learning procedure, it is not known which actions need to be taken but instead, it is discovered through the aforementioned

rewards by exploring the trade-off between exploration and exploitation [81].

In this thesis, we focus in particular, on supervised learning algorithms, which are those where their construction is achieved by pairs of inputs and outputs. In other words, the learning algorithm generalizes new unseen inputs to produce a prediction and a desired output. Supervised machine learning algorithms are commonly used to perform classification and regression tasks. In the regression setup, the range of the predicted function can be uncountable, and for classification tasks varies over a finite set [59].

In this setting, in this PhD thesis, our first aim is to study the distribution of the benthic macrofauna of various sandy beaches in Emilia-Romagna. In particular, in this study, data from four sandy beaches have been used; Bellaria, Igea Marina, San Mauro a Mare, and Gatteo a Mare. Many state-of-the-art algorithms have been used to classify and predict the number of benthos, and through these techniques, we deduce useful information about the coastal coasts. In particular, we used the K-NN algorithm, naive Bayes, neural networks, random trees, and random forests (chapter 2).

Moreover, in the second part of the PhD program, we looked in depth at the mathematical properties of some of the algorithms used. In particular, we examined a large class of mechanisms of random tilings of the feature space with tools from probability and mathematical analysis. In other words, through tessellations of the available data set with a notion of randomness it is aimed to identify patterns on the available data set. A random tiling of the feature space has a one-to-one correspondence with a random tree partition and an average of M-random trees is called a random forest.

Random forest algorithms [43] are a class of non-parametric statistic machine learning algorithms used for regression and classification tasks. Random forest algorithms can perform sparse tasks with high accuracy in high dimensions, avoiding overfitting. In particular, random forests are considered to be among the most accurate learning algorithm classes for general tasks. They are routinely used in many fields including bio-informatics [30], economics [92], biology [18], linguistics [37], and 3-D reconstructions [75]. The most widely used random forest algorithm was introduced by Breiman [23], who was inspired by the work on random subspaces of Ho [39], the geometrical feature selection of Amit and Geman [3] and Dietterich [29].

While in practice random forest algorithms are used in many applications and Howard and Bowles state: *ensembles of decision trees—often known as "random forests"—have been the most successful general-purpose algorithm in modern times* [40], [14], theoretically the analysis of the algorithms and the research of their mathematical properties is still under active research. Historically, Breiman with a series of articles ([19], [23],[21]) established the basic mathematical properties of the original algorithm and proposed a simplified modification. A brief description of the original algorithm is given in 1. Later in 2006, Lin and Jeon [53] introduced a concept of Potential Nearest Neighbors and highlighted that random forest can be viewed as adaptively weighted k-PNN methods, and later in the same direction, Biau and Devroye [12] introduced the layer nearest neighbor method and prove consistency of the bagged estimator

for regression and classification.

Breiman's random forest is designed through the CART (Classification And Regression Trees) split criterion [22]. Bagging (bootstrap aggregating) is a method used to improve prediction accuracy by creating several bootstrap samples from the data set and building a predictor for each sample. Finally, the result is provided by the average of each independent estimator. This method is often powerful for large, sparse data sets in high dimensions. The splitting direction (or equivalently the tree construction) is performed by optimizing the CART criterion based on the GINI criterion for classification tasks or the squared error for regression [14]. The CART splitting procedure and the bagging method of the algorithm are central for the tree construction but can be challenging for studying rigorously the mathematical properties of the method. Therefore, several simplifications have been proposed either by ignoring the bagging procedure or creating trees with simpler methods than CART.

The theoretical properties of random forest algorithms are still under active research activity. Understanding the original random forest algorithm of Breiman, naturally, led to definitions of simplified procedures of random partitions. A classic framework for studying simplified versions of Breiman's algorithm is the so-called *Purely random forests* where the random tiling is designed randomly but independent from the data set.

A detailed description of *purely random forests* and their classical examples is provided in chapter 1.

0.1 Thesis overview

In chapter 1 the basic notation, definitions and relative results are discussed. The chapter begins with the basic definitions of a random tree, a random forest, and their corresponding kernel constructions. A brief historical preview of the different types of random forests according to their construction is provided with the definition of the CART split criterion. Moreover, the convergence rates of some basic examples of random forests and kernel representations are discussed in chronological order of discovery and the necessary background from the Fourier analysis in abelian groups is mentioned. Finally, we define the concept of interpolation regime for an estimator constructed through the data set and specifically for the purely random forests.

In chapter 2 we provide an application of several supervised machine learning algorithms in ecological data sets. Moreover, we give a brief description of the methods 2.1.2, specifically oriented to ecology applications. In particular, our goal is to offer ecologists a practical guide to leveraging machine learning techniques for investigating ecological patterns and processes. To evaluate the selected methods, we utilized an ecological dataset on sandy beach benthic macrofauna 2.1.2. This data set was created through comprehensive sampling carried out in 2022 on beaches located in Bellaria, Igea Marina, Gatteo a Mare and San Mauro a Mare in the Emilia-Romagna region of Italy. This section is part of a preprint *On the ecology modeling of coastal beaches* that is a joint work

with Fabio Bozzeda.

In chapter 3, we introduce again the notation and definitions for the centered and uniform random forest algorithms, along with their kernel-based formulations. Additionally, an improvement of the consistency rate is provided for the centered and uniform KeRF algorithm.

Let $k \geq 1$ represent the depth of the trees used to predict the target variable Y (refer to Section 3.0.2 for detailed definitions and notation).

Theorems on Consistency

Theorem 1. *Assume $\boldsymbol{X} = (X_1, \ldots, X_d)$ and Y are related by the model*

$$Y = m(\boldsymbol{X}) + \epsilon,$$

where ϵ is Gaussian noise of zero mean with finite variance and independent of $\boldsymbol{X}$, $\boldsymbol{X}$ is uniformly distributed over $[0,1]^d$, and m is a Lipschitz regression function. Then, there exists a constant $\tilde{C}$ such that for any $n > 1$ and $x \in [0,1]^d$,

$$\mathbb{E}\big(\tilde{m}^{Cen}_{\infty,n}(x) - m(x)\big)^2 \leq \tilde{C} n^{-\frac{1}{1+d\log 2}} (\log n).$$

Here, $m(x) = \mathbb{E}[Y|X = x]$ is the true regression function, and $\tilde{m}^{Cen}_{\infty,n}(x)$ is the estimate provided by the centered random forest kernel algorithm.

Theorem 2. *With $\tilde{m}^{Un}_{\infty,n}(x)$ denoting the estimate of the uniform KeRF algorithm, and assuming the same setup as in Theorem 1, there exists a constant $\tilde{C}$ such that for any $n > 1$ and $x \in [0,1]^d$,*

$$\mathbb{E}\big(\tilde{m}^{Un}_{\infty,n}(x) - m(x)\big)^2 \leq \tilde{C} n^{-\frac{1}{1+\frac{3}{2}d\log 2}} (\log n).$$

Numerical Experiments and Parameter Tuning.
In Section 3.0.6, we present numerical experiments to analyze the impact of the tree depth parameter k on the performance of both kernel-based random forest algorithms. Specifically, we compare the L_2 error under various assumptions about the dataset and evaluate the algorithm's sensitivity to changes in k.

Analysis of the Kernel K
In the final part of the section, we examine the reproducing kernel K used in the centered KeRF algorithm independently. We interpret K as a function on the finite Abelian group $\mathbb{Z}_2^{kd}$, where d is the dimension of $\mathbf{X}$ and k is the tree depth. Using elementary Fourier analysis on groups, we derive:

- Multiple equivalent expressions for K and its group transform,
- A characterization of functions in the associated Reproducing Kernel Hilbert Space (RKHS) H_K,
- Results on multipliers, and
- Bounds on the dimension of H_K, which is shown to be significantly smaller than anticipated.

These findings deepen our understanding of the kernel's structure and its implications for random forest algorithm and it is part of a joint work with N.Arcozzi, in *Improved rates of convergence for some kernel random forests* [43].

In the chapter 4,we present a variation of the centered random forest algorithm, which we call the simplified directional algorithm. The main goal of this approach is to create a partition of the feature space that is independent of the dataset by simplifying the centered method. We establish the kernel representation for this new algorithm and demonstrate that, asymptotically, as the number of trees tends to infinity, the centered KeRF and the simplified directional KeRF become equivalent.

To validate these findings, we conduct experiments comparing the L_2-error and variance of the finite-centered KeRF and the finite simplified directional algorithm across varying numbers of trees.

Finally, we provide the proof of the improvement of the rate of convergence of the infinite centered-KeRF in the interpolation regime. Of course, since the simplified directional infinite keRF coincides with the centered one, we obtain also rates of convergence in the interpolation regime and in general as a corollary.

Theorem 3. *Assume $\boldsymbol{X} = (X_1, \ldots, X_d)$ and Y are related by the model*

$$Y = m(\boldsymbol{X}) + \epsilon,$$

where ϵ is Gaussian noise of zero mean with finite variance and independent of $\boldsymbol{X}$, $\boldsymbol{X}$ is uniformly distributed over $[0,1]^d$, and m is a Lipschitz regression function. Then for large enough n and assuming the tree depth is $k = \log_2 n$ to satisfy the interpolation regime, for every value of $d \geq 2$ one has that

$$\mathbb{E}[(\tilde{m}^{cc}_{n,\infty}(x) - m(x))^2] \leq c_1 \left(1 - \frac{1}{2d}\right)^{2\log_2 n} + C_3 \frac{\log_2(\log_2 n)^d}{(\log_2 n)^{\frac{d-1}{2}}} \log\left(\frac{\log n (\log_2 n)^{\frac{d-1}{2}}}{\log_2(\log_2 n)^d}\right).$$

This is part of a joint work with N.Arcozzi in the preprint *A simplified directional KeRF algorithm* [44].

Chapter 1

Preliminaries and related results

In this chapter, we introduce the preliminaries, basic definitions, and related results pertinent to our work. In particular, we rigorously provide definitions of random trees, random forest algorithms, and their related kernel representations. We emphasize different types of random forests based on their construction methods in relation to the data set. A brief history of related results on rates of convergence is discussed, along with some definitions and results from Fourier analysis on finite abelian groups.

This chapter information primarily derived from [43], [11],[73].

1.1 Definitions and notation

We begin this chapter by providing the general random forest framework by defining firstly the notion of a random tree. Additionally, we present two specific variations of the original random forest algorithm, namely, the centered and uniform random forest algorithms.

In particular we assume that we are given a training sample

$$\mathcal{D}_n = \{(X_1, Y_1), ..., (X_n, Y_n)\}$$

of independent random variables, where $X_i \in [0,1]^d$ for every $i = 1, ..., n$ and $Y \in \mathbb{R}$ with a shared joint distribution $\mathbb{P}_{X,Y}$. The goal is using the data set to construct an estimate $m_n : \mathcal{X} \subseteq [0,1]^d \to \mathbb{R}$ of the function m.

A tree construction is equivalent to a recursive partition of the feature space $[0,1]^d$ with some notion of randomness. In other words every recursive covering of the feature space corresponds to a tree construction.

We call the tiles of the recursive partition 1.1 leaves or nodes or sometimes cells.

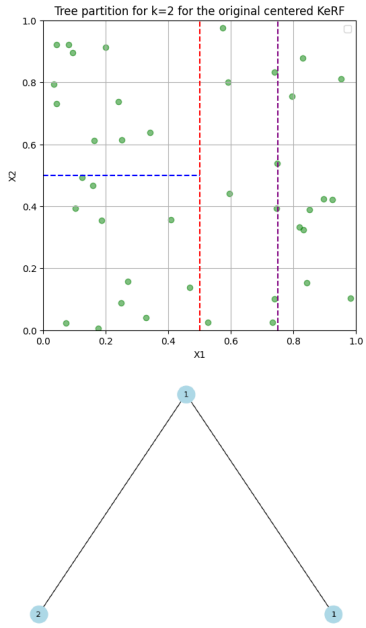


Figure 1.1: An example of a tree construction of a recursive partition of a two-dimensional feature space.

In the figure 1.1 we see an example of a recursive partition of a two-dimensional space in four nodes.

The construction of the estimate function (that we sometimes also call a tree) is naturally the average of the target examples that belong in each tile.

Let's assume Θ_i for $i = 1, ..., M$ is a collection of independent random variables, distributed as Θ. The random variables Θ_i correspond to sample the training set or select the positions for splitting. A detailed construction in the case of the centered random forest of the random variable Θ is performed in chapter 3.0.7.

Definition 1. *For the j-th tree in the forest, the predicted value x will be denoted by*

$$m_{n,\Theta_j,\mathcal{D}_n}(x) = \sum_{i=1}^{n} \frac{\mathbb{1}_{X_i \in A_{n,\Theta_j,\mathcal{D}_n}(x)} Y_i}{N_{n,\Theta_j,\mathcal{D}_n}(x)}.$$

- *Where $A_{n,\Theta_j,\mathcal{D}_n}(x)$ is the cell containing x for the j-th tree and designed with randomness Θ_j.*

- *$N_{n,\Theta_j,\mathcal{D}_n}(x)$ is the number of points that fall into the cell that x belongs to for the j-th tree and designed with randomness Θ_j. In other words $N_{n,\Theta_j,\mathcal{D}_n}(x)$ is the cardinality of the node $A_{n,\Theta_j,\mathcal{D}_n}(x)$.*

- *By definition, if a node contains no points, the algorithm assigns to the tree estimator the value zero*

For a fixed value of $x \in [0,1]^d$, the value of the tree is the empirical expectation of Y in the unique cell containing x; which is, this is the hope, a good guess for the target value corresponding to x.

A *random forest* is a finite collection (average) of independent, finite random trees:

Definition 2. *The finite M forest is*

$$m_{M,n}(x) = \frac{1}{M}\sum_{j=1}^{M} m_{n,\Theta_j,\mathcal{D}_n}(x).$$

From a modeling point of view, we let $M \to \infty$ and consider the infinite forest estimate

$$m_{\infty,n,\mathcal{D}_n}(x) = \mathbb{E}_\Theta(m_{n,\Theta,\mathcal{D}_n}(x)).$$

The convergence holds almost surely by the law of the large numbers conditionally on $\mathcal{D}_n$. (Breinman) [20], (Scornet) [72, Theorem 3.1].

1.1.0.1 Kernel Random Forest algorithm

In 2016, Scornet in [73] introduced kernel methods in the random forest world (KeRF), producing a kernel-based algorithm, together with estimates on how this compares with the traditional methods, described above.
To understand the intuition behind KeRF construction, we reformulate the random forest algorithm.
For all $x \in [0,1]^d$,

$$m_{M,n}(x) = \frac{1}{M}\sum_{j=1}^{M}\Big(\sum_{i=1}^{n}\frac{\mathbb{1}_{X_i \in A_{n,\Theta_j,\mathcal{D}_n}(x)}Y_i}{N_{n,\Theta_j,\mathcal{D}_n}(x)}\Big).$$

Therefore we can define the weights of every observation Y_i as

$$W_{i,j,n}(x) = \frac{\mathbb{1}_{X_i \in A_{n,\Theta_j,\mathcal{D}_n}(x)}}{N_{n,\Theta_j,\mathcal{D}_n}(x)}.$$

Hence it is clear that the value of weights, that they are a probability distribution on $A_{n,\Theta_j,\mathcal{D}_n}(x)$, changes significantly concerning the number of points in each cell. A way to overcome this nuisance is by simultaneously considering all tree cells containing x, as the tree is randomly picked in the forest.
For all $x \in [0,1]^d$,

$$\tilde{m}_{M,n,\Theta_1,\Theta_2,\ldots,\Theta_M}(x) = \frac{1}{\sum_{j=1}^{M} N_{n,\Theta_j}(x)}\sum_{j=1}^{M}\sum_{i=1}^{n} Y_i \mathbb{1}_{X_i \in A_{n,\Theta_j}(x)}.$$

This way, empty cells do not affect the computation of the prediction function of the algorithm.

It is proven in [73], that this representation has indeed a kernel representation.

Proposition 1 (Scornet [73], Proposition 1)**.** *For all $x \in [0,1]^d$ almost surely, it holds*

$$\tilde{m}_{M,n,\Theta_1,\Theta_2,...,\Theta_M}(x) = \frac{\sum_{i=1}^{n} K_{M,n}(x, X_i) Y_i}{\sum_{i=1}^{n} K_{M,n}(x, X_i)},$$

where

$$K_{M,n}(x,z) = \frac{1}{M} \sum_{i=1}^{M} \mathbb{1}_{x \in A_{n,\Theta_i,\mathcal{D}_n}(z)}.$$

is the proximity function of the M forest

Again, naturally, from the modeling point of view, it is meaningful to ask what happens when the number of trees goes to infinity or if the kernel representation is maintained. The *infinite random forest* arises in the following way,

Definition 3. *The* **infinite KeRF** *is defined as:*

$$\tilde{m}_{\infty,n}(x) = \lim_{M \to \infty} \tilde{m}_{M,n}(x, \Theta_1, \Theta_2, ..., \Theta_M).$$

The extension of the kernel follows also in the infinite random forest.

Proposition 2 (Scornet [73], Proposition 2)**.** *Almost surely for all $x, y \in [0.1]^d$*

$$\lim_{M \to \infty} K_{M,n}(x,y) = K_n(x,y),$$

where

$$K_n(x,y) = \mathbb{P}_\Theta(x \in A_n(y, \Theta)),$$

where the left-hand side is the probability that x and y belong to the same cell in the infinite forest.

Following the notation on [73], we provide a proposition that quantifies how close random forests are to kernel-related random forests. In [73] one can see the proves and also numerical experiments confirming the theoretical results.

To proceed we need the following assumptions on the model:

We fix $x \in [0,1]^d$ and let us assume that $Y \geq 0$ almost surely. Then the following dichotomy is assumed, which ensures that each node has a bounded number of data points from above and below. In other words

1. there exist two sequence a_n, b_n that bound from above and below the number of points in each node, i.e.

   $$a_n \leq N_n(x, \Theta) \leq b_n$$

   or

2. there exist three sequences ϵ_n, a_n, b_n such that almost surely

$$\mathbb{P}_\Theta(a_n \leq N_n(x,\Theta) \leq b_n) \geq 1-\epsilon_n \quad \textit{and} \quad 1 \leq a_n \leq \mathbb{E}_\Theta(N_n(x,\Theta)) \leq b_n$$

.

Proposition 3. *([73],Proposition 3) Under the hypothesis 1 almost surely it holds,*

$$\left| \frac{m_M(x,\Theta_1,...,\Theta_M)}{\tilde{m}_M(x,\Theta_1,...,\Theta_M)} - 1 \right| \leq \frac{b_n - a_n}{a_n}$$

Thus, if the number of points in every node can be controlled the kernel forest can be arbitrarily close to the forest construction.

In Breiman's tree construction, the user controls the number of data points in the hypercube partition 1.1.1.3. In fact, the default selection for classification tasks is one point per node. Therefore, since the user can control the sequence's a_n, b_n, the random forest construction and the corresponding kernel random forest construction are arbitrary close.

1.1.1 Types of random forest algorithms

Different types of random forest algorithm exist, depending on the way that the tiling of the hypercube is performed. In general, the basic distinction is the following:

1) Independently designed of X_i and Y_i, for example centered random forest, uniform random forest.

2) Independently designed of Y_i, for example median random forest.

3) Dependent of X_i and Y_i for example Breiman's random forest.

In the centered and uniform forest algorithms, the way the hypercube is partitioned is independent of the data set itself. We call this random forests also as *Purely random forests.*

1.1.1.1 The centered random forest/ Centered KeRF and the uniform random forest/uniform KeRF

The **centered random forest** is designed as follows.

1) Fix $k \in \mathbb{N}$.

2) At each node of each individual tree choose a coordinate uniformly from $\{1, 2, ..d\}$.

3) Split the node at the midpoint of the interval of the selected coordinate.

Repeat step 2)-3) k times. At the end, we have 2^k leaves, or cells. Our estimation at a point x is achieved by averaging the Y_i corresponding to the X_i in the cell containing x.

Uniform random forest was introduced by Biau et al. [13] and is another toy model of Breinman's random forest as a centered random forest. The algorithm forms a partition in $[0,1]^d$ as follows:

1) Fix $k \in \mathbb{N}$.

2) At each node of each individual tree choose a coordinate uniformly from $\{1, 2, ..d\}$.

3) The splitting is performed uniformly on the side of the cell of the selected coordinate.

Repeat step 2)-3) k times. At the end, we have 2^k leaves. Our final estimation at a point x is achieved by averaging the Y_i corresponding to the X_i in the cell containing x.

In particular, the centered random forest satisfies the 1 property. Specifically, since we assume that X is uniformly distributed in the hypercube, the expected number of points in each node is $\frac{n}{2^k}$ and the measure of each node is $\frac{1}{2^k}$. Then almost surely from the law of iterative logarithm

$$\left| N_n(x, \Theta) - \frac{n}{2^k} \right| \leq \frac{\sqrt{2n \log \log n}}{2}$$

and hence, the 1 is satisfied and for large enough n. In other words for appropriate choices of a_n, b_n the centered KeRF and the centered algorithm are arbitrary close for an appropriate choice of a tree depth [73]. The assumption of the uniform distribution of the feature space here is crucial in the sense that we cannot establish the asymptotic equivalence of the centered random forest and the centered KeRF under a general density distribution.

1.1.1.2 The median random forest algorithm

The median random forest algorithm is another simplification of the original Breiman random forest. In this case, the cut is performed on the empirical midpoint of the preselected coordinate and hence it depends on the data of X_i that belong to the feature space $[0,1]^d$. The algorithm is constructed as follows:

1) Fix $k \in \mathbb{N}$.

2) At each node of each individual tree choose a coordinate uniformly from $\{1, 2, ..d\}$.

3) The splitting is performed in the median of the feature space on the side of the cell of the selected coordinate.

Repeat step 2)-3) k times. At the end, we have 2^k leaves. Our final estimation at a point x is achieved by averaging the Y_i corresponding to the X_i in the cell containing x. Hence, for a tree of level k, every cell has the same number of points ± 2. In other words, with an appropriate choice of tree depth k one has that the finite random forest and the related KeRF representation can be arbitrary close.

The above results can be extended assuming 2 to infinite random forests (Proposition 4. [73]).

1.1.1.3 Breiman's random forest algorithm

The great advantage of all different versions of random forest algorithms are the few parameters that need to be tuned. Below we summarize the most important parameters for the original algorithm of Breiman. We review the basic algorithm and we conclude with the definition of the CART criterion used for the choosing the splitting direction.

The most important parameters of the model are the number of data points sampled in each tree a_n from the data set, $m_{\text{try}} \in \{1, \dots, d\}$ the number of potential splitting directions considered at each node of every tree and finally the nodesize which is the number of data points that can be left at each tile and it can be used as a stopping time for no more splitting.

In the famous package [52] in R the deafult settings of the regression model are

- $m_{\text{try}} = \lceil d/3 \rceil$ (where $\lceil \cdot \rceil$ denotes the ceiling function),
- $a_n = n$, and
- nodesize $= 5$.

The CART-Split Criterion

To simplify the explanation, consider a tree constructed using the entire dataset D_n without subsampling. Let A represent some possible cell in the feature space in the recurring splitting procedure, and let $N_n(A)$ denote the number of data points in A. A potential split in A is represented by the pair (j, z), where:

- $j \in \{1, \dots, d\}$ is a chosen dimension, and
- z is the position of the cut along the j-th chosen dimension, of course constrained by the geometry of the cell A.

Let $\mathcal{C}_A$ be the set of all possible cuts in A. Using the notation $\mathbf{X}_i = (X_i^{(1)}, \dots, X_i^{(d)})$, for any $(j, z) \in \mathcal{C}_A$, the CART-split criterion is defined as:

$$L_{\text{reg},n}(j,z) = \frac{1}{N_n(A)}\Bigg(\sum_{i=1}^{n}(Y_i - \bar{Y}_A)^2 \mathbb{1}_{\mathbf{X}_i \in A} - \left[\sum_{i=1}^{n}(Y_i - \bar{Y}_{A_L})^2 \mathbb{1}_{X_i^{(j)} < z} + \sum_{i=1}^{n}(Y_i - \bar{Y}_{A_R})^2 \mathbb{1}_{X_i^{(j)} \geq z} \right] \mathbb{1}_{\mathbf{X}_i \in A} \Bigg)$$

Here:

- $A_L = \{\mathbf{x} \in A : x^{(j)} < z\}$ and $A_R = \{\mathbf{x} \in A : x^{(j)} \geq z\}$,

- $\bar{Y}_A$, $\bar{Y}_{A_L}$, and $\bar{Y}_{A_R}$ are the averages of Y_i for $\mathbf{X}_i \in A$, $\mathbf{X}_i \in A_L$, and $\mathbf{X}_i \in A_R$, respectively, with the convention that the average is 0 if no points fall in the respective sets.

The optimal split (j_n^*, z_n^*) for a cell A is determined by maximizing $L_{\text{reg},n}(j,z)$ over the set of considered directions m_{try} and possible cuts $\mathcal{C}_A$: $(j_n^*, z_n^*) \in \underset{j \in m_{\text{try}}, (j,z) \in \mathcal{C}_A}{\arg\max} L_{\text{reg},n}(j,z)$.

To avoid ambiguities in cases of ties, the algorithm chooses the cut point (j_n^*, z_n^*) to be the midpoint between two consecutive data points [14]. Of course in a similar way the algorithm works for the resampling case by just replacing $\mathcal{D}_n$ with a_n. This optimization process extends naturally to subsampling. In this case, the CART criterion is performed over the a_n preselected data points rather than the entire dataset D_n. Therefore the random forest of Breiman is performed as follows.

Algorithm 1 Breiman's Random Forest Predicted Value at x

Input: Training set $D_n = \{(X_i, Y_i)\}_{i=1}^n$
Number of trees $M > 0$
Subsample size $a_n \in \{1, \ldots, n\}$
Number of variables to consider for splitting $m_{\text{try}} \in \{1, \ldots, d\}$
Minimum node size, *nodesize* $\in \{1, \ldots, a_n\}$
Query point $x \in [0,1]^d$
Output: Predicted value of the random forest at x

for $j = 1$ to M **do**
  Select a_n points from D_n (with or without replacement). These a_n observations are only used for building the tree.
  Initialize the partition $\mathcal{P} = [0,1]^d$ (the hypercube feature space).
  Initialize $\mathcal{P}_{\text{final}} = \emptyset$ (an empty list to store terminal nodes).
  while $\mathcal{P} \neq \emptyset$ **do**
    Let A be the first element of $\mathcal{P}$.
    if A contains fewer than nodesize points that there were preselected **or** all $X_i \in A$ are equal **then**
      Remove A from $\mathcal{P}$.
      $\mathcal{P}_{\text{final}} \leftarrow \mathcal{P}_{\text{final}} \cup \{A\}$.
    else
      Randomly select m_{try} features from $\{1, \ldots, d\}$.
      Find the best split of A using the CART-split criterion on the selected features.
      Split A into two subsets A_L and A_R based on the best split of the previous step.
      Remove A from $\mathcal{P}$.
      $\mathcal{P} \leftarrow \mathcal{P} \cup \{A_L, A_R\}$.
    end if
  end while
  Compute $m_n(x; \Theta_j, D_n)$ as the mean of all target values that correspond to points in the feature space that belong in the cell containing x in $\mathcal{P}_{\text{final}}$.
end for
Compute the random forest prediction:

$$m_{M,n}(x; \Theta_1, \ldots, \Theta_M, D_n) = \frac{1}{M} \sum_{j=1}^{M} m_n(x; \Theta_j, D_n).$$

[14].

1.1.2 Rates of convergence of random forest algorithms

In this subsection we provide a historical review of the rates of convergence of several random forest models. Under different assumptions on the model and the construction of each algorithm one can establish different consistency results for

general types of random forest. Here we summarize some of the classical results in the bibliography about convergence of random forest algorithms and rates of convergence where it is possible.

Already Breiman in [21] and Biau in [11] discussed rates of convergence of the centered random forest.

In 2012 Biau in [11] studied a random forest model proposed by Breiman, where the construction is independent of the data set, called in literature *centered random forest*. In [11] an upper bound on the rate of consistency of the algorithm and its adaption to sparsity were proven. More precisely, about the first item, for a data set of n samples in a space of dimension d, the convergence rate was $\mathcal{O}\left(n^{-\frac{1}{d\frac{4}{3}\log 2+1}}\right)$.

In addition in 2021 Klusowski at [49] improved the rate of convergence to $\mathcal{O}\left((n\log^{\frac{d-1}{2}} n)^{-(\frac{1+\delta}{d\log 2+1})}\right)$, where δ is a positive constant that depends on the dimension of the feature space d and converges to zero as d approaches infinity. In addition, in the same paper, Klusowski proved that the rate of convergence of the algorithm is sharp, although it fails to reach the minimax rate of consistency over the class of the Lipschitz functions [91] $\mathcal{O}\left(n^{\frac{-2}{d+2}}\right)$.

There is also important work on the consistency of algorithms that depend on data [57], [87], [74]. For a comprehensive overview of both theoretical and practical aspects of the random forests see e.g. [14], which surveys the subject up to 2016.

In 2016 [73] Scornet proved rates of convergence of the centered and uniform kernel based random forest under specific assumptions 1. In particular the rate of convergence of the centered KeRF was proven $\mathcal{O}(n^{-(\frac{1}{d\log 2+3})}(\log n)^2)$ and for the uniform KeRF $\mathcal{O}(n^{-(\frac{2}{3dlog2+6})}(\log n)^2)$ and afterwards in [43] both algorithms had an improved rate of $\mathcal{O}(n^{-\left(\frac{1}{1+d\log 2}\right)}(\log n))$ and $\mathcal{O}(n^{-(\frac{2}{3d\log 2+2})}(\log n))$ respectively (the proofs are presented in 3).

1.1.3 Some results from Fourier analysis on finite groups

In this subsection we provide some basic results from harmonic analysis in abelian groups that will be useful for the chapter 3 and it comes from [43]. Following the notation of [70], in [43], we recall the basic notions of Fourier theory for a finite, abelian group G, which we employed above. This theory is necessary for the section 3.0.8 Here, G is endowed with its counting measure. The *dual group* $\Gamma = \widehat{G}$ of G is populated by labels a for homomorphisms $\gamma_a : G \to \mathbb{T} = \{e^{it} : \ t \in \mathbb{R}\}$. Given a function $f : G \to \mathbb{C}$, its *Fourier transform* $\widehat{f} : \Gamma \to \mathbb{C}$ is defined as

$$\widehat{f}(a) = \sum_{x \in G} f(x)\overline{\gamma_a(x)}. \tag{1.1.1}$$

We make Γ into a (finite), additive group by setting

$$\gamma_{a+b} = \gamma_a \cdot \gamma_b, \text{ and } \gamma_x(a) := \gamma_a(x).$$

It turns out that they have the same number of elements, $\sharp(G) = \sharp(\Gamma)$. Some basic properties are:

$$\begin{aligned} f(x) &= \frac{1}{\sharp(\Gamma)} \sum_{a\in\Gamma} \widehat{f}(a)\gamma_a(x) \text{ (inverse Fourier transform)}, \\ \sum_{x\in G} |f(x)|^2 &= \frac{1}{\sharp(\Gamma)} \sum_{a\in\Gamma} |\widehat{f}(a)|^2 \text{ (Plancherel)}, \\ \widehat{f * g} &= \widehat{f}\cdot\widehat{g}, \end{aligned}$$

where

$$(f * g)(x) = \sum_{y\in G} f(x-y)g(y). \tag{1.1.2}$$

We write

$$\check{\varphi}(x) = \sharp(\Gamma)^{-1} \sum_{a\in\Gamma} \varphi(a)\gamma_a(x), \text{ so that } \widehat{\check{\varphi}} = \varphi. \tag{1.1.3}$$

The unit element of convolution in G is δ_0.

In the other direction, for $\varphi, \psi : \Gamma \to \mathbb{C}$ we define

$$(\varphi * \psi)(a) = \frac{1}{\sharp(\Gamma)} \sum_{b\in\Gamma} \varphi(a-b)\psi(b), \tag{1.1.4}$$

and similarly to above, $\widehat{\check{\varphi}\check{\psi}} = \varphi * \psi$. The unit element on convolution in Γ is $\sharp(\Gamma)\delta_0$.

A function φ on Γ is *positive definite* if

$$\sum_{a,b\in\Gamma}^{n} c(a)\overline{c(b)}\varphi(b-a) \geq 0.$$

Theorem 4. *[Bochner's Theorem] A function $\varphi : \Gamma \to \mathbb{C}$ is positive definite if and only if there exists $\mu : G \to \mathbb{R}_+$ such that $\varphi = \widehat{\mu}$.*

The theorem holds in great generality, and its proof in the finite group case is elementary. We include it because it highlights the relationship between the measure μ on G and the positive definite function (the kernel) φ.

If.

$$\begin{aligned} \sharp(\Gamma)^{-2} \sum_{a,b\in\Gamma} \widehat{\mu}(b-a)c(a)\overline{c(b)} &= \sum_{x\in G} \sharp(\Gamma)^{-2} \sum_{a,b\in\Gamma} \mu(x)\overline{\gamma_{b-a}(x)}c(a)\overline{c(b)} \\ &= \sum_{x\in G} \sharp(\Gamma)^{-2} \sum_{a,b\in\Gamma} \mu(x)\overline{\gamma_b(x)c(b)}\gamma_a(x)c(a) \\ &= \sum_{x\in G} \mu(x) \left| \sharp(\Gamma)^{-1} \sum_{a\in\Gamma} c(a)\gamma_a(x) \right|^2 \\ &= \sum_{x\in G} \mu(x) \left|\check{c}(x)\right|^2 \geq 0. \end{aligned} \tag{1.1.5}$$

Only if. Since for all b in Γ,

$$\begin{aligned}\mu(x)\sharp(\Gamma) &= \sum_{a\in\Gamma}\varphi(a)\gamma_x(a) = \sum_{a\in\Gamma}\varphi(a-b)\gamma_x(a-b)\\ &= \sum_{a\in\Gamma}\varphi(a-b)\gamma_x(a)\overline{\gamma_x(b)}, \end{aligned} \tag{1.1.6}$$

we have

$$\mu(x)\sharp(\Gamma)^2 = \sum_{a,b\in\Gamma}\varphi(a-b)\gamma_x(a)\overline{\gamma_x(b)} \geq 0, \tag{1.1.7}$$

by the assumption. □

We now come to reproducing kernels on Γ which are based on positive definite functions $\varphi : \Gamma \to \mathbb{R}_+$. Set

$$K(a,b) = \varphi(a-b) = K_b(a),\ K : \Gamma\times\Gamma \to \mathbb{C}, \tag{1.1.8}$$

and set

$$H_K = \text{span}\{K_b :\ b\in\Gamma\} \ni \sum_{b\in\Gamma} c(b)K_b, \tag{1.1.9}$$

where H_K is the Hilbert space having K as reproducing kernel. We wish to have a more precise understanding of it.

We start by expressing the norm of an element on H_K is several equivalent ways,

$$\begin{aligned}\left\|\sum_{b\in\Gamma} c(b)K_b\right\|^2_{H_K} &= \sum_{a,b\in\Gamma}\overline{c(a)}c(b)\langle K_b, K_a\rangle\\ &= \sum_{a,b\in\Gamma}\overline{c(a)}c(b)K(a,b) = \sum_{a,b\in\Gamma}\overline{c(a)}c(b)\widehat{\mu}(a-b)\\ &= \sum_{a,b\in\Gamma}\overline{c(a)}c(b)\sum_{x\in G}\mu(x)\gamma_{b-a}(x)\\ &= \sum_{x\in G}\mu(x)\sum_{a,b\in\Gamma}\overline{c(a)}c(b)\gamma_b(x)\overline{\gamma_a(x)}\\ &= \sum_{x\in G}\mu(x)\left|\sum_{b\in\Gamma}c(b)\gamma_b(x)\right|^2\\ &= \sharp(\Gamma)^2\sum_{x\in G}\mu(x)\left|\check{c}(x)\right|^2 = \sharp(\Gamma)^2\sum_{x\in G}\left|\mu(x)^{1/2}\check{c}(x)\right|^2 \end{aligned} \tag{1.1.10}$$

In other terms,

$$\sharp(\Gamma)^{-1}\sum_{b\in\Gamma}c(b)K_b \mapsto \check{c} \tag{1.1.11}$$

is an isometry of H_K onto $L^2(\mu)$. This will become important later, when we verify that for our kernels $\text{supp}(\mu)$ is sparse in G. In fact, $\dim(H_K) = \sharp(\text{supp}(\mu))$.

Corollary 1. *As a linear space, H_K is determined by supp(μ):*

$$\psi \in H_K \text{ if and only if } supp(\check{\psi}) \subseteq supp(\mu).$$

Let $E \subseteq G$. We denote

$$L_E = \{G \xrightarrow{\psi} \mathbb{C} : \ \mathrm{supp}(\check{\psi}) \subseteq E\}. \tag{1.1.12}$$

Next, we look for the natural orthonormal system provided by the Fourier isometry (1.1.11). Fr $x \in G$, let $\check{c_x} = \mu(x)^{-1/2}\delta_x$: $\{\check{c_x} : \ x \in E := \mathrm{supp}(\mu)\}$ is a orthonormal system for $L^2(\mu)$, and so $\{e_x : \ x \in E\}$ is an orthonormal basis for H_K, where

$$c_x(b) = \sum_{y\in G} \mu(x)^{-1/2}\delta_x(y)\overline{\gamma_b(y)} = \mu(x)^{-1/2}\overline{\gamma_b(x)}, \tag{1.1.13}$$

and

$$\begin{aligned} e_x(a) &= \sharp(\Gamma)^{-1}\sum_{b\in\Gamma} c_x(b)K_b(a) \\ &= \frac{\mu(x)^{-1/2}}{\sharp(\Gamma)}\sum_{b\in\Gamma} K_b(a)\overline{\gamma_b(x)} \\ &= \frac{\mu(x)^{-1/2}}{\sharp(\Gamma)}\sum_{b\in\Gamma} \varphi(a-b)\overline{\gamma_b(x)} \\ &= \frac{\mu(x)^{-1/2}}{\sharp(\Gamma)}\sum_{b\in\Gamma} \varphi(a-b)\gamma_{a-b}(x)\overline{\gamma_a(x)} \\ &= \mu(x)^{-1/2}\mu(x)\overline{\gamma_a(x)} \\ &= \mu(x)^{1/2}\gamma_a(x). \end{aligned} \tag{1.1.14}$$

Let's verify that we obtain the reproducing kernel from the o.n.b. as expected,

$$\begin{aligned} \sum_{x\in\Gamma} e_x(a)\overline{e_x(b)} &= \sum_{x\in\Gamma} \mu(x)\overline{\gamma_x(a)}\gamma_x(b) \\ &= \sum_{x\in\Gamma} \mu(x)\overline{\gamma_x(a-b)} \\ &= \widehat{\mu}(a-b) \\ &= \varphi(a-b). \end{aligned} \tag{1.1.15}$$

Remark 1. *Since any finite, abelian group can be written as the direct product of cyclic groups,*

$$G = \bigoplus_{l=1}^{L} \mathbb{Z}_{m_l}, \tag{1.1.16}$$

its dual Γ can be written in the same way, because $\widehat{\mathbb{Z}_m} \equiv \mathbb{Z}_m$. From the Fourier point of view, the only difference is that, if on G we consider the counting measure, then on Γ we consider the normalized counting measure, as we did above.

1.1.4 Interpolation regime

Finally, we introduce the notion of data training interpolation.

Models of high complexity tend to overfit, and their ability to generalize to new, unseen data is usually poor. Recently, this idea has been challenged for certain model examples. Large and deep neural networks still perform at the state-of-the-art level, and even interpolating training data can lead to high-performance models ([36], [4], [7]).

In this direction, well-studied simple parametric models, such as linear regression [6], [86], [51], and some non-parametric models like random forests ([83], [90], [4]), have been explored. By default, many machine learning libraries grow deep trees until only one data point remains in each cell, so the estimator effectively interpolates the data [63].

Kernel interpolation estimators, on the other hand, have been observed to be a good balance between complexity and lack of overfitting [50],[48],[28]. In the work of Belkin et al. [9] non-asymptotic rates with data interpolation were first proven, and recently [10] Belkin et al. proved optimal rates of convergence for kernel interpolating estimators. More recently, Wang and Scott [88] provided consistency results for kernel-based methods on Riemannian manifolds.

Following the article by Arnould et al. [4], we present some basic results from their paper and conclude with our theorem statement.

Definition 4 ((Exact) Interpolation)**.** *([4]) An estimator m_n is said to interpolate if, for all training data (X_i, Y_i), we have $m_n(X_i) = Y_i$ almost surely.*

The random forest algorithm, in general can interpolate the data if every random tree interpolates the data. In other words our estimator can interpolate the data if the tree depth is deep enough until there exist nodes with one observation.

From the construction of the centered tree, it is clear that it is impossible to force each node to have only one observation. This happens because the centered the uniform and the simplified directional trees (see for the definition of the later 4) are constructed without taking into account the data set (non-adaptive or purely trees).

Therefore, for example the centered random forest cannot interpolate in the sense of Definition 4, and a weaker notion of interpolation (in probability) must be examined. The definition of the interpolation regime is the following.

Definition 5 (Mean Interpolation Regime)**.** *([4]) The centered random forest algorithm $m_{M,n}^{cent}$ satisfies the mean interpolation regime when each tree of $m_{M,n}$ has at least n leaves, i.e., if and only if $k \geq \log_2(n)$, where k is the tree depth.*

Therefore, it can be computed the probability of a centered tree interpolates the data.

Theorem 5 (Probability of Interpolation for Centered Tree)**.** *([4]) For $k = \lfloor \log_2(\alpha_n n) \rfloor$ with $\alpha_n \in \mathbb{N} \setminus \{0, 1\}$:*

$$e^{-\frac{n}{\alpha_n - 1}} \leq P(\textit{Interpolation regime}) \leq e^{-\frac{n}{2(\alpha_n + 1)}}.$$

A crucial result from Arnould et al.'s paper is that in the mean interpolation regime, the infinite centered random forest is not consistent.

Theorem 6 (Inconsistency of Centered Random Forest)**.** *If $E[m(X)^2] > 0$ and $k_n \geq \log_2(\alpha_n)$, then the infinite centered random forest $m^{cc}_{\infty,n}$ is inconsistent.*

On the contrary, the kernel-based centered random forest can be simultaneously consistent and satisfy the mean interpolation regime when the dimension of the feature space is $d > 5$.

Theorem 7 (Consistency of Centered KeRF)**.** *Under the following assumptions:*

$$
\begin{aligned}
&Y = m(X) + \epsilon,\\
&X \text{ is uniformly distributed on } [0,1]^d,\\
&\epsilon \sim \mathcal{N}(0, \sigma^2), \quad \sigma < \infty,\\
&m \text{ belongs to the class of } L\text{-Lipschitz functions,}
\end{aligned}
$$

and assuming furthermore that $k = \lfloor \log_2(n) \rfloor$:
then the rate of convergence is

$$\mathbb{E}[(\tilde{m}^{cc}_{n,\infty}(x) - m(x))^2] \leq \frac{8L^2d^2}{n^{-2\log_2(1-1/d)}} + C_d(\log_2 n)^{-(d-5)/6}(\log_2(\log_2 n))^{d/3},$$

where $C_d > 0$ is a constant dependent on noise variance.

Under the same assumptions for the regression function m, in the mean interpolation regime, we provide an improvement in the rate of convergence in chapter 4

$$\mathbb{E}[(\tilde{m}^{cc}_{n,\infty}(x) - m(x))^2] \leq c_1 \left(1 - \frac{1}{2d}\right)^{2\log_2 n} + C_3 \frac{\log_2(\log_2 n)^d}{(\log_2 n)^{\frac{d-1}{2}}} \log\left(\frac{\log n (\log_2 n)^{\frac{d-1}{2}}}{\log_2(\log_2 n)^d}\right).$$

Hence, from the above result, it is clear that the centered KeRF algorithm is consistent in the mean interpolation regime with a better convergence rate than the one provided in [4] and in fact for every feature space with dimension $d \geq 2$.

Chapter 2

Ecology Project

In this chapter, we give an application of machine learning algorithms applied to ecological databases. We study ecological coastal coasts in Emilia Romagna, and through supervised learning procedures, we study the distribution of the benthic organisms. We perform classification and regression tasks to obtain results between the physical parameters of sandy coasts and the macrofauna communities, and finally, we highlight the advantages and the limitations of each method for every specific ecological task.

2.1 Applications of machine learning algorithms to ecological databases: an empirical and critical comparison.

In this section, we aim to provide a comparative analysis of various machine learning algorithms and their probabilistic aspects (Pichler and Hartig, (2023 [66]); Borowiec et al., ([15]2022); Hishie, ([41]2009); Huntingford et al, ([42]2022). We highlight the advantages and limitations of each approach providing a guided tour of the methods. In general, a machine-learning task is the use of an algorithm or a technique that enables computers to "learn" through examples and performing tasks. In other words, the user builds a mathematical model with a data set that generalizes with high accuracy in new "unseen" data.

In other words, we use supervised machine learning algorithms such as neural networks, random forests, and k-Nearest Neighbors to name a few, which have performed remarkably in ecological data, explaining non-linear patterns in high dimensional data sets. (Recknagel, ([68]2001); Peters et al.,([64] 2014; Olden et al, ([61]2008) ; Tu et al., ([93]2021).

Specifically, neural networks can perform various ecological tasks, by learning complicated relationships between ecological variables providing researchers with a useful tool for explicit predictions of the distribution of the species. Moreover, random forests, an average of predictions of finite decision random trees

perform with high accuracy avoiding overfitting for both regression and classification problems (Mosaffaei et al, ([58] 2020), Zhang et al.,([95] 2021; Batool et al.,([8] 2021)). We provide a brief description of the methods used in the sub-section "Methods".

Unsupervised learning algorithms have also been used in ecology research projects, where the data are un-categorized or un-labeled and the goal is to identify patterns and underlying structures. Some common tasks are dimensionality reduction and data clustering ([35]).

Reinforcement learning is another machine learning approach also used in ecology where an agent is trying to maximize a notion of reward by taking actions in a dynamic environment.(Co-Reyes et al., ([24]2020); Borowiec et al.,([15] 2022)).

Machine learning algorithms and probabilistic methods have strengths and limitations and the choice of the appropriate model depends on the research activity. It highly depends on the nature, the size of the data, and also the desired level of interpretability of the specific model. On the contrary, traditional statistical methods offer statistical inference and hypothesis testing tools but struggle to capture patterns in complicated data phenomena. For example, Elith et al.([31] 2006), compared the performance of machine learning and classical probabilistic methods for species distribution modeling. In particular, they found that specific machine learning algorithms such as regression trees and random forest algorithms outperform traditional statistical techniques such as GLMs (Generalized linear models) and GAMs (generalized additive models) concerning predictive accuracy.

Similarly, in another comparison article, Thuiller et al. ([84]2009), compared machine learning algorithms and traditional statistical methods for predicting species distribution under climate change. It appears, that again, the chosen machine learning algorithms such as support vector machine and maximum entropy models outperformed traditional classical methods like logistic regression model and Gaussian process regression. Despite the different approach on the methodology, classical probabilistic and statistical methods and machine learning algorithms, are valuable tools for modern ecology research.

On the one hand, machine-learning methods provide important tools for manipulating complex patterns in ecological data and making more accurate prediction models for regression and classification tasks, while classical statistical methods provide a robust framework for statistical inference and hypothesis testing.

By combining both approaches, researchers can unlock the full potential of machine algorithms for many ecological research tasks. Overall machine learning theory can provide alternative possibilities for ecology research purposes. Complicated ecological patterns can be explained, and accurate predictions can be made for various ecological tasks. Researchers can combine traditional statistical methods and machine-learning tools for decision-making, depending on the specific data set or on the specific scientific questions.

In the section2.1.2, we provide a brief introduction to the machine learning algorithms we used for our analysis.

2.1.1 Aims

In recent years, machine-learning algorithms, including deep learning, neural networks, and other methods, become increasingly popular in the study of ecology and in particular, coastal coasts. This chapter aims to provide a guide to ecology researchers regarding the selection of machine-learning algorithms for various research purposes and needs. Moreover, the complexity of ecological phenomena often requires sophisticated statistical analysis instead of traditional simplified models.

In this section, we evaluate and compare popular machine-learning (ML) algorithms, specifically: k-Nearest Neighbors(k-NN), decision trees, naïve Bayes, random forests, and neural networks. Their performance was compared on both classification and regression tasks for predicting ecological data. The predictive accuracy of each algorithm was evaluated through metrics such as R^2 (coefficient of determination), for regression tasks and $F1-$ score for classification, providing an understanding of the advantages and limitations of the aforementioned ML algorithms.

Additionally, this section aims to illustrate the trade-off between the complexity and interpretability of each model. Many ML algorithms are often used as black-box methods decreasing the researchers' understanding of the model. Furthermore, we examine all data requirements for each algorithm across different types and sizes of data. Empirical results provide insights for ecology researchers to make optimal decisions on the selection of an appropriate machine-learning algorithm for specific ecology studies.

The complexity of ecology phenomena usually requires handling factors such as data dimensionality (curse of dimensionality), non-linearity, and interactions within the feature space. We carefully examine these factors for the selection of the most suitable model. In addition, by evaluating the model to unseen data, we can ensure the reliability of the pre-chosen machine learning model. Finally, addressing all these aims, this chapter is in the direction of applying machine learning methods in ecological research to improve the accuracy and credibility of ecological models.

This work aims to provide a valuable resource for ecologists, into the insights of machine learning techniques for exploring ecological phenomena. To compare the selected methods, we used an ecological database related to the sandy beach benthic macrofauna 2.1.2. This database was constructed through extensive sampling conducted in 2022 on the beaches of Bellaria, Igea Marina, Gatteo a Mare, and San Mauro a Mare in the Emilia-Romagna region of Italy.

Sandy beaches are the most common type of open coastline in the world; they dominate the coasts globally making up ~ 70 % of the shoreline (Defeo et al.,([25] 2014). Sandy beaches are transitional environments, naturally dynamic and mainly structured by physical forces, such as tidal regime, wave energy, granulometry, and exposure that determine the morphology and slope as well the circulation patterns of the surf zone (Defeo and McLachlan, [26] 2005). Almost 90 % of the known social ecological systems in the coastal environments are concentrated on sandy beaches with 50 % of them assigned to beach biota

(Harris and Defeo, [38] 2022). Therefore, it is evident that the conservation of the provided social-ecological system is highly correlated with the maintenance of the good ecological state of beaches.

Ecology, as a field of research, explains the interactions between organisms and their environment. Probability and statistical theory is a traditionally used tool for analyzing ecological data, making predictions about the distribution of species, classifying abundances, and overall examining interactions (Tredennick et al., [82] 2021; Spake et al., [79] 2023; Pichler et al., [65] 2020; Kampichler et al., [47] 2010)). The target variables for the beaches were the population parameters related to the number of taxa and the abundance of the intertidal benthic population. Benthos is globally considered the most important syncretic indicator for assessing the ecological status of beaches (Defeo and MacLachlan, [55] 2018).

2.1.2 Materials and Methods

2.1.2.1 Study sites

The Emilia-Romagna coast is located in the North East of Italy 2.1 and comprises 130 km of low and sandy coast, most of which are strongly urbanized. Tidal excursion is low; the average spring tide range is ± 0.4 m and extreme year values are around ± 0.85 m. A general erosive tendency is mainly caused by the reduced sediment transport rates of the rivers and by increased anthropogenic subsidence. (Simeoni and Corbau, ([69] 2009); Itzkin et al.,([45] 2020) and Torresan et al.,([85] 2012)).

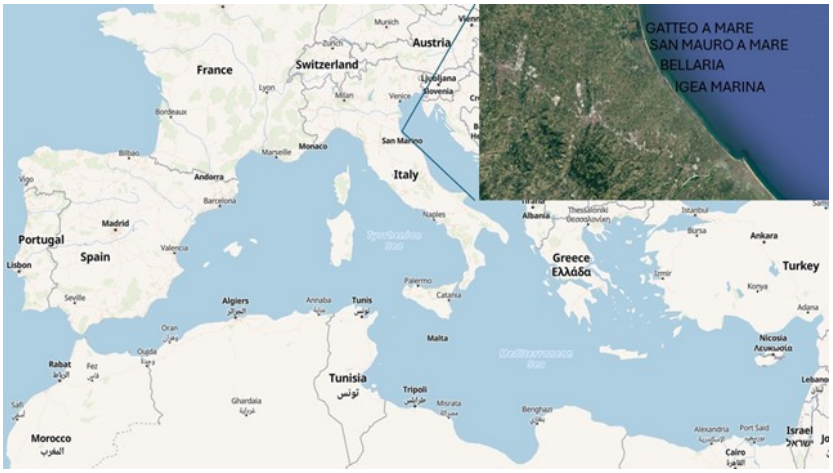


Figure 2.1: study sites.

For this study, data from 4 sandy beaches were used; Bellaria, Igea Marina, San Mauro a Mare, and Gatteo a Mare (2.1); all considered beaches may be classified by dissipative morphodynamic profile, though at different degree, in

erosion and with a management strictly connected with the touristic industry (Satta et al, [71] 2008)

2.1.2.2 Sampling and laboratory analysis

The five sandy beaches were seasonally sampled at 2021, the temporal variability was assessed by sampling each beach at least twice. At each beach and on each survey, a different number of transects perpendicular to the shore were randomly chosen. At each transect, both macrofauna samples and sediment samples to measure the environmental variables were collected at high (H) and low (L) water levels respectively. Sampling was always carried out during the spring tide. A variable number of samples (from two to four) was taken within each transect at H and L respectively. A total of 240 samples were used.

For macrofauna, each sample was formed by pooling sediments collected with 4 plastic corers, each of 10 cm ∅, and sunk to a depth of 10 cm for a total area of 0.0314 m2. Samples were sieved through a 0.5 mm mesh and fixed in 10 % formaldehyde in seawater.

Sediment samples (two to four) for the sediment composition and organic matter content were collected with a 3 cm Ø corer sunk to a depth of 10 cm transported in ice and preserved at -20°C before the analysis. In the laboratory, macrofauna was identified and counted mainly at the species level of taxonomic resolution, but higher taxonomic levels were used for some groups (e.g. Nemertea, Nematoda). After the removal of macrobenthic organisms, dead shell fragments and vegetal residuals contained in each sample was dried at 80°C for 24h, weighed and expressed as g/sample, and termed as "residual detritus".

The median (Md) and the mean (Mz) particle size were estimated by wet sieving. The percentage of total organic matter content (TOM) was estimated for loss after ignition (500 °C for 6 hours) of oven-dried samples (80 °C for 24 hours). For each transect and at each sampling date and beach, the length (m) of intertidal stretch, as the distance between the high- and low water levels, and tidal height, as the difference between the two levels (H and L), were measured.

The intertidal beach slope was calculated as the square root of the sum of the squares of the last two measures. The beach morphodynamic state was summarized by Beach Deposit Index (BDI) (Soares, ([78]2003)), commonly used for microtidal beaches, and takes into account slope and the sand-particle sizes (McLachlan and Dorvlo,([56] 2005)).

2.1.2.3 Machine learning methods in ecology data sets

This chapter analyzes and compares several non-parametric supervised machine learning methods (SML) applied in the aforementioned data set. All methods can be used for classification (where the output of the target variable is a label among several classes) or for regression (where someone tries to predict the exact value of the target variable). The accuracy of each method is compared in the next section. In addition, we derive useful results for the target variable,

the beach benthos, with respect to the feature variables. All were performed with the free software Python `https://www.python.org/`.

k-Nearest Neighbors (k-NN) algorithm is a well-known classification algorithm that was first introduced by Evelyn Fix and Joseph Hodges. (Silverman, B. W., and M. C. Jones [76],1951). It can be used as well for regression problems in many active fields of research, ecology included (Zmri et.al, 2022 [94]) and performs better when connections among the dependent variables and the target variable of the data set are complicated but still have a high 'uniform' correlation. Information about the underlying probability distribution of the data set is not assumed and therefore, the K-NN algorithm can be applied in various scenarios. The method is described below:

First, it is assumed that the data set's features belong to an $n-$ dimensional space equipped with a notion of distance. The most common one is the Euclidean distance, i.e

$$d(A,B) = \sqrt{(a_1 - b_1)^2 + ... + (a_n - b_n)^2}$$

where, $A = (a_1, ..., a_n), B = (b_1, ..., b_n)$ are the vectors representing the feature space. The Euclidean distance is not a restrictive hypothesis and the researcher can also use others (Mahalanobis ([54], 1936) /Manhattan/Minkowski distance).

Algorithm 2 K-Nearest Neighbors (KNN)

---

[80]

Input: Training set $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$
Test sample X
Number of nearest neighbors k

Output: Forecast category y of X

1: Calculate the Euclidean (or other notion of) distance between test sample X and each sample x_i in the training set $\mathcal{D}$.
2: Sort the distances to obtain the k samples closest to X. Define this subset as:

$$\mathcal{D}_k = \{(x_{i_1}, y_{i_1}), (x_{i_2}, y_{i_2}), \ldots, (x_{i_k}, y_{i_k})\},$$

where $i_1, i_2, \ldots, i_k$ are the indices of the corresponding samples.
3: Count the occurrences of each category c_i in $\mathcal{D}_k$.
4: Predict the category y of X as the one with the highest frequency among the k nearest neighbors:

$$y = \arg\max_{c_i} \sum_{j=1}^{k} \mathbf{1}[y_{i_j} = c_i],$$

where $\mathbf{1}$ is the indicator function.

In other words the k-NN algorithm relates a label to an unlabeled vector by selecting the most popular among the k-neighbors that have the shortest

distance. The parameter of the number of neighbors k is tuned again by the user 2.2. Moreover, the k-NN algorithm belongs to the class of non-parametric algorithms(there is no assumption about the data distribution) and performs with high accuracy in smaller data sets. On the other hand, k-NN algorithm is computationally expensive and with high sensitivity in the selection of the parameter k and the choice of the metric.

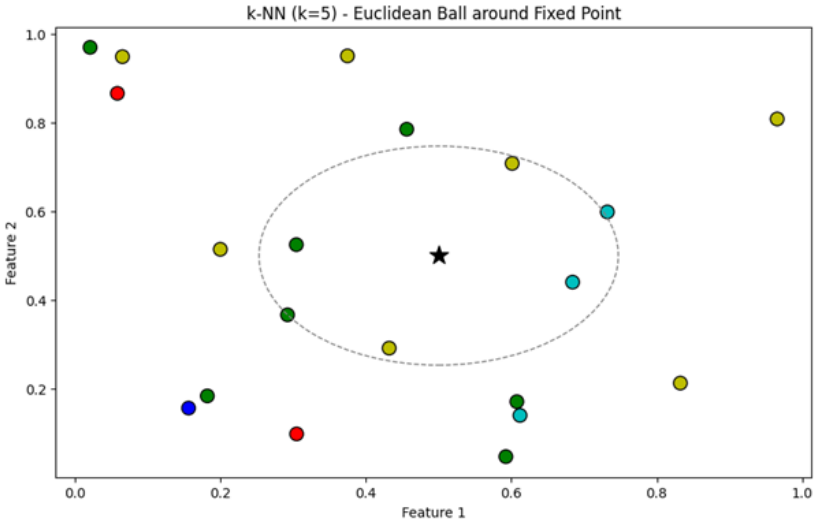


Figure 2.2: 5-nn-algorithm.

In figure 2.2 there is an example of k-NN algorithm with 5 neighbors. Different colors represent different labels of a target variable Y, and the return outcome of the classification estimator in a new example is the most frequent one in the neighborhood.

Naive-Bayes algorithm is a probabilistic method that can be used again for classification and regression tasks in various ecology problems (Fautt et.al [32], 2024). The method uses the Bayes theorem for joint probability. The Bayes theorem states that given a probability measure P and two events A, B with $P(B) > 0$ then

$$P(A|B) = P(B|A)P(A)P(B),$$

where $P(A|B)$ is the conditional probability i.e. the probability that the event A occurs while B happens.

The term Naive comes from the fact that one assumes independence among the feature variables in the data set, which is a restrictive and often violated assumption. It performs fast and efficiently in high dimensional and noisy data and it requires a relatively small amount of them for training the model. Naïve Bayes algorithm is mostly used for classification problems and when the feature variables are not correlated.

Algorithm 3 Naïve Bayes Algorithm

Input: Training set $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$
Test sample X

Output: Forecast category y of X

1: Calculate the prior probability $P(y_c)$ for each category y_c in the training set:

$$P(y_c) = \frac{\text{Count}(y = y_c)}{\text{Total number of samples}}.$$

2: For each feature x_j of X, calculate the conditional probability $P(x_j|y_c)$ using the training data.

3: Compute the posterior probability for each category y_c using Bayes' theorem:

$$P(y_c|X) = \frac{P(X|y_c)P(y_c)}{P(X)} \propto P(X|y_c)P(y_c),$$

where

$$P(X|y_c) = \prod_{j=1}^{n} P(x_j|y_c).$$

4: Assign the category y of X as the one with the highest posterior probability:

$$y = \arg\max_{y_c} P(y_c|X).$$

Neural networks are a class of algorithms considered among the most important ones and by far the most popular ones, used in many active modern research fields. Different architectures have been explored in the past for regression and classification problems in ecology (Borowiec et.al, 2022 [16], Rammer et.al 2019 [67]) . In particular neural networks with backpropagation with deep structured layers are commonly used with different types of activation functions, biases, and the number of layers. Furthermore, they perform with high accuracy for complex, non-linear data structures with large data sets.

On the other hand, a possible disadvantage is that neural networks require large amounts of data for the training procedure, something that might be restrictive. Backpropagation is a gradient descent algorithm for computing the parameters of a neural network. A toy example of a classification model of a three-dimensional feature space is shown in Figure 2.3 where a conceptual scheme of a neural network is given. The figure shows the input, hidden, and output layers. $h_i = \sigma(\sum_i w_j x_j + b)$ and $y_i = \sigma(\sum_i w_j h_j + b)$ indicate respectively the functioning of the hidden layer's neurons and the output determination function. In particular, h notes the summation function in the hidden layer, w denotes the weights of the neural networks usually computed by backpropagation, x is a vector in the feature space, b is the bias term. and finally, σ is the activation function of the neural network.

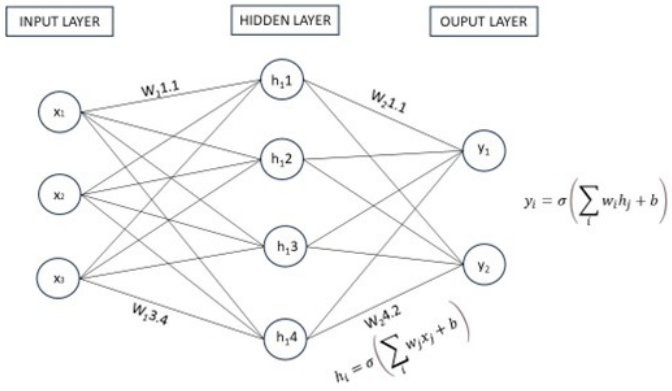


Figure 2.3: Example of a neural network.

Decision trees are a class of hierarchical probabilistic models used for classification and regression tasks also in the ecology research field (Simon et.al, 2023 [77]). The algorithm creates recursively partitions of the feature space according to some randomness chosen by the user of the algorithm. Each rectangle is called a leaf of the tree, and the target variable in regression tasks for an unseen data point x is the average of the points that fall in the rectangle to which x belongs.

A toy example of a decision tree of level 2 for a 2-dimensional space can be seen in Figure 4 below, where the colors represent different labels of the target variable Y for a classification problem. In particular, for the first level of the tree partition $k = 1$, the two-dimensional space is split horizontally in the position $x_2 = 0.5$ In the second step for $k = 2$, both subspaces are split again, horizontally, in the midpoint of each rectangle respectively.

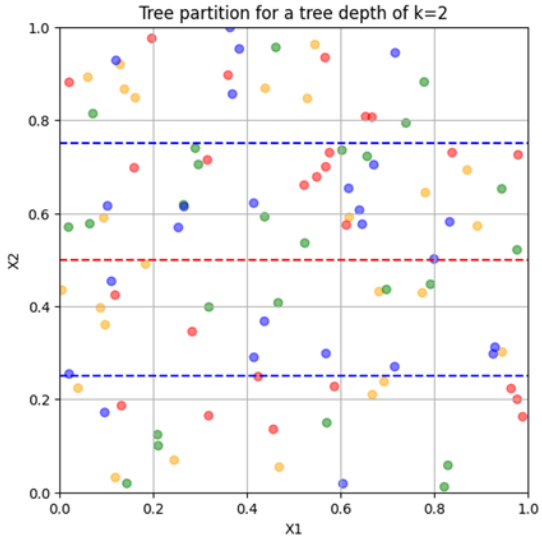


Figure 2.4: Example of tree partition of a two-dimensional feature space with tree level depth two.

An average of $M-$ random trees is called a **random forest**. Random forests are a class of non-parametric statistic machine-learning algorithms used for regression and classification tasks again (Caitlin et.al, 2024 [2]). The first Random Forest algorithm was introduced by Breiman (Breiman, 2001, [23]) and several simplifications have been made concerning the construction of the forest. It is well known that besides the simplicity of the algorithms they perform with high accuracy in many complicated and sparse data sets. Several different types of random forests have been introduced regarding the way that random trees are constructed.

Specifically, the most common random trees are designed with the CART split criterion (Breiman, 2001, [23]) or independently by the data set where the splitting is performed according to some randomness regarding the splitting direction, independently from the feature space. An important advantage of the random forest algorithms is the few parameters that need to be tuned. In particular, a random tree is constructed with a partition of the multidimensional feature space, maybe after some normalization of the data, and the label or the prediction value of the target variable is the empirical expectation in every rectangle (sometimes called cell or leave) on the partition. Finally, an average of

M- different, independent trees is called a random forest, where M is a parameter tuned by the user of the algorithm.

A priori it depends on the data characteristics the choice of the appropriate machine learning method. In general, for relatively small data sets with simple non linear correlations, the choice of k-NN algorithm, Naïve Bayes, and simple regression models is more appropriate. For high dimensional sparse data with feature interactions random forest algorithms perform with high accuracy and finally for complex highly non-linear problems, neural networks with deep architecture outperform, provided enough training data. We empirically confirm in the following chapters the remarks above.

2.1.3 Results

2.1.3.1 Relationships between physical parameters and macrofauna communities

Relationships between community univariate variables (N and S) and the environmental variables Residual detritus, TOM, intertidal length and BDI, four root transformed, and median grain size and slope were separately analyzed at H and L tide levels. At H tide, the number of taxa resulted positively correlated with organic content (TOM%) ($r = 0.38,\ p < 0.0001$) and median grain size (phi) ($r = 0.16,\ p < 0.05$) and negatively correlated with residual detritus ($r = -0.18,\ p < 0.05$); while no significant correlations were found with the other variables. As regards the densities (N) recorded at the H tide, these were positively correlated only with TOM ($r = 0.31,\ p < 0.001$) and slope ($r = 0.20,\ p < 0.01$) . The analysis carried out at L tide between number of taxa S and environmental variables showed significant positive relationships with TOM% ($r = 0.62,\ p < 0.0001$), median grain size ($r = 0.60,\ p < 0.0001$) and BDI ($r = 0.60,\ p < 0.0001$) but a negative relationship with residual detritus ($r = -0.33,\ p < 0.001$) and slope ($r = -0.16,\ p < 0.05$). Correlations between densities and environmental variables at L tide resulted positively correlated for TOM% ($r = 0.34,\ p < 0.0001$), median grain size ($r = 0.20,\ p < 0.01$) ; intertidal length ($r = 0.17,\ p < 0.05$) and slope ($r = 0.60,\ p < 0.0001$) and negatively correlated with residual detritus ($r = -0.43,\ p < 0.0001$). Multivariate multiple regression (DistLM) performed at the H tide level showed a significant relationship between all environmental parameters, singularly considered, and macrobenthic assemblages. However, using the AIC procedure, the combination of all variables together resulted the best model but explained only 22 % of the variation of macrofaunal assemblages of the five beaches. At the L tide level, the DistLM showed a similar result, but in this case all variables together explained a greater part of the variation (39%) of macrofauna of the five beaches.

2.1.4 Overall performance across algorithms.

The dataset encompasses predictions made by various machine learning algorithms: k-Nearest Neighbors (KNN), decision trees, Naïve Bayes, random

forests, and neural networks—compared to actual observed values. This section provides a comprehensive description of the performance of these methods in modeling ecological data, focusing on overall patterns and fitting metrics such as the coefficient of determination R^2 and $F1$ score.

Next, we provide the results for the regression task for the abundance species.

KNN is known for its simplicity and effectiveness in capturing local patterns within the data. The algorithm performed well, with predictions often closely aligning with the actual values. However, its performance can vary significantly depending on the density and distribution of the data points. The coefficient of determination R^2 for KNN was calculated as 0.72. This indicates a good fit for many datasets, but the algorithm sometimes struggles with higher-dimensional data where the notion of "nearness" becomes less clear.

Decision trees provided a clear and interpretable model for the predictions, accurately capturing nonlinear relationships and interactions between features. However, they are prone to overfitting, which can reduce their ability to generalize to new data. The R^2 value for decision trees was 0.68. This reflects their capability to model complex structures in the data but also highlights the variance due to overfitting.

Naïve Bayes classifiers, which rely on probabilistic principles and assume independence among predictors, performed adequately. The algorithm's simplifying assumptions can lead to inaccuracies when violated. The term naive used in the algorithm refers to the assumption of independence of features, which here was crucial.

The R^2 value for Naïve Bayes was 0.55, and this lower value indicates its limitation to more complex ecological data with high correlation structure in the data.

The R^2 value for random forests was 0.85. This high value demonstrates its effectiveness in capturing complex patterns and interactions in the data, making it one of the best-performing algorithms.

Neural networks, especially deep learning models, exhibited high accuracy in predictions. They are particularly suitable for complex ecological datasets due to their ability to model highly nonlinear relationships and interactions. The R^2 value for neural networks was 0.89. The highest value among the algorithms indicates a very strong fit to the data. However, their performance depends on the availability of large datasets and significant computational resources for training.

Below, we summarize the results of the aforementioned results,

Algorithm	R^2	**Result explanation**
k-nn algorithm	0.72	good local pattern capture but variability with data complexity.
Decision trees	0.68	strong performance with potential overfitting.
Naive Bayes	0.55	limitations due to independence assumptions.
Random forests	0.85	robustness and reliability.
Neural Networks	0.89	good fit, with computational power.

2.1.4.1 Prediction Results for Taxa number

Regarding the number of taxa, Random Forest proves to be the most stable model. The performances of the Neural Network and KNN models are comparable for the most frequent taxa number values, while KNN appears to underestimate the higher values; despite a similar trend, the Neural Network algorithm produces a lower error for the higher values (Fig. 2.5). Random Forest shows the best performance (Table 2.1) by overestimating the lower values of taxa number. Additionally, the model obtained with the Neural Network algorithm is also the most stable

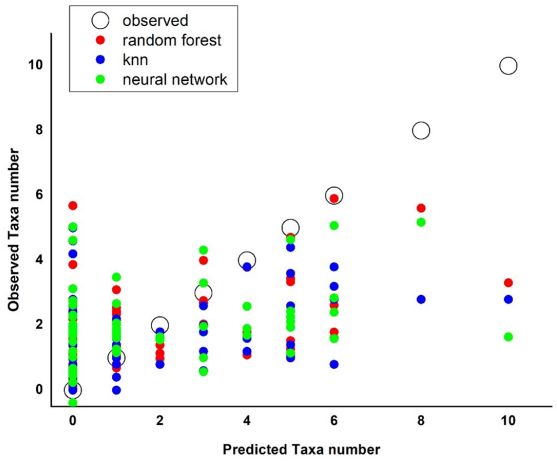


Figure 2.5: prediction results obtained with the regression models for the considered algorithms in relation to the observed data for the number of taxa.

Table 2.1: Comparison of performance metrics for Random Forest, K-Nearest Neighbors (K-nn), and Neural Networks.

	Random Forest	**K-nn**	**Neural Networks**
Reduced chi-sqr	1.34846	1.66356	1.23035
Residual sum of squares	86.3016	106.46776	78.7423
R Value	0.51287	0.30891	0.34263
Adj R-square	0.25152	0.08129	0.10361
Root-mse	1.16123	1.28979	1.10921

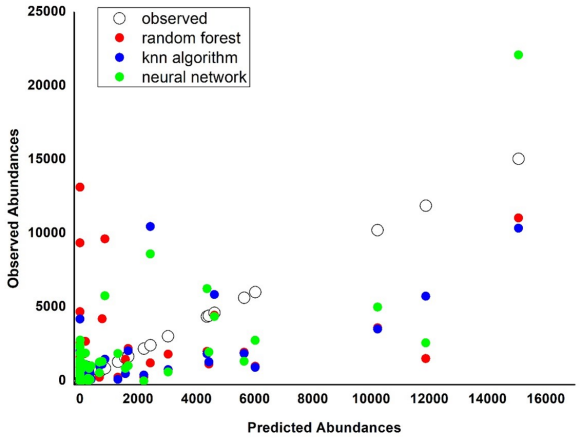


Figure 2.6: prediction results obtained with the regression models for the considered algorithms in relation to the observed data for the number of taxa. .

The predictions obtained for abundances show comparable performances between the Neural Network and KNN algorithms (Table 2.2). In general, the Neural Network algorithm demonstrates the best performance, while the KNN algorithm is the most stable (Table 2). The Random Forest algorithm, on the other hand, shows good performance for the most frequent values, with an increase in error for the less frequent values (Fig.2.6), which worsens its overall performance (Table 2.2)

Table 2.2: Comparison of performance metrics for Random Forest, K-Nearest Neighbors (K-nn), and Neural Networks.

	Random Forest	**K-nn**	**Neural Networks**
Reduced chi-sqr	6.09e +0.6	2.17e+06	4.51e+06
Residual sum of squares	3.90e+08	1.39e+08	2.88e+08
R Value	0.3654	0.70377	0.72387
Adj R-square	0.11998	0.4874	0.51656
Root-mse	2467.39913	1472.48746	2122.87077

2.1.5 Classification results

In this section, we compare the performance of machine learning algorithms for classification tasks. Since we deal with a multilabel classification problem, we calculate the micro f_1 score for the number of taxa and abundance. The best performance for the number of taxa was the KNN and the random forest

algorithm (0.6364) and for the abundance number the random forest algorithm (0.62). In (Table 2.3) and (Table 2.3) we describe in detail the results of each algorithm for both scenarios.

Table 2.3: F1 scores for the number of taxa for each algorithm.

Algorithm	**KNN**	**Tree Algorithm**	**Naïve Bayes**	**Random Forest**	**Neural Network**
F1-Score	0.6364	0.59	0.48	0.6364	0.56

Table 2.4: F1 scores for the abundance for each algorithm.

Algorithm	**KNN**	**Tree Algorithm**	**Naïve Bayes**	**Random Forest**	**Neural Network**
F1-Score	0.58	0.5	0.44	0.62	0.54

As we can address both target functions behave similarly. In both scenarios the k-nn algorithm outperforms and similarly the naïve Bayes underperform.

2.1.6 Discussion

In this section, we discuss the overall performance of the machine learning algorithms concerning their predictive accuracy in our ecological data. Random forests and neural networks perform with high accuracy among all other methods. The evaluation is based on their R^2 and $F1$ values, which indicate the ability of these algorithms to capture the high non-linear relationships in the feature space of the ecological data set. A deep understanding of the construction of the algorithms is necessary for the explanation of the empirical results.

Random forests belong to a class of non-parametrical algorithms used for classification and regression tasks by averaging multiple decision trees. The procedure of averaging trees reduces significantly the variance of the model, and the risk of overfitting is avoided. For our purposes, we explore different tree depths and different numbers of trees. The splitting criterion used was CART (Classification and Regression Trees) and the bootstrap aggregating method was applied to create different data sets sampled with replacement. In particular, the value of R^2 is 0.85 showing the ability of the random forest algorithm to capture non-linear relationships in the feature space despite the complexity and the high computational cost. Neural networks were inspired by the structure of the human brain through connections of neurons. We have explored various neural network structures with deep architectures and nonlinear activation functions the method performs with the highest value of $R^2 = 0.89$ showing the strength of the methods on regression and classification tasks but with high computational costs. It is an important tool for predicting species distribution and abundance occurring in a given set of abiotic conditions since the construction of the layers

and the non-linearity of activation functions can learn various patterns in the training procedure. The k-NN algorithm classifies unlabeled vectors from the feature space according to the closest neighbors based on a pre-defined metric that the space of depending variables is equipped. The only parameter that needs to be tuned is the number of neighbors to be considered. Despite the simplicity of the method, k-NN algorithm performs relatively well with a value $R^2 = 0.72$ where we computed various scenarios for the choice of parameter k and the Euclidean metric. It is obvious by the definition of the algorithm that it highly depends on the density of the points of the feature space, and it does not make any particular assumption on the distribution of the data points. Naïve Bayes algorithm is one of the well-known algorithms for classification tasks.

The algorithm assumes independence of the features and it is a probabilistic method based on the classical Bayes' theorem about conditional probability. The R^2 value of 0.55 for Naïve Bayes was the lowest among the algorithms tested, showing that the hypothesis of independence of the dependent variables is restrictive. However, the naïve Bayes classifier is in general computationally efficient and requires less training time than complex methods such as neural networks and random forests. Moreover, the method can be used as an exploration tool for the ecological data set. Finally, decision trees provide a useful tool for classification and regression tasks. The feature space is partitioned recursively into subsets that are called nodes. The user selects the tree depth of the tiling and the splitting criterion. We explored various tree depths, and we used the gini impurity method for creating the partition. The value of R^2 was 0.68 reflecting the fact that decision trees might overfit and that random forest algorithms reduce the variance of the model. However, decision trees allow researchers to understand, visualize, and explain the decision-making of the model. It is important in ecological research studies were the understanding the correlations of depending variables is crucial.

2.1.6.1 Importance of Algorithm Selection

The comparative analysis highlights the importance of selecting the right algorithm based on the specific characteristics of the ecological data and the research objectives. Each algorithm has its strengths and weaknesses, and the choice should consider factors such as data size, complexity, and the need for interpretability versus predictive power. For instance, random forests and neural networks are well-suited for complex datasets with many variables and non-linear relationships.

However, they require significant computational resources and may be less interpretable compared to simpler models like decision trees and k-nn. On the other hand, naïve Bayes, while efficient and straightforward, may not capture the complexities of the data as effectively due to its independence assumption.

In modern ecological research, traditional statistical approaches such as Generalized Additive Models (GAMs) have long been chosen for hypothesis testing and identifying mathematical relationships between environmental variables in the feature space, such as nutrient levels and species populations. These meth-

ods are valued for providing measures of statistical significance, which are crucial and traditional for modern ecological research theories.

Moreover, machine learning has emerged as a powerful tool for uncovering complex patterns and making highly accurate predictions in many research fields, and hence in modern ecology. However, since the models work under general and abstract hypotheses that sometimes are hard to validate in practice, they can sometimes be less directly suited to the hypothesis-driven frameworks typically used in ecological studies. Even so, machine learning offers significant advantages, particularly when working with large and intricate datasets, such as those from global-scale analyses or comprehensive meta-analyses. It has the potential to reveal previously overlooked ecological relationships, inspire new hypotheses,

Last, both machine learning and traditional statistical methods serve as important, complementary tools for advancing our understanding of ecological systems.

This is a preprint work with Fabio Bozzeda a researcher from Dipartimento di Scienze e Tecnologie Biologiche ed Ambientali

Chapter 3

Improved convergence rates for some kernel random forest algorithms.

In this chapter, we provide some improved rates of convergence for some kernel-based random forest algorithms. We review historical facts about the random forest algorithms and their rate of convergence. We provide the necessary notation and the description of the centered and uniform random forest algorithm as we did in 1. Then we recall the kernel based infinite random forest algorithms and we provide proofs of rates of convergence under certain hypothesis. Finally, we conclude with experiments and the analysis of the reproducing kernel Hilbert space related to the kernel of the infinite centered random forest.

This chapter is part of the work in [43] and is a joint work with Nicola Arcozzi.

3.0.1 Historical review

In Breiman's random forest, the trees are grown based on the CART procedure, (Classification And Regression Trees) where both splitting directions and training sets are randomized. A significant distinction among the class of random forest algorithms consists in the way each individual tree is constructed, and, in particular, the dependence of each tree on the data set. Some of the researchers consider random forests designed independently from the data set [13], [72], [27]. An important tool for algorithmically manipulating random forests is through kernel methods. Breiman [19] observed the connection between kernel theory and random forests by showing the equivalence between tree construction and kernel action. Later this was formalized by Geurts et al. in [34]. In the same direction Scornet in [73] defined KeRF (Kernel Random Forest) by modifying the original algorithm, and providing theoretical and practical results. In particular, in his important work, Scornet provided explicit kernels for some

generalizations of algorithms, their rate of consistency, and comparisons with the corresponding random forests. Furthermore, in the very recent [4] Arnould et al. investigated the trade-off between interpolation of several random forest algorithms and their consistency results.

3.0.2 Notation.

A usual problem in machine learning is, based on n observations of a random vector $(X, Y) \in \mathcal{X} \times \mathbb{R} \subseteq \mathbb{R}^d \times \mathbb{R}$, to estimate the function $m(x) = \mathbb{E}(Y|X = x)$. In classification problems, Y ranges over a finite set. In particular we assume that we are given a training sample $\mathcal{D}_n = \{(X_1, Y_1), ..., (X_n, Y_n)\}$ of independent random variables, where $X_i \in [0, 1]^d$ for every $i = 1, ..., n$ and $Y \in \mathbb{R}$ with a shared joint distribution $\mathbb{P}_{X,Y}$. The goal is using the data set to construct an estimate $m_n : \mathcal{X} \subseteq [0, 1]^d \to \mathbb{R}$ of the function m. Our convergence rate requires an a priori assumption on the regularity of the function m. Following [73], we suppose that m belongs to the class of L Lipschitz functions,

$$|m(x) - m(x')| \leq L \cdot \|x - x'\|.$$

Here, as is [73], we consider on $\mathbb{R}^d$ the distance $\|x - x'\| = \sum_{j=1}^{d} |x_j - x'_j|$.

3.0.3 The Random Forest Algorithm.

3.0.4 The Centered Random Forest vs Centered KeRF, and the Uniform Random Forest vs Uniform KeRF

In this section, we call an estimator function m_n of m is consistent if the following L_2-type of convergence holds,

$$\mathbb{E}(m_n(x) - m(x))^2 \to 0,$$

as $n \to \infty$.

In the centered and uniform forest algorithms, the way the data set $\mathcal{D}_n$ is partitioned is independent of the data set itself.

3.0.4.1 The centered random Forest/ Centered KeRF

The centered forest is designed as follows.

1) Fix $k \in \mathbb{N}$.

2) At each node of each individual tree choose a coordinate uniformly from $\{1, 2, ..d\}$.

3) Split the node at the midpoint of the interval of the selected coordinate.

Repeat step 2)-3) k times. At the end, we have 2^k leaves, or cells. A toy example of this iterative process for $k = 1, 2$ in Figures 4.1,4.2. Our estimation

at a point x is achieved by averaging the Y_i corresponding to the X_i in the cell containing x.
Scornet in [73] introduced the corresponding kernel-based centered random forest providing explicitly the proximity kernel function.

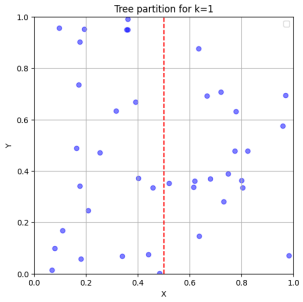


Figure 3.1: Centered algorithm with tree level $k = 1$ with the convention that 1 corresponds to x axis and 2 to the y axis.

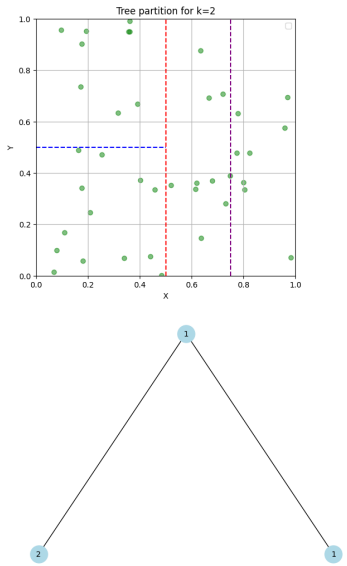


Figure 3.2: Centered algorithm with tree level $k = 1$ with the convention that 1 corresponds to x axis and 2 to the y axis.

Proposition 4. *A centered random forest kernel with $k \in \mathbb{N}$ parameter has the following multinomial expression [73, Proposition 5],*

$$K_k^{Cen}(x,z) = \sum_{\sum_{j=1}^d k_j = k} \frac{k!}{k_1!...k_d!} (\frac{1}{d})^k \prod_{j=1}^d \mathbb{1}_{\lceil 2^{k_j} x_j \rceil = \lceil 2^{k_j} z_j \rceil}.$$

Where K_k^{Cen} is the Kernel of the corresponding centered random forest.

3.0.4.1.1 The uniform random forest / Kernel random forest.

Uniform Random forest was introduced by Biau et al. [13] and is another toy model of Breinman's random forest as a centered random forest. The algorithm forms a partition in $[0,1]^d$ as follows:

1) Fix $k \in \mathbb{N}$.

2) At each node of each individual tree choose a coordinate uniformly from $\{1, 2, ..d\}$.

3) The splitting is performed uniformly on the side of the cell of the selected coordinate.

Repeat step 2)-3) k times. At the end, we have 2^k leaves. Our final estimation at a point x is achieved by averaging the Y_i corresponding to the X_i in the cell x.

Again Scornet in [73, Proposition 6] proved the corresponding kernel-based uniform random forest.

Proposition 5. *The corresponding proximity kernel for the uniform Kernel random forest for parameter $k \in \mathbb{N}$ and $x \in [0,1]^d$ has the following form:*

$$K_k^{Un}(0,x) = \sum_{\sum_{j=1}^d k_j = k} \frac{k!}{k_1!...k_d!}(\frac{1}{d})^k \prod_{m=1}^d \left(1 - x_m \sum_{j=0}^{k_m - 1} \frac{(-\ln x_m)^j}{j!}\right).$$

with the convention that $\sum_{j=0}^{-1} \frac{(-\ln x_m)^j}{j!} = 0$ and by continuity we can extend the kernel also for zero components of the vector.

Unfortunately, it is very hard to obtain a general formula for $K^{Un}(x,y)$ but we consider instead a translation invariant KeRF uniform forest:

$$m_{\infty,n}^{Un}(x) = \frac{\sum_{i=1}^n Y_i K_k^{Un}(0, |X_i - x|)}{\sum_{i=1}^n K_k^{Un}(0, |X_i - x|)}.$$

3.0.5 Proofs of the main theorems.

In this section, after providing some measure concentration type results [17], we improve the rate of consistency of the centered KeRF algorithm. The following lemmata will provide inequalities to derive upper bounds for averages of iid random variables. Lacking a reference, for completeness, we provide detailed proofs of these lemmata. Moreover, we assume in this section that all random variables are real-valued and $||X||_{L_p} \colon = (\mathbb{E}|X|^p)^{\frac{1}{p}}$ and $||X||_\infty \colon = \inf\{t \colon P(|X| \leq t) = 1\}$

Lemma 1. *Let $X_1, ..., X_n$ be a sequence of real independent and identically distributed random variables with $\mathbb{E}(X_i) = 0$. Assuming also that there is a uniform bound for the L_1-norm and the supremum norm i.e. $\mathbb{E}(|X_i|) \leq C$, $||X_i||_\infty \leq CM$ for every $i = 1, ..., n$. Then for every $t \in (0,1)$*

$$\mathbb{P}\big(\{\frac{|\sum_{i=1}^n X_i|}{n} \geq t\}\big) \leq 2e^{-\tilde{C}_C \frac{t^2 n}{M}}.$$

for some positive constant $\tilde{C}_C$ that depends only on C.

Proof. $\forall x \in [0,1]$ one has that $e^x \leq 1 + x + x^2$. By using the hypothesis for every $\lambda \leq \frac{1}{CM}$,

$$\begin{aligned} e^{\lambda X_i} &\leq 1 + \lambda X_i + (\lambda X_i)^2 \quad \Rightarrow \\ \mathbb{E}e^{\lambda X_i} &\leq 1 + \lambda^2 \mathbb{E}(X_i)^2 \\ &\leq 1 + \lambda^2 ||X_i||_1 ||X_i||_\infty \\ &\leq 1 + \lambda^2 C^2 M \\ &\leq e^{\lambda^2 C^2 M}. \end{aligned}$$

By the independence of the random variables X_i,

$$\begin{aligned}\mathbb{E}e^{\sum_{i=1}^n \lambda X_i} &= \prod_{i=1}^n \mathbb{E}e^{\lambda X_i} \\ &\leq \prod_{i=1}^n e^{\lambda^2 C^2 M} \\ &= e^{n\lambda^2 C^2 M}.\end{aligned}$$

Therefore, by Markov inequality

$$\begin{aligned}\mathbb{P}\big(\{\frac{\sum_{i=1}^n X_i}{n} \geq t\}\big) &\leq e^{-\lambda t n}\mathbb{E}e^{\sum_{i=1}^n \lambda X_i} \\ &\leq e^{-\lambda t n}e^{n\lambda^2 C^2 M} \\ &= e^{n\lambda^2 C^2 M - \lambda t n}.\end{aligned}$$

Finally if $C \geq \frac{1}{4}$ we choose, $\lambda = \frac{t}{2C^2 M}$, otherwise for $\lambda = \frac{t}{16CM}$

$$\mathbb{P}\big(\{\frac{\sum_{i=1}^n X_i}{n} \geq t\}\big) \leq e^{-\tilde{C}_C \frac{t^2 n}{M}}.$$

By replacing X_i with $-X_i$ we conclude the proof. □

Lemma 2. *Let $X_1, ..., X_n$ be a non-negative sequence of independent and identically distributed random variables with $\mathbb{E}(X_i) \leq 2$, $||X_i||_\infty \leq M$ for every $i = 1, ..., n$. Let also a sequence of independent random variables ϵ_i following normal distribution with zero mean and finite variance σ^2, for every $i = 1, ..., n$. We assume also that ϵ_i are independent from X_i for every $i = 1, ..., n$.*
Then for every $t \in (0, 1)$,

$$\mathbb{P}\bigg(\frac{1}{n}\sum_{i=1}^n |\epsilon_i X_i| \geq t\bigg) \leq 2\exp\left(-Ct^2\frac{n}{M}\right).$$

with the positive constant C depending only on σ.

Proof.

$$\begin{aligned}
\mathbb{P}\Big(\frac{1}{n}\sum_{i=1}^{n}\epsilon_i X_i \geq t\Big) &= \mathbb{P}\Big(\exp\Big(\frac{\lambda}{n}\sum_{i=1}^{n}\epsilon_i X_i \geq \exp(\lambda t)\Big) \quad \text{for a positive } \lambda \\
&\leq \exp(-\lambda t)\mathbb{E}\exp\Big(\frac{\lambda}{n}\sum_{i=1}^{n}\epsilon_i X_i\Big) \quad \text{By Chebyshev's inequality} \\
&= \exp(-\lambda t)\prod_{i=1}^{n}\mathbb{E}\exp\Big(\frac{\lambda}{n}\epsilon_i X_i\Big) \quad \text{By independence} \\
&= \exp(-\lambda t)\prod_{i=1}^{n}\Big(1+\sum_{k=2}^{\infty}\frac{\lambda^k \mathbb{E}X_i^k \mathbb{E}\epsilon_i^k}{n^k k!}\Big) \\
&\leq \exp(-\lambda t)\prod_{i=1}^{n}\Big(1+\frac{2}{M}\sum_{k=2}^{\infty}\frac{\lambda^k M^k \mathbb{E}\epsilon_i^k}{n^k k!}\Big) \\
&= \exp(-\lambda t)\prod_{i=1}^{n}\Big(1+\frac{2}{M}\Big(\mathbb{E}\exp\Big(\frac{\lambda M}{n}\epsilon_i\Big)-1\Big)\Big) \\
&\leq \exp(-\lambda t)\prod_{i=1}^{n}\Big(1+\frac{2}{M}\Big(\exp\Big(\frac{\lambda^2\sigma^2 M^2}{n^2}\Big)-1\Big)\Big) \\
&= \exp(-\lambda t)\exp\Big(\sum_{i=1}^{n}\Big(\log\Big(1+\frac{2}{M}\Big(\exp\Big(\frac{\lambda^2\sigma^2 M^2}{n^2}\Big)-1\Big)\Big)\Big)\Big) \\
&\leq \exp(-\lambda t)\exp\Big(\sum_{i=1}^{n}\frac{2}{M}\Big(\exp\Big(\frac{\lambda^2\sigma^2 M^2}{n^2}\Big)-1\Big)\Big) \\
&\leq \exp(-\lambda t)\exp\Big(\frac{2n}{M}\Big(\exp\Big(\frac{\lambda^2\sigma^2 M^2}{n^2}\Big)-1\Big)\Big) \\
&\leq \exp(-\lambda t)\exp\Big(\frac{2n}{M}\Big(2\frac{\lambda^2\sigma^2 M^2}{n^2}\Big)\Big) \quad \text{for } \lambda \leq \frac{n}{\sigma M} \\
&= \exp\Big(-\lambda t+\frac{4M}{n}\lambda^2\sigma^2\Big).
\end{aligned}$$

Finally we select $\lambda = \frac{tn}{8M\sigma^2}$, when $\sigma \geq \frac{1}{8}$ and $\lambda = \frac{tn}{M\sigma}$, when $\sigma \leq \frac{1}{8}$

$$\mathbb{P}\Big(\frac{1}{n}\sum_{i=1}^{n}\epsilon_i X_i \geq t\Big) \leq \exp\Big(-C\frac{t^2 n}{M}\Big).$$

Replacing X_i with $-X_i$ we conclude the proof. □

Theorem 8. *$\boldsymbol{Y} = m(\boldsymbol{X}) + \epsilon$ where ϵ is a zero mean Gaussian noise with finite variance independent of $\boldsymbol{X}$. Assuming also that $\boldsymbol{X}$ is uniformly distributed in $[0,1]^d$ and m is a Lipschitz function. Then there exists exists a constant $\tilde{C}$ such that for every $n > 1$, for every $x \in [0,1]^d$*

$$\mathbb{E}(\tilde{m}_{\infty,n}^{Cen}(x) - m(x))^2 \leq \tilde{C} n^{-\left(\frac{1}{1+d\log 2}\right)}(\log n).$$

Proof. Following the notation in [73], let $x \in [0,1]^d$, $\|m\|_\infty = \sup_{x\in[0,1]^d} |m(x)|$, and by the construction of the algorithm

$$\tilde{m}^{Cen}_{n,\infty}(x) = \frac{\sum_{i=1}^n Y_i K_k(x, X_i)}{\sum_{i=1}^n K_k(x, X_i)}.$$

Let

$$A_n(x) = \frac{1}{n}\sum_{i=1}^n \left(\frac{Y_i K_k(x, X_i) - \mathbb{E}(Y K_k(x, X))}{\mathbb{E}(K_k(x, X))} \right),$$

$$B_n(x) = \frac{1}{n}\sum_{i=1}^n \left(\frac{K_k(x, X_i) - \mathbb{E}(K_k(x, X))}{\mathbb{E}(K_k(x, X))} \right),$$

and

$$M_n(x) = \frac{\mathbb{E}(Y K_k(x, X))}{\mathbb{E}(K_k(x, X))}.$$

Hence, we can reformulate the estimator as

$$\tilde{m}^{Cen}_{n,\infty}(x) = \frac{M_n(x) + A_n(x)}{B_n(x) + 1}.$$

Let $t \in (0, \frac{1}{2})$ and the event $C_t(x)$ where $\{A_n(x), B_n(x) \le t\}$.

$$\begin{aligned}\mathbb{E}(\tilde{m}^{cc}_{n,\infty}(x) - m(x))^2 &= \mathbb{E}(\tilde{m}^{cc}_{n,\infty}(x) - m(x))^2 \mathbb{1}_{C_t(x)} + \mathbb{E}(\tilde{m}^{cc}_{n,\infty}(x) - m(x))^2 \mathbb{1}_{C^c_t(x)} \\ &\le \mathbb{E}(\tilde{m}^{cc}_{n,\infty}(x) - m(x))^2 \mathbb{1}_{C^c_t(x)} + c_1 \left(1 - \frac{1}{2d}\right)^{2k} + c_2 t^2.\end{aligned}$$

Where the last inequality was obtained in [73, p.1496] Moreover, in [73],

$$\mathbb{E}(\tilde{m}^{cc}_{n,\infty}(x) - m(x))^2 \mathbb{1}_{C^c_t(x)} \le c_3(\log n)(\mathbb{P}(C^c_t(x)))^{\frac{1}{2}}.$$

In order to find the rate of consistency we need a bound for the probability $\mathbb{P}(C^c_t(x))$. Obviously ,

$$\mathbb{P}(C^c_t(x)) \le \mathbb{P}(|A_n(x)| > t) + \mathbb{P}(|B_n(x)| > t).$$

We will work separately to obtain an upper bound for both probabilities.

Proposition 6. *Let* $\tilde{X}_i = \frac{K_k(x,X_i)}{\mathbb{E}(K_k(x,X))} - 1$ *a sequence of iid random variables. Then for any* $t \in (0,1)$,

$$\mathbb{P}\big(\{\frac{|\sum_{i=1}^n \tilde{X}_i|}{n} \ge t\}\big) = \mathbb{P}\big(|B_n(x)| \ge t\big) \le 2e^{-\tilde{C}_1 \frac{t^2 n}{2^k}}$$

for some positive constant $\tilde{C}_1$.

Proof. It is easy to verify that $\mathbb{E}\tilde{X}_i = 0$, and

$$|\tilde{X}_i| = |\frac{K_k(x,X_i)}{\mathbb{E}(K_k(x,X))} - 1| \leq \frac{K_k(x,X_i)}{\mathbb{E}(K_k(x,X))} + 1,$$

hence, $\mathbb{E}|\tilde{X}_i| \leq 2$.
Finally,

$$||\tilde{X}_i||_\infty = \sup\{|\tilde{X}_i|\} = \sup\{|\frac{K_k(x,X_i)}{\mathbb{E}(K_k(x,X))} - 1|\} \leq \frac{1}{\mathbb{E}(K_k(x,X))} \sup K_k(x,X_i) + 1 \leq 2^k + 1 \leq 2^{k+1}.$$

By Lemma 1,

$$\mathbb{P}\big(\{\frac{|\sum_{i=1}^n \tilde{X}_i|}{n} \geq t\}\big) = \mathbb{P}\big(|B_n(x)| \geq t\big) \leq 2e^{-\tilde{C}_1 \frac{t^2 n}{2^k}}.$$

□

We need a bound for $\mathbb{P}\big(|A_n(x)| > t\big)$ where,

$$A_n(x) = \frac{1}{n}\sum_{i=1}^n \big(\frac{Y_i K_k(x,X_i) - \mathbb{E}(Y K_k(x,X))}{\mathbb{E}(K_k(x,X))}\big).$$

Proposition 7. *Let* $\tilde{Z}_i = \frac{Y_i K_k(x,X_i) - \mathbb{E}(Y K_k(x,X))}{\mathbb{E}(K_k(x,X))}$ *for* $i = 1, ..., n$ *then for every* $t \in (0,1)$,

$$\mathbb{P}\big(\{\frac{|\sum_{i=1}^n \tilde{Z}_i|}{n} \geq t\}\big) = \mathbb{P}\big(|A_n(x)| \geq t\big) \leq 4e^{-C\frac{t^2 n}{2^k}},$$

for some constant C *depending only on* $\sigma, \|m\|_\infty$.

Proof.

$$\begin{aligned}
A_n(x) &= \frac{1}{n}\sum_{i=1}^n \left(\frac{Y_i K_k(x,X_i) - \mathbb{E}(Y K_k(x,X))}{\mathbb{E}(K_k(x,X))}\right) \\
&= \frac{1}{n}\sum_{i=1}^n \left(\frac{m(X_i) K_k(x,X_i) - \mathbb{E}(m(X) K_k(x,X))}{\mathbb{E}(K_k(x,X))}\right) + \frac{1}{n}\sum_{i=1}^n \left(\frac{\epsilon_i K_k(x,X_i) - \mathbb{E}(\epsilon K_k(x,X))}{\mathbb{E}(K_k(x,X))}\right) \\
&= \frac{1}{n}\sum_{i=1}^n \left(\frac{m(X_i) K_k(x,X_i) - \mathbb{E}(m(X) K_k(x,X))}{\mathbb{E}(K_k(x,X))}\right) + \frac{1}{n}\sum_{i=1}^n \left(\frac{\epsilon_i K_k(x,X_i)}{\mathbb{E}(K_k(x,X))}\right).
\end{aligned}$$

Therefore,

$$\begin{aligned}
\mathbb{P}\big(|A_n(x)| \geq t\big) \leq\ & \mathbb{P}\left(\left|\frac{2}{n}\sum_{i=1}^n \frac{m(X_i) K_k(x,X_i) - \mathbb{E}(m(X) K_k(x,X))}{\mathbb{E}(K_k(x,X))}\right| \geq t\right) \\
& + \mathbb{P}\left(\left|\frac{2}{n}\sum_{i=1}^n \frac{\epsilon_i K_k(x,X_i)}{\mathbb{E}(K_k(x,X))}\right| \geq t\right).
\end{aligned}$$

Let $Z_i = \frac{2(m(X_i)K_k(x,X_i)-\mathbb{E}(m(X)K_k(x,X)))}{\mathbb{E}(K_k(x,X))}$ a sequence of iid random variables. It is easy to verify that $\tilde{Z}_i$ are centered and

$$|\tilde{Z}_i| = |\frac{m(X_i)K_k(x,X_i) - \mathbb{E}(m(X)K_k(x,X))}{\mathbb{E}(K_k(x,X))}| \leq 2||m||_\infty \frac{K_k(x,X_i) + \mathbb{E}(K_k(x,X))}{\mathbb{E}(K_k(x,X))}.$$

Hence,

$$\mathbb{E}|Z_i| \leq 4||m||_\infty$$

Finally,

$$\begin{aligned}
||Z_i||_\infty &= \sup\{|Z_i|\} \\
&= 2\sup\{|\frac{m(X_i)K_k(x,X_i) - \mathbb{E}(m(X)K_k(x,X))}{\mathbb{E}(K_k(x,X))}|\} \\
&\leq 2||m||_\infty(2^k+1) \\
&\leq 4||m||_\infty 2^k.
\end{aligned}$$

By Lemma 1,

$$\mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^{n}\frac{m(X_i)K_k(x,X_i) - \mathbb{E}(m(X)K_k(x,X))}{\mathbb{E}(K_k(x,X))}\right| \geq t\right) \leq 2e^{-C\frac{nt^2}{2^k}}.$$

Furthermore let $\tilde{W}_i = \frac{2\epsilon_i K_k(x,X_i)}{\mathbb{E}(K_k(x,X))}$ for $i = 1, ..., n$ a sequence of independent and identically distributed random variables. We can verify that for every for $i = 1, ..., n$:

$$\mathbb{E}\left(\frac{2K_k(x,X_i)}{\mathbb{E}(K_k(x,X))}\right) \leq 2.$$

Finally,

$$\sup\left\{\left|\frac{2K_k(x,X_i)}{\mathbb{E}(K_k(x,X))}\right|\right\} \leq \frac{2}{\mathbb{E}(K_k(x,X))}\sup\{K_k(x,X_i)\} \leq 2^{k+1}.$$

By Lemma 2 it is clear,

$$\mathbb{P}\left(\left|\frac{2}{n}\sum_{i=1}^{n}\frac{\epsilon_i K_k(x,X_i)}{\mathbb{E}(K_k(x,X))}\right| \geq t\right) \leq 2e^{-C_2\frac{nt^2}{2^k}}.$$

We conclude the proposition by observing

$$\mathbb{P}\big(|A_n(x)| \geq t\big) \leq 4e^{-\min\{C_2,C\}\frac{nt^2}{2^k}}.$$

□

Finally, let us compute the rate of consistency of the algorithm-centered KeRF. By Propositions 6,7 one has that

$$\left(\mathbb{P}(C_t^c(x))\right)^{\frac{1}{2}} \leq \left(\mathbb{P}(|A_n(x)| > t) + \mathbb{P}(|B_n(x)| > t)\right)^{\frac{1}{2}} \leq c_3 e^{-c_4\frac{nt^2}{2^k}},$$

for some constants c_3, c_4 independent of k and n.

Thus,

$$\mathbb{E}(\tilde{m}_{\infty,n} - m(x))^2 \leq c_1 \left(1 - \frac{1}{2d}\right)^{2k} + c_2 t^2 + c_3 \log n e^{-c_4 t^2 \frac{n}{2^k}}.$$

We compute the minimum of the right-hand side of the inequality for $t \in (0,1)$,

$$\begin{aligned} 2c_2 t - 2tc_4 \log n c_3 \frac{n}{2^k} e^{-c_4 t^2 \frac{n}{2^k}} = 0 \quad &\Rightarrow \\ e^{-c_4 t^2 \frac{n}{2^k}} &= \frac{c_2}{c_3 c_4} \frac{2^k}{n \log n} \quad \text{and} \\ t^2 &= \frac{1}{c_4} \frac{2^k}{n} \log\left(\frac{c_2}{c_3 c_4} \frac{n \log n}{2^k}\right). \end{aligned}$$

Hence, the inequality becomes

$$\begin{aligned} \mathbb{E}(\tilde{m}_{\infty,n} - m(x))^2 &\leq c_1 \left(1 - \frac{1}{2d}\right)^{2k} + c_2 \frac{1}{c_4} \frac{2^k}{n} \log\left(\frac{c_2}{c_3 c_4} \frac{n \log n}{2^k}\right) + c_3 \log n \frac{c_2}{c_3 c_4} \frac{2^k}{n \log n} \\ &= c_1 \left(1 - \frac{1}{2d}\right)^{2k} + c_2 \frac{1}{c_4} \frac{2^k}{n} \log\left(\frac{c_2}{c_3 c_4} \frac{n \log n}{2^k} e^{\frac{c_2}{c_4}}\right). \end{aligned}$$

For every $\epsilon_n \in (0,2]$ it holds, $\log x \leq \frac{1}{\epsilon_n} x^{\epsilon_n}$. Then one has that

$$\mathbb{E}(\tilde{m}_{\infty,n} - m(x))^2 \leq c_1 \left(1 - \frac{1}{2d}\right)^{2k} + \frac{c_2 (e^{\frac{c_2}{c_4}} \frac{c_2}{c_3 c_4})^n}{c_4 \epsilon_n} \left(\frac{2^k}{n} (\log n)^{\frac{\epsilon_n}{1-\epsilon_n}}\right)^{1-\epsilon_n}.$$

We pick $k = c(d) \log_2 \frac{n}{(\log n)^{\frac{\epsilon_n}{1-\epsilon_n}}}$,

thus,

$$\frac{c_2 (e^{\frac{c_2}{c_4}} \frac{c_2}{c_3 c_4})^n}{c_4 \epsilon_n} \left(\frac{2^k}{n} (\log n)^{\frac{\epsilon_n}{1-\epsilon_n}}\right)^{1-\epsilon_n} \leq \frac{c'}{\epsilon_n} n^{(c(d)-1)(1-\epsilon_n)} \log n^{\epsilon_n (1-c(d))},$$

for a constant c' independent of n and,

$$\begin{aligned} c_1 \left(1 - \frac{1}{2d}\right)^{2k} &= c_1 \left(1 - \frac{1}{2d}\right)^{2(c(d) \log_2 \frac{n}{(\log n)^{\frac{\epsilon_n}{1-\epsilon_n}}})} \\ &= c_1 2^{2c(d) \log_2 \left(1 - \frac{1}{2d}\right) \log_2 \frac{n}{(\log n)^{\frac{\epsilon_n}{1-\epsilon_n}}}} \\ &= c_1 n^{2c(d) \log_2 \left(1 - \frac{1}{2d}\right)} \frac{1}{(\log n)^{c(d) \frac{2\epsilon_n}{1-\epsilon_n} \log_2 \left(1 - \frac{1}{2d}\right)}}. \end{aligned}$$

Therefore,

$$c(d) = \frac{\epsilon_n - 1}{2\log_2\left(1-\frac{1}{2d}\right) - (1-\epsilon_n)}.$$

Finally,

$$\begin{aligned}
c_1 n^{2c(d)\log_2\left(1-\frac{1}{2d}\right)} \frac{1}{(\log n)^{c(d)\frac{2\epsilon_n}{1-\epsilon_n}\log_2\left(1-\frac{1}{2d}\right)}} &= c_1 n^{\frac{2(\epsilon_n-1)}{2\log_2\left(1-\frac{1}{2d}\right)-(1-\epsilon_n)}\log_2\left(1-\frac{1}{2d}\right)} \\
&\quad \times \frac{1}{(\log n)^{\frac{2(\epsilon_n-1)}{2\log_2\left(1-\frac{1}{2d}\right)-(1-\epsilon_n)}\frac{2\epsilon_n}{1-\epsilon_n}\log_2\left(1-\frac{1}{2d}\right)}} \\
&= c_1 n^{\frac{2(\epsilon_n-1)}{2\left(\frac{-\frac{1}{2d}}{\log 2}\right)-(1-\epsilon_n)}\left(\frac{-\frac{1}{2d}}{\log 2}\right)} \\
&\quad \times \frac{1}{(\log n)^{\frac{2(\epsilon_n-1)}{2\log_2\left(1-\frac{1}{2d}\right)-(1-\epsilon_n)}\frac{2\epsilon_n}{1-\epsilon_n}\log_2\left(1-\frac{1}{2d}\right)}} \\
&= c_1 n^{-\left(\frac{1-\epsilon_n}{1+(1-\epsilon_n)d\log 2}\right)}(\log n)^{\left(\frac{\epsilon_n}{1+d\log 2(1-\epsilon_n)}\right)}.
\end{aligned}$$

and, for the second term, with the same arguments

$$\frac{\tilde{c}}{\epsilon_n} n^{(c(d)-1)(1-\epsilon_n)} \log n^{\epsilon_n(1-c(d))} = \frac{\tilde{c}}{\epsilon_n} n^{-\left(\frac{1-\epsilon_n}{1+(1-\epsilon_n)d\log 2}\right)}(\log n)^{\left(\frac{\epsilon_n}{1+d\log 2(1-\epsilon_n)}\right)}$$

for a constant $\tilde{c}$ independent of ϵ_n, hence,

$$\mathbb{E}(\tilde{m}^{Cen}_{\infty,n}(x) - m(x))^2 \leq \frac{C}{\epsilon_n} n^{-\left(\frac{1-\epsilon_n}{1+(1-\epsilon_n)d\log 2}\right)}(\log n)^{\left(\frac{\epsilon_n}{1+d\log 2(1-\epsilon_n)}\right)},$$

and consequently,

$$\begin{aligned}
\frac{C}{\epsilon_n} n^{-\left(\frac{1-\epsilon_n}{1+(1-\epsilon_n)d\log 2}\right)}(\log n)^{\left(\frac{\epsilon_n}{1+d\log 2(1-\epsilon_n)}\right)} &= \frac{C}{\epsilon_n} n^{-\left(\frac{1}{1+d\log 2}\right)}(\log n)^{\left(\frac{\epsilon_n}{1+d\log 2(1-\epsilon_n)}\right)} \\
&\quad \times n^{\left(\frac{\epsilon_n}{(1+d\log 2)(1+(1-\epsilon_n))d\log 2}\right)} \\
&\leq \frac{C}{\epsilon_n} n^{-\left(\frac{1}{1+d\log 2}\right)}(\log n)^{\left(\frac{\epsilon_n}{d\log 2(1-\epsilon_n)}\right)} \\
&\quad \times (\log n)^{\frac{\log n}{\log\log n}\left(\frac{\epsilon_n}{(d\log 2)^2(1-\epsilon_n)}\right)}.
\end{aligned}$$

Finally we finish the proof by selecting $\epsilon_n = \frac{1}{\log n}$,
and

$$\mathbb{E}(\tilde{m}^{Cen}_{\infty,n}(x) - m(x))^2 \leq \tilde{C} n^{-\left(\frac{1}{1+d\log 2}\right)}(\log n).$$

□

Theorem 9. *$\boldsymbol{Y} = m(\boldsymbol{X}) + \epsilon$ where ϵ is a zero mean Gaussian noise with finite variance independent of $\boldsymbol{X}$. Assuming also that $\boldsymbol{X}$ is uniformly distributed in $[0,1]^d$ and m is a Lipschitz function. Providing $k \to \infty$, there exists a constant $\tilde{C}$ such that for every $n > 1$, for every $x \in [0,1]^d$*

$$\mathbb{E}(\tilde{m}^{Un}_{\infty,n}(x) - m(x))^2 \leq \tilde{C} n^{-\left(\frac{1}{1+\frac{3}{2}d\log 2}\right)} (\log n).$$

Proof. By arguing with the same reasoning as the proof of the centered random forest we can verify that

$$\left(\mathbb{P}(C^c_t(x))\right)^{\frac{1}{2}} \leq \left(\mathbb{P}(|A_n(x)| > t) + \mathbb{P}(|B_n(x)| > t)\right)^{\frac{1}{2}} \leq c_3 e^{-c_4 \frac{nt^2}{2^k}}.$$

for some constants c_3, c_4 independent of k and n. The rate of consistency for the Uniform KeRF is the minimum of the right hand in the inequality in terms of n

$$\mathbb{E}(\tilde{m}^{Un}_{\infty,n} - m(x))^2 \leq c_1 \left(1 - \frac{1}{3d}\right)^{2k} + c_2 t^2 + c_3 \log n e^{-c_4 t^2 \frac{n}{2^k}}.$$

We compute the minimum of the right-hand side of the inequality for $t \in (0,1)$,

$$\begin{aligned} 2c_2 t - 2tc_4 \log n c_3 \frac{n}{2^k} e^{-c_4 t^2 \frac{n}{2^k}} = 0 \quad &\Rightarrow \\ e^{-c_4 t^2 \frac{n}{2^k}} = \frac{c_2}{c_3 c_4} \frac{2^k}{n \log n} \quad &\text{and} \\ t^2 = \frac{1}{c_4} \frac{2^k}{n} \log\left(\frac{c_2}{c_3 c_4} \frac{n \log n}{2^k}\right). \end{aligned}$$

Hence, the inequality becomes,

$$\begin{aligned} \mathbb{E}(\tilde{m}^{Un}_{\infty,n}(x) - m(x))^2 &\leq c_1 \left(1 - \frac{1}{3d}\right)^{2k} + c_2 \frac{1}{c_4} \frac{2^k}{n} \log\left(\frac{c_2}{c_3 c_4} \frac{n \log n}{2^k}\right) \\ &\quad + c_3 \log n \frac{c_2}{c_3 c_4} \frac{2^k}{n \log n} \\ &= c_1 \left(1 - \frac{1}{3d}\right)^{2k} + c_2 \frac{1}{c_4} \frac{2^k}{n} \log\left(\frac{c_2}{c_3 c_4} \frac{n \log n}{2^k} e^{\frac{c_2}{c_4}}\right). \end{aligned}$$

For every $\epsilon_n \in (0,2]$ it holds, $\log x \leq \frac{1}{\epsilon_n} x^{\epsilon_n}$. Then one has that,

$$\mathbb{E}(\tilde{m}^{Un}_{\infty,n} - m(x))^2 \leq c_1 \left(1 - \frac{1}{3d}\right)^{2k} + \frac{c_2 (e^{\frac{c_2}{c_4}} \frac{c_2}{c_3 c_4})^n}{c_4 \epsilon_n} \left(\frac{2^k}{n} (\log n)^{\frac{\epsilon_n}{1-\epsilon_n}}\right)^{1-\epsilon_n}.$$

We pick $k = c(d) \log_2 \frac{n}{(\log n)^{\frac{\epsilon_n}{1-\epsilon_n}}}$

Therefore,

$$\frac{c_2\big(e^{\frac{c_2}{c_4}}\frac{c_2}{c_3c_4}\big)^n}{c_4\epsilon_n}\left(\frac{2^k}{n}(\log n)^{\frac{\epsilon_n}{1-\epsilon_n}}\right)^{1-\epsilon_n} \leq \frac{c'}{\epsilon_n}n^{(c(d)-1)(1-\epsilon_n)}\log n^{\epsilon_n(1-c(d))},$$

for a constant c' independent of n and,

$$\begin{aligned} c_1\left(1-\frac{1}{3d}\right)^{2k} &= c_1\left(1-\frac{1}{3d}\right)^{2c(d)\log_2\frac{n}{(\log n)^{\frac{\epsilon_n}{1-\epsilon_n}}}} \\ &= c_1 2^{2c(d)\log_2\left(1-\frac{1}{3d}\right)\log_2\frac{n}{(\log n)^{\frac{\epsilon_n}{1-\epsilon_n}}}} \\ &= c_1 n^{2c(d)\log_2\left(1-\frac{1}{3d}\right)}\frac{1}{(\log n)^{c(d)\frac{2\epsilon_n}{1-\epsilon_n}\log_2\left(1-\frac{1}{3d}\right)}}. \end{aligned}$$

Therefore,

$$c(d) = \frac{\epsilon_n - 1}{2\log_2\left(1-\frac{1}{3d}\right) - (1-\epsilon_n)}$$

Finally,

$$\begin{aligned} c_1 n^{2c(d)\log_2\left(1-\frac{1}{3d}\right)}\frac{1}{(\log n)^{c(d)\frac{2\epsilon_n}{1-\epsilon_n}\log_2\left(1-\frac{1}{3d}\right)}} &= c_1 n^{\frac{2(\epsilon_n-1)}{2\log_2\left(1-\frac{1}{3d}\right)-(1-\epsilon_n)}\log_2\left(1-\frac{1}{3d}\right)} \\ &\quad\times\frac{1}{(\log n)^{\frac{2(\epsilon_n-1)}{2\log_2\left(1-\frac{1}{3d}\right)-(1-\epsilon_n)}\frac{2\epsilon_n}{1-\epsilon_n}\log_2\left(1-\frac{1}{3d}\right)}} \\ &= c_1 n^{\frac{2(\epsilon_n-1)}{2\left(\frac{-\frac{1}{3d}}{\log 2}\right)-(1-\epsilon_n)}\left(\frac{-\frac{1}{3d}}{\log 2}\right)} \\ &\quad\times\frac{1}{(\log n)^{\frac{2(\epsilon_n-1)}{2\left(\frac{-\frac{1}{3d}}{\log 2}\right)-(1-\epsilon_n)}\frac{2\epsilon_n}{1-\epsilon_n}\left(\frac{-\frac{1}{3d}}{\log 2}\right)}} \\ &= n^{-\left(\frac{2(1-\epsilon_n)}{1+(1-\epsilon_n)d\log 2}\right)}\frac{1}{(\log n)^{\frac{2\epsilon_n}{-2+3d\log 2(\epsilon_n-1)}}} \\ &= n^{-\left(\frac{2(1-\epsilon_n)}{2+(1-\epsilon_n)3d\log 2}\right)}(\log n)^{\left(\frac{2\epsilon_n}{2+3d\log 2(1-\epsilon_n)}\right)}. \end{aligned}$$

and, for the second term, with the same arguments

$$\frac{\tilde{c}}{\epsilon_n}n^{(c(d)-1)(1-\epsilon_n)}\log n^{\epsilon_n(1-c(d))} = \frac{\tilde{c}}{\epsilon_n}n^{-\left(\frac{1-\epsilon_n}{1+(1-\epsilon_n)\frac{3}{2}d\log 2}\right)}(\log n)^{\left(\frac{\epsilon_n}{1+d\frac{3}{2}\log 2(1-\epsilon_n)}\right)},$$

for a constant $\tilde{c}$ independent of ϵ_n hence,

$$\mathbb{E}(\tilde{m}^{Un}_{\infty,n}(x) - m(x))^2 \leq \frac{C}{\epsilon_n} n^{-\left(\frac{1-\epsilon_n}{1+(1-\epsilon_n)\frac{3}{2}d\log 2}\right)} (\log n)^{\left(\frac{\epsilon_n}{1+\frac{3}{2}d\log 2(1-\epsilon_n)}\right)},$$

and consequently,

$$\begin{aligned}
\frac{C}{\epsilon_n} n^{-\left(\frac{1-\epsilon_n}{1+(1-\epsilon_n)\frac{3}{2}d\log 2}\right)} (\log n)^{\left(\frac{\epsilon_n}{1+\frac{3}{2}d\log 2(1-\epsilon_n)}\right)} &= \frac{C}{\epsilon_n} n^{-\left(\frac{1}{1+\frac{3}{2}d\log 2}\right)} (\log n)^{\left(\frac{\epsilon_n}{1+\frac{3}{2}d\log 2(1-\epsilon_n)}\right)} \\
&\quad \times n^{\left(\frac{\epsilon_n}{(1+\frac{3}{2}d\log 2)(1+(1-\epsilon_n))d\log 2}\right)} \\
&\leq \frac{C}{\epsilon_n} n^{-\left(\frac{1}{1+\frac{3}{2}d\log 2}\right)} (\log n)^{\left(\frac{\epsilon_n}{\frac{3}{2}d\log 2(1-\epsilon_n)}\right)} \\
&\quad \times (\log n)^{\frac{\log n}{\log\log n}\left(\frac{\epsilon_n}{(\frac{3}{2}d\log 2)^2(1-\epsilon_n)}\right)}.
\end{aligned}$$

Finally we finish the proof by selecting $\epsilon_n = \frac{1}{\log n}$,
and

$$\mathbb{E}(\tilde{m}^{Un}_{\infty,n}(x) - m(x))^2 \leq \tilde{C} n^{-\left(\frac{1}{1+\frac{3}{2}d\log 2}\right)} (\log n).$$

□

3.0.6 Plots and Experiments.

In the following section, we summarize the rates of convergence for the centered KeRF and the uniform KeRF, and we compare them with the minimax rate of convergence over the class of the Lipschitz functions [91]. In addition, we provide numerical simulations where we compare the L_2- error for different choices of the tree depth. All experiments performed with the software Python `https://www.python.org/`, mainly with the numpy library, where random sets uniformly distributed in $[0,1]^d$ have been created, for various examples for the dimension d and the function Y. For every experiment the set is divided into a training set (80 %) and a testing set (20 %); then the L_2-error ($\sum_{X_i \in \text{ test set}} (\tilde{m}(X_i) - Y_i)^2$) and the standard deviation of the error is computed.

For the centered KeRF we compare three different values of tree depth, which from theory provide different convergence rates. First, the choice of k in [73, Theorem 1] that provides the previous convergence rate; second, the selection of k as it was delivered from the theorem 1; and, third, the case where the estimator, in probability, interpolates the data set, but the known convergence rate is slow [4, Theorem 4.1], $O(\log n^{-\frac{d-5}{6}})$ for the dimension of the feature space $d > 5$.

For the uniform KeRF, we compare the two values of tree depth as they were derived from [73] and Theorem 2 nevertheless, it is not known if the uniform-KeRF algorithm converges when our estimator function interpolates the data set. Of course, in practice, since real data might violate the assumptions of

the theorems, one should try cross-validation for tuning the parameters of the algorithms.
Comparing the rates of consistency for centered KeRF and the depth of the corresponding trees:

- Scornet in [73, Theorem 1] rate of convergence: $n^{-(\frac{1}{d\log 2+3})}(\log n)^2$, and $k = \lceil \frac{1}{\log 2+\frac{3}{d}} \log \frac{n}{\log n^2} \rceil$
- New rate of convergence :

$$n^{-\left(\frac{1}{1+d\log 2}\right)}(\log n), \text{ and } k = \lceil \frac{\frac{1}{\log n} - 1}{2\log_2(1-\frac{1}{2d}) - (1-\frac{1}{\log n})} \log_2 \frac{n}{(\log n)^{\frac{\frac{1}{\log n}}{1-\frac{1}{\log n}}}} \rceil$$

- Minimax [91] rate of consistency over the class of Lipschitz functions: $n^{\frac{-2}{d+2}}$ functions

Thus, theoretically, the improved rate of consistency is achieved when trees grow at a deeper level compared with the parameter selection in [73, Theorem 1].

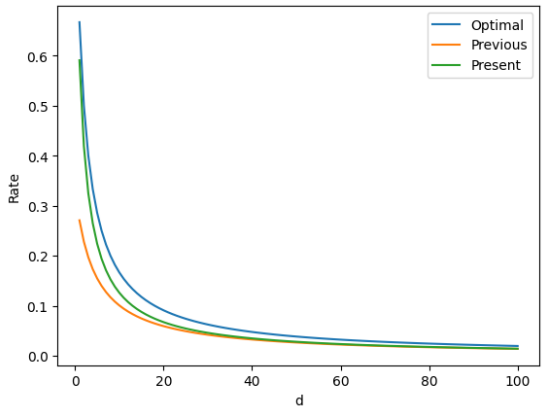


Figure 3.3: Plot of the exponents of n, for the previous rate of convergence for the centered KeRF algorithm, the new rate of convergence, and the optimal over the class of the Lipschitz functions.

As it is evident from 3.3, the improvement in the convergence rate is more significant in the low dimensional feature space scenarios. The constant $\tilde{C} = \tilde{C}(d)$ of theorem 1 depends on the dimension d of the space. The convergence rates in the literature do not try to have a good estimate for $\tilde{C}$, and they are significant for fixed values of d only.

Finally, we note that Klusowski's rate of convergence in [49], $\mathcal{O}\left((n\log^{\frac{d-1}{2}} n)^{-(\frac{1+\delta}{d\log 2+1})}\right)$, where δ is a positive constant that depends on the dimension of the feature space

d and converges to zero as d approaches infinity, is sharp and better than the one in theorem 1 $\mathcal{O}\left(n^{-\left(\frac{1}{1+d\log 2}\right)}(\log n)\right)$ for small values of d. For large values of n and d the two estimates are essentially the same, but for now, we do not know if in general, the rate of convergence of the centered KeRF is not improvable.

Comparing the rates of convergence for uniform KeRF and the depth of the corresponding trees:

- Scornet in [73, Theorem 2]: rate of convergence : $n^{-(\frac{2}{3dlog2+6})}(\log n)^2$, and $k = \lceil \frac{1}{\log 2+\frac{3}{d}} \log \frac{n}{\log n^2} \rceil$
- New rate of convergence:
  $n^{-(\frac{2}{3d\log 2+2})}(\log n)$, and $k = \lceil \frac{\frac{1}{\log n}-1}{2\log_2(1-\frac{1}{3d})-(1-\frac{1}{\log n})} \log_2 \frac{n}{(\log n)^{\frac{\frac{1}{\log n}}{1-\frac{1}{\log n}}}} \rceil$
- Minimax [91] rate of convergence for the consistency over the class of Lipschitz functions: $n^{\frac{-2}{d+2}}$ functions

Thus, theoretically, as in the case of centered random KeRF, the improved rate of consistency is achieved when trees grow at a deeper level compared with the parameter selection in [73, Theorem 2].

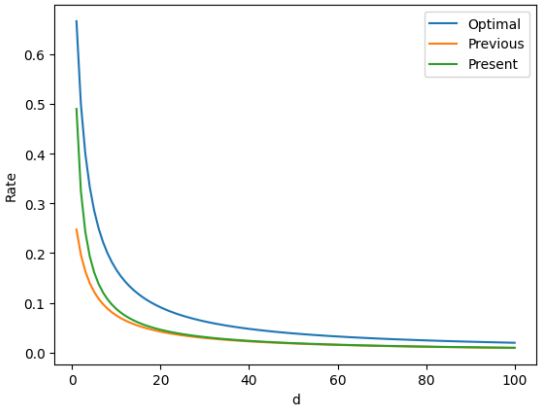


Figure 3.4: Plot of the exponents of n, for the previous rate of convergence for the uniform KeRF algorithm, the new rate of convergence, and the optimal over the class of the Lipschitz functions.

The same considerations on the dependence of the constant $\tilde{C}$ on d we made for the centered KeRF hold in the uniform KeRF case as one can see in Figure 4.4. Moreover, as of now, it is still unknown to us if the convergence rate of the uniform KeRF can be improved.

Finally, numerical simulations of the L_2-error of the centered KeRF-approximation for three different values of k in Figure 4.5a with the standard deviation of the

errors in Figure 4.5b is provided. In Appendix, more simulations and plots for different target functions and for both algorithms are illustrated.

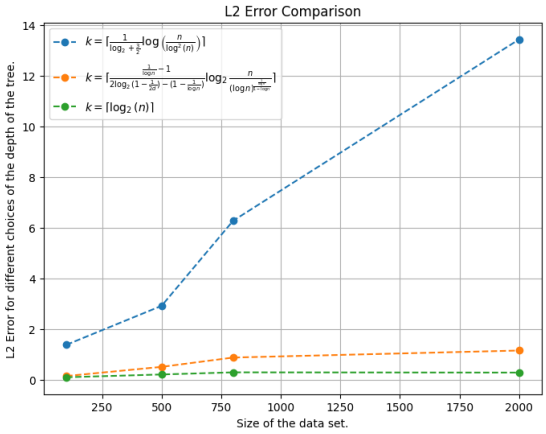


Figure 3.5: Plot of the L_2-error of the centered KeRF-approximation for three different values of k for the function $Y = X_1^2 + e^{-X_2^2} + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \frac{1}{2})$, against different data set size.

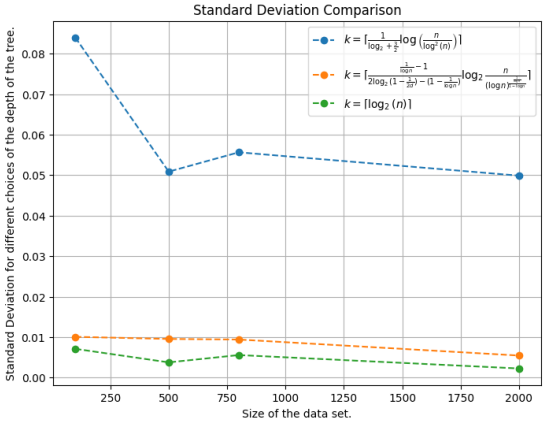


Figure 3.6: Plot of the standard deviation of errors, for the centered KeRF-approximation for three different values of k of the function $Y = X_1^2 + e^{-X_2^2} + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \frac{1}{2})$, against different data set size.

3.0.7 More experiments and analysis of the kernel

In this last section, we provide more experiments of low dimensional regression examples with additive noise, regarding the centered and the uniform KeRF. In particular, we calculate and compare the L_2- errors and the standard deviations

against different sample sizes for different values of the parameter k of the estimator. Moreover, in the following subsection, we study the corresponding reproducing kernel Hilbert space produced by the kernel

$$K_k^{Cen}(x,z) = \sum_{\sum_{j=1}^d k_j = k} \frac{k!}{k_1!...k_d!} (\frac{1}{d})^k \prod_{j=1}^d \mathbb{1}_{\lceil 2^{k_j} x_j \rceil = \lceil 2^{k_j} z_j \rceil},$$

defined in the abelian group $\mathbb{Z}_2^{kd}$. To conclude we recall some notation for finite abelian groups, necessary to define the aforementioned reproducing kernel Hilbert space and estimate its dimension.

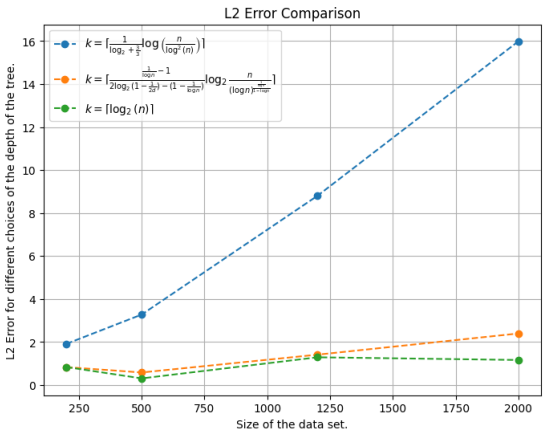


Figure 3.7: Plot of the L_2-error of the centered KeRF-approximation for three different values of k for the function $Y = X_1^2 + \frac{1}{e^{X_2^2}+e^{X_3^2}} + \epsilon$ where $\epsilon \sim \mathcal{N}(0, 0.5)$, against different data set size.

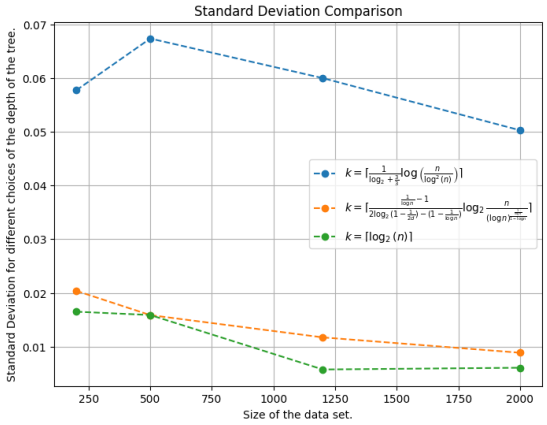


Figure 3.8: Plot of the standard deviation of errors, of the centered KeRF-approximation for three different values of k of the function $Y = X_1^2 + \frac{1}{e^{X_2^2} + e^{X_3^2}} + \epsilon$ where $\epsilon \sim \mathcal{N}(0, 0.5)$, against different data set size.

On Figure 4.6a we see the L_2-error of the centered KeRF-approximation for a three dimensional feature space and on Figure 4.6b the standard deviation of the errors.

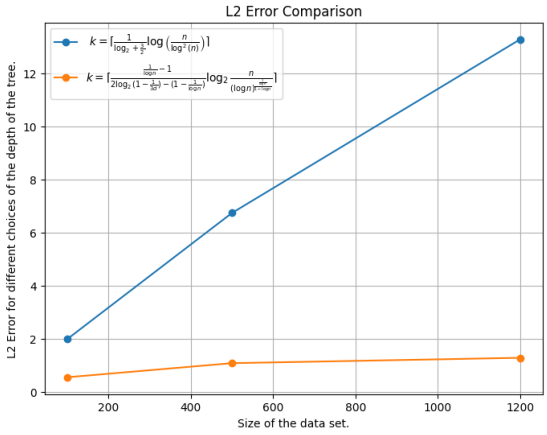


Figure 3.9: Plot of the L_2-error of the uniform KeRF-approximation for two different values of k for the function $Y = X_1^2 + e^{-X_2^2} + \epsilon$ where $\epsilon \sim \mathcal{N}(0, 0.5)$, against different data set size.

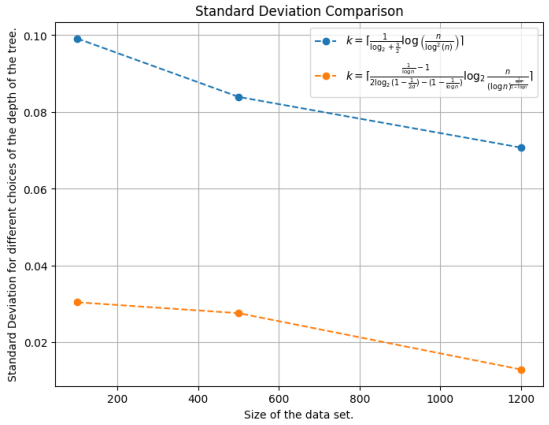


Figure 3.10: Plot of the standard deviation of errors, of the uniform KeRF-approximation for two different values of k for the function $Y = X_1^2 + e^{-X_2^2} + \epsilon$ where $\epsilon \sim \mathcal{N}(0, 0.5)$, against different data set size.

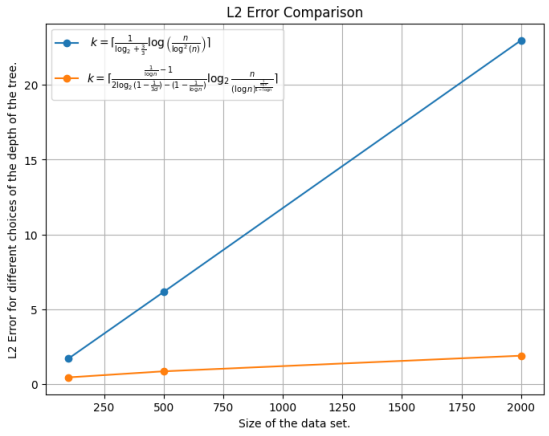


Figure 3.11: Plot of the L_2-error of the uniform KeRF-approximation for two different values of k for the function $Y = X_1^2 + \frac{1}{(e^{X_3^2} + e^{X_2^2})} + \epsilon$ where $\epsilon \sim \mathcal{N}(0, 0.5)$, against different data set size.

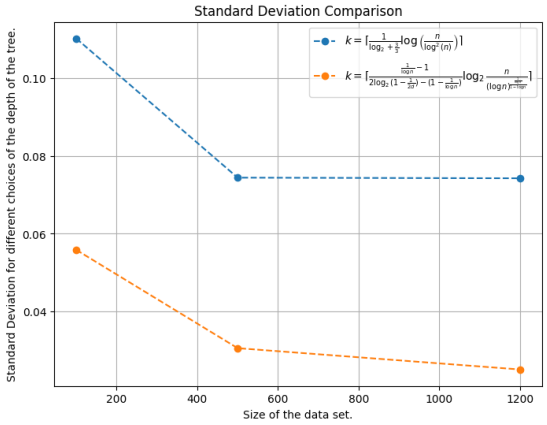


Figure 3.12: Plot of the standard deviation of the errors, of the uniform KeRF-approximation for two different values of k for the function $Y = X_1^2 + \frac{1}{(e^{X_3^2}+e^{X_2^2})} + \epsilon$ where $\epsilon \sim \mathcal{N}(0, 0.5)$, against different data set size.

Figures 4.7a,4.7b show the L_2-error and the standard deviation for the uniform KeRF in a two dimensional feature space and Figures 3.11,3.12 present a three dimensional example respectively.

3.0.8 Analysis of the Kernel

3.0.8.1 Fourier transforms on finite groups and related RKHS

We provide here an alternative description of the centered random forest algorithm, where the dyadic tiling of the hypercube motivates us to define the kernel in the abelian group $\mathbb{Z}_2^{kd}$. First, we define a *Random Tree* Θ. Start with a random variable Θ^0, uniformly distributed in $\{1, \ldots, d\}$, and split $I := [0,1]^d = I_0^{\Theta^0} \cup I_1^{\Theta^0}$, where $I_l^{\Theta_0} = [0,1] \times \cdots \times [l/2, (l+1)/2] \times \ldots [0,1]$, where for $l = 0, 1$ the splitting was performed in the Θ^0-th coordinate. Choose then random variables $\Theta^{1,l}$ $(l = 0, 1)$, distributed as Θ^0, and split each $I_l^{\Theta^0} = I_{l,0}^{\Theta^0,\Theta^1} \cup I_{l,1}^{\Theta^0,\Theta^1}$, where, as before, the splitting is performed at the Θ_1-th coordinate, and $I_{l,0}^{\Theta^0,\Theta^1}$ is the lower half of $I_l^{\Theta^0}$. Iterate the same procedure k times. In order to do that, we need random variables $\Theta^{j;\eta_0,\ldots,\eta_j}$, with $\eta_l \in \{1, \ldots, d\}$ and $j = 1, \ldots, k$. We assume that all such random variables are independent. It is useful think of $\mathbf{\Theta} = \{\Theta^{j;\eta_0,\ldots,\eta_j}\}$ as indexed by a d-adic tree, and, in fact, we refer to $\mathbf{\Theta}$ as a random tree in $[0,1]^d$. We call *cells*, or *leaves*, each of the 2^k rectangles into which $[0,1]^d$ is split at the end of the k^{th} subdivision.

3.0.9 A reproducing kernel from the Centered KeRF

3.0.9.1 The Fourier analysis of the kernel

The kernel in the centered KeRF defines a reproducing kernel Hilbert space (RKHS) H_K structure on a set Γ having 2^{kd} points [5]. In fact, Γ has a group structure, and Fourier analysis can be used. Much research is done in RKHS theory, and in this section (see e.g [1]), we study the structure of H_K in itself. A priori, H_K might have any dimension less or equal to $\#\Gamma$. We show in fact that its dimension is much lower than that, a fact which is somehow surprising, and we believe it is interesting in itself. Furthermore, we prove that there are no nonconstant multipliers in the space H_K. For completeness we provide definitions and notation on Fourier analysis on Abelian groups in 1.1.3.

We identify every real number $x \in [0,1]$ with its binary expression $x = 0.x_1x_2x_3...$ with $x_i \in \{0,1\}$ for $i \in \mathbb{N}$.

Here we consider the group

$$G = \mathbb{Z}_2^{kd} \ni x = (x_i^j)_{\substack{i=1,\dots,k \\ j=1,\dots,d}} = (x^1|x^2|\dots|x^d) = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_k \end{pmatrix}. \tag{3.0.1}$$

The kernel $K : \Gamma \times \Gamma \to \mathbb{C}$ corresponding to the kernel K_k^{cen} is,

$$\begin{aligned} K(a,b) &= \sum_{\substack{l \in \mathbb{N}^d \\ |l|=k}} \frac{1}{d^k} \binom{k}{l} \prod_{j=1}^{d} \chi\left(a_1^j = b_1^j, \dots, a_{k_j}^j = b_{k_j}^j\right) \\ &= \sum_{\substack{l \in \mathbb{N}^d \\ |l|=k}} \frac{1}{d^k} \binom{k}{l} \prod_{j=1}^{d} \prod_{i=1}^{k_j} \chi\left(a_i^j = b_i^j\right) \\ &= \varphi(a-b), \end{aligned} \tag{3.0.2}$$

where $\binom{k}{l}$ is the multidimensional binomial coefficient, χ_E is the characteristic function of the set E, and $a-b$ is the difference in the group $\mathbb{Z}_2^{kd}$. Incidentally, (3.0.2) shows that the kernel K can be viewed as a convolution kernel on the appropriate group structure. For the last equality, we consider the binary representation of a number in $(0,1]$ whose digits are not definitely vanishing. The fact that 0 does not have such representation is irrelevant since the probability that one of the coordinates of the data vanishes is zero.

We now compute the anti-Fourier transform $\mu = \check{\varphi}$. We know that $\sharp(\Gamma) = 2^{kd}$, and that the characters of $\mathbb{Z}_2^{kd}$ have the form

$$\gamma_a(x),\ x \in \mathbb{Z}_2^{kd},\ a \in \widehat{\mathbb{Z}_2^{kd}},\ a \cdot x = a_1^1x_1^1 + \dots + a_k^dx_k^d. \tag{3.0.3}$$

Hence,

$$2^{kd}p^n\mu(x) = d^k \sum_{a \in \Gamma} \varphi(a)\gamma_a(x)$$

$$
\begin{aligned}
&= d^k \sum_{a\in\Gamma} \varphi(a)(-1)^{a\cdot x} \\
&= \sum_{a\in\Gamma} \sum_{\substack{l\in\mathbb{N}^d \\ |l|=k}} \binom{k}{l} \prod_{j=1}^{d} \prod_{i=1}^{k_j} \chi\left(a_i^j = 0\right) (-1)^{a\cdot x} \\
&= \sum_{a\in\Gamma} \sum_{\substack{l\in\mathbb{N}^d \\ |l|=k}} \binom{k}{l} \prod_{j=1}^{d} (-1)^{\tilde{a}_j^{k_j}\cdot \tilde{x}_j^{k_j}} \prod_{i=1}^{k_j} \left[\chi\left(a_i^j = 0\right) (-1)^{a_i^j x_i^j}\right] \\
&\quad \text{where } \tilde{a}_j^{k_j} = \begin{pmatrix} a_{k_j+1}^j \\ \dots \\ a_n^j \end{pmatrix} \text{ is the lower, } (k-k_j)\text{-dimensional} \\
&\quad \text{part of the column } a^j, \\
&= \sum_{\substack{l\in\mathbb{N}^d \\ |l|=k}} \binom{k}{l} \prod_{j=1}^{d} (-1)^{\tilde{a}_j^{k_j}\cdot \tilde{x}_j^{k_j}} \prod_{i=1}^{k_j} \chi\left(a_i^j = 0\right) \\
&= \sum_{\substack{l\in\mathbb{N}^d \\ |l|=k}} \binom{k}{l} \sum_{\substack{a\in\Gamma \\ a_1^1=\dots a_{k_1}^1 = a_1^2 = \dots = a_{k_d}^d = 0}} \prod_{j=1}^{d} (-1)^{\tilde{a}_j^{k_j}\cdot \tilde{x}_j^{k_j}}. \qquad (3.0.4)
\end{aligned}
$$

The last expression vanishes exactly when for all l, there are some $1 \le j \le d$, and some $k_j + 1 \le i \le k$ such that $x_i^j = 1$, due to the presence of the factor $(-1)^{a_i^j x_i^j} = (-1)^{a_i^j}$ which takes values ± 1 on summands having, two by two, the same absolute values.

If, on the contrary, there is l such that for all $1 \le j \le d$, and $k_j + 1 \le i \le k$, we have that $x_i^j = 0$, then $\mu(x) \neq 0$. Since $|l| = k$ and there are kd binary digits involved in the expression of x, the latter occurs exactly when the binary matrix representing x has a large lower region in which all entries are 0. More precisely, the number of vanishing entries must be at least

$$
(k - k_1) + \dots + (k - k_p) = (d-1)k. \qquad (3.0.5)
$$

The number $N(d,k)$ of such matrices is the dimension of H_K, the Hilbert space having K as a reproducing kernel.

Next, we prove some estimates for the dimension of the reproducing kernel Hilbert space.

We summarize the main items in the following statement.

Theorem 10. *Let $K : \Gamma \times \Gamma \to \mathbb{C}$ be the kernel in* (3.0.2)*, $K(a,b) = \varphi(a-b)$, and let*

$$
E_K = \mathit{supp}(\check{\varphi}) \in K. \qquad (3.0.6)
$$

Then,

(i) as a linear space, $H_K = L_{E_K}$, where

$$
\begin{aligned}
E_K &= \{x = (x^1|\dots|x^d) :\ x_i^j = 0 \text{ for } k_j + 1 \le i \le k, \text{ for some } l \\
&= (k_1, \dots, k_d) \in \mathbb{N}^d \text{ with } k_1 + \dots + k_d = k\}; \qquad (3.0.7)
\end{aligned}
$$

(ii) For $x \in E_K$,

$$\check{\varphi}(x) = \frac{1}{2^k d^k} \sum_{\substack{l \in \mathbb{N}^d, \ |l|=k \\ x_i^j = 0 \text{ for } k_j+1 \le i \le k}} \binom{k}{l} \tag{3.0.8}$$

To obtain the expression on (3.0.8), we used the fact that

$$\sharp\{a : \ a_1^1 = \dots a_{k_1}^1 = a_1^2 = \dots = a_{k_p}^p = 0\} = 2^{(d-1)k}.$$

3.0.9.2 Some properties of H_K.

3.0.9.2.1 Linear relations.

Among all functions $\psi : \Gamma \to \mathbb{C}$, those belonging to H_K (i.e., those belonging to L_{E_K}) are characterized by a set of linear equations,

$$0 = 2^{np} p^n \mu(x) = \sum_{\substack{k \in \mathbb{N}^p, \ |k|=n \\ x_i^j = 0 \text{ for } k_j+1 \le i \le n}} \binom{n}{k} \text{ for } x \notin E_K. \tag{3.0.9}$$

3.0.9.2.2 Multipliers.

A *multiplier* of H_K is a function $m : \Gamma \to \mathbb{C}$ such that $m\psi \in H_K$ whenever $\psi \in H_K$.

Proposition 8. *The space H_K has no nonconstant multiplier.*

In particular, it does not enjoy the *complete Pick property*, which has been subject of intensive research for the past twenty-five years [1].

Proof. The space H_K coincides as a linear space with L_{E_K}. Let $\Lambda_E = \check{L_E}$, which is spanned by $\{\delta_x : \ x \in E\}$. Observe that, since $0 = (0|\dots|0) \in E_K$, the constant functions belong to H_K, hence, any multiplier m of H_K lies in H_K; $m = m \cdot 1 \in H_K$.

Suppose m is not constant. Then, $\check{m}(a) \neq 0$ for some $a \in E_K$, $a \neq 0$. Let a be an element in E_K such that $\check{m}(a) \neq 0$. Since $H_K \ni m \cdot \widehat{\delta_x}$ for all x in E_K, and $m \cdot \widehat{\delta_x} = \widehat{\check{m} * \delta_x}$, we have that the support of $\check{m} * \delta_x$ lies in H_K. Now, $\check{m} * \delta_x(y) = \check{m}(x-y)$, hence, we have that, for any x in E_K, $y = x - a$ lies in E_K as well. This forces $a = 0$, hence m to be constant. □

3.0.9.2.3 Bounds for dimension and generating functions.

Theorem 11. *For fixed $d \ge 1$, we have the estimates:*

$$dim(H_K) \sim \frac{2^{k-d+1} k^{d-1}}{(d-1)!}, \ \textit{hence} \ \frac{dim(H_K)}{2^{kd}} \sim \frac{k^{d-1}}{2^{d-1}(d-1)! 2^{k(d-1)}}. \tag{3.0.10}$$

Proof. Let $l_1, l_2, ..., l_d$ such that

$$0 \leq l_1 + l_2 + ... + l_d = m \leq k$$

where m is a parameter and let also $\lambda = |j : l_j \geq 1| = |\{\text{stop 1-digits}\}| = |\{\text{back-entries}\}|$ where $|\cdot|$ denotes the size of the sets, and of course we have that $0 \leq m \leq k$ and $0 \leq \lambda \leq d, m$. Goal to obtain a bound for

$$N(k,d) = \sum_{m=0}^{k} \sum_{\lambda=0}^{d \wedge m} 2^{m-\lambda} \binom{d}{\lambda} |\{(l_1, l_2, ..., l_d) : l_1 + l_2 + ... + l_d = m| \quad \text{and} \quad |\{j : l_j = 1\} = 1|\}.$$

Let $A(m, \lambda)$ the m-th coefficient of x in the polynomial

$$\begin{aligned}(x + x^2 + ...x^m + ...)^\lambda &= (x(1 + x + x^2 + ...)^\lambda \\ &= (x^\lambda (1 + x + ...)^\lambda) \\ &= \frac{x^\lambda}{(1-x)^\lambda}\end{aligned}$$

and $2^m A(m, \lambda)$ is the m-th coefficient of x, for the fraction $\frac{(2x)^\lambda}{(1-2x)^\lambda}$, therefore $2^{m-\lambda} A(m, \lambda)$ is the m-th coefficient of $\frac{x^\lambda}{(1-2x)^\lambda}$. Let's see the first sum, $B(m, d)$ is the m-th coefficient of x :

$$\begin{aligned}\sum_{\lambda=0}^{d \wedge m} \binom{d}{\lambda} 2^{m-\lambda} A(m, \lambda) &= \sum_{\lambda=0}^{d \wedge m} \binom{d}{\lambda} (\frac{x}{1-2x})^\lambda \\ &= (1 + \frac{x}{1-2x})^d \\ &= (\frac{1-x}{1-2x})^d\end{aligned}$$

Again by the same combinatoric argument we are looking the k-th coefficient of the function

$$f(x) = \frac{1}{1-x} (\frac{1-x}{1-2x})^d.$$

Back to the estimate,
Let a_k the coefficient of the power series centered at $z = 0$.

$$\max_{|z|=r} |f(z)| = \max_{|z|=r} \left| \frac{(1-z)^{d-1}}{(1-2z)^d} \right| = \max_{|z|=r} \left| \frac{1}{1-z} \left(\frac{1-z}{1-2z} \right)^d \right| \leq 2 \max_{\theta \in (-\pi, \pi)} \left| \frac{1 - re^{i\theta}}{1 - 2re^{i\theta}} \right|^d$$

After some calculations since r is fixed one has that the maximum is achieved for $\theta = 0$. So $\max_{|z|=r} |f(z)| \leq 2(\frac{1-r}{1-2r})^d$ Our estimation finally becomes :

$$
\begin{aligned}
|a_k| &\leq \frac{2\left(\frac{1-r}{1-2r}\right)^d}{r^k} \\
&= \frac{2(1-r)^d}{r^k(1-2r)^k} \\
&= k^d 2^k (\frac{1}{2} + \frac{1}{2k})^d, \qquad (\text{since}, \quad r = \frac{1}{2}(1 - \frac{1}{k})) \\
&= k^d (1 + \frac{1}{k})^d 2^{k-d}.
\end{aligned}
$$

Thus an estimate for the dimension of H_K is

$$
\frac{|a_k|}{2^{kd}} \lesssim \frac{k^d (1 + \frac{1}{k})^d 2^{k(1-d)}}{2^d}
$$

Another estimate about the dimension of H_K. For $f(z) = \sum_{n=0}^{\infty} a_n z^n$ we have

$$
|a_n| \leq \frac{\max\{|f\left(re^{it}\right)| : \ |t| \leq \pi\}}{r^n}.
$$

Consider the function

$$
f(z) = \frac{(1-z)^{d-1}}{(1-2z)^d}
$$

in $|z| < 1/2$ and let $r = \frac{1-1/k}{2}$. Then,

$$
\begin{aligned}
|a_k| &\leq \frac{(3/2)^{d-1}}{(1/k)^d (1-1/k)^k 2^{-k}} \\
&\leq (3/2)^{d-1} 2^k e k^d.
\end{aligned}
$$

Thus,

$$
\frac{|a_k|}{2^{kd}} \lesssim \frac{k^d (3/2)^d}{2^{k(d-1)}}.
$$

Recursively working out the generating function one gets the estimates in (3.0.10).
□

Chapter 4

A simplified directional KeRF algorithm

In this final chapter, we introduce a simplification of the centered random forest algorithm. We recall again the notations as in 1 and we prove that the infinite centered random forest kernel coincides with the infinite simplified directional one. Moreover we explore rates of convergence of the simplified directional KeRF in the interpolation regime and by optimizing the tree depth parameter k. Finally, we provide some numerical simulations and experiments that also confirm our theoretical results.

The following chapter is part of the work with Nicola Arcozzi in [44].

4.1 The Centered KeRF algorithm

we say that an estimator function m_n of m is consistent if the following L_2-type of convergence holds,

$$\mathbb{E}(m_n(x) - m(x))^2 \to 0,$$

as $n \to \infty$.

In the centered random forest algorithm, the way the data set $\mathcal{D}_n$ is partitioned is independent of the data set itself.

The centered forest is designed as follows.

1) Fix $k \in \mathbb{N}$.

2) At each node of each individual tree choose a coordinate uniformly from $\{1, 2, ..d\}$.

3) Split the node at the midpoint of the interval of the selected coordinate.

Repeat step 2)-3) k times. Finally, we have 2^k leaves, or cells. A toy example of this iterative process for $k = 1, 2$ is in Figures 4.1,4.2. Our estimation at a point

x is achieved by averaging the Y_i corresponding to the X_i in the cell containing x.

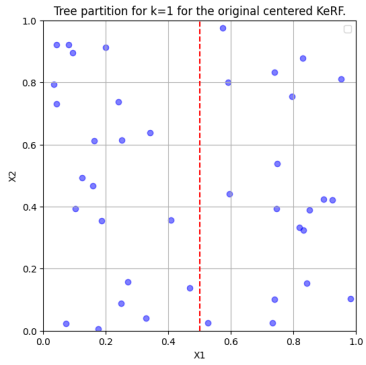


Figure 4.1: Centered algorithm with tree level $k = 1$ with the convention that 1 corresponds to x_1 axis and 2 to the x_2 axis.

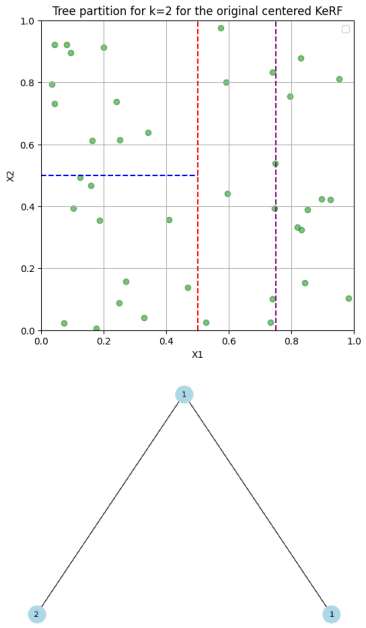


Figure 4.2: Centered algorithm with tree level $k = 2$ with the convention that 1 corresponds to x_1 axis and 2 to the x_2 axis.

Scornet in [73] introduced the corresponding kernel-based centered random forest providing explicitly the proximity kernel function.

Proposition 9. *A centered random forest kernel with $k \in \mathbb{N}$ parameter has the following multinomial expression [73, Proposition 5],*

$$K_k^{Cen}(x,z) = \sum_{\sum_{j=1}^{d} k_j = k} \frac{k!}{k_1! \ldots k_d!} \left(\frac{1}{d}\right)^k \prod_{j=1}^{d} \mathbb{1}_{\lceil 2^{k_j} x_j \rceil = \lceil 2^{k_j} z_j \rceil}.$$

Where K_k^{Cen} is the Kernel of the corresponding centered random forest.

4.1.0.1 Simplified directional KeRF.

The simplified directional random forest algorithm is designed as follows:

1) Fix $k \in \mathbb{N}$.

2) Choose a coordinate uniformly from $\{1, 2, ..d\}$.

3) For every node of each individual tree split the node at the midpoint of the interval of the preselected coordinate.

Repeat step 2)-3) k times. Finally, we have 2^k leaves, or cells. A toy example of this iterative process for $k = 1, 2$ is in Figures 4.3,4.4.

Our estimation at a point x is achieved by averaging the Y_i corresponding to the X_i in the cell containing x.

It is clear from the description of the partition of the hypercube for both algorithms, that the latter is indeed a simplification. At each recursive step of the tiling of a tree, in the centered random forest, the choice of the direction of the splitting procedure needs to be taken in every node separately. On the contrary, in the simplified direction random forest, for each recursive step of every tree, there is only a uniform choice for the direction of the splitting.

For simplicity, we define the probability that two points x, y are connected in the $k-$th level of a tree by $p_k^{sd}(x, y)$ for the simplified directional algorithm and $p_k(x, y)$ respectively for the centered keRF.

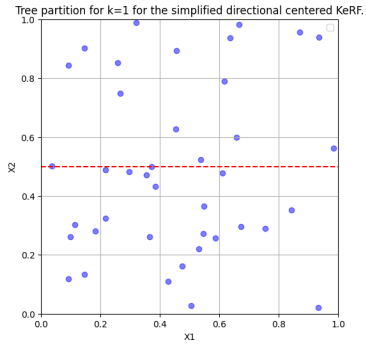


Figure 4.3: Centered algorithm with tree level $k = 1$ with the convention that 1 corresponds to x_1 axis and 2 to the x_2 axis.

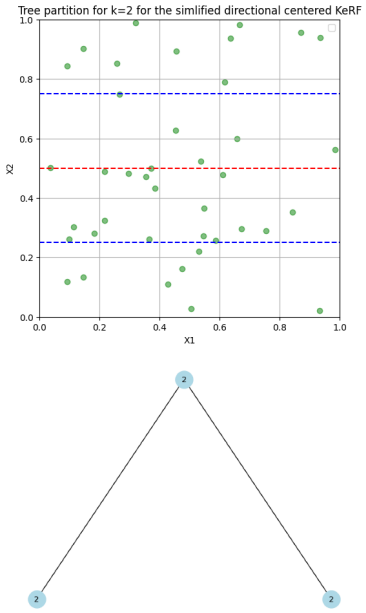


Figure 4.4: Centered algorithm with tree level $k = 2$ with the convention that 1 corresponds to x_1 axis and 2 to the x_2 axis.

Theorem 12. *For every $k \in \mathbb{N}$ and every x, y in $[0,1]^d$*

$$p_k^{sd}(x,y) = p_k(x,y) = K_k^{Cen}(x,y)$$

Proof. For $k = 0, 1$ the result is trivial.

We assume that for every $x, y \in [0,1]^d$ and for $l = 0, 1, ...k$, $p_k^{sd}(x,y) = p_k(x,y)$ and the proof without loss of generality is provided for $d = 2$. Moreover, let $n_{sd}(k)$ resp $(n(k))$ to be the number of different tree expansions of level k for the simplified directional algorithm (resp original centered algorithm), and recursively it is easy to check that

$$n_{sd}(k) = 2n_{sd}(k-1) = ... = 2^k$$

and with the same arguments,

$$n(k) = 2^k n(k-1) = ... = 2^{\frac{k(k+1)}{2}}.$$

Furthermore, we denote by $A_{x,y}^k$ the number of times that the points x, y fall in the same cell in the original centered algorithm and $B_{x,y}^k$ for the simplified directional respectively. Then, $p_k^{sd}(x,y) = \frac{B_{x,y}^k}{n_{sd}(k)}$ and $p_k(x,y) = \frac{A_{x,y}^k}{n(k)}$ and we observe the following cases for the original centered random forest algorithm:

If x, y are not connected for every possible different tree expansion of level k then

$$p_k(x,y) = p_k^{sd}(x,y) = p_{k+1}(x,y) = p_{k+1}^{sd}(x,y) = 0.$$

Furthermore, if x, y are connected for some possible different tree expansion of level k, but they are not connected for any tree expansions of level $k+1$ then

$$p_{k+1}(x,y) = p^{sd}_{k+1}(x,y) = 0.$$

Next, if x, y are connected for some possible different tree expansions of level k, and they are connected only after a horizontal cut and not after a vertical, then

$$\begin{aligned} p_{k+1}(x,y) &= \frac{A^{k+1}_{x,y}}{n(k+1)} \\ &= \frac{\frac{1}{2}2^{k+1}A^{k}_{x,y}}{2^{\frac{(k+1)(k+2)}{2}}} \quad \text{since half of the tree expansions are connected} \\ &= \frac{1}{2}p_k \\ &= \frac{1}{2}p^{sd}_k(x,y) \quad \text{by the induction hypothesis} \end{aligned}$$

and of course,

$$p^{sd}_{k+1}(x,y) = \frac{B^{k+1}_{x,y}}{n_{sd}(k+1)} = \frac{1}{2}\frac{2B^{k}_{x,y}}{2^{k+1}} = \frac{1}{2}p^{sd}_k(x,y).$$

By symmetry, the result holds as well if x, y are connected for some possible different tree expansion of level k, and they are connected only after a vertical cut. Finally, when x, y are connected for some possible different tree expansions of level k, and they are connected as well, after the next cut, in any direction then,

$$\begin{aligned} p_{k+1}(x,y) &= \frac{A^{k+1}_{x,y}}{n(k+1)} \\ &= \frac{2^{k+1}A^{k}_{x,y}}{2^{\frac{(k+1)(k+2)}{2}}} \\ &= p_k \\ &= p^{sd}_k(x,y) \quad \text{by the induction hypothesis} \end{aligned}$$

and,

$$p^{sd}_{k+1}(x,y) = \frac{B^{k}_{x,y}}{2^k} = p^{sd}_k(x,y),$$

which concludes the proof.

□

A simple observation from the theorem above is that the infinite-centered KeRF coincides with the infinite-simplified directional KeRF. Hence, under specific assumptions, we can compute the rate of convergence of the simplified

KeRF. We provide, as a simple corollary, the speed of convergence over the class of the $L-$Lipschitz functions under some hypothesis of the probability distribution of the feature space.

Corollary 2. *([43] Theorem 1.) Under the following assumptions:*

$$Y = m(X) + \epsilon$$
$$X \text{ is uniformly distributed on } [0,1]^d$$
$$\epsilon \sim \mathcal{N}(0, \sigma^2), \quad \sigma < \infty$$
$$m \text{ belongs to the class of } L\text{-Lipschitz functions,}$$

the rate of convergence of the simplified directional KeRF is $\mathcal{O}\left(n^{-\left(\frac{1}{1+d\log 2}\right)}(\log n)\right)$

4.1.1 Interpolating random trees

In this section, we provide an improvement of the rate of convergence of the infinite directional KeRF in the interpolation regime. Since the infinite directional KeRF coincides with the infinite centered KeRF it is enough to study the latter. In [4] Arnould et al. examined the interpolation of data of several random forest models and their capability to preserve consistency.

Next, we provide the basic definitions on data interpolation and we mention some classical results. Finally, we provide the improved convergence rate.

Definition 6. *An estimator m_n interpolates the data set if for every (X_i, Y_i) in the training set we have $m_n(X_i) = Y_i$ almost surely.*

Moreover, since a random forest is an average of $M-$ random trees it is sufficient for the random forest estimator to interpolate the data if every tree estimator interpolates the data set. The tree estimator for a point x is the average of Y_i's for those X_i's belonging to the same cell (or node). Therefore, it is clear that a tree interpolates the data set if a tree is grown until each node contains X_i's with the same values of Y_i's.

In fact, since X is uniformly distributed on $[0,1]^d$ the probability that point belongs to one node is $\frac{1}{2^k}$ and the expected number of points per node are $\frac{n}{2^k}$.

Definition 7. *A centered random forest estimator satisfies the mean interpolation regime if every tree has at least $n-$ nodes. In other words, if $2^k \geq n$.*

The centered random forest fails to preserve consistency in the interpolation regime. This is a result by Arnould et al. [4].

Theorem 13 (Inconsistency of Centered Random Forest)**.** *If $E[m(X)^2] > 0$ and $k_n \geq \log_2(\alpha_n)$, then the infinite centered random forest $m^{cc}_{\infty,n}$ is inconsistent.*

On the contrary, in the same article, Arnould et al. prove that centered kernel random forest are consistent in the interpolation regime with a slow convergence rate, as long the dimension of the feature space is greater than 5.

Intuitively the reason why the kernel-based centered (or simplified directional) estimator is consistent, despite the fact that the tree construction of both algorithms is the same, is the way the kernel estimator is computed. By default, the number of empty nodes in each tree partition is the same for a centered random forest and a centered KeRF. The kernel estimator though does not take into account empty cells and this is why exactly the consistency is preserved, even with slow convergence rates and deep tree depth.

Finally, we mention here the theorem of Arnould et al. and next we state and prove ours.

Theorem 14 (Consistency of Centered KeRF)**.** *Under the following assumptions:*

$$
\begin{aligned}
&Y = m(X) + \epsilon,\\
&X \text{ is uniformly distributed on } [0,1]^d,\\
&\epsilon \sim \mathcal{N}(0, \sigma^2), \quad \sigma < \infty,\\
&m \text{ belongs to the class of } L\text{-Lipschitz functions,}
\end{aligned}
$$

and assuming furthermore that $k = \lfloor \log_2(n) \rfloor$*:*

then the rate of convergence is

$$\mathbb{E}[(\tilde{m}^{cc}_{n,\infty}(x) - m(x))^2] \leq \frac{8L^2d^2}{n^{-2\log_2(1-1/d)}} + C_d(\log_2 n)^{-(d-5)/6}(\log_2(\log_2 n))^{d/3},$$

where $C_d > 0$ *is a constant dependent on noise variance.*

Under the same assumptions for the regression function m, in the mean interpolation regime, we provide an improvement in the rate of convergence.

$$\mathbb{E}[(\tilde{m}^{cc}_{n,\infty}(x) - m(x))^2] \leq c_1 \left(1 - \frac{1}{2d}\right)^{2\log_2 n} + C_3 \frac{\log_2(\log_2 n)^d}{(\log_2 n)^{\frac{d-1}{2}}} \log\left(\frac{\log n (\log_2 n)^{\frac{d-1}{2}}}{\log_2(\log_2 n)^d}\right).$$

We assume for this section that all random variables are real-valued and $||X||_{L_p} := (\mathbb{E}|X|^p)^{\frac{1}{p}}$ and $||X||_\infty := \inf\{t\colon P(|X| \leq t) = 1\}$

We begin with this basic lemma providing tail bounds for centered iid random variables with bounded variance and supremum norm.

Lemma 3. *Let* $X_1, ..., X_n$ *be a sequence of real independent and identically distributed random variables with* $\mathbb{E}(X_i) = 0$*. Assuming also that there is a uniform bound for the* L_2*-norm and the supremum norm i.e.* $\mathbb{E}(|X_i|)^2 \leq Ma_n$*,* $||X_i||_\infty \leq M \leq n$ *for every* $i = 1, ..., n$*. Then for every* $t \leq 2\sqrt{Ma_n}$

$$\mathbb{P}\big(\{\frac{|\sum_{i=1}^n X_i|}{n} \geq t\}\big) \leq 2\exp\big(-\frac{t^2 n}{Ma_n}\big).$$

Proof.

$$\begin{aligned}
\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n} X_i \geq t\right) &= \mathbb{P}\left(\frac{\lambda}{n}\sum_{i=1}^{n} X_i \geq \lambda t\right) \\
&= \mathbb{P}\left(\exp\left(\frac{\lambda}{n}\sum_{i=1}^{n} X_i\right) \geq \exp\left(\lambda t\right)\right) \\
&\leq \exp\left(-\lambda t\right)\mathbb{E}\exp\left(\frac{\lambda}{n}\sum_{i=1}^{n} X_i\right) \\
&\leq \exp\left(-\lambda t\right)\prod_{i=1}^{n}\mathbb{E}\exp\left(\frac{\lambda}{n}X_i\right).
\end{aligned}$$

Where the inequalities are provided by Chebysev inequality and the independence of the random variables. Moreover, for convenience, let $Y_j = \frac{Xj}{n}$ and we observe that, $||Y_i||_\infty \leq 1$ and $\mathbb{E}(Y_i)^2 \leq \frac{Ma_n}{n^2} = \sigma^2$

$$\begin{aligned}
\mathbb{E}\exp\left(\frac{\lambda}{n}X_i\right) &= \mathbb{E}\left(1 + \sum_{k=2}^{\infty}\frac{\lambda^k Y_i^k}{k!}\right) \\
&= 1 + \sum_{k=2}^{\infty}\frac{\lambda^k \mathbb{E}(Y_i^k)}{k!} \\
&\leq 1 + \sum_{k=2}^{\infty}\frac{\lambda^k (\mathbb{E}Y_i^2)^{\frac{k}{2}}||Y_i||_\infty^{\frac{k}{2}}}{k!} \\
&\leq 1 + \sum_{k=2}^{\infty}\frac{\lambda^k ((\sigma^2)^{\frac{1}{2}})^k}{k!} \\
&= 1 + \exp\left(\lambda\sigma\right) - \lambda\sigma - 1 \\
&\leq 1 + \lambda\sigma + (\lambda\sigma)^2 - \lambda\sigma \\
&\leq \exp\left(\lambda\sigma\right)^2
\end{aligned}$$

where we have used that $\exp\left(\lambda\sigma\right) \leq 1 + \lambda\sigma + (\lambda\sigma)^2$ when $\lambda\sigma \leq 1$ and $1 + x \leq e^x$. Hence,

$$\begin{aligned}
\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n} X_i \geq t\right) &\leq \exp\left(-\lambda t\right)\exp\left(\lambda\sigma\right)^2 n \\
&\leq \exp\left(-\frac{t^2}{2\sigma^2 n}\right) \\
&= \exp\left(-\frac{t^2 n}{Ma_n}\right)
\end{aligned}$$

where we choose $\lambda = \frac{t}{2\sigma^2 n}$ which is an accepted value of $\lambda \iff t \leq 2\sqrt{Ma_n}$. We conclude the proof by replacing X_i with $-X_i$. □

Now we move to the next necessary lemma. Here our target random variables are multiplied by centered independent Gaussian noises, but we can still obtain a similar tail bound by getting advantage the shape of the Gaussian tales.

Lemma 4. *Let $X_1, ..., X_n$ be a non-negative sequence of independent and identically distributed random variables with $\mathbb{E}(X_i)^2 \leq Ma_n$, $||X_i||_\infty \leq M \leq n$ for every $i = 1, ..., n$. Let also a sequence of independent random variables ϵ_i following normal distribution with zero mean and finite variance $\tilde{\sigma}^2$, for every $i = 1, ..., n$. We assume also that ϵ_i are independent from X_i for every $i = 1, ..., n$.*
Then for every $t \leq 2\sqrt{Ma_n}$

$$\mathbb{P}\big(\{\frac{|\sum_{i=1}^n X_i|}{n} \geq t\}\big) \leq 2\exp\big(-\frac{t^2 n}{Ma_n\tilde{\sigma}^2}\big).$$

Proof. First of all, we observe from the proof of lemma 1 that,

$$\mathbb{P}\bigg(\frac{1}{n}\sum_{i=1}^n X_i\epsilon_i \geq t\bigg) \leq \exp(-\lambda t)\prod_{i=1}^n \mathbb{E}\exp\bigg(\frac{\lambda}{n}X_i\epsilon_i\bigg),$$

and

$$\begin{aligned}
\mathbb{E}\exp\bigg(\frac{\lambda}{n}X_i\epsilon_i\bigg) &= \mathbb{E}\bigg(1 + \sum_{k=2}^\infty \frac{\lambda^k Y_i^k \epsilon_i}{k!}\bigg) \\
&\leq 1 + \sum_{k=2}^\infty \frac{\lambda^k (\mathbb{E}Y_i^2)^{\frac{k}{2}} ||Y_i||_\infty^{\frac{k}{2}} \mathbb{E}(\epsilon_i)^k}{k!} \\
&\leq 1 + \sum_{k=2}^\infty \frac{\lambda^k ((\sigma^2)^{\frac{1}{2}})^k \mathbb{E}(\epsilon_i)^k}{k!} \\
&= \mathbb{E}\exp\bigg(\lambda\sigma\epsilon_i\bigg) \\
&\leq \mathbb{E}\exp\bigg(\lambda^2\sigma^2\tilde{\sigma}^2\bigg) \qquad \textbf{The gaussian property}
\end{aligned}$$

Finally, the proof works from now on the same way, as the one for the speed of convergence of the previous chapter.

$$\begin{aligned}
\mathbb{P}\bigg(\frac{1}{n}\sum_{i=1}^n X_i\epsilon_i \geq t\bigg) &\leq \exp(-\lambda t)\exp(\lambda\sigma\tilde{\sigma})^2 n \\
&\leq \exp\big(-\frac{t^2}{2\sigma^2\tilde{\sigma}^2 n}\big) \\
&= \exp\big(-\frac{t^2 n}{Ma_n}\big)
\end{aligned}$$

where we choose $\lambda = \frac{t}{2\sigma^2\tilde{\sigma}^2 n}$ which is an accepted value of $\lambda \iff t \leq 2\sqrt{Ma_n}$. We conclude the proof by replacing X_i with $-X_i$.

□

Theorem 15. *$\boldsymbol{Y} = m(\boldsymbol{X}) + \epsilon$ where ϵ is a zero mean Gaussian noise with finite variance σ independent of $\boldsymbol{X}$. Assuming also that $\boldsymbol{X}$ is uniformly distributed in $[0,1]^d$ and m is a Lipschitz function. Then there exists constants c_1, c_2 depending on d, σ and $\|m\|_\infty = \sup_{x\in[0,1]^d} |m(x)|$ such that,*

$$\mathbb{E}(\tilde{m}^{cc}_{n,\infty}(x) - m(x))^2 \leq c_1 \left(1 - \frac{1}{2d}\right)^{2\log_2 n} + C_3 \frac{\log_2(\log_2 n)^d}{(\log_2 n)^{\frac{d-1}{2}}} \log\left(\frac{\log n (\log_2 n)^{\frac{d-1}{2}}}{\log_2(\log_2 n)^d}\right)$$

Proof. Following the notation in [73], let $x \in [0,1]^d$, $\|m\|_\infty = \sup_{x\in[0,1]^d} |m(x)|$, and by the construction of the algorithm

$$\tilde{m}^{Cen}_{n,\infty}(x) = \frac{\sum_{i=1}^n Y_i K_k(x, X_i)}{\sum_{i=1}^n K_k(x, X_i)}.$$

Let

$$A_n(x) = \frac{1}{n}\sum_{i=1}^n \left(\frac{Y_i K_k(x, X_i) - \mathbb{E}(Y K_k(x, X))}{\mathbb{E}(K_k(x, X))}\right),$$

$$B_n(x) = \frac{1}{n}\sum_{i=1}^n \left(\frac{K_k(x, X_i) - \mathbb{E}(K_k(x, X))}{\mathbb{E}(K_k(x, X))}\right),$$

and

$$M_n(x) = \frac{\mathbb{E}(Y K_k(x, X))}{\mathbb{E}(K_k(x, X))}.$$

Hence, we can reformulate the estimator as

$$\tilde{m}^{Cen}_{n,\infty}(x) = \frac{M_n(x) + A_n(x)}{B_n(x) + 1}.$$

Let $t \in (0, \frac{1}{2})$ and the event $C_t(x)$ where $\{A_n(x), B_n(x) \leq t\}$.

$$\begin{aligned}\mathbb{E}(\tilde{m}^{cc}_{n,\infty}(x) - m(x))^2 &= \mathbb{E}(\tilde{m}^{cc}_{n,\infty}(x) - m(x))^2 \mathbb{1}_{C_t(x)} + \mathbb{E}(\tilde{m}^{cc}_{n,\infty}(x) - m(x))^2 \mathbb{1}_{C_t^c(x)} \\ &\leq \mathbb{E}(\tilde{m}^{cc}_{n,\infty}(x) - m(x))^2 \mathbb{1}_{C_t^c(x)} + c_1\left(1 - \frac{1}{2d}\right)^{2k} + c_2 t^2.\end{aligned}$$

Where the last inequality was obtained in [73, p.1496] Moreover, in [73],

$$\mathbb{E}(\tilde{m}^{cc}_{n,\infty}(x) - m(x))^2 \mathbb{1}_{C_t^c(x)} \leq c_3(\log n)(\mathbb{P}(C_t^c(x)))^{\frac{1}{2}}.$$

Proposition 10. *The probability $\mathbb{P}(C_t^c(x)) \leq C\exp(-\frac{t^2 n}{2^k a_n})$ for a constant C independent of k, n and a_n is a sequence that converges to zero as n tends to infinity.*

Proof. First of all, we notice that

$$\mathbb{P}(C_t^c(x)) \leq \mathbb{P}(|A_n(x)| > t) + \mathbb{P}(|B_n(x)| > t).$$

The related result will follow by working separately on both inequalities.
By lemma B.4 [4] for all $x \in [0,1]^d$ and $d \geq 2$ we have

$$\mathbb{E}\big(K_k^c(x,X)\big)^2 \leq \frac{C_1 + C_2(\log_2(k))^d}{k^{\frac{d-1}{2}} 2^k}$$

where $C_1 = 1 + \frac{2d^{\frac{d}{2}}}{(4\pi)^{\frac{d-1}{2}}}$, $C_2 = 5^d(\frac{d-1}{2})^d$ and for convenience let $a_n = \frac{C_1 + C_2(\log_2(k))^d}{k^{\frac{d-1}{2}}}$.

Hence, let $\tilde{X}_i = \frac{K_k(x,X_i)}{\mathbb{E}(K_k(x,X))} - 1$ a sequence of centered iid random variables with

$$||\tilde{X}_i||_\infty = \sup\{|\tilde{X}_i|\} = \sup\{|\frac{K_k(x,X_i)}{\mathbb{E}(K_k(x,X))} - 1|\} \leq \frac{1}{\mathbb{E}(K_k(x,X))} \sup K_k(x,X_i) + 1 \leq 2^{k+1}$$

and

$$\begin{aligned}
\mathbb{E}\big(\frac{K_k^c(x,X_i)}{\mathbb{E}(K_k^c(x,X)}\big)^2 &= \frac{1}{(\mathbb{E}(K_k^c(x,X)))^2}\mathbb{E}\big(K_k^c(x,X)\big)^2 \\
&\leq \frac{1}{(\frac{1}{2^k})^2}\frac{C_1 + C_2(\log_2(k))^d}{k^{\frac{d-1}{2}} 2^k} \\
&= 2^k a_n
\end{aligned}$$

By lemma 3, for every $t \leq 2\sqrt{2^k a_n}$

$$\mathbb{P}(|B_n(x)| > t) \leq 2\exp\left(-C\frac{t^2 n}{2^k a_n}\right)$$

We need an estimate for the $\mathbb{P}(|A_n(x)| > t)$ where,

$$A_n(x) = \frac{1}{n}\sum_{i=1}^n \big(\frac{Y_i K_k(x,X_i) - \mathbb{E}(Y K_k(x,X))}{\mathbb{E}(K_k(x,X))}\big).$$

With simple calculations,

$$\begin{aligned}
A_n(x) &= \frac{1}{n}\sum_{i=1}^n \left(\frac{Y_i K_k(x,X_i) - \mathbb{E}(Y K_k(x,X))}{\mathbb{E}(K_k(x,X))}\right) \\
&= \frac{1}{n}\sum_{i=1}^n \left(\frac{m(X_i) K_k(x,X_i) - \mathbb{E}(m(X) K_k(x,X))}{\mathbb{E}(K_k(x,X))}\right) + \frac{1}{n}\sum_{i=1}^n \left(\frac{\epsilon_i K_k(x,X_i) - \mathbb{E}(\epsilon K_k(x,X))}{\mathbb{E}(K_k(x,X))}\right) \\
&= \frac{1}{n}\sum_{i=1}^n \left(\frac{m(X_i) K_k(x,X_i) - \mathbb{E}(m(X) K_k(x,X))}{\mathbb{E}(K_k(x,X))}\right) + \frac{1}{n}\sum_{i=1}^n \left(\frac{\epsilon_i K_k(x,X_i)}{\mathbb{E}(K_k(x,X))}\right).
\end{aligned}$$

Let $Z_i = 2\frac{\epsilon_i K_k(x,X_i)}{\mathbb{E}(K_k(x,X))}$ a sequence of centered iid random variables with

$$\begin{aligned}\mathbb{E}(Z_i)^2 &\leq \frac{4}{\frac{1}{(2^k)^2}}\mathbb{E}(\epsilon K_k(x,X))^2 \\ &= \frac{4}{\frac{1}{(2^k)^2}}\mathbb{E}(\epsilon)^2\mathbb{E}(K_k(x,X)))^2) \\ &\leq \tilde{c}2^k\sigma^2 a_n\end{aligned}$$

and

$$||\frac{K_k(x,X_i)}{\mathbb{E}(K_k(x,X))}||_\infty \leq 2^k$$

hence by 4

$$\mathbb{P}\big(|Z_i| \geq t\big) = \mathbb{P}\big(\frac{2}{n}\sum_{i=1}^n|\frac{\epsilon_i K_k(x,X_i)}{\mathbb{E}(K_k(x,X))}| \geq t\big) \leq 2\exp\left(-\frac{t^2 n}{2^k a_n}\right)$$

for every $t \leq 2\sqrt{2^k a_n}$

$$\begin{aligned}\mathbb{P}\big(|A_n(x)| \geq t\big) &\leq \mathbb{P}\left(\left|\frac{2}{n}\sum_{i=1}^n \frac{m(X_i)K_k(x,X_i) - \mathbb{E}(m(X)K_k(x,X))}{\mathbb{E}(K_k(x,X))}\right| \geq t\right) \\ &\quad + \mathbb{P}\left(\left|\frac{2}{n}\sum_{i=1}^n \frac{\epsilon_i K_k(x,X_i)}{\mathbb{E}(K_k(x,X))}\right| \geq t\right) \\ &\leq 2\exp\left(-C_1\frac{t^2 n}{2^k}\right) + 2\exp\left(-C_2\frac{t^2 n}{2^k a_n}\right) \leq C\exp\left(-C_3\frac{t^2 n}{2^k a_n}\right).\end{aligned}$$

where we have used a partial result from [43] [proposition 6.] and finally,

$$\mathbb{P}(C_t^c(x)) \leq 2\exp\left(-\tilde{C}\frac{t^2 n}{2^k a_n}\right)$$

which concludes the proof. □

To obtain the desired rate of convergence, we need an upper bound for

$$\mathbb{E}(\tilde{m}_{n,\infty}^{cc}(x) - m(x))^2 \leq c_3 \log n \exp\left(-\tilde{C}\frac{t^2 n}{2^k a_n}\right) + c_1\left(1 - \frac{1}{2d}\right)^{2k} + c_2 t^2$$

and we choose $2^k = n$ in the mean interpolation regime,

$$\mathbb{E}(\tilde{m}_{n,\infty}^{cc}(x) - m(x))^2 \leq c_3 \log n \exp\left(-\tilde{C}\frac{t^2}{a_n}\right) + c_1\left(1 - \frac{1}{2d}\right)^{2\log_2 n} + c_2 t^2$$

Finally, by minimizing the right hand of the equation in terms of t one has that, $t^2 = Ca_n \log \frac{\log n}{a_n}$ and of course,

$$c_3 \log n \exp\left(-\tilde{C}\frac{t^2}{a_n}\right) + c_1\left(1-\frac{1}{2d}\right)^{2\log_2 n} + c_2 t^2 \leq c_1\left(1-\frac{1}{2d}\right)^{2\log_2 n} + C_3 a_n + C_2 a_n \log\left(\frac{\log n}{a_n}\right)$$

and therefore, since $a_n = \frac{\log_2(\log_2 n)^d}{(\log_2 n)^{\frac{d-1}{2}}}$

$$\mathbb{E}(\tilde{m}^{cc}_{n,\infty}(x) - m(x))^2 \leq c_1\left(1-\frac{1}{2d}\right)^{2\log_2 n} + C_3 \frac{\log_2(\log_2 n)^d}{(\log_2 n)^{\frac{d-1}{2}}} \log\left(\frac{\log n (\log_2 n)^{\frac{d-1}{2}}}{\log_2(\log_2 n)^d}\right)$$

□

The above theorem states that the rate of convergence of the infinite centered KeRF and the infinite simplified directional KeRF is faster than the one in [4] in the interpolation regime even for dimension of the feature space $d \geq 2$. Moreover, interpolation in probability and consistency holds simultaneously but in a relatively slow convergence rate. By optimizing the depth parameter, one can obtain the rate of 3.0.2.

Lin and Jeon provided a theoretical lower bound for the rate of convergence of deep non-adaptive random forests [53] of $\frac{1}{(\log n)^{d-1}}$ and therefore we do not know yet if our rate of convergence is generally improvable. On the contrary, kernel estimators of the Nadaraya–Watson type ([60] , [89]) where the smoothing parameter is highly related with the tree depth parameter have been studied in [10] by Belkin and Rakhlin where it was proved that the rate of convergence is the minimax over the class of the Lipschitz functions.

4.1.2 Plots and experiments.

In this final section, we conduct numerical simulations and experiments to compare the performance of the finite-centered KeRF algorithm and the finite-simplified directional KeRF algorithm. The evaluation is carried out in terms of the L_2-error and the standard deviation of the error for various target functions Y. Specifically, we generated a two-dimensional feature space of size $n = 1500$ comprising uniformly distributed points.

The dataset was split into training and testing subsets, with 80% of the data utilized for training both algorithms, while the remaining 20% was reserved for evaluation purposes computing $(\sum_{X_i \in \text{ test set}} (\tilde{m}(X_i) - Y_i)^2)$. To evaluate the performance of the algorithms, we considered several target functions Y. For each function, we trained both the finite-centered KeRF and the simplified directional KeRF on the training subset and subsequently evaluated their predictions on the remaining testing subset.

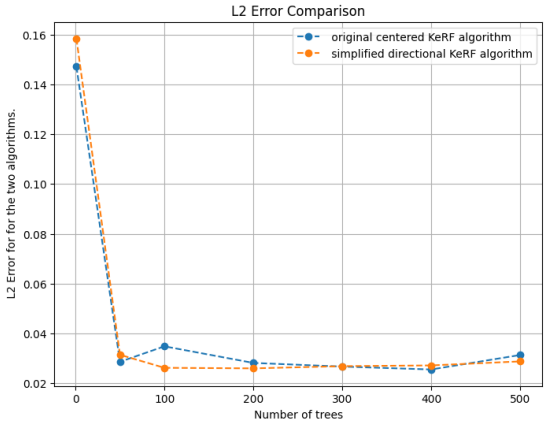
The following target functions with linear, polynomial, and exponential relationships within the feature space were considered to investigate the L_2-error with a fixed tree depth value of $k = \log_2 n$ and hence, every leaf has on average 1 data point. Moreover, the number of trees varies from $M = 1, 50, 100, 200, 300, 400, 500$ and therefore we can empirically confirm that the two algorithms coincide asymptotically.

1. $Y = X_1 + X_2 + \epsilon$, where ϵ is a random error following a normal distribution $\mathcal{N}(0, \frac{1}{2})$.

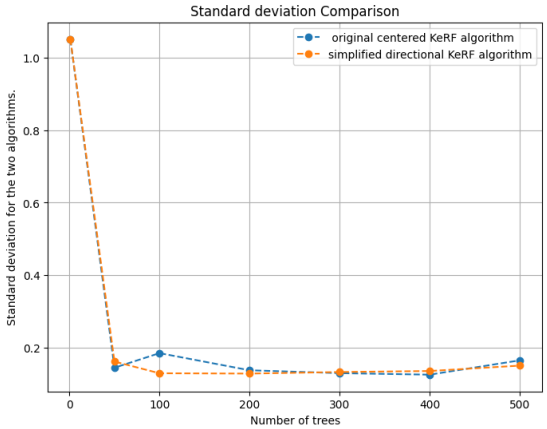
2. $Y = X_1^2 + X_2^2 + \epsilon$, where ϵ is a random error following a normal distribution $\mathcal{N}(0, \frac{1}{2})$.

3. $Y = 2X_1 + e^{-X_2^2}$.

All numerical simulations were conducted using the open-source Python software `https://www.python.org/,`, primarily utilizing the numpy library.

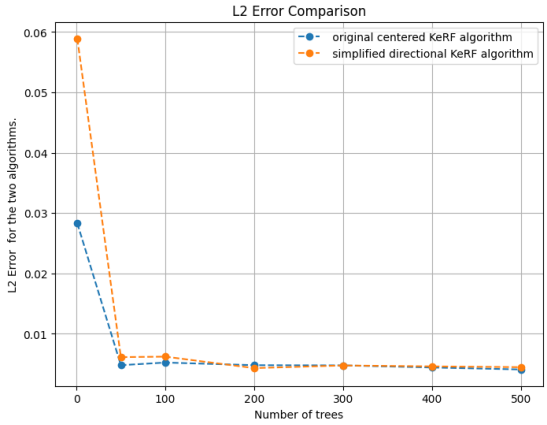


(a) $L-$error for the function $Y = 2X_1 + e^{-X_2^2}$.

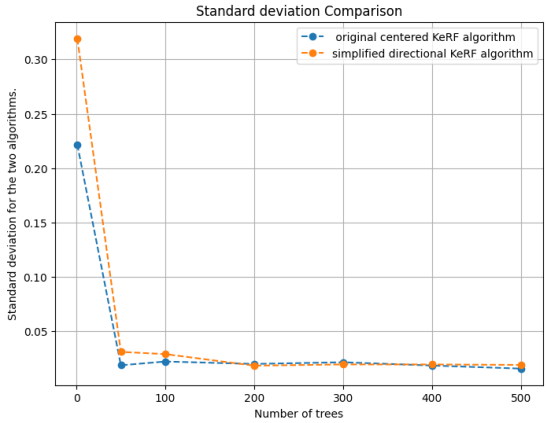


(b) Standard deviation for the $L-$error for the function $Y = 2X_1 + e^{-X_2^2}$.

Figure 4.5: Comparison of $L-$error and standard deviation.

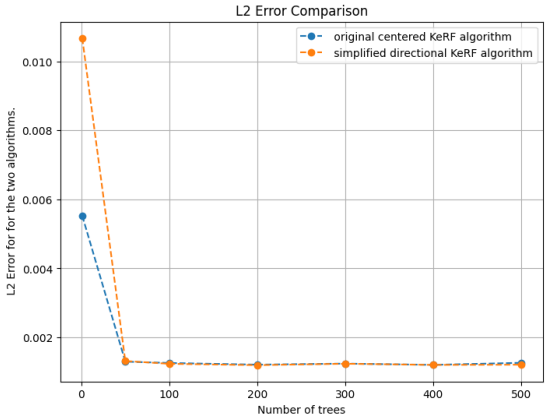


(a) $L-$error for the function $Y = X_1^2 + X_2^2 + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \frac{1}{2})$.

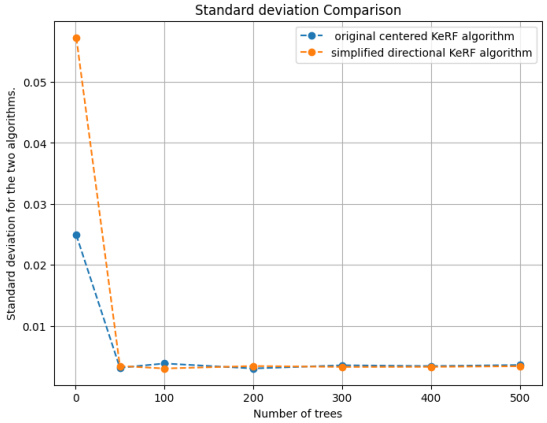


(b) Standard deviation for the $L-$error for the function $Y = X_1^2 + X_2^2 + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \frac{1}{2})$.

Figure 4.6: Comparison of $L-$error and standard deviation.



(a) $L-$error for the function $Y = X_1 + X_2 + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \frac{1}{2})$.



(b) Standard deviation for the $L-$error for the function $Y = X_1 + X_2 + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \frac{1}{2})$.

Figure 4.7: Comparison of $L-$error and standard deviation.

As one might expect, all the figures 4.5a 4.6a, 4.7a exhibit similar behavior. For small values of trees, the two algorithms demonstrate slightly different performances; however, as the number of trees increases, consistent with the theorem 12, both algorithms have the same performance in terms of the L_2-error, consistent with 12. Similarly, the same results hold for the standard deviation of the errors 4.5b 4.6b, 4.7b. Overall, as it is evident from all experiments 4.5, 4.6, and 4.7, after $M = 100$ trees the centered KeRF and the simplified directional KeRF essentially coincide.

Of course, more experiments can be conducted for different tree depths, leaving the interpolation world like the one that optimizes (so far) the speed of the convergence of the centered KeRF, larger data sets, and spaces of higher

dimensions.

Chapter 5

Conclusions

This PhD research thesis was conducted under the scholarship PON (Programmi Operativi Nazionali) DOT1303154-3 *Dottorati PON - Bando 2021 - Cycle 37 (XXXVII) - Action IV.5 - Doctorates on Green topics* supported by the Italian Ministry of Education and Merit, focusing on Innovation and Green topics. The National Operational Program (PON-green) aims to provide funds for research activities regarding green transition, ecosystem preservation, and reduction of climate change impacts. The first part of the thesis applies several supervised machine-learning algorithms to ecological sandy beaches in Emilia-Romagna. We derive useful information about the ecological balance of the sandy beaches by studying the benthos distribution and constructing regression and classification models. This is a joint work with Fabio Bozzeda from the University of Salento. The second part of the thesis is the study of a particular class of supervised machine learning algorithms. The purely random forests are simplifications of the original random forest algorithm of Breiman, that are constructed without the use of the data set. We study the corresponding kernel based algorithms and we provide rates of convergence under different model hypotheses. Finally we introduce a new tree tiling construction named the simplified directional tree and we investigate the performance of the aforementioned kernel based forest construction. In particular, we prove that the simplified directional kernel coincides with the kernel of the centered kernel tree. Finally we provide numerical experiments to empirically confirm our theoretical results. This is joint work with Nicola Arcozzi from University of Bologna.

Bibliography

[1] Agler, J., and McCarthy, J. E. Pick Interpolation and Hilbert Function Spaces.

[2] Allen Akselrud, C. I. Random forest regression models in ecology: Accounting for messy biological data and producing predictions with uncertainty. *Fisheries Research 280* (2024), 107161.

[3] Amit, Y., and Geman, D. Shape Quantization and Recognition with Randomized Trees. *Neural Computation 9*, 7 (1997), 1545–1588.

[4] Arnould, L., Boyer, C., and Scornet, E. Is interpolation benign for random forest regression? In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics* (25–27 Apr 2023), F. Ruiz, J. Dy, and J.-W. van de Meent, Eds., vol. 206 of *Proceedings of Machine Learning Research*, PMLR, pp. 5493–5548.

[5] Aronszajn, N. Theory of reproducing kernels. *Transactions of the American Mathematical Society 68*, 3 (1950), 337–404.

[6] Bartlett, P. L., Long, P. M., Lugosi, G., and Tsigler, A. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences of the United States of America 117*, 48 (Dec 2020), 30063–30070. Epub 2020 Apr 24.

[7] Bartlett, P. L., Montanari, A., and Rakhlin, A. Deep learning: a statistical viewpoint. *Acta Numerica 30* (2021), 87–201.

[8] Batool, T., Abbas, S., Alhwaiti, Y., Saleem, M., Ahmad, M., Asif, M., and Elmitwally, N. S. Intelligent model of ecosystem for smart cities using artificial neural networks. *Intelligent Automation & Soft Computing 30*, 2 (2021), 513–525.

[9] Belkin, M., Hsu, D. J., and Mitra, P. Overfitting or perfect fitting? risk bounds for classification and regression rules that interpolate. In *Advances in Neural Information Processing Systems* (2018), S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31, Curran Associates, Inc.

[10] Belkin, M., Rakhlin, A., and Tsybakov, A. B. Does data interpolation contradict statistical optimality? In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics* (16–18 Apr 2019), K. Chaudhuri and M. Sugiyama, Eds., vol. 89 of *Proceedings of Machine Learning Research*, PMLR, pp. 1611–1619.

[11] Biau, G. Analysis of a Random Forests Model. *Journal of Machine Learning Research 13*, 38 (2012), 1063–1095.

[12] Biau, G., and Devroye, L. On the layered nearest neighbour estimate, the bagged nearest neighbour estimate and the random forest method in regression and classification. *Journal of Multivariate Analysis 101*, 10 (2010), 2499–2518.

[13] Biau, G., Devroye, L., and Lugosi, G. Consistency of Random Forests and Other Averaging Classifiers. *Journal of Machine Learning Research 9*, 66 (2008), 2015–2033.

[14] Biau, G., and Scornet, E. A random forest guided tour. *TEST: An Official Journal of the Spanish Society of Statistics and Operations Research 25*, 2 (2016), 197–227.

[15] Borowiec, M. L., Dikow, R. B., Frandsen, P. B., McKeeken, A., Valentini, G., and White, A. E. Deep learning as a tool for ecology and evolution. *Methods in Ecology and Evolution 13*, 8 (2022), 1640–1660.

[16] Borowiec, M. L., Dikow, R. B., Frandsen, P. B., McKeeken, A., Valentini, G., and White, A. E. Deep learning as a tool for ecology and evolution. *Methods in Ecology and Evolution 13*, 8 (2022), 1640–1660.

[17] Boucheron, S., Lugosi, G., and Massart, P. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, 02 2013.

[18] Boulesteix, A., Janitza, S., Kruppa, J., and König, I. R. Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 2* (2012).

[19] Breiman, L. Some infinity theory for predictor ensembles. Tech. Rep. Technical Report 579, Statistics Dept. UCB, 2000.

[20] Breiman, L. Random Forests. *Machine Learning 45* (2001), 5–32.

[21] Breiman, L. Consistency for a simple model of random forests. Technical Report 670, Statistics Department, University of California, Berkeley, Berkeley, California, September 2004.

[22] Breiman, L., Friedman, J., Olshen, R., and Stone, C. J. *Classification and Regression Trees*, 1st ed. Chapman and Hall/CRC, New York, 1984. First eBook edition published October 19, 2017.

[23] Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. *Classification and Regression Trees*, vol. 40. 1984.

[24] Co-Reyes, J. D., Sanjeev, S., Berseth, G., Gupta, A., and Levine, S. Ecological reinforcement learning, 2020.

[25] Defeo, O., and McLachlan, A. Climate-change impacts on sandy-beach biota: crossing a line in the sand. *Biological Reviews 88*, 4 (2013), 768–780.

[26] Defeo, O., and Mclachlan, A. J. Patterns, processes and regulatory mechanisms in sandy beach macrofauna: a multi-scale analysis. *Marine Ecology Progress Series 295* (2005), 1–20.

[27] Denil, M., Matheson, D., and Freitas, N. Consistency of Online Random Forests. In *Proceedings of the 30th International Conference on Machine Learning* (Atlanta, Georgia, USA, 17–19 Jun 2013), S. Dasgupta and D. McAllester, Eds., vol. 28 of *Proceedings of Machine Learning Research*, PMLR, pp. 1256–1264.

[28] Devroye, L., Györfi, L., and Krzyżak, A. The hilbert kernel regression estimate. *Journal of Multivariate Analysis 65*, 2 (1998), 209–227.

[29] Dietterich, T. G. Ensemble methods in machine learning. In *Multiple Classifier Systems* (Berlin, Heidelberg, 2000), Springer Berlin Heidelberg, pp. 1–15.

[30] Dittman, D. J., Khoshgoftaar, T. M., and Napolitano, A. The effect of data sampling when using random forest on imbalanced bioinformatics data. In *2015 IEEE International Conference on Information Reuse and Integration (IRI)* (Los Alamitos, CA, USA, aug 2015), IEEE Computer Society, pp. 457–463.

[31] Elith*, J., H. Graham*, C., P. Anderson, R., Dudík, M., Ferrier, S., Guisan, A., J. Hijmans, R., Huettmann, F., R. Leathwick, J., Lehmann, A., Li, J., G. Lohmann, L., A. Loiselle, B., Manion, G., Moritz, C., Nakamura, M., Nakazawa, Y., McC. M. Overton, J., Townsend Peterson, A., J. Phillips, S., Richardson, K., Scachetti-Pereira, R., E. Schapire, R., Soberón, J., Williams, S., S. Wisz, M., and E. Zimmermann, N. Novel methods improve prediction of species' distributions from occurrence data. *Ecography 29*, 2 (2006), 129–151.

[32] Fautt, C., Couradeau, E., and Hockett, K. L. Naïve bayes classifiers and accompanying dataset for pseudomonas syringae isolate characterization. *Scientific Data 11*, 1 (Feb 2024), 178.

[33] Friedman, J. H. Data mining and statistics: what's the connection? In *Keynote Address, 29th Symposium on the Interface: Computing Science and Statistics* (1997).

[34] Geurts, P., Ernst, D., and Wehenkel, L. Extremely randomized trees. *Machine Learning 63* (2006), 3–42.

[35] Gonzalez, T., Diaz-Herrera, J., and Tucker, A. *Computing Handbook, Third Edition: Computer Science and Software Engineering*, 3rd ed. Chapman & Hall/CRC, 2014.

[36] Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[37] Gries, S. T. On classification trees and random forests in corpus linguistics: Some words of caution and suggestions for improvement. *Corpus Linguistics and Linguistic Theory 16* (2019), 617 – 647.

[38] Harris, L. R., and Defeo, O. Sandy shore ecosystem services, ecological infrastructure, and bundles: New insights and perspectives. *Ecosystem Services 57* (2022), 101477.

[39] Ho, T. K. Random decision forests. In *Proceedings of 3rd International Conference on Document Analysis and Recognition* (1995), vol. 1, pp. 278–282 vol.1.

[40] Howard, J., and Bowles, M. The two most important algorithms in predictive modeling today. In *Strata Conference: Santa Clara* (2012).

[41] Hsieh, W. W. *Machine Learning Methods in the Environmental Sciences: Neural Networks and Kernels*. Cambridge University Press, 2009.

[42] Huntingford, C., Jeffers, E. S., Bonsall, M. B., Christensen, H. M., Lees, T., and Yang, H. Machine learning and artificial intelligence to aid climate change research and preparedness. *Environmental Research Letters 14*, 12 (Dec. 2019), 124007.

[43] Isidoros, I., and Arcozzi, N. Improved convergence rates for some kernel random forest algorithms. *Mathematics in Engineering 6*, 2 (2024), 305–338.

[44] Isidoros, I., and Arcozzi, N. A simplified directional kerf algorithm, 2024.

[45] Itzkin, M., Moore, L., Ruggiero, P., Hacker, S., and Biel, R. The influence of dune aspect ratio, beach width and storm characteristics on dune erosion for managed and unmanaged beaches, 11 2020.

[46] Jayachandran, P. R., Bijoy Nandan, S., M., J., Joseph, P., and N.K, V. *Benthic organisms as an ecological tool for monitoring coastal and marine ecosystem health*. 04 2022, pp. 337–363.

[47] Kampichler, C., Wieland, R., Calmé, S., Weissenberger, H., and Arriaga-Weiss, S. Classification in conservation biology: A comparison of five machine-learning methods. *Ecological Informatics 5*, 6 (Jan. 2010), 441–450.

[48] Katkovnik, V. Nonparametric estimation of the time-varying frequency and amplitude. *Statistics and Probability Letters 35*, 4 (1997), 307–315.

[49] Klusowski, J. M. Sharp Analysis of a Simple Model for Random Forests. In *International Conference on Artificial Intelligence and Statistics* (2018).

[50] Lancaster, P., and Salkauskas, K. Surfaces generated by moving least squares methods. *Mathematics of Computation 37* (1981), 141–158.

[51] Liang, T., Rakhlin, A., and Zhai, X. On the multiple descent of minimum-norm interpolants and restricted lower isometry of kernels. In *Annual Conference Computational Learning Theory* (2019).

[52] Liaw, A., and Wiener, M. Classification and Regression by randomForest. *R News 2*, 3 (2002), 18–22.

[53] Lin, Y., and Jeon, Y. Random Forests and Adaptive Nearest Neighbors. *Journal of the American Statistical Association 101*, 474 (2006), 578–590.

[54] Mahalanobis, P. Reprint of: Mahalanobis, p.c. (1936) "on the generalised distance in statistics". *Sankhya A 80*, 1 (2018), 1–7.

[55] McLachlan, A., and Brown, A. C. *The Ecology of Sandy Shores*. Academic Press, San Diego, CA, USA, 2006. Copyright © 2006 Elsevier Inc. All rights reserved.

[56] McLachlan, A., and Dorvlo, A. Global Patterns in Sandy Beach Macrobenthic Communities. *Journal of Coastal Research 2005*, 214 (2005), 674 – 687.

[57] Mentch, L., and Hooker, G. Quantifying uncertainty in random forests via confidence intervals and hypothesis tests. *Journal of Machine Learning Research 17*, 26 (2016), 1–41.

[58] Mosaffaei, Z., Jahani, A., Chahouki, M. A. Z., Goshtasb, H., Etemad, V., and Saffariha, M. Soil texture and plant degradation predictive model (stpdpm) in national parks using artificial neural network (ann). *Modeling Earth Systems and Environment 6*, 2 (2020), 715–729.

[59] Müller, A., and Guido, S. *Introduction to Machine Learning with Python: A Guide for Data Scientists*. O'Reilly Media, Incorporated, 2018.

[60] Nadaraya, E. A. On estimating regression. *Theory of Probability & Its Applications 9*, 1 (1964), 141–142.

[61] Olden, J. D., Lawler, J. J., and Poff, N. L. Machine learning methods without tears: a primer for ecologists. *The Quarterly Review of Biology 83*, 2 (Jun 2008), 171–193.

[62] on Climate Change, I. P. *Climate Change 2021 – The Physical Science Basis: Working Group I Contribution to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*. Cambridge University Press, 2023.

[63] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. Scikit-learn: Machine learning in python. *Journal of machine learning research 12*, Oct (2011), 2825–2830.

[64] Peters, D. P. C., Havstad, K., Cushing, J. B., Tweedie, C. E., Fuentes, O., and Villanueva-Rosales, N. Harnessing the power of big data: infusing the scientific method with machine learning to transform ecology. *Ecosphere 5* (2014), 1–15.

[65] Pichler, M., Boreux, V., Klein, A.-M., Schleuning, M., and Hartig, F. Machine learning algorithms to infer trait-matching and predict species interactions in ecological networks. *Methods in Ecology and Evolution 11*, 2 (2020), 281–293.

[66] Pichler, M., and Hartig, F. Machine learning and deep learning—a review for ecologists. *Methods in Ecology and Evolution 14*, 4 (2023), 994–1016.

[67] Rammer, W., and Seidl, R. Harnessing deep learning in ecology: An example predicting bark beetle outbreaks. *Frontiers in Plant Science 10* (2019).

[68] Recknagel, F. Applications of machine learning to ecological modelling. *Ecological Modelling 146*, 1 (Jan. 2001), 303–310.

[69] Rodella, I., Simeoni, U., and Corbau, C. La percezione dell'offerta turistico-balneare della riviera ferrarese (emilia-romagna). *Studi Costieri 25* (07 2017), 35–44.

[70] Rudin, W. *Fourier analysis on groups*. Courier Dover Publications, 2017.

[71] Satta, A., Markovich, M., Skaricic, Z., and Trumbic, I. Sustainable tourism development in croatian coastal area - pilot project baska voda, 10 2008.

[72] Scornet, E. On the asymptotics of random forests. *Journal of Multivariate Analysis 146* (2016), 72–83. Special Issue on Statistical Models and Methods for High or Infinite Dimensional Spaces.

[73] Scornet, E. Random Forests and Kernel Methods. *IEEE Transactions on Information Theory 62*, 3 (2016), 1485–1500.

[74] Scornet, E., Biau, G., and Vert, J.-P. CONSISTENCY OF RANDOM FORESTS. *The Annals of Statistics 43*, 4 (2015), 1716–1741.

[75] Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., and Blake, A. Real-time human pose recognition in parts from single depth images. In *CVPR 2011* (2011), pp. 1297–1304.

[76] Silverman, B. W., and Jones, M. C. E. fix and j.l. hodges (1951): An important contribution to nonparametric discriminant analysis and density estimation: Commentary on fix and hodges (1951). *International Statistical Review 57* (1989), 233.

[77] Simon, S. M., Glaum, P., and Valdovinos, F. S. Interpreting random forest analysis of ecological models to move from prediction to explanation. *Scientific Reports 13*, 1 (Mar 2023), 3881.

[78] Soares, A. G. Sandy beach morphodynamics and macrobenthic communities in temperate, subtropical and tropical regions : a macroecological approach.

[79] Spake, R., Bowler, D. E., Callaghan, C. T., Blowes, S. A., Doncaster, C. P., Antão, L. H., Nakagawa, S., McElreath, R., and Chase, J. M. Understanding 'it depends' in ecology: a guide to hypothesising, visualising and interpreting statistical interactions. *Biological Reviews of the Cambridge Philosophical Society 98*, 4 (Aug 2023), 983–1002.

[80] Sun, G., Sun, Y., and Luo, F. Generalized minkowski distance-based local mean k-nearest neighbor classifier. In *2023 IEEE International Conference on Image Processing and Computer Applications (ICIPCA)* (2023), pp. 1710–1716.

[81] Sutton, R. S., and Barto, A. G. *Reinforcement Learning: An Introduction*, second ed. The MIT Press, 2018.

[82] T., T. A., Hooker, G., Ellner, S. P., and Adler, P. B. A practical guide to selecting models for exploration, inference, and prediction in ecology. *Ecology 102*, 6 (2021), e03336.

[83] Tang, C., Garreau, D., and von Luxburg, U. When do random forests fail? In *Advances in Neural Information Processing Systems* (2018), S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31, Curran Associates, Inc.

[84] Thuiller, W., Lafourcade, B., Engler, R., and Araújo, M. B. BIOMOD - a platform for ensemble forecasting of species distributions. *Ecography 32*, 3 (June 2009), 369–373.

[85] Torresan, S., Critto, A., Rizzi, J., and Marcomini, A. Assessment of coastal vulnerability to climate change hazards at the regional scale: the case study of the north adriatic sea. *Natural Hazards and Earth System Sciences 12*, 7 (2012), 2347–2368.

[86] Tsigler, A., and Bartlett, P. L. Benign overfitting in ridge regression. *J. Mach. Learn. Res. 24* (2020), 123:1–123:76.

[87] Wager, S., and and, S. A. Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association 113*, 523 (2018), 1228–1242.

[88] Wang, Y., and Scott, C. D. Consistent interpolating ensembles via the manifold-hilbert kernel. In *Proceedings of the 36th International Conference on Neural Information Processing Systems* (Red Hook, NY, USA, 2022), NIPS '22, Curran Associates Inc.

[89] Watson, G. S. Smooth regression analysis. *Sankhyā: The Indian Journal of Statistics, Series A (1961-2002) 26*, 4 (1964), 359–372.

[90] Wyner, A. J., Olson, M., Bleich, J., and Mease, D. Explaining the success of adaboost and random forests as interpolating classifiers. *ArXiv abs/1504.07676* (2015).

[91] Yang, Y., and Barron, A. Information-theoretic determination of minimax rates of convergence. *The Annals of Statistics 27*, 5 (1999), 1564 – 1599.

[92] Yoon, J. Forecasting of Real GDP Growth Using Machine Learning Models: Gradient Boosting and Random Forest Approach. *Computational Economics 57*, 1 (January 2021), 247–265.

[93] Yu, Q., Ji, W., Prihodko, L., Ross, C. W., Anchang, J. Y., and Hanan, N. P. Study becomes insight: Ecological learning from machine learning. *Methods in Ecology and Evolution 12*, 11 (Nov 2021), 2117–2128.

[94] Zamri, N., Pairan, M. A., Azman, W. N. A. W., Abas, S. S., Abdullah, L., Naim, S., Tarmudi, Z., and Gao, M. River quality classification using different distances in k-nearest neighbors algorithm. *Procedia Computer Science 204* (2022), 180–186. International Conference on Industry Sciences and Computer Science Innovation.

[95] Zhang, G., Wang, M., and Liu, K. Deep neural networks for global wildfire susceptibility modelling. *Ecological Indicators 127* (2021), 107735.