Functional Data Representation and Discrimination Employing Locally Linear Embedding

Genaro Daza Santacoloma

Universidad Nacional de Colombia
Faculty of Engineering and Architecture
Department of Electrical, Electronic and Computer Engineering
Manizales
2010

Functional Data Representation and Discrimination Employing Locally Linear Embedding

Genaro Daza Santacoloma

Thesis for the degree of
Ph. D. in Engineering — Automatics

Advisor
Prof. César Germán Castellanos Domínguez

Universidad Nacional de Colombia
Faculty of Engineering and Architecture
Department of Electrical, Electronic and Computer Engineering
Manizales
2010

Representación y discriminación de datos funcionales empleando inmersiones localmente lineales

Genaro Daza Santacoloma

Tesis para optar al título de
Ph. D. en Ingeniería — Automática

Director
Prof. César Germán Castellanos Domínguez

Universidad Nacional de Colombia
Facultad de Ingeniería y Arquitectura
Departamento de Ingeniería Eléctrica, Electrónica y Computación
Manizales
2010

Contents

List of Tables

List of Figures

Acknowledgments

Thanks God for all these things that You have helped me to achieve!.

I would like to express my gratitude to my advisor Prof. Germán Castellanos for his guide and orientation during this research. His suggestions were incentive for continuously improving this work, and hours of pleasant academic discussion. I thank him by the funds and computational equipment that he arranged for me.

I would like to thank Prof. Carlos D. Acosta, who helped me to resolve a lot of questions taking out time from his busy schedule. Without his support I would not have reached all the objectives of this work.

I would like to thank Prof. Jose Principe from University of Florida, who gave me the opportunity for developing a research internship at CNEL. I am very pleased for his advices and support. I learned there many things, and specially, I could strengthen my work. Academic discussions and seminars that have been occurred at CNEL were very helpful for me.

A very special thanks goes out to my friend Luis G. Sánchez. He gave me his full support, and I think that without his help the academic internship at University of Florida would not have been possible. Also I thank him by help me to resolve a lot of doubts about my research. I must also express my gratitude to Andrés Álvarez and Juliana Valencia, who always were on my side for supporting and discussing academic ideas about the project.

I would also like to thank my family (my father, my mother, my brother, aunts and uncles) for the support they provided me through my entire life. I want to thank Luisa for her love and encouragement, which made me life easier, inclusive whether I had to spend hours at lab or in front of a computer.

Thanks also goes out to the members of the research group Signal Processing and Recognition at Universidad Nacional de Colombia, specially to Lina and Julián, who always were with me. I would also like to thank my friends in the Computational NeuroEngineering Lab: Erion, Sohan, Jihye, Stefan, Vero, Alex, Iago, Abhishek, Memming, Lin, Austin and Shalom (thanks guys, I will always have fond memories!).

I recognize that this research would not have been possible without the financial assistance provided by the project *Representación y discriminación de datos funcionales empleando inmersiones localmente lineales* funded by Universidad Nacional de Colombia, and a Ph. D. scholarship funded by Colciencias.

Genaro Daza Santacoloma
July, 2010

Abstract

In this work, three specific improvements for the nonlinear dimensionality reduction technique called locally linear embedding (LLE) are proposed. Firstly, an objective way to choose the free parameters of the LLE algorithm is introduced, particularly, new methods for choosing the regularization parameter and the number of nearest neighbors are developed. This makes possible that low dimensional representations obtained by means of LLE to be consistent. Secondly, it is presented a new technique for nonlinear dimensionality reduction (NLDR) called correntropy locally linear embedding, that improves the performance of the LLE algorithm. This technique replaces the Euclidean distance as similarity measure by the correntropy similarity measure in the core of the NLDR algorithm, which is very useful when noisy functional data are employed, allowing to correctly determine the local neighborhoods, which are the basis for suitable embedding results. And finally, as third topic, it is formulated an improved version of the LLE technique, which allows us to construct a NLDR algorithm that preserves the local geometry of the data, and provides a supervised strategy during the embedding procedure, improving the visualization and/or classification results in comparison to conventional LLE or other topologically constrained NLDR techniques. Our approaches are experimentally assessed on artificial and real-world data sets, which allow us to visually and quantitatively confirm whether the embedding results were correctly calculated.

The conjunction of these advances conforms a method for training pattern recognition systems, which is a full automatized nonlinear dimensionality reduction method that allows to use of functional representations, to preserve the local relations among the high dimensional input data, and to provide a supervised scheme for the dimensionality reduction. In this sense, the proposed supervised NLDR technique is efficient and competitive, outperforming other similar methods. Besides, it shows the ability of computing low dimensional representations of several manifolds at the same time.

Resumen

En este trabajo, son presentadas tres mejoras específicas para la técnica de reducción de dimensión no lineal llamada Inmersión Localmente Lineal (*Locally Linear Embedding* – LLE). Primero, se expone una forma objetiva para escoger los parámetros libres del algoritmo LLE, particularmente, son desarrollados métodos nuevos para escoger el parámetro de regularización y el número de vecinos más cercanos. Esto hace posible que las representaciones de baja dimensión obtenidas por medio de LLE sean consistentes. Segundo, se presenta una nueva técnica para reducción de dimensión no lineal (*Nonlinear Dimensionality Reduction* – NLDR) llamada Inmersión Localmente Lineal con Correntropía, esto mejora el desempeño del algoritmo LLE. Esta técnica está en reemplazar la distancia Euclídea como medida de similitud por la medida de similitud dada por la correntropía, al interior del algoritmo de reducción de dimensión no lineal, lo cual es muy útil cuando son emplados datos funcionales ruidosos, permitiendo determinar correctamente las vecindades locales, que son las bases para obtener resultados de inmersión adecuados. Y finalmente, como tercer tópico, es formulada una versión mejorada de la técnica LLE, la cual nos permite construir un algoritmo de reducción de dimensión no lineal que preserva la geo-metría local de los datos y provee una estrategia supervisada durante el procedimiento de inmersión, mejorando los resultados de visualización y/o clasificación en comparación con LLE convencional u otras técnicas de reducción de dimensión no lineal restringidas topológicamente. Nuestras aproximaciones son evaluadas experimentalmente en conjuntos de datos artificiales y reales, lo cual permite confirmar visual y cuantitativamente si los resultados de la inmersión fueron calculados correctamente.

La unión de estos avances conforma un método para el entrenamiento de sistemas de reconocimiento de patrones, el cual es un método de reducción de dimensión no lineal totalmente automatizado que permite usar representaciones funcionales, preservar las relaciones locales entre los datos de entrada de alta dimensión, y proveer un esquema supervisado para la reducción de dimensión. En este sentido, la técnica de reducción de dimension no lineal supervisada es eficiente y competitiva, superando otros métodos semejantes. Además, esta técnica exhibe la habilidad de calcular representaciones de baja dimensión para múltiples variedades de manera simultánea.

Part I

Preliminary

Chapter 1

Introduction

Pattern recognition studies how an automatized system can watch the environment, learn to distinguish patterns, and make decisions. The identification, description, classification, visualization and clustering of patterns are important problems for engineering developments and scientific issues such as biology, medicine, economy, artificial vision, artificial intelligence, industrial production, etcetera [1].

In computational research on perceptual categorization, it is generally taken for granted that it is possible to obtain meaningful features, which will be either related to the raw sensory input or to some procedure for feature extraction once the signals are acquired, in both cases the amount of provided features can be intractable. In order to obtain more useful representations of the information in these features for subsequent operations such as classification or visualization, it is necessary to discover underlying structures behind the observed data. In this sense techniques for dimensionality reduction appears as solution to obtain more compact representations of the original data that capture the information necessary for suitable data description and higher-level decision making.

Indeed, to apply unsupervised or supervised classification techniques directly to these measured features is problematic due to the parameter estimation problems inherent in applying learning methods to high-dimensional data sets with a limited number of samples. Before such techniques can be applied with a reasonable hope of generalization, a small number of useful features will have to be extracted. That is, the dimensionality of the feature space will have to be reduced [2].

Some canonical forms of dimensionality reduction are principal component analysis (PCA), linear discriminant analysis (LDA), and multidimensional scaling (MDS), which are simple to implement, and their optimizations are well understood and not prone to local minima. These virtues account for the widespread use of PCA, LDA and MDS [3]. Nevertheless, these methods are inappropriate when working with non-linear structured data. To cope with this problem, a technique called Locally Linear Embedding (LLE) is proposed [4], which calculates a non-supervised embedding to a low dimensional space, such that closely points in the high dimensional space remain nearby and similarly co-located with respect to one another in the low dimensional space [3]. The following advantages of LLE are worthy of consideration: 1) preservation of local geometry of high dimensional data in the low dimensional space, 2) construction of a single global coordinate system in a low dimensional space, 3) optimization problem has analytic solution avoiding

local minima [5], and 4) only three parameters are needed to be set by user.

Particularly, the three parameters to be set by the user are the dimensionality of embedding space m, the regularization parameter α, and the number of nearest neighbors k for local analysis [2]. These parameters are highly important and they can not be ignored or poorly chosen, because they have a strong influence in the embedding results. Although several approaches have been proposed for choosing them, the embedding results achieved employing these techniques for free parameters tuning are not encouraging. Thus seems to arise the necessity of formulate a new methodology for an automatic and objective way of computing the free parameters of the LLE algorithm.

It is also well known that when the LLE algorithm is used for finding out an underlying structure, it is necessary to ensure that chosen neighborhoods appropriately represent the manifold, these neighborhoods actually must be well-sampled and lying in locally linear patches. The suitable selection of the neighborhoods fundamentally depends on the distance measure employed inside the algorithm and its associated parameters.

The LLE algorithm traditionally uses the Euclidean distance measure to determine neighbors within local patches on the manifold. While the Euclidean distance measure is appropriate for measuring the distance between objects characterized by discrete attributes (static features [6]), it is less appropriate for measuring functional data similarity [7]. Due to the increase of the computation capacities, functional data representation is today a very common way for characterizing the objects, because it captures much more information about the analyzed objects than the static feature representations.

Besides, functional data representations that are coming from biological and industrial systems are usually corrupted by artifacts or missing values, these observations are considered as outliers for LLE algorithm employing Euclidean distance, producing low-density and unconnected neighborhoods, which distorts the relations among neighbors and leads to unappropriated embeddings. Actually, the problem with using Euclidean distance is that artifacts in the observations get more importance and missing values are intractable. This reason makes necessary to suggest a measure for comparing functional data representations, that allow conforms suitable neighborhoods for LLE.

On the other hand, although LLE have shown to be an appropriate technique for NLDR, specially in visualization, it has some limitations when data proceeds from different manifolds or when data is divided into separated groups [8], which are common cases in pattern recognition. LLE does not consider class label information, which can be helpful for improving data representation on this kind of data.

Indeed, we are often interested in analyzing data which do not constitute just one manifold, but a single process, e.g. an electrocardiographic signal which is describing normal and pathological behaviors. In these cases, looking for a single unfolded manifold that represents the whole data set can be an unfeasible procedure. Then, it is required to extend the conventional LLE approach to deal with several manifolds (patterns), employing class labels as extra information to guide the procedure of dimensionality reduction allowing to figure out a suitable representation for each one of them.

For all these reasons, in this work, three specific improvements for the LLE algorithm are proposed. Firstly, an objective way to choose the free parameters of the LLE algorithm is presented, particularly, new methods for choosing the regularization parameter and the number of nearest neighbors are developed. This makes possible that low dimensional

representations obtained by means of LLE to be consistent.

Secondly, inspired by information-theoretic learning (ITL) [9], we propose a new technique for nonlinear dimensionality reduction (NLDR) called correntropy locally linear embedding (Correntropy-LLE), that improve the performance of the LLE algorithm, when functional data are employed allowing to correctly determine the neighborhoods, which are the basis of a suitable embedding.

Finally, we formulate an improved version of the LLE technique, which allows us to construct a NLDR algorithm that preserves the local geometry of the data, and provides a discriminative strategy during the embedding procedure, improving the visualization and/or classification results in comparison to conventional LLE or other topologically constrained NLDR techniques.

Chapter 2

Objectives

2.1 General Objective

Develop a method for training pattern recognition systems, which allows the use of functional data by employing Euclidean space embedding that preserves local relations among data and improves the performance in the classification. The improvement can be reached taking into account class label information during the embedding procedure, which provides a discriminative scheme to the learning strategy.

2.2 Specific Objectives

- Develop a methodology for an objective estimation of each one of the free parameters of the locally linear embedding algorithm, which allows to reduce the variability and uncertainty about the embedding results.
- Suggest a cost function for comparing functional data that can be employed by non-linear reduction algorithms, particularly for the locally linear embedding algorithm. The cost function must define suitable neighborhoods and improve the quality of the embedding.
- Construct a non-linear dimensionality reduction algorithm, which allows the use of functional data, preserves the local geometry of the data, and provides a discriminative strategy during the embedding procedure.

Part II

Theoretical framework

Chapter 3

Non-Linear Dimensionality Reduction

In many pattern recognition problems the characterization stage generates a large amount of data. There are several important reasons for reducing the feature space dimensionality, such as, improve the classification performance, diminish irrelevant or redundant information, find out underlying data structures, obtain a graphical data representation for visual analysis, etc. Dimensionality reduction techniques try to discover underlying structures in low dimensional spaces from data lying on high dimensional spaces. Conventional methods that are essentially linear include feature subset selection and linear mapping. Some methods are principal component analysis (PCA), linear discriminant analysis (LDA), and multidimensional scaling (MDS). Nevertheless, these methods are inappropriate when working with non-linear structured data.

Non-Linear Dimensionality Reduction (NLDR) is the search for intrinsically low dimensional structures embedded nonlinearly in high dimensional observations. The problem involves mapping high dimensional inputs into a low dimensional feature space with as many coordinates as observed modes of variability.

In the followings sections some of the most well known techniques for NLDR based-on distance or topology preservation are related. Besides, a discussion around each one of them are presented, this discussion clarifies concepts, highlights the advantages and warn us about potential problems of the NLDR techniques reviewed.

3.1 Isometric Feature Mapping

Isometric Feature Mapping (ISOMAP) is a NLDR technique build on classical MDS but seeks to preserve the intrinsic geometry of the data, as captured in the geodesic distances between all pairs of data points [10], that is, it preserves the global geometric properties of the manifold as characterized by the geodesic distances between faraway points [11]. The crux is estimating the geodesic distance between faraway points, given only input–space distances [10].

The ISOMAP procedure consists of three main steps, each of which might be carried out by more or less sophisticated techniques. ISOMAP assumes that distance between

points in observation space is an accurate measure of manifold distance only locally and must be integrated over paths on the manifold to obtain global distances. As preparation for computing manifold distances, a discrete representation of the manifold in the form of a topology-preserving network is constructed. Given this network representation, then compute the shortest–path distance between any two points in the network, which is a good approximation of the actual manifold distances. Finally, from these manifold distances, construct a global geometry-preserving map of the observations $\mathbf{Y}$ in low–dimensional Euclidean space, using multidimensional scaling. The algorithm is presented below (Algorithm 1).

Algorithm 1 – Isometric Feature Mapping

Require: Input data matrix $\mathbf{X}$, number of neighbor k or radius ϵ, dimensionality of the output space m.

1: Construct neighborhood graph. Define the graph G_x over all data points by connecting points i and j (as measured by Euclidean distance $d_x(i,j)$) if they are closer than a value ϵ, or if i is one of the k nearest neighbors of j. Set edge lengths equal to $d_x(i,j)$.

2: Compute shortest–paths. Initialize the graph distances $d_G(i,j)$ equal to $d_x(i,j)$ if i,j are linked by an edge, else $d_G(i,j) = \infty$. Repeat this process for all input samples, and replace all entries $d_G(i,j)$ by

$$\min\left\{d_G\left(i,j\right), d_G\left(i,\hat{k}\right) + d_G\left(\hat{k},j\right)\right\} \tag{3.1}$$

where $\hat{k}$ are intermediate points between i and j. The matrix of final values $D_G = \{d_G(i,j)\}$ will contain the shortest–path distances between all pair of points in G_x.

3: Construct m–dimensional embedding. Apply classical nonmetric MDS [12] to the matrix D_G. MDS finds a configuration of m–dimensional feature vectors, corresponding to the high–dimensional sample vectors of $\mathbf{X}$, that minimizes the stress function

$$S = \min_{d_G^{ij}} \sqrt{\frac{\sum_{i<j}\left(d_Y^{ij} - \hat{d}_G^{ij}\right)^2}{\sum_{i<j}\left(d_Y^{ij}\right)^2}} \tag{3.2}$$

where $d_Y^{ij} = \|\mathbf{y}_i - \mathbf{y}_j\|$ is the Euclidean distance between feature vectors i and j in the output space, and $\hat{d}_G^{ij}$ are some monotonic transformation of the graph distances d_G^{ij}.

3.1.1 Discussion about ISOMAP

– The embedding cost for ISOMAP is dominated by the (geodesic) distances between faraway inputs at the expense of distortions in the local geometry.

– This approach is capable of discovering the nonlinear degrees of freedom that underlie complex natural observations. It is guaranteed asymptotically to recover the

true dimensionality and geometric structure of a larger class of nonlinear manifolds. These are manifolds whose intrinsic geometry is that of a convex region of Euclidean space, but whose ambient geometry in the high–dimensional input space may be highly folded, twisted, or curved.

- For non–Euclidean manifolds such as a hemisphere or the surface of a doughnut, ISOMAP still produces a globally suitable low–dimensional Euclidean representation.

- ISOMAP will not be applicable to every data manifold. However, ISOMAP is appropriate for manifolds with no holes and no intrinsic curvature.

- In practice, for finite data sets $d_G(i, j)$ may fail to approximate the actual geodesic distance for a small fraction of points that are disconnected from the component of the neighborhood graph. These outliers are easily detected as having infinite graph distances from the majority of other points and can be deleted from further analysis.

- ISOMAP may be applied wherever nonlinear geometry complicates the use of PCA or MDS.

- The scale–invariant parameter k is typically easier to set than ϵ, but may yield misleading results when the local dimensionality varies across the data set. When available, additional constraints such as temporal ordering of observations may also help to determine neighbors.

- ISOMAP requieres a considerable amount of points for a suitable estimation of the geodesic distance, but at the same time if the number of samples is very huge the last step of the algorithm (classical MDS computation) can become intractable.

3.2 Locally Linear Embedding

Locally Linear Embedding (LLE) [4] is an unsupervised learning algorithm that attempts to compute a low–dimensional embedding with the property that nearby points in the high–dimensional space remain nearby and similarly co–located with respect to one another in the low–dimensional space. The embedding is optimized to preserve the local configurations of nearest neighbors [13]. Besides, LLE recovers global structure from locally linear fits.

Let $\mathbf{X}$ be the input data matrix of size $n \times p$, where the sample vectors $\mathbf{x}_i \in \mathbb{R}^p$, $i = 1, \ldots, n$ are given, assuming that data lies on or close to a non–linear manifold that is well–sampled. Besides, each point and its neighbors lie on a locally linear hpatch (or nearby). High–dimensional points (input data) can be approximated as linear combinations of their nearest neighbors [8] and then be mapped to lower dimensional space ($m, m \leq p$) which preserves data local geometry.

LLE algorithm has 3 steps: At the beginning, the k nearest neighbors per point are searched, as measured by Euclidean distance (see Figure 3.1).

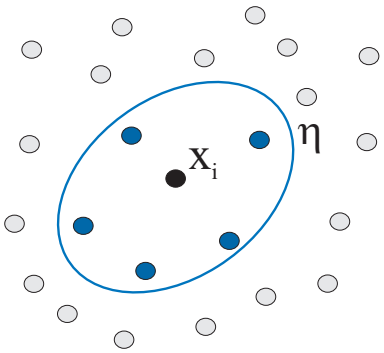

Figure 3.1: Example of neighborhood, $k = 5$

Secondly, each point is represented as a weighted linear combination of its neighbors, that is, we calculate weights $\mathbf{W}$ that minimize reconstruction error

$$\varepsilon(\mathbf{W}) = \sum_{i=1}^{n} \left\| \mathbf{x}_i - \sum_{j=1}^{n} w_{ij}\mathbf{x}_j \right\|^2, \tag{3.3}$$

subject to an sparseness constraint $w_{ij} = 0$ if $\mathbf{x}_j$ is not $k-$neighbor of $\mathbf{x}_i$, and an invariance constraint $\sum_{j=1}^{n} w_{ij} = 1$.

Now, considering a particular data point $\mathbf{x} \in \mathbb{R}^p$ and its k nearest neighbors $\boldsymbol{\eta}_j, j = 1, \ldots, k$. It is possible to rewrite (3.3) as

$$\varepsilon = \left\| \sum_{j=1}^{k} w_j \left(\mathbf{x} - \boldsymbol{\eta}_j\right) \right\|^2, \tag{3.4}$$

then

$$\varepsilon = \left\langle \sum_{j=1}^{k} w_j \left(\mathbf{x} - \boldsymbol{\eta}_j\right), \sum_{l=1}^{k} w_l \left(\mathbf{x} - \boldsymbol{\eta}_l\right) \right\rangle. \tag{3.5}$$

Let $\mathbf{G}$ the Gram matrix of size $k \times k$ with elements

$$G_{jl} = \left\langle \left(\mathbf{x} - \boldsymbol{\eta}_j\right), \left(\mathbf{x} - \boldsymbol{\eta}_l\right) \right\rangle, \tag{3.6}$$

then equation (3.5) can be written as

$$\varepsilon = \mathbf{w}^\top \mathbf{G}\mathbf{w} \quad \text{s.t.} \quad \sum_{j=1}^{k} w_j = 1. \tag{3.7}$$

Employing Lagrange theorem for minimizing (3.3), it is obtained that

$$2\mathbf{G}\mathbf{w} = \lambda \mathbf{1}, \tag{3.8}$$

where $\mathbf{1}$ is a vector of size $k \times 1$ (at least until something else is said), then

$$\mathbf{w} = \frac{\lambda}{2}\mathbf{G}^{-1}\mathbf{1}, \quad \text{being} \quad \lambda = \frac{2}{\mathbf{1}^\top \mathbf{G}^{-1}\mathbf{1}}. \tag{3.9}$$

In the third step, the input data are mapped to a low–dimensional space. Using $\mathbf{W}$, the low–dimensional output $\mathbf{Y}$ is found by minimizing (3.10)

$$\Phi(\mathbf{Y}) = \sum_{i=1}^{n} \left\| \mathbf{y}_i - \sum_{j=1}^{n} w_{ij}\mathbf{y}_j \right\|^2, \tag{3.10}$$

subject to $\sum_{i=1}^{n} \mathbf{y}_i = \mathbf{0}$ and $\sum_{i=1}^{n} \mathbf{y}_i\mathbf{y}_i^\top/n = \mathbf{I}_{m\times m}$, where $\mathbf{Y}$ is the output data $n \times m$ matrix (being $m \leq p$), and $\mathbf{y}_i \in \mathbb{R}^m$ is the output vector

$$\mathbf{y}_i = \begin{bmatrix} y_{i1} & y_{i2} & \cdots & y_{im} \end{bmatrix}. \tag{3.11}$$

By setting

$$\mathbf{M} = \left(\mathbf{I}_{n\times n} - \mathbf{W}^\top\right)\left(\mathbf{I}_{n\times n} - \mathbf{W}\right), \tag{3.12}$$

we rewrite (3.10) to find $\mathbf{Y}$ in the form

$$\Phi(\mathbf{Y}) = \operatorname{tr}\left(\mathbf{Y}^\top\mathbf{M}\mathbf{Y}\right) \quad \text{s.t.} \quad \begin{cases} \mathbf{1}_{1\times n}\mathbf{Y} = \mathbf{0}_{1\times n} \\ \frac{1}{n}\mathbf{Y}^\top\mathbf{Y} = \mathbf{I}_{m\times m} \end{cases} \tag{3.13}$$

It is possible to calculate $m+1$ eigenvectors of $\mathbf{M}$, which are associated to $m+1$ smallest eigenvalues. First eigenvector is the unit vector with all equal components, which is discarded. The remaining m eigenvectors constitute the m embedding coordinates found by LLE. A graphic representation of the whole LLE algorithm 2 can be shown in Figure 3.2.

Algorithm 2 – Locally Linear Embedding

Require: $\mathbf{X}$, k, m

1: Find the k nearest neighbors for each point $\mathbf{x}_i$, $i = 1, \ldots, n$.
2: Compute the weight matrix $\mathbf{W}$ by minimizing (3.7).
3: Compute the output matrix $\mathbf{Y}$ by minimizing (3.13), that is, to calculate the $m+1$ eigenvectors of $\mathbf{M}$ and then to discard the first one.

3.2.1 Generalization of LLE for New Points

LLE provides an embedding for the fixed set of training data to which the algorithm is applied. Often, however, it is necessary to generalize the results of the LLE algorithm to new locations in the input space. For example, suppose that it is desired to compute the output $\mathbf{y}_{new}$ corresponding to a new input $\mathbf{x}_{new}$. In principle, it is possible to rerun the entire LLE algorithm with the original data set augmented by the new input. For large data sets of high dimensionality, however, this approach is prohibitively expensive.

A realistic option is to compute the output $\mathbf{y_{new}}$ for a new input $\mathbf{x}_{new}$, by means of the following procedure: 1) identify the k nearest neighbors of $\mathbf{x}_{new}$ among the training inputs; 2) compute the linear weights $\mathbf{w}_j$ that best reconstruct $\mathbf{x}_{new}$ from its neighbors, subject to the sum-to-one constraint, $\sum_{j=1}^{n} \mathbf{w}_j = 1$; 3) output $\mathbf{y}_{new} = \sum_{j=1}^{n} \mathbf{w}_j\mathbf{y}_j$, where the sum is over the outputs corresponding to the neighbors of $\mathbf{x}_{new}$.

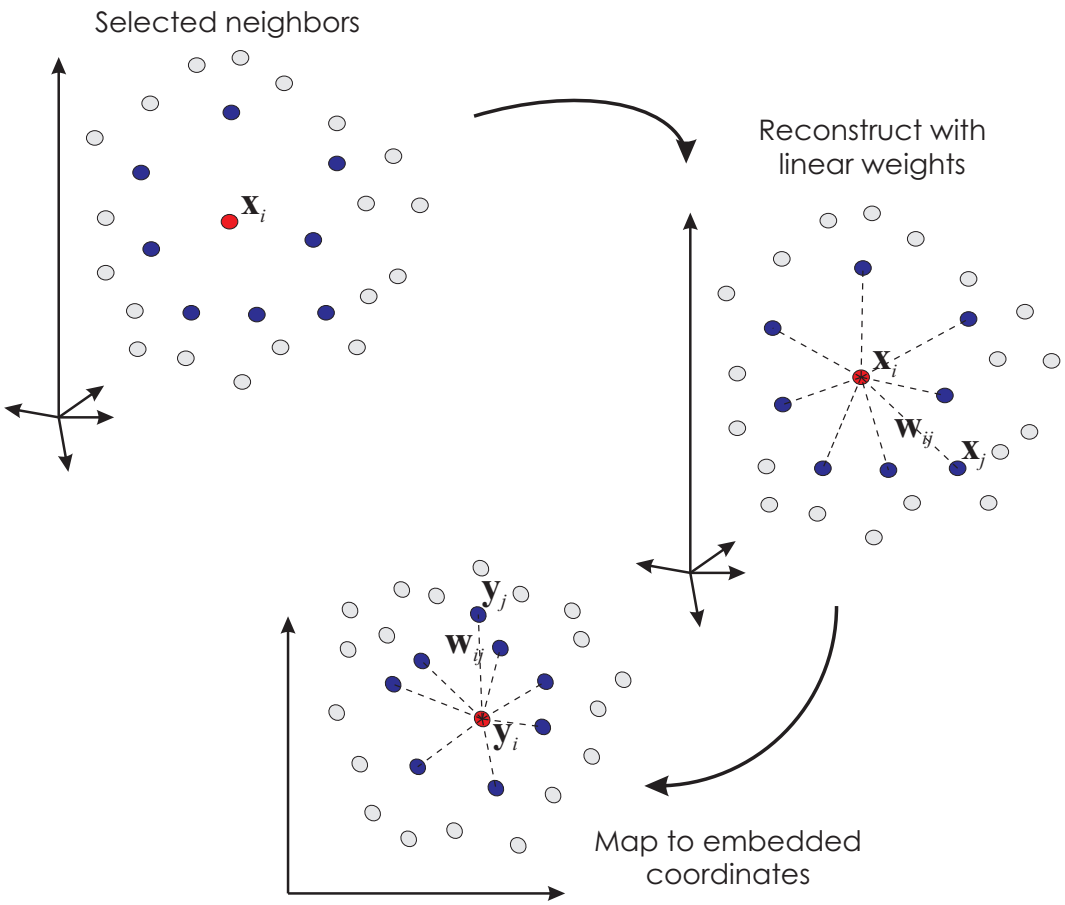

Figure 3.2: LLE algorithm

3.2.2 Discussion about LLE

- This method requires to manually set up three free parameters, the number of nearest neighbors k, the dimensionality of the embedding space m, and a regularization parameter.
- If k is set too small, the mapping will not reflect any global properties; if it is too high, the mapping will lose its nonlinear character and behave like traditional PCA [2]. The authors of LLE [3] suggest $k = 2m$, but this value for k can not be suitable for unfolding every kind of manifolds.
- The results of LLE are tipically stable over a range of neighborhood sizes. The size of that range depends on various features of the data, such as sampling density, the manifold geometry [3], and the noise in the signal.
- The algorithm can only be expected to recover embeddings whose dimensionality, m, is strictly less than the number of neighbors k.
- When $k > p$ some further regularization must be added to break the degeneracy, because the matrix $\mathbf{G}$ (3.6) does not have full rank. The choice of the regularization parameter plays an important role in the embedding results. What is more,

optimal values for this parameter can vary over a wide range; it depends on particular applications, which is partially explained by changes over the scale of the input data [2].

- One of the causes that make LLE fail is that the local geometry exploited by the reconstruction weights is not well–determined, since the constrained least square (LS) problem involved for determining the local weights may be ill–conditioned. A Tikhonov regularization is generally used for the ill–conditioned LS problem. However, a regularized solution may not be a good approximation to the exact solution if the regularization parameter in not suitably selected [14].

- To determine the neighborhood an Euclidean distance can be carried out. Other criteria, however, can also be used to choose the neighbors.

- The rows of the weight matrix $\mathbf{W}$ are constrained to sum one but may be either positive or negative. One can additionally constraint these weights to be non–negative, thus forcing the reconstruction of each data point to lie within the convex hull of its neighbors. The latter constraint tends to increase the robustness of linear fits to outliers. Nevertheless, it can degrade the reconstruction of data points that lie on the boundary of a manifold and outside the convex hull of their neighbors.

- The intrinsic dimensionality m can affect the mapping quality. If m is set too high, the mapping will enhance noise (due to the constraint $(1/n)\,\mathbf{Y}^\top\mathbf{Y} = \mathbf{I}$); if it is set too low, distinct parts of the data set might be mapped on top of each other.

- For estimating the dimensionality m of the output space it is possible to choose m by the number of eigenvalues comparable in magnitude to the smallest non–zero eigenvalues of the matrix $\mathbf{M}$. This procedure works only for contrived data. More generally, there is not evidence of the reliability of this procedure.

- One way to estimate the dimensionality of the embedding is assessing the eigenvalue spectra of local covariance matrices. Performing in essence a local PCA in the neighborhood of each data point, we can then ask whether these analyses yield a consistent estimate of the intrinsic dimensionality.

- Not all manifolds are suitable for LLE, even in the asymptotic limit of infinite data. Manifolds that do not admit a uniformly continuous mapping to the plane are very difficult to deal.

- Bootstrapping methods and self-consistency constraints can be used to improve the algorithm's performance on smaller data sets.

3.3 Locally Linear Embedding for Classification

The main capabilities of the LLE algorithm are not related to classification tasks, indeed LLE leads to poor classification performance when it is employed as a feature extractor [2]. There are possible reasons for this behavior. First, LLE fails when data is divided