Università di Bologna

Dipartimento di Scienze Statistiche "Paolo Fortunati"
Dottorato di Ricerca in
Metodologia Statistica per la Ricerca Scientifica

Automated Local Linear Embedding with an application to microarray data

Elisa Grilli

Relatore:
Chiar.mo Prof. Angela Montanari

Coordinatore di Dottorato:
Chiar.mo Prof. Daniela Cocchi

XIX ciclo
Anno Accademico 2005/2006

UNIVERSITÀ DEGLI STUDI DI BOLOGNA
DIPARTIMENTO DI
SCIENZE STATISTICHE "PAOLO FORTUNATI"

DOTTORATO DI RICERCA IN
METODOLOGIA STATISTICA PER LA RICERCA
SCIENTIFICA

Data: 15 Marzo 2007

Research Supervisor:	Angela Montanari
Coordinator:	Daniela Cocchi
External Examiner:	Ernst Wit
Scientific Committee:	Silvano Bordignon Carla Rampichini Maurizio Vichi

UNIVERSITÀ DEGLI STUDI DI BOLOGNA

Date: **15 Marzo 2007**

Author: **Elisa Grilli**

Title: **Automated Local Linear Embedding with an application to microarray data**

Department: **Scienze Statistiche "Paolo Fortunati"**

Degree: **Ph.D.** Convocation: **Marzo** Year: **2007**

Signature of Author

Acknowledgements

The essential work reported in this thesis was carried out during my visiting periods at the University of Glasgow in the Autumn 2005 and at the Lancaster University in May 2006. I would like to express my gratitude to Ernst Wit for being my supervisor and for his precious support and his helpful advices during my work.

I would like to give my sincere and warm thanks to Angela Montanari, Cinzia Viroli and Marilena Pillati for their supervision and guidance of my PhD thesis and for their helpful criticism on this work.

It has been a pleasure to exchange ideas with the other PhD student at the University of Glasgow and Bologna.

Finally, I would like to give many thanks to my husband, my parents and my friends for their patient and for supporting me during the three years of my PhD period.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The problem of dimensionality reduction arises in many fields of information processing, including machine learning, scientific visualization, pattern recognition and neural computation. Dimensionality reduction is an important operation to deal with multi-dimensional data. Its goal is to obtain a compact representation of the original high-dimensional data while eliminating noisy factors which dramatically hide meaningful relationship and correlations. Since a part of information is lost during the dimensionality reduction, it is important for the resulting low-dimensional data to preserve the original structure and relationship of the high-dimensional data. It is highly desirable that the low-dimensional projected space preserves the local geometry of the original space, that is, close points in the high dimensional space must remain close in the embedded space.

When the embedded structures are linear subspaces, linear techniques such as Principal Component Analysis (PCA) and MultiDimensional Scaling (MDS) can be used. Both PCA and MDS are eigenvector methods designed to model linear structures in high-dimensional data. In PCA, second order statistics (variance and covariance) of the data are considered, by searching for directions in which the variances are maximised. In classical (or metric) MDS, low-dimensional embedding that best preserves pairwise distances between data points is computed. If these distances correspond to euclidean distances, the result of metric MDS is equivalent to PCA. Both methods are simple to implement, and their optimizations do not involve local minima. These virtues account for the widespread use of PCA and MDS, despite their inherent limitations as linear methods.

When the observed data can not be properly modelled by linear structures, linear dimensionality reduction performed by PCA and MDS fail to preserve the global geometry of the high-dimensional space. In other words, they often map distant points in the original space into close points in the embedded space.

Thus, in case of nonlinear manifolds one needs to seek some methods reducing the dimensionality of the data in a nonlinear manner. Several methods are suitable for this purpose. Among them are Sammon's Mapping (SM) [37, 26] which is inspired to Multidimensional Scaling techniques [11], Curvilinear Component Analysis (CCA) and Curvilinear Distance Analysis (CDA) [15, 35], Generative Topographic Mapping (GTM) [8], Self-Organizing Map (SOM) [29] and Visualization Induced Self-Organizing Map (ViSOM) [49].

All these methods, though are workable, have too many parameters to be set by the user and in addition, some of these, were tested on rather artificial than real world data.

Two more recents unsupervised learning algorithms that allow to perform dimensionality reduction based on the idea that global geometry of the high-dimensional data can be retained in a collection of local geometries when projecting the data to a low-dimensional space, are the "isomap" (J. B. Tenenbaum *et al.*, 2000) and the "local linear embedding" (LLE) (S.T. Roweis and L.K. Saul, 2000).

The "isomap" [46] uses easily measured local metric information to learn the underlying global geometry of a data set. The approach builds on classical MDS but seeks to preserve the intrinsic geometry of the data as captured in the geodesic manifold distances between all pairs of data points.

The other procedure has been proposed in the same year by Roweis and Saul. The goal of their "local linear embedding" [43, 45] is to recover the nonlinear structure of high-dimensional data. The idea behind this algorithm is that nearby points in the high-dimensional space remain nearby and similarly co-located with respect to one another in the low-dimensional embedding. Starting from this intuitive consideration, authors proceed by approximating each data point by a weighted linear combination of its neighbours. The algorithm derives its name from the nature of these local and linear reconstructions: it is *local*, in the sense that only neighbours contribute to each reconstruction, and *linear*, in the sense that reconstruction are confined to linear subspaces. The LLE algorithm is an attractive method for the following main reasons: 1) a good preservation of the local geometry of the high-dimensional space in the low-dimensional space, 2) only two parameters to be set, 3) a single global coordinate system of the embedded space, 4) avoids the problems with local minima that plague many other iterative techniques. Since LLE method has recently developed, is possible to seek still few literature in [43, 45, 1, 7, 20, 34, 30, 31, 33, 32, 41].

This thesis focuses on some extensions of Local Linear Embedding algorithm and applications to microarray data and simulated data.

The first purpose is to develop a procedure for the automatic selection of the two free parameters of the model. The first natural question is how choosing the optimal number of k nearest neighbours since this parameter noticeably influences the final data projection. If one chooses a large k, it produces smoothing or eliminates the scale structure in the data, as well as if one chooses a small k, it can falsely divide the continuous data manifold into disjointed components. In order to overcome this issue some criteria for the automatic selection of the optimal number of k nearest neighbours are proposed.

The dimensionality d of the projection space is the second free parameter of the algorithm. Initially the LLE method emerged with the purpose of visualizing the high-dimensional datasets. Since human observer can not visually perceive a high-dimensional representation of the data, its dimensionality was reduced to one, two or three and so the dimensionality of the projection space was automatically fixed. In general, when the goal of the dimensionality reduction technique is not confined to data visualization, the dimensionality of the embedded space can not be a priori fixed. Thus, some procedures for the automatic selection of the parameter d are proposed.

The second purpose of the thesis is to test the proposed criteria for the automatic determination of the model parameters on simulated datasets. The data was created by randomly sampling from two uniform independent variables and by generating nonlinear combinations of them. Several simulations have been conducted to test the proposed procedures by providing similar results. Thus, the results on just one simulation study are presented.

The finally purpose is to apply the procedures on datasets arisen from microarray study. DNA Microarray technology leads to a new class of biological experiments where data acquisition of gene activity is possible on a large scale. A typical microarray data matrix contains the expression levels of thousands of genes across different experimental samples. In this context, where the number of D genes is much greater than the experimental conditions n, the standard techniques are difficult to employ and dimensionality reduction methods are required in order to obtain a compact representation of the high-dimensional data in fewer dimensions.

The thesis is structured as follows.

The second chapter deals with the original Local Linear Embedding method. After a brief introduction to the main idea behind the method, we present the LLE algorithm by reproducing some classical examples in which the powerful of LLE against the PCA technique is compared. The major extensions of LLE method are presented. In particular we propose a review on some techniques for the automatic determination of the model parameters.

The third chapter deals with our proposal for the automatic selection of the optimal number of k nearest neighbours. After an introduction about the reference context in which we work, the three different criteria for the automatic determination of the model free parameter are proposed and their formulation in the work context derived.

The fourth chapter deals with our proposal for the automatic determination of d embedded coordinates able to represent the high-dimensional original data space. We propose three criteria for the determination of the model free parameter afterward the reference context has been illustrated.

The fifth chapter deals with the simulation study. After a brief description on the data generation we present the results for the choice of the optimal number of k neighbours and we propose a validation measure based on the Procustes Analysis. This technique compares the shape differences between two configurations after that, location, scale and rotational effect are filtered out by minimizing distance. Once computed the optimal number of k neighbours, we proceed by presenting the results over the three proposed criteria for the automatic selection of d embedded coordinates.

The next chapter concerns with an introduction to microarray data analysis. We review a description of the microarray technology process, by illustrating a typical measure to detect differentially expressed genes in order to obtain the final microarray data matrix. Finally, the analysis of gene expression data is treated.

The seventh chapter deals with the results obtained by the application of the methods proposed on three public available microarray data sets: the mammary data set of Wit and McClure (2004), the lymphoma data set of Alizadeth *et al.* (2000), the leukemia data set of Golub *et al.* (1999).

In the last chapter the conclusions of the thesis are presented.

Chapter 2

Locally Linear Embedding

2.1 Introduction

Dimensionality reduction can be done either by feature selection or by feature extraction. Feature selection methods choose the most informative features among those given, therefore low-dimensional data representation possesses a physical meaning. Feature extraction methods, indeed, obtain informative projection by applying certain operation to the original features. The advantage of feature extraction over feature selection methods is that, given the same dimensionality of reduced data representation, the transformed features might provide better results in further data analysis.

There are two possibilities to reduce dimensionality of data: supervised, when data labels are provided, and unsupervised when no data labels are given. In most cases in practice, no prior knowledge about data is available, since it is very expensive to assign labels to the data samples. Therefore, nowadays, unsupervised methods discovering the hidden structure of the data are of prime interest.

Here, we describe the Locally Linear Embedding algorithm (S.T. Roweis and L.K. Saul, 2000) [43]. This is an unsupervised non linear feature extraction technique that analyses high-dimensional data sets and reduces their dimensionalities while preserving local topology, that is, close points in the high dimensional space remain close in the low-dimensional space. LLE obtains a low dimensional data representation by assuming that even if the high-dimensional data forms a non linear manifold it still can be considered locally linear if each data point and its neighbours lie on or close to a locally linear patch of the manifold. Because of the assumption that local patches are linear, then each of them can be approximated by a linear hyperplane and each data point can be represented by a weighted linear combination of its k nearest neighbours. The coefficients of this approximation characterize local geometries in the high-dimensional space and they are used to find low-dimensional embeddings preserving the geometries in the low-dimensional space.

Sections 2.2 and 2.3 introduce LLE by stating initial conditions to be satisfied and by explaining the main idea of this method.

Section 2.4 describes in details the LLE algorithm and some classical non linear dimensionality reduction examples in which the power of LLE method against linear technique as PCA is shown. In this section particular attention is given to the regularization problem which arises when the number of k nearest neighbours, needed to reconstruct each data point, outnumber the input dimensionality of the data.

In section 2.5 some extensions of the LLE technique are presented. They try to overcome the main drawbacks that affect the original LLE algorithm.

2.2 Initial Conditions

LLE produces a low-dimensional data representation by assuming that even if the high dimensional data forms a non linear manifold it still can be considered locally linear if each data point and its neighbours lie on or close to a locally linear patch of the manifold. The simplest example of such cases, proposed in [31], is the Earth. Its global manifold is a sphere which is described by a non linear equation $x^2+y^2+z^2=r^2$ where (x, y, z) are the coordinates of three dimensional space and r is the radius of the sphere, while locally it can be considered as a linear two dimensional plane $ax + by = 0$ where (x, y) are the coordinates and a, b are the coefficients. Hence, the sphere can be locally approximated by linear planes instead of convex ones. Unfortunately, LLE cannot deal with closed data manifolds such as sphere. In this case, one should manually cut the manifold before applying the LLE algorithm, for example by deleting a pole from the sphere. In Fig. 2.1 are proposed two examples of "good" manifolds satisfying to this requirement. In other words, we assume that n points embedded in a D-dimensional space are sampled from some d-dimensional manifold ($d \ll D$) so that the sampled points represent the manifold sufficiently well. We want to emphasize that the data come from *one* manifold. The internal structure of this manifold must be flat in a sense that it cannot be anything like a sphere. In this case, when unfolded, it becomes flat.

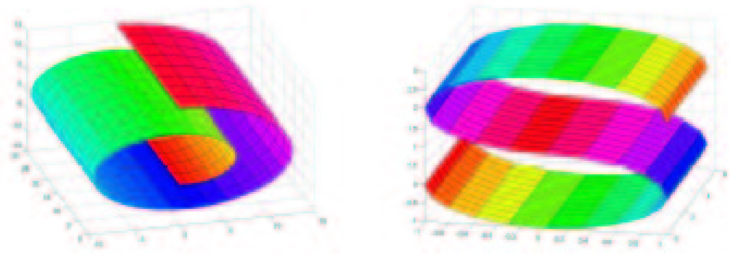

Figure 2.1. *Flat 2-D manifolds embedded in 3-D space .*

2.3 Main Idea

LLE takes a set of high-dimensional data and maps them into a low-dimensional euclidean space preserving *local* structure of the data. The key assumption related to LLE is that even if the manifold embedded in a high-dimensional space is nonlinear when considered as a whole, it still can be assumed *locally linear* if each data point and its neighbours lie on or close to a locally linear patch of the manifold. That is, the manifold can be covered with a set of locally linear (possibly overlapping) patches which, when analyzed together, can yield information about global geometry of the manifold. Because of the assumption that local patches are linear, each of them can be approximated by a linear hyperplane so that each data point can be represented by a weighted linear combination of its nearest neighbours. Coefficients of this approximation characterize local geometries in a high-dimensional space and they are then used to find low-dimensional embeddings preserving the geometries in a low-dimensional space. The main point in replacing the nonlinear manifold with the linear hyperplanes is that this operation does not bring significant error, because, when locally analyzed, the curvature of the manifold is not large so that the manifold can be considered to be locally flat [31].

The implementation of the Locally Linear Embedding, proposed by Roweis and Saul, consists of three steps. The first step of LLE consists in identifying the neighborhood of each data point. In the simplest formulation of the algorithm, a fixed number of nearest neighbours k per data point can be chosen as measured by euclidean distance. Other criteria, however can also be used to choose the neighbours. For example, by choosing all points within a ball of fixed radius.

The second step of LLE is to reconstruct each data point from its nearest neighbours by computing a weight matrix that minimize the reconstruction error, which adds up the squared distances between all data points and their reconstructions.

The final step of LLE is to compute a low-dimensional embedding y_i based on the reconstruction weights of the high-dimensional inputs x_i. The goal of the method is to find low-dimensional outputs that are reconstructed by the same weights as the high-dimensional inputs. The only information used to construct an embedding is that information captured by the weights.

The attractive features of the algorithm are that it avoids local minima problems, it has only few tuning parameters and the local geometry of high-dimensional data is preserved in the embedded space.

2.4 The Algorithm

Suppose the data consist of n real-valued vectors x_i, each of dimensionality D, assembled in a data matrix $\mathbf{X}$ of size $D \times n$, sampled from some smooth underlying manifold. We assume each data point and its neighbours to lie on or close to some smooth linear or nonlinear manifold. We characterize the local geometry of these patches by linear coefficients that reconstruct each data point from its neighbours. The results of LLE are typically stable over a range of neighborhood size. The size of that range depends on various features of the data, such as the sampling density and the manifold geometry. In the section 2.5.6 is presented a method to determine the optimal number of nearest neighbours proposed in [32].

Once number of neighbours are chosen, we use the neighbours of a data point i to reconstruct it in a linear fashion by means of a set of weights $w^{(i)}$. Considering a $(D \times 1)$ data vector x_i with its associated k nearest neighbours matrix $X^{(i)} = [x_{(i,1)}, x_{(i,2)}, \ldots, x_{(i,j)}, \ldots, x_{(i,k)}]$, where $x_{(i,j)}$ is the *j-th* neighbours of x_i, we can compute the k-dimensional weight vector $w^{(i)}$. Each data point x_i is reconstructed by its locally linear reconstruction $\hat{x_i} = X^{(i)} w^{(i)}$, where the weights $w^{(i)}$ reconstruct the i-th data point. Reconstruction errors are then measured by the cost function:

$$\mathrm{SS}_1(w,k) = \sum_{i=1}^{n} \left| x_i - \sum_{j=1}^{k} X_j^{(i)} w_j^{(i)} \right|^2 , \tag{2.1}$$

To compute the weights, we minimize the cost function in (2.1) subject to two constraint: a *sparseness* constraint and an *invariance* constraint. The *sparseness* constraint is that each data point is reconstructed only from its neighbours, enforcing $w_j^{(i)} = 0$ if $x_{(i,j)}$ does not belong to this set; the *invariance* constraint is that the rows of the weight matrix sum to one $\sum_{j=1}^{k} w_j^{(i)} = 1$. The optimal weights $w^{(i)}$ subject to these constraints are found by minimizing the square problem in (2.1). Computing the reconstruction weights is typically the least expensive step of the LLE algorithm. The weight matrix can be stored as a sparse matrix with nk non-zero elements.

Note that, the constrained weights that minimize these reconstruction errors obey important symmetries: for any particular data point, they are invariant to rotations, rescalings, and translations of that data point and its neighbours. The invariance to rotations and rescalings follows immediately from the form of (2.1); the invariance to translations is enforced by the sum to one constraint on the rows of the weight matrix. A consequence of these simmetries is that the reconstruction weights characterize geometric properties that do not depend on a particular frame of reference. Suppose the data lie on or near a manifold of dimensionality $d \ll D$. We therefore expect that the local geometry in the original data space is equally valid for local patches on the manifold: in particular, the same weights $w^{(i)}$ that reconstruct the i-th data point in D dimensions should also be able to reconstruct its embedded manifold coordinates in d dimensions [43].

LLE constructs a neighborhood preserving mapping based on this idea. In the unusual case where the neighbours outnumber the input dimensionality ($k > D$), (indicating that the original data is itself low dimensional), each data point can be reconstructed perfectly from its neighbours, and the local reconstruction weights are no longer uniquely defined. In this case some further regularization must be added to break the degeneracy. A simple regularizer that authors consider is to penalize the sum of squares of the weights which favors weights that are uniformly distributed in magnitude [43, 44, 45].

In the final step of the algorithm, since the goal is to preserve a local linear structure of a high-dimensional space as accurately as possible in a low dimensional space, the weights $w^{(i)}$ are kept fixed and the following cost function is minimized:

$$\mathrm{SS}_2(y, d) = \sum_{i=1}^{n} \left| y_i - \sum_{j=1}^{k} Y_j^{(i)} w_j^{(i)} \right|^2 , \tag{2.2}$$

where $Y^{(i)}$ is the $d \times k$ matrix of the k nearest neighbours of y_i.

Note that the embedding is computed directly from the weight matrix. The original inputs are not involved in this step of the algorithm. Thus the embedding is determined entirely by the geometric information encoded by the weights.

The embedding cost in (2.2) defines a quadratic form in the vectors. This cost function is minimized when the outputs y_i are reconstructed (or nearly reconstructed) by the same weighted linear combinations of neighbours as computed for the inputs.

To make the problem well-posed, the following constraints are imposed:

$$\sum_{i=1}^{n} y_i = \mathbf{0}_{d\times\mathbf{1}} \tag{2.3}$$

$$\frac{1}{n}\sum_{i=1}^{n} y_i y_i^t = \mathbf{I}_{d\times d} \tag{2.4}$$

The equation (2.3) removes the translation degree of freedom by requiring the outputs to be centered at the origin. The embedded coordinates have to have normalized unit covariance as in (2.4) in order to remove the rotational degree of freedom and to fix the scale. As a result, a unique solution is obtained. To find the embedding coordinates minimizing (2.2) under the constraints given in (2.3) and (2.4), a new matrix is constructed, based on the weight matrix $w^{(i)}$:

$$M = (I_{n\times n} - w^{(i)})^t (I_{n\times n} - w^{(i)}) \tag{2.5}$$

The cost matrix M is sparse, symmetrical and positive semidefinite. LLE then computes the bottom $(d+1)$ eigenvectors of the matrix M, associated with the $(d+1)$ smallest eigenvalues. The first eigenvectors (composed of all elements equal to 1) whose eigenvalue is close to zero is excluded. The remaining d eigenvectors yield the final embedded coordinates.

Note, that while the reconstruction weights for each data point are computed from its local neighborhood—independent of the weights for other data points—the embedding coordinates are computed by an $n \times n$ eigensolver, a global operation that couples all data points in connected components of the graph defined by the weight matrix.

Because M is sparse, eigenvector computation is quite efficient, though for large n it anyway remains the most computationally expensive step of the algorithm.

Implementation of the algorithm is fairly straightforward, as the algorithm has only one free parameter: the number of neighbours per data point, k. Once neighbours are chosen, the optimal weights $w^{(i)}$ and coordinates y_i are computed by standard methods in linear algebra.

The algorithm involves a single pass through three steps and finds global minima of the reconstruction and embedding costs in (2.1) and (2.2). No learning rates or annealing schedules are required during the optimization and no random or local optima affect the final results.

Thus, the LLE algorithm can be summarized as:

1. choose k and select the neighbours of each data point, x_i;
2. Compute the weights $w^{(i)}$ that best reconstruct each data point x_i from its neighbours, minimizing the cost in (2.1) by constrained least squares;
3. Compute the vectors y_i best reconstructed by the weights $w^{(i)}$, minimizing the quadratic form in (2.2) by the bottom eigenvectors.

2.4.1 Step 2: an overview on regularization problem

In the second step of the LLE algorithm, each data point is reconstructed by its nearest neighbours. Considering a particular data point x_i of size $D \times 1$ with k nearest neighbours collected in a matrix $X^{(i)}$ of size $D \times k$ and the vector of the reconstruction weights $w^{(i)}$ of size $k \times 1$ that sum to one, is possible to write the reconstruction error as:

$$\mathrm{SS}_1^{(i)}(w,k) = \left| x_i - \sum_{j=1}^{k} X_j^{(i)} w_j^{(i)} \right|^2 , \tag{2.6}$$

i.e., how well each point x_i can be linearly reconstructed in terms of its neighbours. Since $\sum_{j=1}^{k} w_j^{(i)} = 1$ we can rewrite the Equation (2.6) as:

$$\mathrm{SS}_1^{(i)}(w,k) = \left| \sum_{j=1}^{k} w_j^{(i)}(x_i - X_j^{(i)}) \right|^2 = \sum_{j=1}^{k}\sum_{h=1}^{k} w_j^{(i)} w_h^{(i)} G_{jh}^{(i)}, \tag{2.7}$$

where $G_{jh}^{(i)} = (x_i - X_j^{(i)})^T(x_i - X_h^{(i)})$ is the local Gram[1] matrix where $X_j^{(i)}$ and $X_h^{(i)}$ are neighbours of the point x_i. By construction, this Gram matrix is symmetric and semipositive definite.

The reconstruction error can be minimized analytically using a Lagrange multiplier to enforce the constraint that $\sum_{j=1}^{k} w_j^{(i)} = 1$. In terms of the inverse of the Gram matrix the optimal weights are given by:

$$w_j^{(i)} = \frac{\sum_{h=1}^{k} {G_{jh}^{(i)}}^{-1}}{\sum_{l,m=1}^{k} {G_{lm}^{(i)}}^{-1}}, \tag{2.8}$$

the solution, as written in Equation (2.8), appears to require an explicit inversion of the Gram matrix. In practice, a more efficient and numerically stable way to minimize the error (which yields to same results as above) is simply to solve the linear system of equations [45]:

$$\sum_{h=1}^{k} G_{jh}^{(i)} w_h^{(i)} = 1. \tag{2.9}$$

When the neighbours outnumber the input dimensionality, that is when $k > D$, the local reconstruction weights are no longer uniquely defined since the Gram matrix in Equation (2.9) is singular or nearly singular.

To break the degeneracy, a regularization for the Gram matrix is required, which is done by adding a small positive constant to the diagonal elements of the

[1] given a set V of m vectors of points in $\Re^n$, the Gram matrix G is the matrix of all possible inner products of V, *i.e.* $g_{ij} = \mathbf{v}_i^T \mathbf{v}_j$

matrix [45]. In practice, a regularization parameter r will have to be used for the matrix $G^{(i)}$ before computing its inverse (as its rank is D, certainly for $k > D$):

$$(G^{(i)} + rI)^{-1}. \tag{2.10}$$

Since $G^{(i)}$ is symmetric and semipositive definite, the matrix $G^{(i)} + rI$ has its eigenvalues in $\left[r, r + {G^{(i)}}^2\right]$ and hence a condition number $\leq \frac{r+{G^{(i)}}^2}{r}$ that becomes smaller as r increases. This regularization is known as *Tikonov* regularization [39].

2.4.2 Examples

The embeddings discovered by LLE are easier to visualize for intrinsically two dimensional manifolds. Consider an illustrative example of the nonlinear dimensionality reduction problem which is demonstrated by the two-dimensional manifold in Fig. 2.2. In this example, a linear method as PCA and a nonlinear one as LLE are applied to the data in order to discover the true structure of the manifolds. Figure 2.2(b) and 2.2(c), corresponding to the two-dimensional PCA and LLE projections, allow us to conclude that LLE succeeds in recovering the underlying manifolds whereas PCA creates local and global distortion by mapping faraway points to nearby points in the planes.

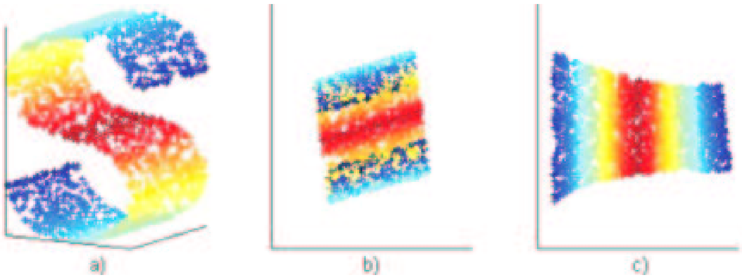

Figure 2.2. *Nonlinear dimensionality reduction problem: a)initial nonlinear data manifold, b)result obtained with the PCA linear method and c)result obtained with the LLE nonlinear method*

Figure 2.3 shows another two dimensional manifold living in a much higher dimensional space. The authors, Roweis and Saul, generated these examples—shown in the middle panel of the figure—by translating the image of a single face across a larger background of random noise. The input to LLE consisted of $n = 961$ grayscale images, with each image containing a 28×20 face superimposed on a 59×51 background of noise. The bottom portion of Fig. 2.3 shows the first two components discovered by LLE, with $k = 4$ neighbours per data point. By contrast, the top portion shows the first two components discovered by PCA. It is clear that the manifold structure in this example is much better modeled by LLE.

Finally, in addition to these examples, for which the true manifold structure was known, the authors also applied LLE to images of lips used in animation of talking heads. The database contained $n = 8588$ color images of lips at 108×84 resolution. The top and the bottom panel of Figure 2.4 show the first two components discovered, respectively, by PCA and LLE with $k = 16$ neighbours per data point. If the lip images described a nearly linear manifold, these two methods would yield similar results; thus, the significant differences in these embeddings reveal the presence of nonlinear structure. Is possible to note that while the linear projection by PCA has a somewhat uniform distribution about its mean, the LLE has a distinctly spiny structure, with the tips of the spines corresponding to extremal configurations of the lips.

2.5 Related works and Extension of the Locally Linear Embedding

Locally Linear Embedding was designed for unsupervised learning. This section starts with a description of some other unsupervised techniques as “isomap” presented in Section 2.5.1 and its variant, “c-isomap” described in Section 2.5.2, and a comparison of them with LLE.

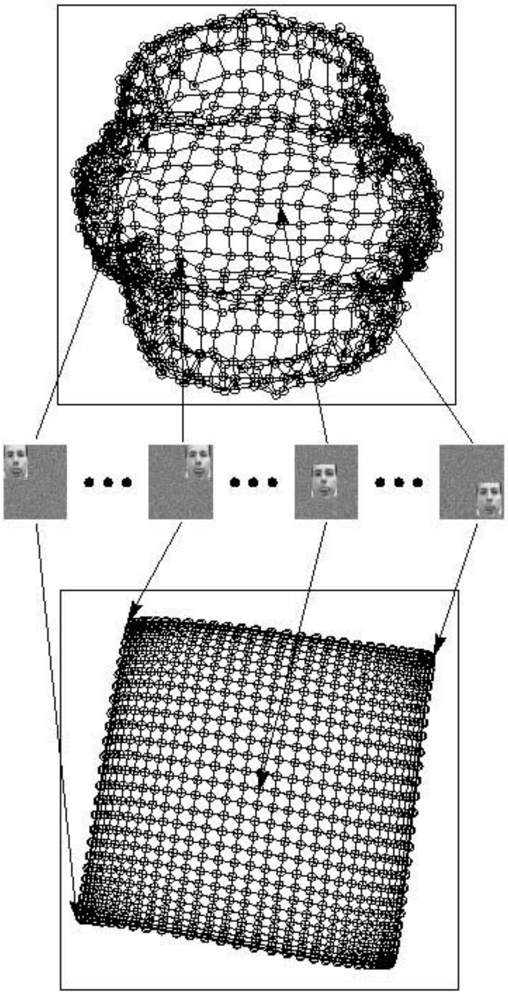

Figure 2.3. *The result of PCA (top) and LLE (bottom) applied to images of a single face translated across a two dimensional background of noise. Note how LLE maps the images with corner faces to the corners of its two dimensional embedding, while PCA fails to preserve the neighborhood structure of nearby images.*

Besides, the LLE is constructed to deal with data mining problems, where the number of classes and relationship between elements of different classes are unknown. To complement the original LLE, a supervised LLE (SLLE), extending the concept of LLE to multiple manifolds is proposed Section 2.5.3.

The original LLE algorithm possesses a number of limitation that make it to

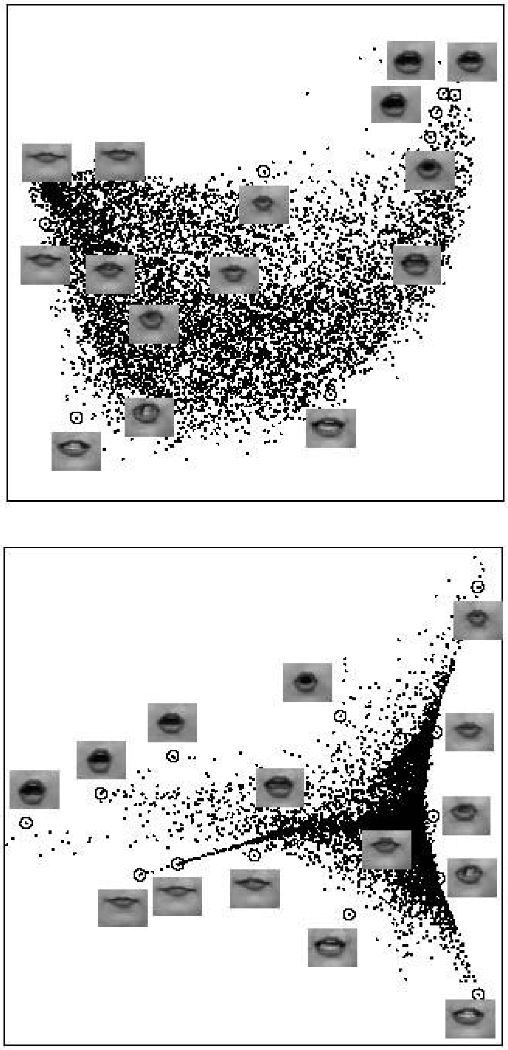

Figure 2.4. *Images of lips mapped into the embedding space described by the first two coordinates of PCA (top) and LLE (bottom). Representative lips are shown next to circled points in different parts of each space. The difference between the two embeddings indicate the presence of nonlinear structure in the data.*

be less attractive for the scientist. Thus, in the last parts of the section we review some methods in order to overcome the main drawbacks of the LLE algorithm:

i) the conventional LLE algorithm operates in a batch mode, that is, it obtains a low-dimensional representation for a certain number of high-dimensional data points to which the algorithm is applied. When new data points arrive, one needs to completely rerun the algorithm. An extension which allows dealing with sequentially incoming data is presented in Section 2.5.4;

ii) the LLE algorithm does not result robust and efficient when the data present some noise or outliers, thus, we consider an approach to make LLE algorithm more robust in Section 2.5.5;

iii) a natural question is how does one choose the number of nearest neighbours k to be considered in LLE since this parameter dramatically affects the resulting projection?. To answer this question a procedure for automatic selection of the optimal value for the parameter k is proposed in Section 2.5.6;

iv) the second LLE parameter to be set is a dimensionality of the projected space, d. It is natural, for visualization purposes that d is 1, 2 or 3; but different choices are required when one needs to preprocess data before applying subsequent operations. In the Section 2.5.7 we present LLE approach for calculating an approximate intrinsic dimensionality (Polito and Perona, 2002) and we compare it with one of the classical methods (Pettis *et al.*, 1979).

2.5.1 Isomap

LLE illustrates a general principle of manifold learning, elucidated by Tenenbaum *et al.*, that overlapping local neighborhoods can provide information about global geometry. Many virtues of LLE are shared by the "isomap" algorithm, which has been successfully applied to similar problems in nonlinear dimensionality reduction.

"Isomap" is a nonlinear generalization of Multi Dimensional Scaling in which embeddings are optimized to preserve "geodesic" distances between pairs of data points. Like LLE, the "isomap" algorithm has three steps: *(i)* construct a graph in which each data point is connected to its nearest neighbours, *(ii)* compute the shortest distance between all pairs of data points among only those paths that connect nearest neighbours, *(iii)* embed the data via MDS so as to preserve these distances, as is possible to see from the example represented in Fig. 2.5 [46].

Though similar in its aims, “isomap” is based on a radically different philosophy than LLE. In particular, “isomap” attemps to preserve the global geometric properties of the manifold, as characterized by the geodesic distances between faraway points, while LLE attemps to preserve the local geometric properties of the manifold as characterized by the linear coefficients of local reconstructions. Depending on the application, one algorithm or the other may be most appropriate.

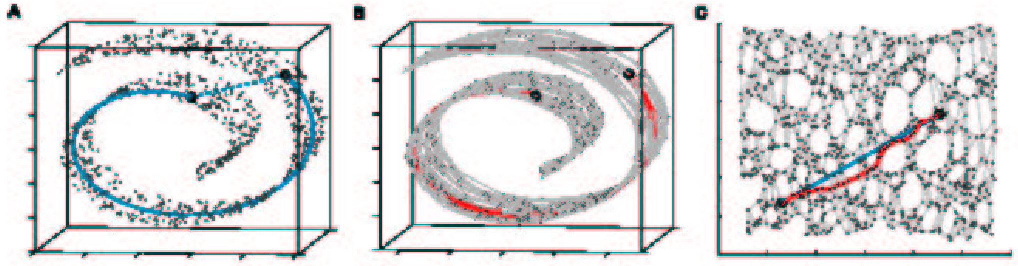

Figure 2.5. *The “Swiss roll” data set, illustrating how “isomap” exploit geodesic paths for nonlinear dimensionality reduction. (A) For two arbitrary points (circled) on a nonlinear mainifold their euclidean distance in the high-dimensional input space (length of dashed line) may not accurately reflect their intrinsic similarity, as measured by geodesic distance along the low-dimensional manifold (length of solid curve). (B) the neighborhood graph constructed in step one of “isomap” allows an approximation (red segment) to the true geodesic path to be computed in step two, as the shortest path. (C) The two dimensional embedding recovered by “isomap” in step three, which best preserves the shortest path distances in the neighborhood graph. Straight lines in the embedding (blue) represent simpler and cleaner approximations to the true geodesic paths than do the corresponding graph paths (red).*

2.5.2 c-isomap

“Isomap” was designed to learn nonlinear mappings which are isometric embeddings of a flat, convex data set, while, under appropriate conditions, LLE can recover conformal mappings, that is, mappings which locally preserve angles but not necessarily distances. Such mappings cannot generally be recovered by “isomap”, whose embeddings explicitily aim to preserve the distance between the inputs.

Noting this, de Silva and Tenenbaum, proposed in 2002 a variant of “isomap” that is able to recover conformal mappings, under the assumption that the data is distributed uniformly or with known density in the low dimensional embedding space. The authors extended the “isomap” approach to a class of intrinsically curved data sets that are conformally equivalent to euclidean space. This allows to learn the structure of manifolds like a fishbowl, as well as the other more complex data manifolds where the conformal assumptions may be approximately valid. The algorithm, called “c-isomap”, uses the observed density in the high dimensional input space to estimate and correct for the local neighborhood scaling factor of the conformal mapping. In general, the effect of “c-isomap” is to magnify regions of the data where the point density is high, and to shrink regions where the point density is low.

However, in those situations where both “isomap” and “c-isomap” are applicable it may be preferable to use “isomap”, being less susceptibile to local fluctuations in the sample density.

2.5.3 Supervised Locally Linear Embedding

To extend the concept of LLE to multiple manifolds, each representing data of one specific class, two supervised variants of LLE were independently proposed in [31, 42]. Being unsupervised, the original LLE does not make use of the class membership of each point to be projected.

To complement the original LLE, a supervised LLE is proposed. Its name implies that membership information influences which points are included in the neighborhood of each data point. That is, the supervised LLE employs prior information about a task to perform feature extraction. The supervised LLE is useful since it can deal with data sets containing multiple and often disjoint manifolds, corresponding to classes. Two approaches to the supervised LLE have been proposed. The first approach, known as 1-SLLE consists in forming the neighborhood of a data point only from those points that belong to the same class [31]. The second approach, α-SLLE, expands the interpoint distance if the points belong to different classes; otherwise, the distance remains unchanged [42]. Either approach modifies only the first step of the original LLE, while leaving the other two steps unchanged. The first step is modified by changing the distance matrix computation, that is, the distances between samples belonging to different classes are increased, but they are left unchanged if samples are from the same class:

$$\Delta = \Delta + \alpha max(\Delta)\Lambda \text{ where } \alpha \in [0, 1]$$

where $\Lambda_{ij} = 0$ if the points x_i and x_j belong to the same class and 1 otherwise. When $\alpha = 0$ we obtain the original LLE, while when $\alpha = 1$ we get the fully supervised LLE (1-SLLE). As α varies between 0 and 1 a partial SLLE (α-LLE) is obtained. Applying the supervised LLE to a number of benchmark data sets, the results confirm that SLLE generally leads to better classification performance than LLE. This is to be expected, as SLLE can extract nonlinear manifolds in a supervised way, and is thereby the most general of the feature extraction methods.

2.5.4 Generalization to new data

Another important weak point of the original LLE is that it is stationary with respect to the data, that is, it requires a whole set of points as an input in order to map them into the embedded space, that is, it operates in a batch mode [31].

When new data points arrive, the only way to map them is to pool both old and new points and rerun LLE again for this pool. In other words, the original LLE lacks generalization to new data. It means that it is not suitable in a changing, dynamic environment [31].

To overcome this weakness, in 2001, an attempt was made by Kouropteva to adapt LLE to a situation when the data come incrementally point by point. It assumed that the dimensionality of the embedded space does not grow after projecting a new point to it, that is, d remains constant. A simple technique was proposed where the adaptation of the embedded space to a new point can be done by means of the weight matrix w. Suppose that the old data, consisting of n points constitute the matrix x_{old}. For a new point x_{n+1}, the euclidean distances to all points in x_{old} are computed. Indices of nearest neighbours for other points in x_{old} do not change and thus neither the recomputation of the matrix of neighbours is needed. Let w_{old} be the weight matrix associated with the original data. In step 2 of LLE, x_{n+1}, instead of the whole matrix x and its neighbours are used to compute w_{n+1}, which is then added to w_{old} by forming the matrix $w_{new} = w_{old} \cup w_{n+1}$. Step 3 of LLE is then performed using w_{new}. Experiments conducted by Kouropteva demonstrated that is obtained approximately identical output matrix y as if the original LLE were applied to the matrix $x_{new} = x_{old} \cup x_{n+1}$. When calculating a square sum of differences between LLE projections in both cases, error was of order 10^{-5} [30].

2.5.5 Robust Locally Linear Embedding

The ability of LLE to deal with large sizes of high dimensional data and non-iterative way to find the embedding makes it more and more attractive to several researchers. All the studies which investigated and experimented this approach have concluded that LLE is a robust and efficient algorithm when the data lie on a smooth and well sampled single manifold [20].

Recently, Hadid and Pietikäinen explored the behavior of the LLE algorithm when the data include some noise or outliers and proposed a method to make LLE more robust. Their method is based on the assumption that all outliers are very far away from the data on the manifold and they themselves form distinct connected components in the neighborhoood graph. Hence the outliers have no effect on the reconstruction of the manifold data points. Apparently, this assumption is not always true for many real-world applications.

To overcome this restricted assumption and make LLE more robust, in 2005

Chang and Yeung proposed a different approach to the problem of outliers. They believed it is crucial to be able to identify the outliers and reduce their influence on the embedding result. Their robust version of LLE, or RLLE, first performs local robust PCA [47] on the data points in the manifold using a weighted PCA algorithm. A reliability score is then obtained for each data point to indicate how likely it is a clean data point (i.e., non-outlier). The reliability scores are then used to constrain the locally linear fitting procedure and generalize the subsequent embedding procedure by incorporating the reliability scores as weights into the cost function. The undesirable effect of outliers on the embedding result can thus be largely reduced. Experimental results on both synthetic and real-world data show the efficacy of RLLE. The RLLE algorithm makes LLE more robust from two aspects. In the first step of the algorithm, the probability of choosing outliers as neighbours is reduced so that the reconstruction weights reflect more accurately the local geometry of the manifold. In the second step, the undesirable effect of outliers on the embedding result is further reduced by incorporating the reliability scores as weights into the cost function.

2.5.6 Automatic determination of the optimal number of nearest neighbours

The algorithm of LLE has two parameters to be set: the number of nearest neighbours k for each data point and the dimensionality of the embedded space d, that is, the intrinsic dimensionality of a manifold or equivalently the minimal number of degrees of freedom needed to generate the original data [34]. One of the aims of multidimensional data analysis is visualization, which often helps to see clustering tendencies in underlying data. Visualization, frequently considered in previous works, mean that d is fixed (1, 2 or 3) for the purpose of a better visualization, so that the only parameter to be estimated is k. The reason for choosing the right k is that a large number of nearest neighbours produces smoothing or eliminating of small-scale structures in the manifold. In contrast, too small number of nearest neighbours can falsely divide the continuous manifold into disjointed sub-manifolds [34].

In Kouropteva *et al.* (2002), a procedure for the automatic selection of the optimal value for the parameter k is proposed. Thus, the aim was to find a measure which could faithfully estimate the quality of input-output mapping, that is, how well the high dimensional structure is represented in the embedded space. The authors considered the residual variance to be suitable for this purpose. It is defined as $1 - \rho^2_{D_x D_y}$ where ρ is the standard linear correlation coefficient taken over all entries of D_x and D_y where D_x and D_y are the matrices of euclidean distances (between pairs of points) in the high dimensional and embedded spaces, respectively. The lower the residual variance is, the better high dimensional data are represented in the embedded space. Hence, the optimal value for k, k_{opt}, can be determined as:

$$k_{opt} = \arg\min_k (1 - \rho^2_{D_x D_y})$$

In order to select the value of k_{opt}, first a set of potential candidates to become k_{opt} is selected without proceeding through all steps of LLE, followed by computing the residual variance for each candidate and picking that candidate for which this measure is minimal. As a result, the most time-consuming operation of LLE, that is, the eigenvector computation, is carried out only few times, which leads to a significant speed up. Result obtained with face images and with wood images demonstrate that the method is accurate [32]. Details of the method can be found in [34].

2.5.7 Estimation of the Intrinsic Dimensionality

In the situation where the intrinsic dimensionality of a high dimensional data set is not enforced to 1, 2 or 3 for a better visualization purpose, one might be interested in estimating the intrinsic dimensionality . As evidenced above, the goal of this intrinsic dimensionality estimation is to find the number of independent parameter needed to represent a data sample.

The PCA strategy to find the intrinsic dimensionality for linear manifolds is based on computing the linear projections of greatest variance from the top eigenvectors of the covariance matrix for the data.

In 2002, Polito and Perona, proposed a similar strategy for LLE, that is, they try to estimate d by the number of eigenvalues that are appreciable in magnitude to the second smallest nonzero eigenvalue of the cost matrix M, from Equation (2.5). The main difference between PCA and LLE is that the former method concentrates on finding eigenvectors corresponding to the largest eigenvalues, while the latter one searches for the bottom eigenvectors. Hence, in the case of LLE, one has to deal with ill-conditioned eigenvalues and eigenvectors.

In spite of the ill-conditioning of the individual eigenvectors, those corresponding to a cluster of close eigenvalues are better conditioned together, and they span the invariant subspace [7]. As evidenced by foregoing, it is natural to estimate the intrinsic dimensionality of the data manifold by the number of the smallest eigenvalues that form a cluster as illustrated in [34].

In Saul and Roweis (2003), it was empirically shown that sometimes this procedure of estimating the intrinsic dimensionality does not work for data that has been sampled in a non uniform way. This situation might occur in a case where a gap between ill-conditioned eigenvalues is large. Hence, the eigenspace obtained with the eigenvectors corresponding to the cluster formed by the closest eigenvalues is no longer invariant, leading to wrong intrinsic dimensionality estimations [34].

Thus, they have found more useful to rely on classical methods [27] which should be used prior to the final step of LLE for estimating the intrinsic dimensionality d of a data set. Further details can be found in [34].

Chapter 3

Automatic determination of k neighbours

In this chapter we deal with the determination of the optimal number of k nearest neighbours since this parameter dramatically affects the resulting projection of the high dimensional data into a low dimensional space. In fact a large number of nearest neighbours causes smoothing or eliminates the scale structures in the data whereas a small number of nearest neighbours can falsely divide the continuous manifold into disjoint sub-manifolds.

3.1 Our proposal for the choice of k neighbours

For each data point let us assume that

$$x_i = X^{(i)} w^{(i)} + {}_1e_i$$

where ${}_1e_i \sim \mathcal{N}(0, \Sigma)$ and Σ is the diagonal covariance matrix that doesn't depend on a particular data point i and for which we assume the homoscedasticity assumption:

$$\Sigma_{D\times D} = \begin{bmatrix} \sigma^2 & 0 & 0 & 0 \\ 0 & \sigma^2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \sigma^2 \end{bmatrix}.$$

Thus, for a data matrix of size $D \times n$, we deal with n models of multiple regression and we want to solve a problem of model selection in order to identify the best number of k neighbours able to reconstruct each data point by minimizing the error.

In order to solve this issue we investigate three criteria for the automatic determination of the optimal number of nearest neighbours:

- $\bar{R}^2$, the adjusted coefficient of multiple determination,
- The Akaike's Information Criterion,
- The Bayesian's Information Criterion.

3.1.1 The adjusted coefficient of multiple determination

The adjusted $\bar{R}^2$ is a measure of how well the independent variables predict the dependent variable. The adjusted $\bar{R}^2$ penalizes R^2 for the number of explanatory variables included into the model, in fact, the determination index R^2 has the disadvantage that always increases when a new regressor is added to the model. As regressors are added to the model, each predictor will explain some of the variance in the dependent variable simply due to chance.

Therefore, the adjusted $\bar{R}^2$ is a better comparison between models with different number of independent variables and it is required when the number of regressors is high relative to the number of points. It is computed as:

$$\bar{R}^2 = 1 - \left[(1 - R^2)\frac{n-1}{n-p-1}\right]$$

where n are the points and p the number of regressors included into the model. As is possible to note from the above formula, when the number of observation is small and the number of regressor is large, there will be a much greater difference between $\bar{R}^2$ and R^2 because the ratio $\frac{n-1}{n-p-1}$ will be less than 1. By contrast, when the number of observation is very large compared to the number of regressor the value of $\bar{R}^2$ and R^2 will be much closer because the ratio $\frac{n-1}{n-p-1}$ will approach to 1.

For these reasons the adjusted $\bar{R}^2$ seems to be suitable for model selection where each of n model considered has a number of points relative small respect to the number of regressors included in the model at each time.

Assuming a model $x_i = X^{(i)}w^{(i)} + {}_1e_i$, where the point x_i is a *D-dimensional* vector, the adjusted $\bar{R}^2$ is defined as:

$$\bar{R}^2 = 1 - \left[\frac{\sum_{d=1}^{D} \left(x_{id} - \sum_{j=1}^{k} X_j^{(i)} w_j^{(i)} \right)^2 / (D-k)}{\sum_{d=1}^{D} \left(x_{id} - \bar{x}_i \right)^2 / (D-1)} \right],$$

where $SSE = \sum_{d=1}^{D} \left(x_{id} - \sum_{j=1}^{k} X_j^{(i)} w_j^{(i)} \right)^2$ is the residual sum of squares and $(D-k)$ its degrees of freedom and $SST = \sum_{d=1}^{D} \left(x_{id} - \bar{x}_i \right)^2$ is the total sum of squares of the model with $(D-1)$ degrees of freedom. A higher adjusted $\bar{R}^2$ indicates a better model.

3.1.2 The Akaike's Information Criterion

The Akaike's information criterion (AIC), developed by Hirotugu Akaike in 1971 and proposed in 1974, is a measure of the goodness of fit of an estimated statistical model. It is grounded in the concept of entropy and it is an operational way of trading off the complexity of an estimated model against how well the model fits the data. It is defined as:

$$AIC = -2 \max \log \text{likelihood} + 2p$$

where p is the number of the model free parameters. Details can be found in [2].

In 1951 Kullback and Leibler had been addressed the issue of finding which model would be best approximate reality given the data recorded. In other words, they had been tried to minimize the loss of information and so they had been developed a measure, the Kullback–Leibler Information, [1] to represent the information lost, when approximating reality.

Hirotugu Akaike in 1971 has been developed a measure for model selection using Kullback–Leibler Information. This establishes a relationship between the maximum likelihood, which is an estimation method used in many statistical analysis, and the Kullback–Leibler Information. Later in 1981, he declared in [4]:

> [*...On the morning of March 16, 1971, while taking a seat in a commuter train, I suddenly realized that the parameters of the factor analysis model were estimated by maximizing the likelihood and that the mean value of the logarithmus of the likelihood was connected with the Kullback-Leibler Information number. This was the quantity that was to replace the mean squared error of prediction. A new measure of the badness of a statistical model with parameters determined by the method of maximum likelihood was then defined by the formula* $AIC = (-2)log_e(maximum\ likelihood) + 2(number\ of\ parameters)$. *AIC is an acronym for "an information criterion" and was first introduced in 1971. A model with a lower value of AIC is considered to be a better model. ...*].

[1] In probability theory and information theory, the Kullback-Leibler divergence (or information divergence) is a natural distance measure from a true probability distribution P to an arbitrary probability distribution Q. Typically P represents data, observations, or a precise calculated probability distribution. The measure Q typically represents a theory, a model, a description or an approximation of P.For probability distributions P and Q of a discrete variable the KullbackLeibler divergence of Q from P is defined as:$D_{KL}(P||Q) = \sum_i P(i)log\frac{P(i)}{Q(i)}$. For distributions P and Q of a continuous random variable the summations give way to integrals, so that: $D_{KL}(P||Q) = \int_{-\infty}^{\infty} p(x)log\frac{p(x)}{q(x)}\,dx$ where p and q denote the densities of P and Q. (Wikipedia)

Considering a model $x_i = X^{(i)}w^{(i)} +_1 e_i$ we proceed to compute the log likelihood of the model:

$$\begin{aligned} l(\Sigma \setminus x, w) &= \sum_{i=1}^{n}\sum_{d=1}^{D}\left[-\log(\sigma\sqrt{2\pi}) - \frac{1}{2}\left(\frac{x_{id} - \sum_{j=1}^{k} X_j^{(i)} w_j^{(i)}}{\sigma}\right)^2\right] \\ &= \sum_{i=1}^{n}\sum_{d=1}^{D}\left[-log\sqrt{2\pi} - log\sigma - \frac{1}{2}\left(\frac{x_{id} - \sum_{j=1}^{k} X_j^{(i)} w_j^{(i)}}{\sigma}\right)^2\right] \\ &= C - \frac{nD}{2}log\sigma^2 - \frac{1}{2}\sum_{i=1}^{n}\sum_{d=1}^{D}\frac{\left(x_{id} - \sum_{j=1}^{k} X_j^{(i)} w_j^{(i)}\right)^2}{\sigma^2} \end{aligned}$$

where C stands for a constant equal to $-log\sqrt{2\pi}$.

Then, we proceed to maximize the log likelihood over σ^2:

$$\frac{\delta}{\delta\sigma^2} l(\Sigma \setminus x, w) = -\frac{nD}{2\sigma^2} + \frac{1}{2}\frac{\sum_{i,d}\left(x_{id} - \sum_{j=1}^{k} X_j^{(i)} w_j^{(i)}\right)^2}{(\sigma^2)^2}.$$

In order to obtain the maximum estimate of σ^2, we set it to zero obtaining the following log likelihood estimate:

$$-\frac{nD}{2\sigma^2} + \frac{1}{2}\frac{\sum_{i,d}\left(x_{id} - \sum_{j=1}^{k} X_j^{(i)} w_j^{(i)}\right)^2}{(\sigma^2)^2} = 0$$

$$\Longrightarrow \hat{\sigma}^2 = \frac{\sum_{i,d}\left(x_{id} - \sum_{j=1}^{k} X_j^{(i)} w_j^{(i)}\right)^2}{nD}$$

Finally, we obtain the maximum log likelihood as:

$$\begin{aligned} l(\hat{\Sigma}, \backslash x, w) &= C - \frac{nD}{2} log \sum_{i,d} \left(x_{id} - \sum_{j=1}^{k} X_j^{(i)} w_j^{(i)} \right)^2 - log(nD) - \frac{1}{2} \sum_{i,d} nD \\ &= C + C^* - \frac{nD}{2} log \sum_{i,d} \left(x_{id} - \sum_{j=1}^{k} X_j^{(i)} w_j^{(i)} \right)^2 \end{aligned}$$

where C^* is the constant equal to $\left(-log(nD) - \frac{1}{2}\sum_{i,d} nD\right)$.

Replacing the maximum log likelihood in the Akaike's Information Criterion we achieve:

$$AIC(k) = nDlog \sum_{i=1}^{n} \sum_{d=1}^{D} \left(x_{id} - \sum_{j=1}^{k} X_j^{(i)} w_j^{(i)} \right)^2 + 2(nk + 1)$$

where $nk + 1$ is the number of the model free parameters, that is $w_j^{(i)}$ for $i = 1, \ldots, n$; $j = 1, \ldots, k$ and σ^2.

In order to select the optimal number of nearest neighbours we choose those k^* that minimize the AIC value:

$$k^* = \arg\min_{(1 \leq k \leq n-1)} AIC(k)$$

3.1.3 The Bayesian's Information Criterion

The Bayesian's Information Criterion (BIC) proposed by Akaike in 1978 has become a popular criterion for model selection in the last few years [3]. The BIC was developed to provide a measure of the weight of evidence favoring one model over another, or Bayes factor.

To combine the maximum likelihood (data fitting) and the choice of model, the maximum log likelihood would be penalized with a term related to the model complexity. The typical penalty term is like αp where p is the number of the model free parameters.

When $\alpha = 2$ the Akaike's Information Criterion formula is obtained as described in the previous section.

When $\alpha = logn$ the Bayesian's Information Criterion is achieved:

$$BIC = -2 \max \log \text{likelihood} + p \log n$$

which is characterized by a higher penalty term respect AIC since it involves also the sample dimension n.

For the model $x_i = X^{(i)}w^{(i)} +_1 e_i$ the Bayesian's Information Criterion results:

$$BIC(k) = nDlog \sum_{i=1}^{n} \sum_{d=1}^{D} \left(x_{id} - \sum_{j=1}^{k} X_j^{(i)} w_j^{(i)} \right)^2 + (nk+1)log(n).$$

In order to select the optimal number of nearest neighbours we choose those k^* that minimize the BIC value:

$$k^* = \arg\min_{(1 \leq k \leq n-1)} BIC(k).$$

Chapter 4

Automatic Determination of d coordinates

The original LLE algorithm was proposed in the context of visualizing the speech and audio signals characterized by a high dimensionality. Since is not possible for the human observer visually perceive a high dimensional representation of the data, the dimensionality was automatically reduced to one, two or three. In each other situations where the purpose of the dimensionality reduction technique can not reduced to data visualization, is needed to determine the intrinsic dimensionality of the data, that is, the number of free parameters needed to represent the original high-dimensional space. In this chapter we propose some criteria to solve the issue.

4.1 Proposal for the automatic determination of parameter d

Once the optimal number of nearest neighbours able to reconstruct each data point has been chosen, the purpose is that nearby points in the high-dimensional space remain nearby and similarly co-located in the low-dimensional embedding. Effectively we want that the location between points in the low-dimensional space is similar to the location between points in the original space.

The idea was to measure the location between points in order of distances. Thus, let D_X and D_Y be the $n \times n$ euclidean distance matrices measured in the input and output spaces respectively. We will typically use the euclidean distance, but one can also make other choice. Then, we transform the matrices into the H-dimensional vectors δ^X and δ^Y by considering only the distances between the points and their neighbours[1]. The dimension of the vectors depends by the number of points and neighbours considered, so that it results $H \leq kn$.

We assume that:

$$\delta^Y = b + c\delta^X + {}_2e,$$

where b is the intercept of the model and ${}_2e \sim \mathcal{N}(0, \sigma_d^2)$.

As the dimensionality of X and Y are intrinsically different, it is crucial to introduce a constant c, which is able to capture the difference in scale, in fact, if $\delta^Y = c\delta^Y$, then Y is a good representation of X. When d increases, we expect σ_d^2 to decrease as a result of an increasingly better fit.

We propose three different indices to evaluate the decrease of the error ${}_2e$:

1. the coefficient of determination R^2,

2. The Akaike's Information Criterion,

3. The Bayesian's Information Criterion.

[1] Let d_{ij} the euclidean distance between pairs of points x_i and x_j. We assume $d_{ij} = 0$ if the points x_i and x_j are not close to each other; otherwise, we assume $d_{ij} = 1$ if x_i is close to x_j but x_j is not close to x_i and again we suppose $d_{ij} = 1$ if x_i is close to x_j and vice versa. The sum of these distances define the dimensionality of the vectors δ^X and δ^Y.

4.1.1 The coefficient of determination

As already said in Section 3.1.1, the coefficient of determination R^2 is a statistic widely used to determine the goodness of the fit. It can be computed as:

$$R^2 = \frac{Dev(Y)_{regr}}{Dev(Y)} = 1 - \frac{Dev(Y)_{disp}}{Dev(Y)}.$$

R^2 can also be computed as:

$$R^2 = \frac{Codev(X,Y)^2}{Dev(X)Dev(Y)}.$$

Following the latter formulation we determine the expression of R^2 for the regression model $\delta^Y = c\delta^X + {}_2e$:

$$R^2 = \frac{Codev(\delta^X,\delta^Y)^2}{Dev(\delta^X)Dev(\delta^Y)},$$

where

$$Codev(\delta^X,\delta^Y) = \sum_{h=1}^{H} \delta_h^X \delta_h^Y - H\bar{\delta}^X\bar{\delta}^Y,$$

$$Dev(\delta^X) = \sum_{h=1}^{H} (\delta_h^X)^2 - H(\bar{\delta}^X)^2,$$

$$Dev(\delta^Y) = \sum_{h=1}^{H} (\delta_h^Y)^2 - H(\bar{\delta}^Y)^2,$$

and H is the set of all the neighbours (with $H \leq kn$).

In order to select d^* embedded coordinates able to represent in a lower space the high-dimensional data by minimizing the error, we choose the larger value of R^2 as:

$$d^* = \arg\max_{(1 \leq d \leq D)} R^2.$$

4.1.2 The Akaike's Information Criterion

In order to compute the AIC value for the model $\delta^Y = b + c\delta^X + {}_2e$ where ${}_2e \sim \mathcal{N}(0, \sigma_d^2)$ we proceed to determine the log likelihood of the model:

$$l(\sigma_d^2 \setminus \delta^X) = \sum_{h=1}^{H} \left[-log\left(\sigma_d\sqrt{2\pi}\right) - \frac{1}{2}\frac{\left(\delta_h^Y - c\delta_h^X\right)^2}{\sigma_d^2} \right]$$

Then, we maximize the log likelihood over σ_d^2:

$$\frac{\delta}{\delta\sigma_d^2} l(\sigma_d^2 \setminus \delta^X) = -\frac{H}{2\sigma_d^2} + \frac{1}{2}\frac{\sum_{h=1}^{H}\left(\delta_h^Y - c\delta_h^X\right)^2}{\left(\sigma_d^2\right)^2}$$

To obtain the maximum estimate of σ_d^2, we set it to zero obtaining the following log likelihood estimate:

$$-\frac{H}{2\sigma_d^2} + \frac{1}{2}\frac{\sum_{h=1}^{H}\left(\delta_h^Y - c\delta_h^X\right)^2}{\left(\sigma_d^2\right)^2} = 0$$

$$\Longrightarrow \hat{\sigma}_d^2 = \frac{1}{H}\sum_{h=1}^{H}\left(\delta_h^Y - c\delta_h^X\right)^2$$

Thus, we can write the Akaike's Information Criterion as:

$$AIC(d) = -2 \max \log \text{likelihood} + 2p$$

$$\Downarrow$$

$$AIC(d) = -2\sum_{h=1}^{H}\left[-\log\left(\hat{\sigma}_d\sqrt{2\pi}\right) - \frac{1}{2}\frac{\left(\delta_h^Y - c\delta_h^X\right)^2}{\hat{\sigma}_d^2} \right] + 2^{*}3$$

where c and σ_d^2 are the model free parameters.

We choose that number of d^* embedded dimensions that satisfies the following relationship:

$$d^* = \arg\min_{(1\leq d\leq D)} AIC(d).$$

4.1.3 The Bayesian's Information Criterion

For the model $\delta^Y = b + c\delta^X + {}_2e$ representing the distances in the low dimensional embedding as expression of the distances in the high dimensional space plus an error with $Normal$ distribution ${}_2e \sim \mathcal{N}(0, \sigma_d^2)$ we derive the formulation for the Bayesian's Information Criterion as:

$$BIC(d) = -2 \max \log \text{likelihood} + p \log n$$

$$\Downarrow$$

$$BIC(d) = -2 \sum_{h=1}^{H} \left[-\log\left(\hat{\sigma}_d \sqrt{2\pi}\right) - \frac{1}{2} \frac{\left(\delta_h^Y - c\delta_h^X\right)^2}{\hat{\sigma}_d^2} \right] + 3^* \log(H)$$

where c and σ_d^2 are the model free parameters. The value of the parameter that satisfies the following relationship:

$$d^* = \arg\min_{(1\leq d\leq D)} BIC(d)$$

represents the d^* embedded coordinates that the Bayesian's Information Criterion indicates as better reconstruction of the original high dimensional space by minimizing the reconstruction error in term of the distances.

Chapter 5

Simulation Study

In order to test the proposed criteria for the automatic selection of the two free parameters of the model, some simulation studies over a different synthetic high-dimensional data sets have been conducted.

Section 5.1 presents the creation of several data sets by generating the d-dimensional manifold embedded in a D-dimensional space with $d \ll D$.

Section 5.2 illustrates the regularization problem for the simulation study, pointing out the choice for the penalty term in order to obtain an accurate selection for the parameter k.

Section 5.3 describes the results over a simulation study for the automatic determination of the optimal number of k nearest neighbours by applying the three proposed criteria. In this section a validation measure for the three methods computed, is proposed and described in details.

Section 5.4 presents the results for the automatic selection of the intrinsic dimensionality of the data manifold. The results are shown over the same simulation study generated for the choice of the optimal number of nearest neighbours.

5.1 Simulation data

The data have been generated by randomly sampling from two independent uniform variables as $u_1 \sim \mathcal{U}(0,1)$ and $u_2 \sim \mathcal{U}(0,1)$ and by generating 10 nonlinear combinations of the two-dimensional manifold.
Each data set consists of $n = 500$ points embedded in $D = 10$ non linear dimensions and 100 different data sets have been simulated. In other words, the points can be considered as a 2-dimensional manifold embedded in a 10-dimensional space. The true dimension of this manifold is represented by the two independent uniform variables, whereas other dimensions might be safely ignored.

For each data set and for each data point x_i, $i = 1, \ldots, n$, we proceed by ordering all other $n-1$ points in according to their proximity to x_i, based on euclidean distance. Proximity indices for all points are collected in a matrix "index" of size $n \times n$. Columns of this matrix correspond to points and rows correspond to the nearest neighbours, the smaller row index correspond to the nearest neighbours. For example, $index_{3,4} = 7$ means that the third nearest neighbour of the point x_4 is the point x_7.
The distance between pairs of point x_i and x_j is computed as:

$$d_{ij} = \sum_{d=1}^{D} (x_{id} - x_{jd})^2,$$

that corresponds to the quadratic euclidean distance. We will typically use the euclidean distance as in the original formulation of LLE algorithm, but other choices are however possible.

Once the distances among the data points have been computed, k nearest neighbours for each data point can be found by an automatic determination of the model parameter.

5.2 The regularization problem on simulation study

Having found k nearest neighbours for each data point, the next step is to assign a weight to every point of neighbouring points. This weight characterizes a degree of closeness of two points [31].

As mentioned in Section 2.4.1, the LLE algorithm reconstructs each data point x_i from k neighbours of x_i by means of the weight matrix as $\hat{x}_i = X^{(i)}w^{(i)}$, where $\hat{x}_i$ is the estimate of the data point x_i, $X^{(i)}$ is the matrix of k neighbours of a data point and $w^{(i)}$ is the vector of the weights of *i-th* point. The optimal weights $w^{(i)}$ are found by solving a least squares problem minimizing the reconstruction error:

$$\mathrm{SS}_1(w,k) = \sum_{i=1}^{n} \left| x_i - X^{(i)}w^{(i)} \right|^2 . \tag{5.1}$$

Considering a $(D \times 1)$ data vector x_i with its associated k nearest neighbours matrix $X^{(i)} = [x_{(i,1)}, x_{(i,2)}, \ldots, x_{(i,j)}, \ldots, x_{(i,k)}]$, where $x_{(i,j)}$ is the *j-th* neighbours of x_i, we compute the k-dimensional weight vector $w^{(i)}$ as:

$$w^{(i)} = \frac{(X_c^{(i)t} X_c^{(i)} + \Lambda)^{-1}\mathbf{1}}{\mathbf{1}^t (X_c^{(i)t} X_c^{(i)} + \Lambda)^{-1}\mathbf{1}},$$

where $X_c^{(i)} = [x_{(i,1)} - x_i, \ldots, x_{(i,k)} - x_i]$ is the centralized nearest-neighbour matrix and Λ is a diagonal matrix where the elements of the diagonal are $\lambda = \varepsilon \sum_{j=1}^{k} \lambda_j$ where λ_j is the j-th eigenvalue of $X_c^{(i)t} X_c^{(i)}$ and ε a small given tolerance. In order to define the solution for $k > D$ a regularization Λ is needed. In the original algorithm, this regularization was only applied in those circumstances. However, in order to make a fair choice of the appropriate number of neighbours k, we apply the regularization for any k. We typically use a small tolerance equal to $\varepsilon = 0.001$, as proposed by Roweis and Saul in the original LLE, but one can also make other choices.

5.3 Simulation results for the parameter k

5.3.1 Results on $\bar{R}^2$

In order to choose the optimal number of nearest neighbours we proceed by computing for each value of k that varies between 1 up to 20 and for every data point x_i, $(i = 1, \dots, n)$, n value of the adjusted coefficient $\bar{R}^2$ in according with the following formula:

$$\bar{R}^2 = 1 - \left[\frac{\sum_{d=1}^{D} (x_{id} - X_d^{(i)} w_d^{(i)})^2 / D}{\sum_{d=1}^{D} (x_{id} - \bar{x}_i)^2 / (D-1)} \right], \tag{5.2}$$

where D are the resulting degrees of freedom[1] of the residual sum of squares when the regularization term is applied for every k. We use the regularization term for any k in order to avoid to favour those $k > D$, in fact, if we apply the regularization term only when k become greater than D, follows that

$$\frac{SSE}{SST}\frac{(D-1)}{D} < \frac{SSE}{SST}\frac{(D-1)}{(D-k)}$$

and so $\bar{R}^2_{k>D} > \bar{R}^2_{k<D}$. Thus, by applying the regularization term only when $k > D$ the method would select those neighbours greater than dimension.

Once obtained $n = 500$ values of $\bar{R}^2$, we consider the mean value of $\bar{R}^2$ over n data points in order to summarize the percentage of variation that can be explained by k neighbours, in other words, how well k neighbours can faithfully describe the entire data set. After $k = 20$ values of $\bar{R}^2$ are achieved for each simulation, it is necessary to compute for every k considered, the mean value of $\bar{R}^2$ over 100 simulations. In Figure 5.1 the results are presented. Figure 5.1(a) shows

[1] When the regularization term is applied for every k and not only when $k > D$, the degrees of freedom of the residual sum of square results equal to D. In fact, the resulting degrees of freedom are obtained as the sum of $(D - k)$ effective degrees of freedom of the model plus k bonds needed to estimate each data point when the regularization is introduced. Hence the resulting degrees of freedom are: $(D - k) + k = D$

the mean values of $\bar{R}^2$ for $k = 1, \ldots, 20$ by revealing the decreasing structure of the adjusted coefficient, whereas the Figure 5.1(b) shows the histogram of the relative frequency for every k over 100 simulations.

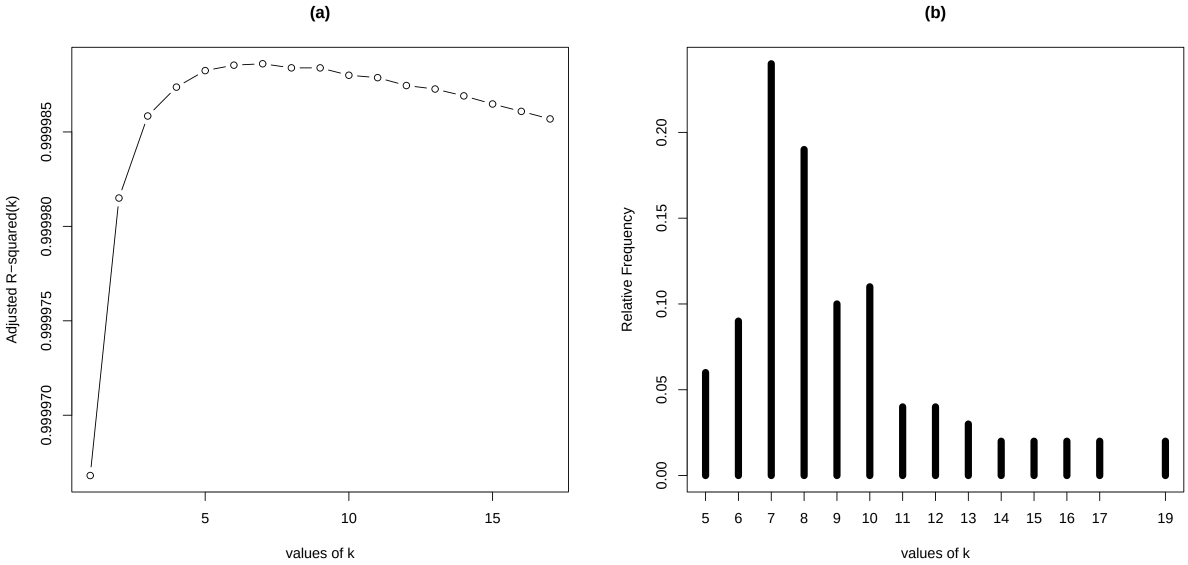

Figure 5.1. *The mean values of* $\bar{R}^2$ *for* $1 \leq k \leq 20$ *and the relative frequency of* k *over* 100 *simulations.*

As is possible to note from Figure 5.1(a), $\bar{R}^2$ rapidly increases until a value of $k = 8$ neighbours, it presents maximum values for $8 \leq k \leq 10$, after that $\bar{R}^2$ shows a decreasing behaviour. This suggests an interval of k values in which every value seems to be able to represent the whole data set by minimizing the reconstruction error. Furthermore, the Figure 5.1(b) shows the relative frequency for every value of k. For example, it explains that in 24% simulations, $k = 7$ neighbours maximize the $\bar{R}^2$ value and in 19% simulations the value of $\bar{R}^2$ is maximized when $k = 8$. Summarizing the results for the histogram, is possible to note that for 64% of the simulations conducted on 100 different data sets the value of $\bar{R}^2$ results maximum for $7 \leq k \leq 10$. Further analysis will deal us to choose only one value to be used in LLE algorithm as presented in Section 5.3.3.

5.3.2 Results on AIC and BIC

In order to define an optimality choice for the parameter[2] k, the Akaike and Bayesian's Information Criteria are computed for every simulated data set in according with the following formula[3]:

$$AIC(k) = nDlog\sum_{i=1}^{n}\sum_{d=1}^{D}\left(x_{id} - X_d^{(i)}w_d^{(i)}\right)^2 + 2(nk+1),$$

$$BIC(k) = nDlog\sum_{i=1}^{n}\sum_{d=1}^{D}\left(x_{id} - X_d^{(i)}w_d^{(i)}\right)^2 + (nk+1)log(n).$$

Thus, the mean values of Akaike and Bayesian's Information Criteria over 100 simulations are calculated, obtaining the results shown in Figure 5.2.

The Akaike's Information Criteria, reported in Figure 5.2(a), rapidly decreases until to achieve the minimum value for $k = 6$ neighbours; after that it presents an increasingly behaviour as k increases. Likewise, as shown in Figure 5.2(b), the Bayesian's Information Criterion achieves the minimum value when $k = 4$ and a very close value to the minimum when $k = 5$; after that it increases as well as k grows.

In both plots the existence of a global minimum for small values of k is caused by the fact that for the data set with many points, their first few neighbours are all close to them and so adding a new neighbour it decreases the reconstruction error every time; but, once achieved the minimum value, the penalty term included into the model works to generate a trade off between the complexity of the model and the goodness of the fit, with the result that the values tend to increase.

The optimal number of neighbours identified by minimizing the Akaike and Bayesian's Information Criteria are different to each other and furthermore they

[2] for computational problems the value of possible k neighbours included into the model is considered up to $k = 20$. This value is enough to define an optimality choice of the model free parameter.

[3] We calculate the Akaike and Bayesian's Information Criteria by introducing, for any k, the small given tolerance term equal to $\varepsilon = 0.001$ as applied to compute $\bar{R}^2$.

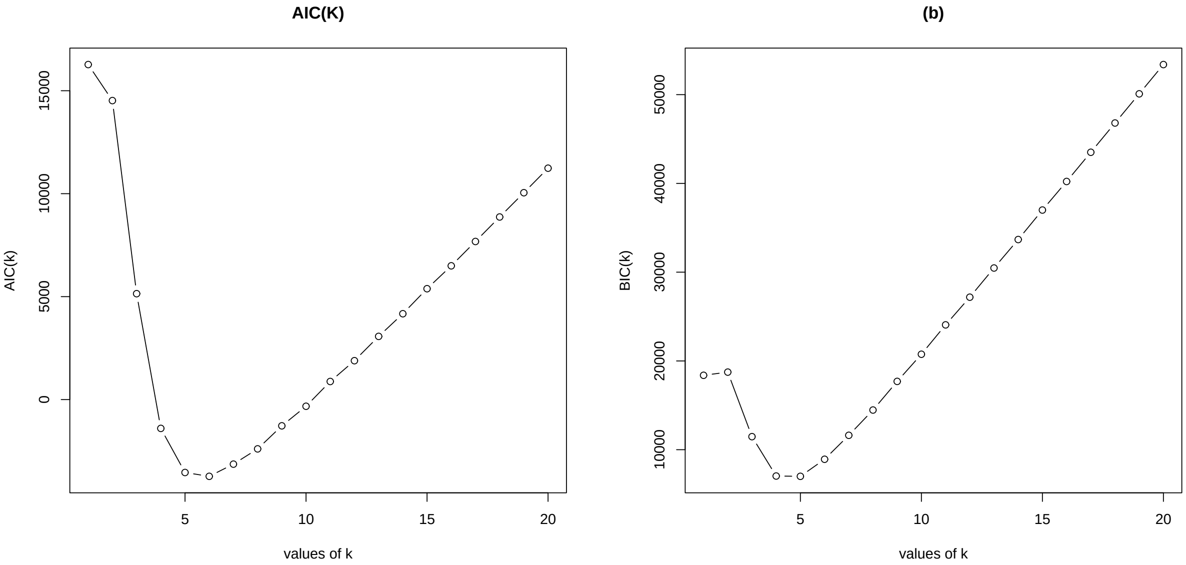

Figure 5.2. *(a)The mean values of Akaike (b) and Bayesian's Information Criteria for* $1 \leq k \leq 20$ *over* 100 *simulations.*

are quite dissimilar to the optimality choice of the model free parameter based on $\bar{R}^2$, which has been selected an interval of optimal values equal to $k \in [8, 10]$. Since we deal with synthetic data sets for which the true 2-dimensional manifold is known, we can obtain a validation measure for the three criteria in order to identify which is the best choice of the optimal number of nearest neighbours. In order to solve this issue we use the Ordinary Procrustes Analysis.

5.3.3 Validation measure to the proposed methods

Effectively, at this point, we want to provide a validation measure by comparing the configurations of the true two-dimensional manifold with those discovered by LLE algorithm for every value of k (with $k \in [1, 20]$). The Ordinary Procrustes Analysis (OPA) seems to be suitable for this purpose.

The Ordinary Procrustes Analysis is a method commonly used for comparing the shape differences between two objects with m landmarks. The shape of an object is defined in a mathematical context as all the geometrical informations that remain after location, scale and rotational effect are filtered out, that is the shape of an object is invariant under the euclidean similarity transformation of translation, scaling and rotation. Each shape is described by a finite number of points which are called landmarks. This Procrustes method estimates the optimal similarity transformation (translation, scale and rotation) parameters by minimizing a least squares criterion. Before comparing two shapes, the location, scale and rotation effect must to filter out.

The translational components can be removed by translating the object so that the mean of all points lie at the origin. The translation essentially moves the shapes to a common center. The origin $(0, 0)$ is the most likely candidate to become that common center, yet not exclusively so.

The scale component can be removed by scaling the object so that the sum of the squared distances from the points to the origin is 1, in other words, the isomorphic scaling is a manipulation technique that transform a shape smaller or larger while maintaining the ratio of the shapes proportion.

When the matrices are aligned and scaled it is time for the rotation step. Removing the rotational components is more complex. Considering two objects with scaling and translational effect removed, let the points of these be $((x_1, y_1), \dots)$ and $((z_1, t_1), \dots)$. Fix one of these and rotate the other around the origin so that the sum of the squared distances between the points is minimized. This distances can be minimized by finding the angle θ which gives the minimum distance.

Thus, let X_1 the reference configuration and X_2 the shape which is to be transformed, the Ordinary Procrustes Analysis finds the similarity transformation to be applied to X_2 which minimize its euclidean distance from the configuration X_1. It finds the similarity parameters s, R and t which minimize:

$$D^2_{OPA}(\mathbf{X_1}, \mathbf{X_2}) = (\mathbf{X_1} - (s\mathbf{X_2}R + \mathbf{1}_m\mathbf{t}^t))^2$$

where s is a scaling parameter, R is a rotation parameter and $\mathbf{t}$ is a translation vector. The minimum of the above equation is denoted by $OSS(\mathbf{X_1}, \mathbf{X_2})$ which stands for the Ordinary Procrustes sum of squares error.

For every value of $k \in [1, 20]$ we proceed by comparing, on the generated 100 data sets, the true data configurations with those obtained by LLE and in this way we compute the mean values of the Ordinary Procrustes sum of squares error over 100 simulations. The results are represented in Figure 5.3.

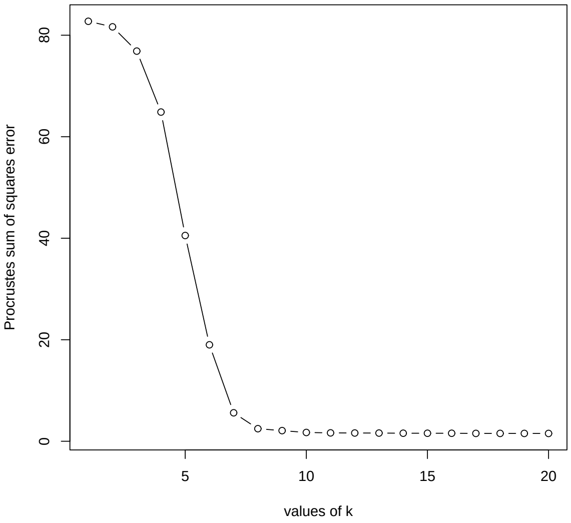

Figure 5.3. *The mean values of the Procrustes sum of squares error on 100 simulations for every value of $k \in [1, 20]$.*

The Ordinary Procrustes sum of squares error quickly decreases until a value of $k = 8$ neighbours, after this, the imperceptible decrease in the error does not justify the increase in the number of neighbours to include into the model. Thus, the plot identifies a value of $k = 8$ neighbours as the optimal neighborhood choice able to minimize the squared error between the true data configurations and those obtained by LLE algorithm.

In the light of the Procrustes Analysis, we can compare this result with those obtained by the three different proposed criteria for the choice of k neighbours.

A summarizing scheme is shown in Table 5.1.

	number of *k* neighbours
$\bar{R}^2$	$[8, 10]$
$AIC(k)$	6
$BIC(k)$	4
Procrustes	8

Table 5.1. *The optimal choice of k nearest neighbours for the three proposed criteria and for the Procrustes Analysis validation.*

The Akaike and Bayesian's Information Criteria underestimate the true parameter value, since their values are equal to $k = 6$ and $k = 4$ respectively. Otherwise the optimal number of nearest neighbours proposed by $\bar{R}^2$ technique seems to be in accordance with the best choice of k suggested via the Procrustes Analysis. In fact, $\bar{R}^2$ identifies an interval of $k \in [8, 10]$ which provides maximum values for the multiple coefficient of determination. The differences among the interval values of k are so imperceptible that we can safely choose $k = 8$ neighbours to be used in the LLE algorithm.

5.4 Simulation results for the parameter d

The intrinsic dimensionality of the data manifold can be expressed as the minimal number of degrees of freedom needed to generate the original data. In order to select the intrinsic dimensionality[4] of the generated 100 data sets with $k = 8$ neighbours per data point, we consider for each data set and for each dimension $d \in [1, 10]$ the euclidean distances matrices, D_X and D_Y, of the input and output data spaces respectively. Then, we transform the matrices into the H-dimensional vectors δ^X and δ^Y by extracting only the distances between the points and their neighbours.

After that, we compute for each dimension the mean values over 100 simulations of R^2, $AIC(d)$ and $BIC(d)$ in accordance with the formula described in Sections 4.1.1, 4.1.2 and 4.1.3:

$$R^2 = \frac{Codev(\delta^X, \delta^Y)^2}{Dev(\delta^X)Dev(\delta^Y)},$$

$$AIC(d) = -2\sum_{h=1}^{H}\left[-\log\left(\hat{\sigma}_d\sqrt{2\pi}\right) - \frac{1}{2}\frac{\left(\delta_h^Y - c\delta_h^X\right)^2}{\hat{\sigma}_d^2}\right] + 2^*3,$$

$$BIC(d) = -2\sum_{h=1}^{H}\left[-\log\left(\hat{\sigma}_d\sqrt{2\pi}\right) - \frac{1}{2}\frac{\left(\delta_h^Y - c\delta_h^X\right)^2}{\hat{\sigma}_d^2}\right] + 3^*\log(H),$$

obtaining the results shown in Figure 5.4.

As the Figure 5.4(a) highlights, the coefficient of determination achieves the maximum value for $d = 2$ dimensions and then decreases as well as d increases, putting in evidence that it is able to reveal the true intrinsic data dimensionality. Moreover, the Akaike and Bayesian's Information Criteria (Figures 5.4(a) and 5.4(b), respectively) are minimized when the dimensionality is equal to 2 implying that also these criteria can faithfully reveal the real structure of the original data sets by detecting the minimal number of degrees of freedom needed to generate the original data spaces. A validation measure to the results provided by three proposed criteria is not needed in this case, since we deal with simulated data sets for which the true intrinsic data dimensionality is known. We conclude

[4] In this part we are not constraining the intrinsic dimensionality to 1, 2 or 3 for visualization purpose, but we are searching for the true 2-dimensionality data space.

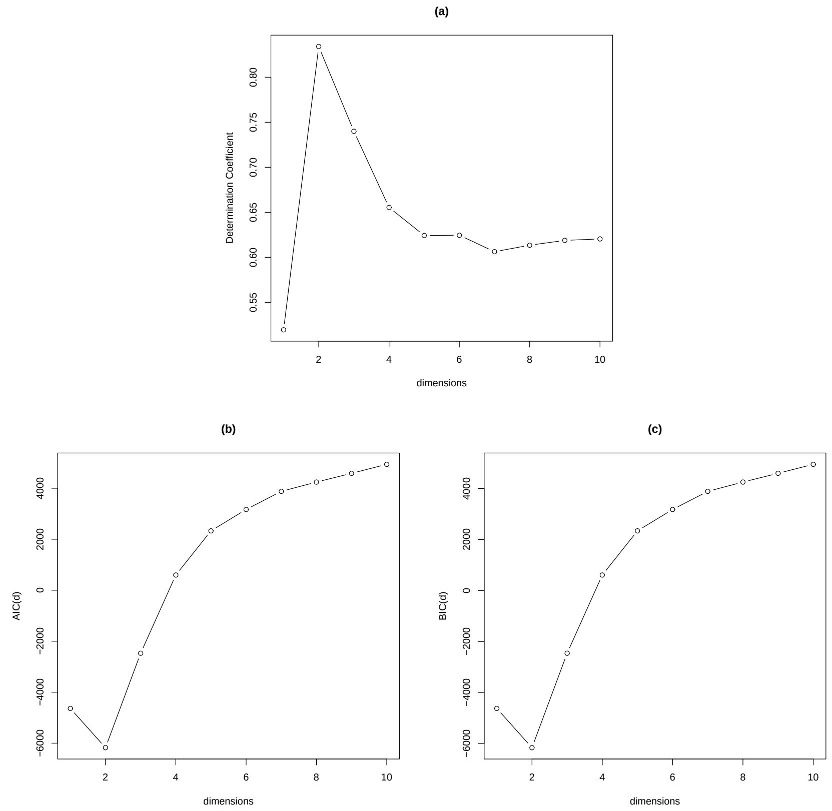

Figure 5.4. *The mean values of (a) R^2, (b) Akaike's Information Criterion and (c) Bayesian's Information Criterion over* 100 *generated data sets.*

that for simulated data, the proposed methods for finding the intrinsic data dimensionality are quite accurate.

Chapter 6

An Introduction to Microarray Data Analysis

In the last decade, molecular biology has seen the rise of a new technology known as DNA microarrays (simplified as *microarrays*) [28]. DNA microarrays involves monitoring the expression levels of thousands of genes simultaneously under a particular condition, called gene expression analysis.

This chapter provides an overview of DNA microarrays technology.

6.1 Nucleic Acids: DNA and RNA

Genes are specific sequences of DNA that determine, for instance, our eye color, hair color, height, *etc.* DNA is described as a double helix. It looks like a twisted long ladder. The sides of the 'ladder' are formed by a backbone of sugar and phosphate molecules, and the 'crosspieces' consist of two nucleotide bases joined weakly in the middle by hydrogen bonds. On either side of the 'rungs' lie complementary bases. Every Adenine base (A) is flanked by a Thymine (T) base, whereas every Guanine base (G) has a Cytosine partner (C) on the other side. Therefore, the strands of the helix are each other's complement. It is this basic chemical fact of complementarity that lies at the basis of each microarray. Microarrays have many single strands of a gene sequence segment attached to their surface, known as probes [48]. Ribonucleic Acid (RNA) delivers DNA's genetic message to the cytoplasm of a cell where proteins are made. Chemically speaking, RNA is similar to a single strand of DNA. The purpose of a microarray is to measure for gene in the genome the amount of message that was broadcast through the RNA. Roughly speaking, colour-labelled RNA is applied to the microarray, and if the RNA finds its complementary sibling on the array, then it naturally binds and sticks to the array. By measuring the amount of colour emitted by the array, one can get a sense of how much RNA was produced for each gene [48].

6.2 Microarrays technology

Microarrays technology has become one of the major tools that many researchers use to monitor genome wide expression levels of genes in a given organism. The goal of many microarray experiments is to identify the genes that are differentially transcribed with respect to different biological conditions of cell cultures or tissue samples.

A microarray is typically a glass (or some other material) slide on to which DNA molecules are fixed in an ordered manner at specific locations called spots (or features). A microarray may contain thousands of spots and each spot may contain a few milion copies of identical DNA molecules that uniquely correspond to a gene. The DNA in a spot may either be genomic DNA or short stretch of oligo-nucleotide strands that correspond to a gene. The spots are printed on to the glass slide by a robot or are synthesised by the process of photolithography [6].

Microarrays may be used to measure gene expression in many ways. One of the most popular application is to compare the expression of a set of genes from a cell mantained in a particular condition (condition A) to the same set of genes from a reference cell under a normal condition (condition B). First, RNA is extracted from the cell and then, RNA molecules in the extract are reverse transcribed into *c*DNA by using an enzyme reverse transcriptase and nucleotides labelled through the incorporation of radioactive markers, such as P, or of fluorescent dyes, such as phy-coerythrin, Cy3, or Cy5. For example, *c*DNA from cells grown in condition A may be labelled with a red dye and from cells grown in condition B with a green dye. Once the samples have been differentially labelled, they are allowed to hybridize onto the same glass slide. At this point any *c*DNA sequence in the sample will hybridize to specific spots on the glass slide containing its complementary sequence. The amount of *c*DNA bound to a spot will be directly proportional to the initial number of RNA molecules present for that gene in both samples [6].

Following the hybridization step, the spots in the hybridized microarray are excited by a laser and scanned at suitable wavelengths to detect the red and green dyes. The amount of fluorescence emitted upon excitation corresponds to the amount of bound nucleic acid. For istance, if *c*DNA from condition A for a particular gene was in greater abundance than that from condition B, one would find the spot to be red. If it was the other way, the spot would be green. If the gene was expressed to the same extent in both conditions, one would find the spot to be yellow, and if the gene was not expressed in both conditions, the spot would be black. Thus, what it seen at the end of the experiment stage is an image of the microarray, in which each spot that corresponds to a gene has an associated fluorescence value representing the relative expression level of that gene [6]. An example of the microarray technology is provided in Figure 6.1.

6.2.1 Image processing and analysis

In the previous section, we saw that the relative expression level for each gene (population of RNA in the two samples) can be stored as an image. The first step in the analysis of microarray data is to process this image. Most manufacturers of microarray scanners provide their own software; however, it is important to understand how data is actually being extracted from images, as this represents

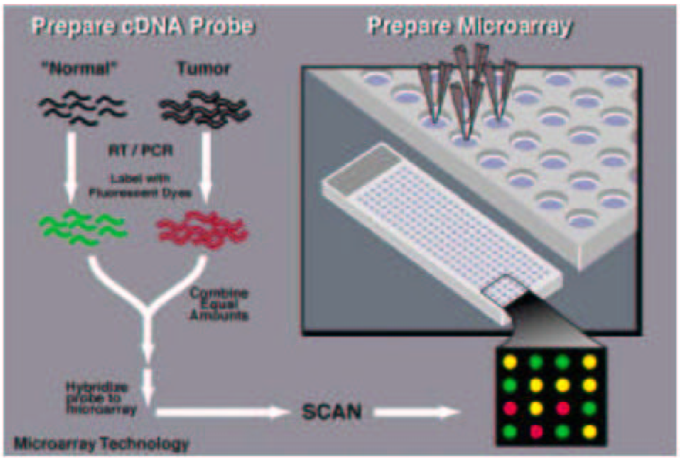

Figure 6.1. *Microarray technology process.*

the primary data collection step and forms the basis of any further analysis.

Image processing involves the following steps:

1. *Identification of the spots and distinguishing them from spurious signals.*

   The microarray is scanned following hybridization and an image file is normally generated. Once image generation is completed, the image is analyzed to identify spots. In the case of microarrays, the spots are arranged in an orderly manner into sub-arrays, which makes spot identification straightforward. Most image processing software requires the user to specify approximately where each sub-array lies and also additional parameters relevant to the spotted array. This information is then used to identify regions that correspond to spots.

2. *Determination of the spot area to be surveyed, determination of the local region to estimate background hybridization.*

   After identifying regions that correspond to sub-arrays, an area within the sub-array must be selected to get a measure of the spot signal and an estimate for background intensity. There are two methods to define the spot signal. The first method is to use an area of a fixed size that is centred on the centre of mass of the spot. This method has an advantage that it is computationally less expensive, but a disadvantage of being more error-prone in estimating spot intensity and background intensity. An alternative method is to precisely define the boundary for a spot and only include pixels within the boundary. This method has an advantage that it can give a better estimate of the spot intensity, but also has a disadvantage of being computationally intensive and time-consuming.

3. *Reporting summary statistics and assigning spot intensity after subtracting for background intensity.*

   Once the spot and background areas have been defined, a variety of summary statistics for each spot in each channel (red and green channels) are reported. Typically, each pixel within the area is taken into account, and the mean, median, and total values for the intensity considering all the pixels in the defined area are reported for both the spot and background. Most approaches use the spot median value, with the background median value subtracted from it, as the metric to represent spot intensity [6].

6.2.2 Expression ratios: the primary comparison

We saw that the relative expression level for a gene can be measured as the amount of red or green light emitted after excitation. The most common metric used to relate this information is called expression ratio. It is denoted as T_l and defined as:

$$T_l = \frac{R_l}{G_l},$$

where l denotes each gene on the array, R_l represents the spot intensity metric for the test sample and G_l represents the spot intensity metric for the reference sample. As mentioned above, the spot intensity metric for each gene can be represented as a total intensity value or a background subtracted median value. If we choose the median pixel value, then the median expression ratio for a given spot is:

$$T_{median} = \frac{R^{spot}_{median} - R^{background}_{median}}{G^{spot}_{median} - G^{background}_{median}}.$$

The expression ratio is a relevant way of representing expression differences in a very intuitive manner. For example, genes that do not differ in their expression level will have an expression ratio of 1 [6].

6.3 Data normalization

In the last section, it was shown that expression ratio is a reasonable measure to detect differentially expressed genes. However, when one compares the expression levels of genes that should not change in the two conditions, what one quite often finds is that an average expression ratio of such genes deviates from 1. This may be due to various reasons, for example, variation caused by differential labelling efficiency of the two fluorescent dyes or different amounts of starting mRNA material in the two samples. Thus, in the case of microarray experiments, as for any large-scale experiments, there are many sources of systematic variation that affect measurements of gene expression levels.

Normalization is a term that is used to describe the process of eliminating such variations to allow appropriate comparison of data obtained from the two samples. The first step in a normalization procedure is to choose a gene-set (which consists of genes for which expression levels should not change under the conditions studied, that is the expression ratio for all genes in the gene-set is expected to be 1). From that set, a normalization factor, which is a number that accounts for the variability seen in the gene-set, is calculated. It is then applied to the other genes in the microarray experiment. One should note that the normalization procedure changes the data, and is carried out only on the background corrected values for each spot [6].

6.4 Analysis of gene expression data

The processed data, after the normalization procedure, can then be represented in the form of a matrix, often called gene expression matrix. Each row in the matrix corresponds to a particular gene and each column could either correspond to an experimental condition or a specific time point at which expression of the genes has been measured. The expression levels for a gene across different experimental conditions are cumulatively called the “gene expression profile”, and the expression levels for all genes under an experimental condition are cumulatively called the “sample expression profile”.

Once we have obtained the gene expression matrix, additional levels of annotation can be added either to the gene or to the sample. Depending on whether the annotation is used or not, analysis of gene expression data can be classified into two different types, namely supervised or unsupervised learning. In the case of a supervised learning, we do use the annotation of either the gene or the sample, and create clusters of genes or samples in order to identify ”objects” with similar expression profiles. In the case of an unsupervised learning, the expression data is analyzed to identify patterns that can group genes or samples into clusters without the use of any form of annotation [6].

6.5 Relating expression data to other biological information

Gene expression profiles can be linked to external information to gain insight into biological processes and to make new discoveries.

6.5.1 Predicting binding sites

It is reasonable to assume that genes with similar expression profiles are regulated by the same set of transcription factors. If this happens to be the case, then genes that have similar expression profiles should have similar transcription factor binding sites upstream of the coding sequence in the DNA. Various research groups have exploited this assumption. The steps involved in such studies are the following:

1. Find a set of genes that have similar expression profiles.
2. Extract promoter sequences of the co-expressed genes.
3. Identify statistically over-represented sequence patterns.
4. Assess quality of the discovered pattern using statistical significance criteria.

6.5.2 Predicting protein interactions and protein functions

Integrating expression data with other external information, for example evolutionary conservation of proteins, have been used to predict interacting proteins, protein complexes, and protein function. The Works by Ge *et al.* (2001) and Jansen and Gerstein (2000) have shown that genes with similar expression profiles are more likely to encode proteins that interact. When this information is combined with evolutionary conservation of proteins, meaningful predictions can be made.

6.5.3 Predicting functionally conserved modules

Genes that have similar expression profiles often have related functions. Instead of studying co-expressed pairs of genes, one can view sets of co-expressed genes that are known to interact as a functional module involved in a particular biological process (Madan Babu*et al.*, 2004). This information, when integrated with the evolutionary conservation of proteins in more than two organisms, provides knowledge of the significance of the functional modules that have been conserved in evolution [6].

Chapter 7

Microarray Study

Due to the ultra high dimensionality nature of microarray data, data dimension reduction plays an important role for such type of data analysis. Dimensionality reduction of microarray data consists in reducing a $n \times D$ input matrix, where n represents the different experimental samples or patients, and D the number of genes, in a new matrix of size $n \times d$, with $d \ll D$, while striving to retain much of the initial information contained in the whole data set.

Dimensionality reduction of microarray data can be applied by two different and feasible approaches: (1) reducing the number of genes or dimensions in the data while maintaining the number of samples constant, or (2) reducing the number of samples while keeping the number of genes constant [28].

In this thesis we focus on the former approach and strive to reduce the dimensions of microarray data in order to obtain a smaller data set that is still representative of the original.

In this chapter we present the results of dimensionality reduction analysis over several public data sets: the mammary data set of Wit and McClure (2004), the lymphoma data set of Alizadeth *et al.* (2000), the leukemia data set of Golub *et al.* (1999). Detail descriptions of these data sets are provided in the Sections 7.1, 7.2 and 7.3 respectively.

7.1 Results on Mammary data set

7.1.1 The data

In order to find out the key genetic determinants associated with aggressive and non-aggressive breast cancer, the researcher John Bartlett of the University of Glasgow, investigated differences in the genomic DNA of breast cancer patients compares to that in controls.

In cancer cells, the genome can undergo change, such as obtaining additional copies of certain genes and losing genetic material from other genes. An increase in the gene number is known as *gene amplification*. When fewer copies are present compared to the genome of normal cells, this is known as *gene deletion*. In order to link gene amplification/deletion information to the aggressiveness of the tumours in the experiment, clinical information is available about each of the patients, such as their Nottingham prognostic index (NPI). This was used to classify the tumours into different severity groups while controlling for non-genomic influences. The work resulted in a sub-classification of breast cancer and it suggested genes that have an effect on the aggressiveness of the cancer. The result of the experiment are detailed in Witton *et al.* (2002).

Genomic DNA from cancer patients is extracted from stored frozen tumours tissue and from female reference DNA. The arrays contain 59 clones, each spotted three times. 57 genes are represented by these 59 clones, since two genes have both $5'$ and $3'$ versions included. In each of the two-channel arrays, reference female DNA is used as a control in one channel. The experiment involves the genetic material from 62 breast cancer patients. To measure the gene profile in all tumours, 62 arrays were used in the experiment. The amplification values are calculated by taking the ratio of tumours samples versus a reference sample.
For each patient, there is also a variety of clinical information available, including the following [48]:

- their survival times (in years) after the tissue was removed;
- Their age at diagnosis (in years);
- The size of their tumours (in mm);
- whether they died from breast cancer;
- whether they are still alive;
- the severity grade of their breast cancer: 1 (low) to 3 (high)
- their NPI score.

In this study we deal with a data matrix consisting of 59 genes whose amplification profiles on 62 patients are considered.

7.1.2 Selection of the number of neighbours

In order to test the proposed criteria for the automatic determination of the optimal number of nearest neighbours on the mammary cancer data we proceed by computing for each $k \in [1, 61]$ neighbours[1] the values of the adjusted coefficient of multiple determination[2], Akaike and Bayesian's Information Criteria as:

$$\bar{R}^2 = 1 - \left[\frac{\sum_{d=1}^{D} \left(x_{id} - X_d^{(i)} w_d^{(i)} \right)^2 / D}{\sum_{d=1}^{D} \left(x_{id} - \bar{x}_i \right)^2 / (D-1)} \right],$$

$$AIC(k) = nDlog \sum_{i=1}^{n} \sum_{d=1}^{D} \left(x_{id} - X_d^{(i)} w_d^{(i)} \right)^2 + 2(nk+1),$$

[1] Given the relative small number of samples, in this study we compute the three criteria for the choice of optimal number of k neighbours by considering every patients as possible neighbour at each time.

[2] The regularization term is applied for any k. In this way D are the resulting degrees of freedom of the residual sum of squares.

$$BIC(k) = nDlog \sum_{i=1}^{n} \sum_{d=1}^{D} \left(x_{id} - X_d^{(i)} w_d^{(i)} \right)^2 + (nk+1)log(n).$$

In this study we should apply the regularization term only when $k > D$. However, following this way, the results are unstable and inaccurate just when $k > D$. Thus, we propose to apply a small tolerance to any k in order to make a fair choice of the parameter. The regularization applied is equal to $\varepsilon = 10^{-6}$ even if the choice of the optimal number of neighbours does not change as ε varies between $\varepsilon = 10^{-4}$ up to $\varepsilon = 10^{-11}$. The results are presented in Figure 7.1.

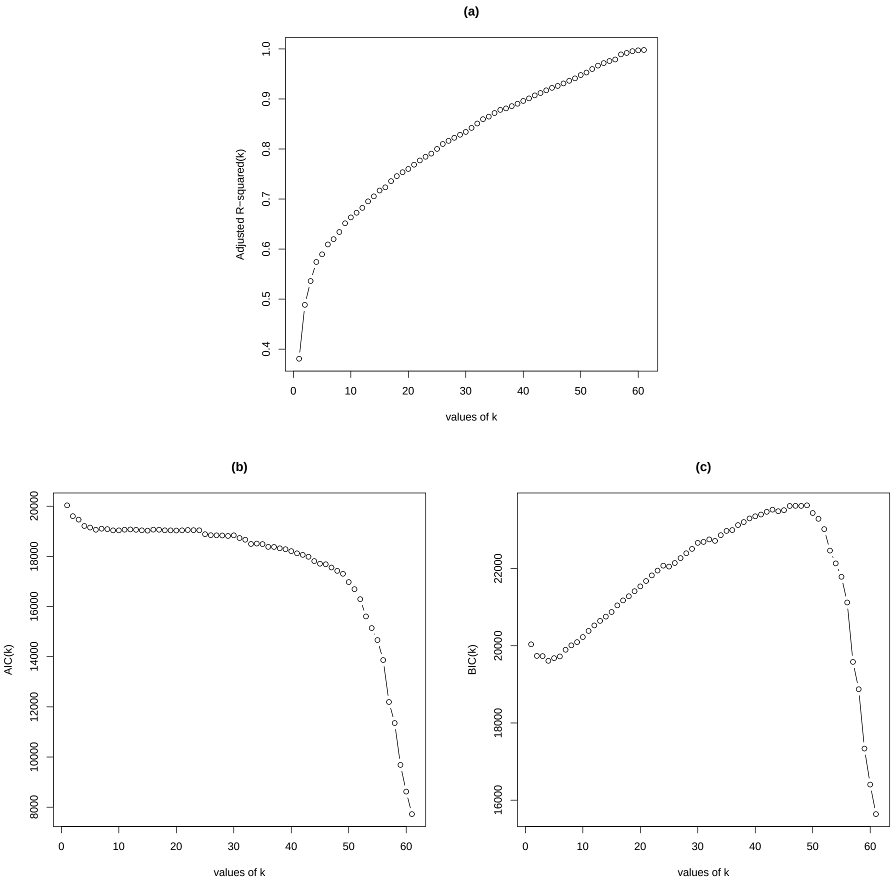

Figure 7.1. *(a) $\bar{R}^2$, (b) Akaike's Information Criterion and (c) Bayesian's Information Criterion for $k \in [1, 61]$ for the mammary cancer data.*

The Figure 7.1(a) evidences that $\bar{R}^2$ always increases as the number of neighbours grows, as the Akaike's Information Criterion (Fig. 7.1(b)) always decreases as k increases, detecting 61 neighbours as the optimal value of the model free parameter. The behaviour of BIC is quite dissimilar to the others. It initially decreases by achieving a local minimum for $k = 4$, then it grows until a value of $k = 50$ neighbours and finally it rapidly tends to its global minimum for $k = 61$ neighbours in accordance with the other criteria, revealing that the higher penalty term applied to BIC works more accurately than the others to penalize the error for the increase of k.

All the criteria applied to this data sets detect $k = 61$ neighbours as the optimal choice of the model free parameter. This suggests that all patients seem to be informative and important to reconstruct the initial data space. In other words, the amplification profile of a breast cancer tumour looks like similar for each patient considered.

7.1.3 Selection of the number of dimensions

Once the optimal number of nearest neighbour have been identified, the purpose of the study is, at this point, to conduct a dimensionality reduction of the mammary data set in order to obtain a compact representation of the original high-dimensional data.

We proceed by computing for each dimension $d \in [1, 59]$ the euclidean distance matrices D_X and D_Y of the input and output spaces, respectively, and consequently the H-dimensional vectors δ^X and δ^Y by extracting only those distances between nearby points and finally by obtaining for each dimension the values of R^2, AIC and BIC as:

$$R^2 = \frac{Codev(\delta^X, \delta^Y)^2}{Dev(\delta^X)Dev(\delta^Y)},$$

$$AIC(d) = -2\sum_{h=1}^{H}\left[-\log\left(\hat{\sigma}_d\sqrt{2\pi}\right) - \frac{1}{2}\frac{\left(\delta_h^Y - c\delta_h^X\right)^2}{\hat{\sigma}_d^2}\right] + 2^{*}3,$$

$$BIC(d) = -2\sum_{h=1}^{H}\left[-\log\left(\hat{\sigma}_d\sqrt{2\pi}\right) - \frac{1}{2}\frac{\left(\delta_h^Y - c\delta_h^X\right)^2}{\hat{\sigma}_d^2}\right] + 3^{*}\log(H).$$

The results for the three proposed criteria are shown in Figure 7.2[3].

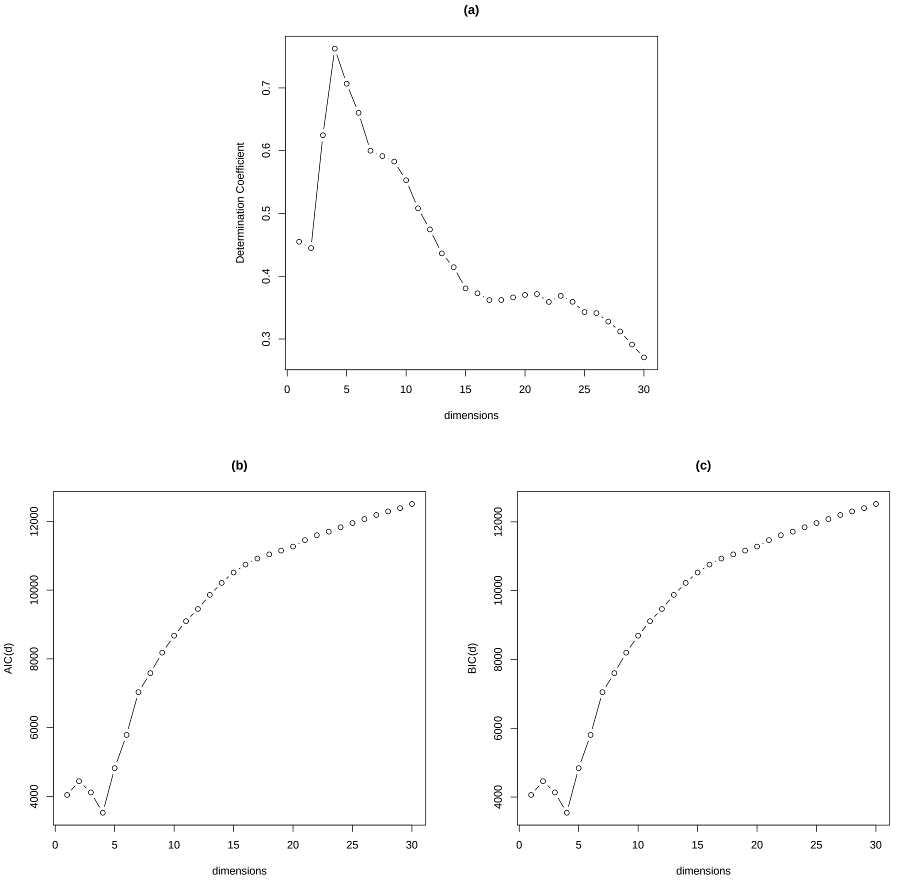

Figure 7.2. *(a) R^2, (b) Akaike's Information Criterion and (c) Bayesian's Information Criterion for $d \in [1, 30]$ for the mammary cancer data.*

The coefficient of determination (Fig. 7.2(a)) attains its maximum value for $d = 4$ dimensions. Same results are provided by AIC (Fig. 7.2(b)) and BIC

[3] The results are presented just for $d \in [1, 30]$ for a better visualization.

(Fig. 7.2(c)) criteria, in which the error is minimized when the data are represented by 4 dimensions. This suggest that the high-dimensional original data space can be faithfully described by considering just 4 coordinates, in other words, the information contained in 59 genes can be carefully represented by 4 metagenes.

7.2 Results on Lymphoma data set

7.2.1 The data

Alizadeth *et al.* (2000) reported a genome-wide gene expression profiling analysis for diffuse large B-cell lymphoma (DLBCL) in which a total of 96 normal and malignant lymphocites samples were profiled over 17.856 cDNA clones. Details can be found in Alizadeth *et al.* [5]. None of the patients included in the study has been treated before obtaining the biopsy samples. After biopsy, the patients were treated at two medical centers using comparable standard chemotherapy regimens. Among 42 patients, 40 of them had followup information, including 22 death with death time ranging from 1.3 to 71.9 months and 18 being still alive with the followup times ranging from 51.2 to 129.9 months. Alizadeth *et al.* first identified 4026 genes which showed large variations across all samples [36].

We deal with a data set consisting of 4026 genes over 62 samples.

7.2.2 Selection of the number of neighbours

By searching the optimal value of free parameter k, we compute for every value of possible k the three criteria in according with the formulas presented in Section 7.1.2, and we achieve the results shown in Figure 7.3.

Every techniques applied to the lymphoma data set identify as an optimal value of the parameter 61 neighbours. In fact, the adjusted coefficient of multiple determination (Fig. 7.3(a)) always increases as k grows, as well as AIC

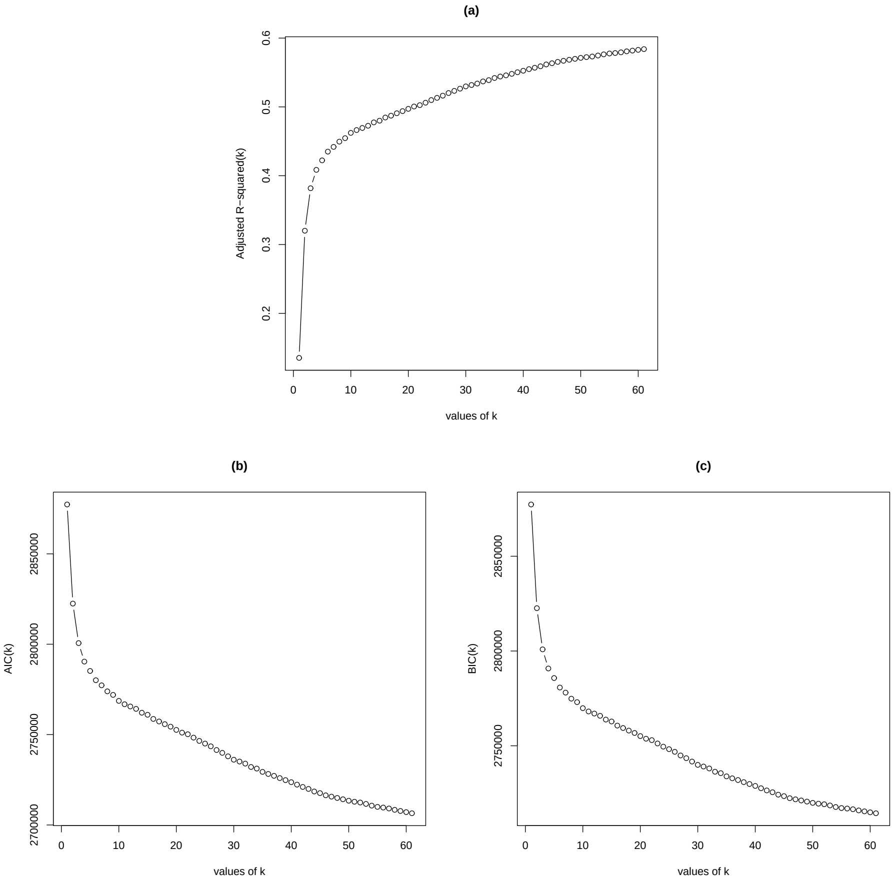

Figure 7.3. *(a) $\bar{R}^2$, (b) Akaike's Information Criterion and (c) Bayesian's Information Criterion for $k \in [1, 61]$ for the lymphoma data.*

(Fig. 7.3(b)) and BIC (Fig. 7.3(c)) criteria decrease as the number of neighbours increases and they attain the minimum error for $k = 61$ neighbours. Every indices computed, suggest to keep 61 neighbours as the parameter value to be used in LLE algorithm.

7.2.3 Selection of the number of dimensions

In order to reduce the high-dimensionality of the lymphoma data set and to make the data more tractable for subsequent operations we calculate the suggested techniques R^2, AIC and BIC in according with the formulas in Section 7.1.3

for several dimensions, achieving the plots presented in Figure 7.4.

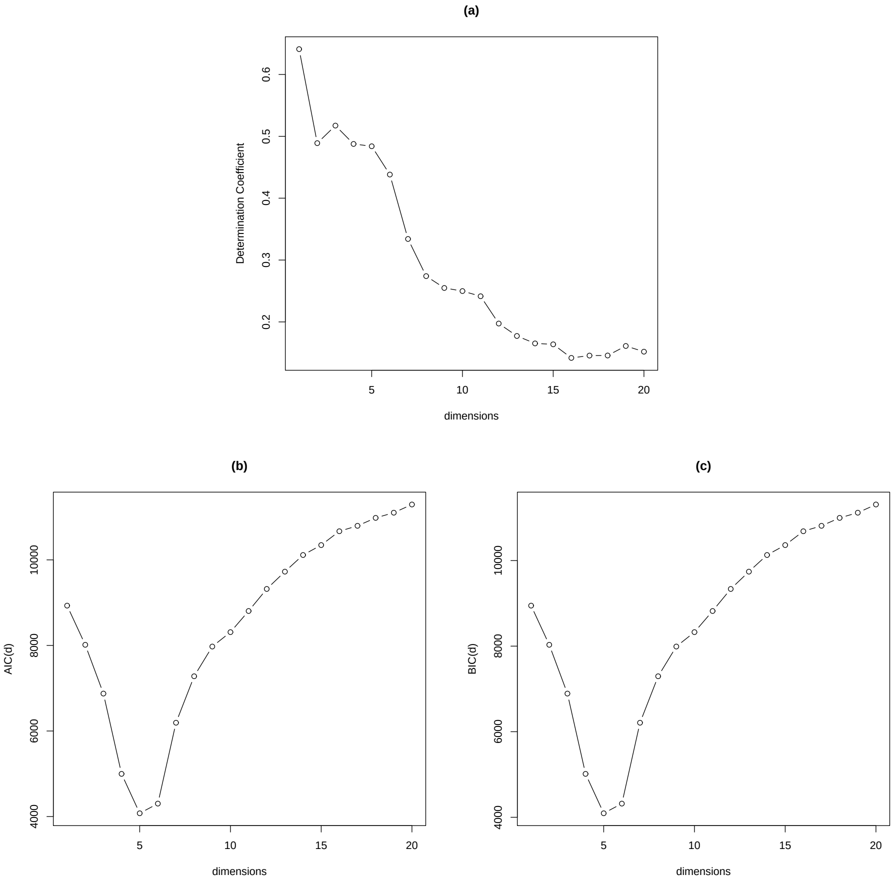

Figure 7.4. *(a) R^2, (b) Akaike's Information Criterion and (c) Bayesian's Information Criterion for $d \in [1, 20]$ for the lymphoma data.*

As is possible to note, the intrinsic dimensionality identified is not the same for the three computed criteria. In particular, for the coefficient of determination the best fit to the data occurs when the original data space is projected into just 1 dimension, or in other words, R^2 suggests that 4026 genes can be represented as a linear combination of 1 meta-gene. Otherwise, AIC and BIC criteria provide a more likely result for the low-dimensional space determination. Both techniques propose, in fact, a dimensionality reduction of the high-dimensional data space into a 5-dimensional space showing that the information contained in 4026 genes can be faithfully represented across 5 meta-genes. We are inclined to favor the results provided by AIC and BIC criteria because even if the coefficient of determination is a widely used measure to determine how well a regression fit is, it does not hold account of any penalty term to increasing of the dimensions. Thus, even if the coefficient of determination seems to work well when the number of dimensions is quite small as in the simulation study and still in the mammary data set, this results underestimated when the number of dimension of a data set is very large as for lymphoma data set in which we deal with 4026 dimensions.

7.3 Results on Leukemia data set

7.3.1 The data

Microarray data was obtained from patients having two types of leukemia, acute lymphoblastic leukemia (ALL) and acute myeloid leukemia (AML). The data was taken from bone marrow samples and the samples were of different cell types, for example B or T cells and different patients genders. Each sample was analyzed using an Affimetrix microarrays containing expression levels of 7129 genes. The data was divided into 38 training data points and 34 test points [16]. Details can be found in Golub *et al.* (1999) [19].

We deal with a data matrix of 2226 genes over 72 patients.

7.3.2 Selection of the number of neighbours

For the leukemia data set the computation of $\bar{R}^2$, AIC and BIC in according with the formulas described in Section 7.1.2 has provided the following results presented in Figure 7.5.

As Figure 7.5 evidences, all techniques applied, select $k = 71$ neighbours as the determination of the optimal number of nearest neighbours per data point needed to reconstruct the initial leukemia data space by minimizing the error.

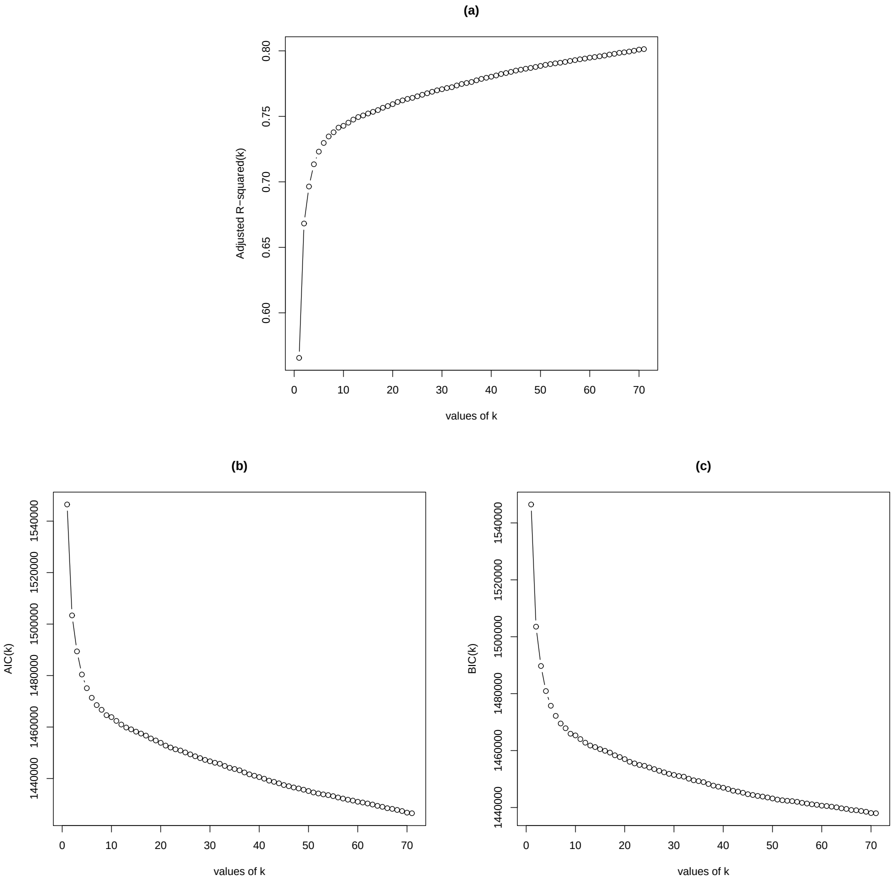

Figure 7.5. *(a) $\bar{R}^2$, (b) Akaike's Information Criterion and (c) Bayesian's Information Criterion for $k \in [1, 71]$ for the leukemia data.*

7.3.3 Selection of the number of dimensions

Once identified 71 neighbours as the optimal value of the model free parameter, we compute for several dimensions the R^2, AIC and BIC values following the formulas in Section 7.1.3 obtaining the results shown in Figure 7.6.

The determination of the optimal number of dimensions provided by three criteria is not the same. In fact, the coefficient of determination (Fig. 7.6(a)) identifies an optimal value of $d = 5$ dimensions, but, as explained in the previous section, when the number of variables in a data set is very large, R^2 represents a biased measure of the fit to the data since it does not make use of any penalty

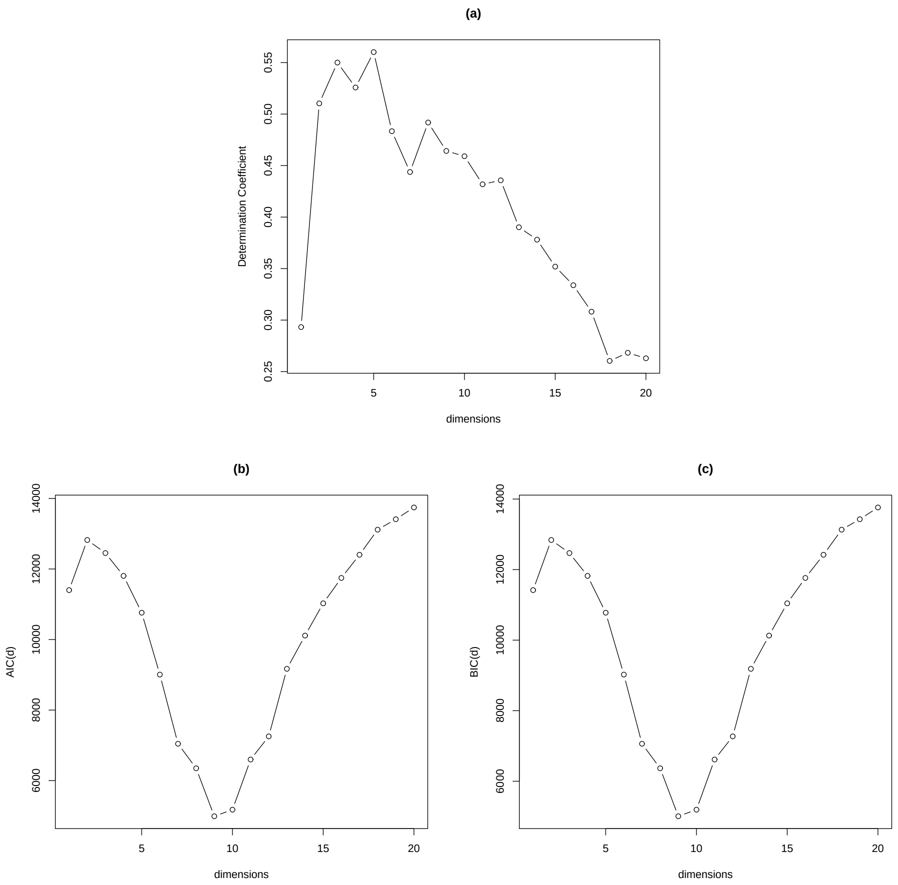

Figure 7.6. *(a)* R^2*, (b) Akaike's Information Criterion and (c) Bayesian's Information Criterion for* $d \in [1, 20]$ *for the leukemia data.*

term to increasing of dimensions. Instead, the other two examined criteria, AIC (Fig. 7.6(b)) and BIC (Fig. 7.6(c)) provide the same result for the choice of the model free parameter. In particular both identify as the intrinsic dimensionality of the leukemia data set $d = 9$ dimensions. For the penalty term included in these criteria, we are inclined to favour the result achieved by AIC and BIC techniques. The original high-dimensional leukemia data set can be represented by a lower 9-dimensional space, that is, 9 meta-genes can be faithfully represent the original 2226 genes.

We then looked at genes that strongest correlate with the 9 discovered embedded dimensions and we found that the $D = 2226$ genes most highly correlated with the new $d = 9$ meta-genes are the same genes identified by Golub *et al.*, 1999 [19], able to distinguish *ALL* from *AML* leukemia as represented in Figure 7.7.

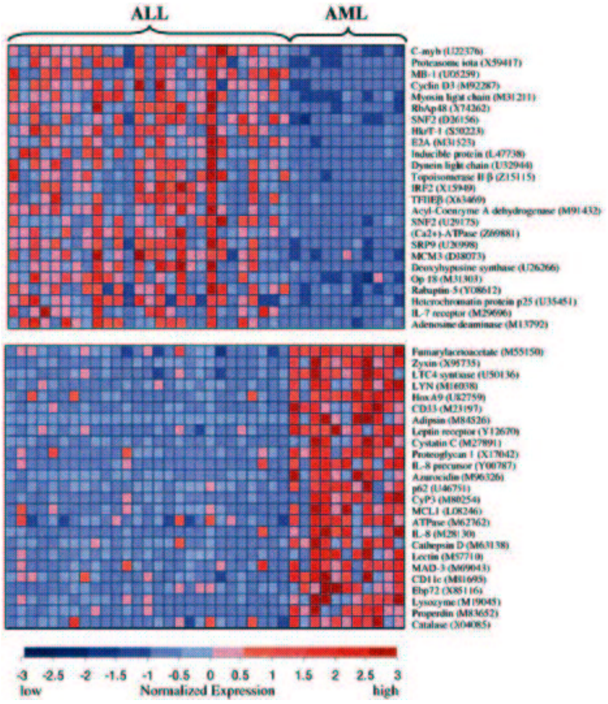

Figure 7.7. *The 50 genes most highly correlated with the ALL-AML class distinction. Each row corresponds to a gene, with the columns corresponding to expression levels in different samples. The top panel shows genes highly expressed in ALL, the bottom panel shows genes more highly expressed in AML.*

Chapter 8

Conclusion

Dimensionality reduction techniques have been used for visualization of high dimensionality data sets. The aim is to obtain a low-dimensional representation of high-dimensional data while preserving the most important characteristics of the data. Dimensionality reduction can be done either by feature extraction or by feature selection. The main advantage of feature extraction is that given the same number of reduced features, the transformed features can provide better results in further data analysis than the selected features [21].

In this thesis, an unsupervised non linear feature extraction technique called Locally Linear Embedding was considered. Its ability to deal with large size of high-dimensional data and its non-iterative way to find the embeddings make it more and more attractive to researchers. Another advantage of LLE comes from only two parameters to be set.

The purpose of the thesis was to develop a procedure for the automatic selection of the two free parameters of the model and to apply it to simulation studies and microarray data.

The low-dimensional representation obtained by LLE algorithm describes the true structure of the original data due to the properties of the reconstruction weights preserving information about the local neighborhoods of the data in the second step of the LLE algorithm. The natural question was how defining an optimal partition of the data manifolds into local patches or in other words how choosing the optimal number of k nearest neighbours able to reconstruct the original data space. In this thesis this problem is solved by proposing three criteria presented in Section 3.

The other free parameter of the LLE algorithm is the number of dimensions able to faithfully represent the high-dimensional data. It is clear that this parameter is set to be equal to two or three in case of visualization since human observer can perceive at most $3D$ space. In general when the aim of the dimensionality reduction technique is not confined to data visualization the dimensionality of the projected data can not be a priori fixed equal to two or three. Therefore one seeks to approximately estimate the intrinsic dimensionality of the data in order to preserve the most important information and reduce the influence of noise and outliers in the following steps of the data analysis. In this thesis some procedures for the automatic extraction of the embedded dimensions are proposed in Section 4.

The proposed criteria for the automatic selection of the two free model parameters have been applied to several simulation studies by generating 10 non linear dimensions as combination of two independent uniform variables. The three procedures for the choice of k neighbours seem to work well. However, by comparing the results with a validation measure as Procrustes Analysis we observed that $\bar{R}^2$ seems to provide a better result than Akaike and Bayesian's Information Criteria since the latter ones underestimate the true value of the parameter. Indeed, all the proposed criteria for the choice of the intrinsic dimensionality of the data seem to be able to reveal the real embedded structure of the high-dimensional data sets.

The finally purpose was to apply the procedures on several data sets arisen from microarray study. In fact, due to the ultra high-dimensionality nature of microarray data, data dimensionality reduction plays an important role for such type of data analysis. In this thesis we presented the results of dimensionality reduction over three different public data sets: the mammary data set of Wit and McClure (2004), the lymphoma data set of Alizadeth *et al.* (2000), the leukemia data set of Golub *et al.* (1999), characterized by different original high-dimensionality. The proposed criteria for the choice of k neighbours always select the maximum value for k, revealing that all patients seem to be informative and important to reconstruct the original data space. It is crucial to note that these results represent only a typical characteristic of the microarray data sets, where the number of genes is much greater than the experimental conditions. On the contrary, in the other data sets the proposed criteria for the choice of the number of neighbours have not provide similar results. As far as the extraction of the intrinsic dimensionality of microarray data sets we observed that the Akaike and Bayesian's Information Criteria seem to be able to reveal the real embedded structure of the original data space, while, on the contrary, the R^2 criterion seems to provide satisfactory results just when the dimensionality of the original space is quite small, while it appears underestimated for those data sets characterized by a original high-dimensionality.

Bibliography

[1] T.L. Ainsworth and J.S. Lee. Optimal polarimetric decomposition variables - nonlinear dimensionality reduction. *In Proc. of IEEE Int. Geoscience and Remote Sensing Symposium*, Sydney:928–930, 2001.

[2] H. Akaike. Use of an information theoretic quantity for statistical model identification. *In Proc. 5th Hawaii International Conference on System Sciences, Western Periodicals Co.*, pages 249–250, 1972.

[3] H. Akaike. A bayesian analysis of the minimum aic procedure. *Ann. Inst. Stats. Math.*, 30,A:9–14, 1978.

[4] H. Akaike. A new look at the statistical model identification. *IEEE Trans. Automat. Contr.*, AC-19:716-723,1974:Tokyo, 1981.

[5] A. Alizadeth, M. Eisen, and R. Davis. Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, 403:503–511, 2000.

[6] M. Madan Babu. An introduction to microarray data analysis. Chapter 11:225–249, 2004.

[7] X. Bai, B. Yin, Q. Shi, and Y. Sun. Face recognition based on supervised locally linear embedding method. *Journal of Information and Computational Science*, (4):641–646, 2005.

[8] C.M. Bishop, M. Svensen, and C.K.I. Williams. Gtm: The generative topographic mapping. *Neural Computation*, 10(1):215–234, 1998.

[9] R. Busa-Fekete and A. Kocsor. Locally linear embedding and its variant for feature extraction.

[10] S. Chao and C. Lihui. Feature dimension reduction for microarray data analysis using locally linear embedding. *School of EEE, Nanyang Technological University*, 2004.

[11] T.F. Cox and M.A.A. Cox. *Multidimensional Scaling.* Chapman and Hall, 1994.

[12] P. Craven and G. Wahba. Smoothing noisy data with spline function. *Numerische Matemathik*, 31:377–390, 1979.

[13] V. de Silva and J.B. Tenenbaum. Unsupervised learning of curved manifolds. citeseer.ist.psu.edu/603956.html.

[14] S. DeBacker, A. Naud, and P. Scheunders. Non-linear dimensionality reduction techniques for unsupervised feature extraction. *Pattern Recognition Letters*, 19(8):711–720, 1998.

[15] P. Demartines and J. Herault. Curvilinear component analysis: a self organizing neural network for nonlinear mapping of data sets. *IEEE Trans. on Neural Networks*, 8(1):148–154, 1998.

[16] J.M. Deutsch. Algorithm for finding optimal gene sets in microarray prediction. *http://stravinsky.ucsc.edu/josh/gesses/*, 2006.

[17] D. Donoho and C. Grimes. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proc. of National Academy of Science*, 100 (10):5591–5596, 2003.

[18] G.H. Golub, M. Heath, and G. Wahba. Generalized cross validation as a method for choosing a good ridge parameter. *Technometrics*, 21:215–223, 1979.

[19] T.R Golub, D.K. Slonim, and P. Tamayo. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999.

[20] A. Hadid, O. Kouropteva, and M. Pietikäinen. Unsupervised learning using locally linear embedding: experiments with face pose analysis. *In Proc. of the 16th Int. Conf. on Pattern Recognition*, Quebec, Canada, 2002.

[21] M. Hall. *Correlation-based feature selection machine learning.* PhD thesis, University of Waikato, New Zealand, 1998.

[22] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning.* Springer, Springer-Verlag New York, 175 Fifth Avenue, 2001.

[23] W. Huber, A. von Heydebreck, and M. Vingron. Analysis of microarray gene expression data. 2003.

[24] X. Huo and J. Chen. Local linear projection (llp). url: citeseer.ist.psu.edu/561947.html.

[25] V. Jain and L.K. Saul. Exploratory analysis and visualization of speech and music by locally linear embedding. *IEEE International Conference on Acoustic, Speech and Signal Processing*, 3:984–987, 2004.

[26] J.W. Sammon Jr. A nonlinear mapping for data structure analysis. *IEEE Trans. on Computers*, 18(5):401–409, 1969.

[27] A. Jain R. Dubes K. Pettis, T. Bailey. An intrinsic dimensionality estimator from nearneighbor information. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pages PAMI–1: 2537, 1979.

[28] R. Kharal. Semidefinite embedding for the dimensionality reduction of dna microarray data. Master's thesis, Master's Thesis of Mathematics in Computer Science. University of Waterloo, Ontario, Canada, 2006.

[29] T. Kohonen. *Self Organization and Associative memory.* Springer-Verlag, Springer-Verlag New York, 175 Fifth Avenue, 3rd edition 1989.

[30] O. Kouropteva. Unsupervised learning with locally linear embedding algorithm: an experimental study. Master's thesis, Master's Thesis. University of Joensuu, Finland, 2001.

[31] O. Kouropteva, O. Okun, A. Hadid, M. Soriano, S. Marcos, and M. Pietikäinen. Beyond locally linear embedding algorithm. Technical Report MVG-01-2002, University of Oulu, Finland, Department of Electrical and Information Engineering, 2002.

[32] O. Kouropteva, O. Okun, and M. Pietikänen. Selection of the optimal parameter value for the locally linear embedding algorithm. *Proc. of International Conference on Fuzzy System and Knowledge Discovery*, pages 359–363, 2002.

[33] O. Kouropteva, O. Okun, and M. Pietikänen. Classification of handwritten digits using supervised locally linear embedding algorithm and support vector machine. *Proc. of the Eleventh European Symposium on Artificial Neural Networks*, pages 229–234, 2003.

[34] O. Kayo Kouropteva. *Locally linear embedding algorithm. Extensions and applications.* PhD thesis, Faculty of technology, Department of electrical and information engineering, University of Oulu, 2006.

[35] J.A. Lee, A. Lendasse, N. Donckers, and M. Verleysen. A robust nonlinear projection method. *In Proc. of European Symposium on Artificial Neural Networks*, Bruges, Belgium:13–20, 2000.

[36] H. Li and Y. Luan. Kernel cox regression models for linking gene expression profiles to censored survival data.

[37] J. Mao and A.K. Jain. Artificial neural networks for feature extraction and multivariate data projection. *IEEE Trans. on Neural Networks*, 6(2):296–317, 1995.

[38] N. Mekuz, C. Bauckhage, and J.K. Tsotsos. Face recognition with weighted locally linear embedding. In *Proceedings of Canadian Conference on Computer and Robot Vision CRV 2005*, pages 290–296. IEEE Computer Society, 2005.

[39] A. Neumaier. Solving ill-conditioned and singular linear systems: A tutorial on regularization. *SIAM Review*, 40(3):636–666, 1998.

[40] M. Pillati and C. Viroli. Supervised locally linear embedding for classification:an application to gene expression data analysis. *Classification and data analysis 2005. Meeting of the classification and data analysis group of the Italian Statistical Society*, pages 147–150, 2005.

[41] M. Polito and P. Perona. Grouping and dimensionality reduction by locally linear embedding. *Advances in Neural Information Processing System*, volume 14, 2002.

[42] D. De Ridder, O. Kouropteva, O. Okun, M. Pietikainen, and R. Duin. Supervised locally linear embedding. *In Proc. Joint Int. Conf. ICANN/ICONIP, Lecture Notes in Computer Science*, 2714:333–341, 2003.

[43] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.

[44] L. K. Saul and S. T. Roweis. An introduction to locally linear embedding. 2002.

[45] L. K. Saul and S. T. Roweis. Think globally, fit locally: unsupervised learning of nonlinear manifolds. Technical Report MS CIS-02-18, University of Pennsylvania, Department of Computer and Information Science, 2002.

[46] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.

[47] F. Torre and M. Black. Robust parameterized component analysis: Theory and applications to 2d facial modeling. 2002.

[48] E. Wit and J. D. McClure. *Statistics for Microarrays: Design, Analysis and Inference*. John Wiley and Sons, Chichester, 2004.

[49] H. Yin. Visom-a novel method for multivariate data projection and structure visualization. *IEEE Trans. on Neural Networks*, 13(1):237–243, 2002.

[50] J. Zhang, H. Shen, and Z. Zhou. Unified locally linear embedding and linear discriminant analysis algorithm (ullelda) for face recognition. url:citeseer.ist.psu.edu/721023.html.