NONLINEAR SUPERVISED DIMENSIONALITY REDUCTION VIA SMOOTH REGULAR EMBEDDINGS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

CEM ÖRNEK

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

FEBRUARY 2018

Approval of the thesis:

NONLINEAR SUPERVISED DIMENSIONALITY REDUCTION VIA SMOOTH REGULAR EMBEDDINGS

submitted by **CEM ÖRNEK** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Gülbin Dural Ünver
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Tolga Çiloğlu
Head of Department, **Electrical and Electronics Engineering** _____

Assist. Prof. Dr. Elif Vural
Supervisor, **Electrical and Electronics Engineering Dept., METU** _____

Examining Committee Members:

Prof. Dr. Abdullah Aydın Alatan
Electrical and Electronics Engineering Department, METU _____

Assist. Prof. Dr. Elif Vural
Electrical and Electronics Engineering Department, METU _____

Assoc. Prof. Dr. Fatih Kamışlı
Electrical and Electronics Engineering Department, METU _____

Assist. Prof. Dr. Sevinç Figen Öktem
Electrical and Electronics Engineering Department, METU _____

Assist. Prof. Dr. Metin Burak Altınoklu
Electrical and Electronics Engineering Department,
Konya Food and Agriculture University _____

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: CEM ÖRNEK

Signature :

ABSTRACT

NONLINEAR SUPERVISED DIMENSIONALITY REDUCTION VIA SMOOTH REGULAR EMBEDDINGS

ÖRNEK, Cem
M.S., Department of Electrical and Electronics Engineering
Supervisor : Assist. Prof. Dr. Elif Vural

February 2018, 74 pages

The recovery of the intrinsic geometric structures of data collections is an important problem in data analysis. Supervised extensions of several manifold learning approaches have been proposed in the recent years. Meanwhile, existing methods primarily focus on the embedding of the training data, and the generalization of the embedding to initially unseen test data is rather ignored. In this work, we build on recent theoretical results on the generalization performance of supervised manifold learning algorithms. Motivated by these performance bounds, we propose a supervised manifold learning method that computes a nonlinear embedding while constructing a smooth and regular interpolation function that extends the embedding to the whole data space in order to achieve satisfactory generalization. The embedding and the interpolator are jointly learnt such that the Lipschitz regularity of the interpolator is imposed while ensuring the separation between different classes. Experimental results on several image data sets show that the proposed method yields quite satisfactory performance in comparison with other supervised dimensionality reduction algorithms and traditional classifiers.

Keywords: Manifold learning, dimensionality reduction, supervised learning, out-of-sample, nonlinear embeddings

ÖZ

YUMUŞAK DÜZENLİ GÖMME İLE DOĞRUSAL OLMAYAN DENETİMLİ BOYUT DÜŞÜRME

ÖRNEK, Cem
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi : Assist. Prof. Dr. Elif Vural

Şubat 2018 , 74 sayfa

Veri topluluklarının esas geometrik yapısını çıkarmak, veri analizinde önemli bir problemdir. Birçok manifold öğrenme yaklaşımlarının denetimli açılımları son yıllarda öne sürülmüştür. Aynı zamanda, var olan metotlar öncelikle eğitim verisinin gömülmesine odaklanmıştır, ve başta elde bulunmayan test verisinin gömülmesiyle ilgilenmemişlerdir. Bu tezde, denetimli manifold öğrenme algoritmalarının genelleme performansı üzerine son zamanlarda elde edilmiş teorik sonuçlara dayanan bir çalışma yapılmıştır. Bu performans sınırlarından hareket ederek, test verilerine başarılı bir şekilde genelleme sağlayabilmek için gömmeyi bütün veri uzayına genişleten yavaş değişime sahip bir interpolasyon fonksiyonu ile, doğrusal olmayan bir gömme hesaplayan denetimli bir manifold öğrenme metodu öne sürülmüştür. Gömme ve interpolasyon fonksiyonu, hem farklı sınıflar arası ayrımı, hem de interpolasyon fonksiyonunun Lipschitz düzenliliğini sağlayacak şekilde öğrenilmiştir. Çeşitli görüntü kümelerinde yapılan deney sonuçları öne sürülen yöntemin diğer denetimli boyut düşürme algoritmaları ve geleneksel sınıflandırıcılara kıyasla oldukça tatminkar bir performans sağladığını göstermiştir.

Anahtar Kelimeler: Manifold öğrenme, boyut düşürme, denetimli öğrenme, örneklem dışılık, doğrusal olmayan gömme

This thesis is dedicated to my family.

ACKNOWLEDGMENTS

I would like to present my sincere gratitude to my thesis advisor Assist. Prof. Elif VURAL for her valuable guidance, advices and support throughout this study.

I would also like to thank my company ASELSAN Inc. for the possibilities that have provided to me.

Finally, I am grateful to my family for their love, trust and support throughout my life.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

1.1 Motivation

Machine learning is a field that applies mathematical and statistical methods to existing data in order to learn models. In the last ten years, machine learning had interesting applications in image recognition, self-driving vehicles, voice recognition, effective web browsing and human genome research. Machine learning algorithms are applied to intelligent robots, text processing, computer vision, medical informatics, voice processing, image processing and data mining.

Machine learning is comprised of algorithms that teach computers to perform tasks that human beings often do naturally on a daily basis. Machine learning algorithms are expected to learn from available data and produce reliable results.

With the recent advances, machine learning has developed drastically in the past few years. However, performance of machine learning is still lower than the human performance in many applications. So, new algorithms are being developed everyday in order to analyze bigger data, perform more complex computations and give more accurate and faster results.

Thanks to the power of machine learning, many industries enjoy more efficient operation, lower-cost and more powerful computational processing, and savings in time, money and data memory storage.

Classification is an important problem, where the purpose is to assign a category or a class to each observation. Applications of classification have a broad fan. Some

examples are listed below.

- Image classification implementations such as object recognition, face recognition and handwritten digit recognition.
- Shape detection, e.g. as in a system where a user draws a shape on a touch screen and the system determines which shape the user draws.
- Sentiment analysis, e.g. classification of comments made about you on the internet as positive, negative or neutral.
- Video classification where a category is assigned to a video. For example, videos can be categorized whether they are proper for children or not.
- Voice classification, e.g., the identification of the person the voice belongs to.
- Spam detection in an inbox, e.g., deciding whether an e-mail is a spam or not.
- Internet traffic interception, e.g. the determination of whether the content of a webpage is legal or not.
- Medical diagnosis, e.g. the detection of diseased cells or diseased organs in medical sector thanks to wearable devices or sensors.
- Fraud detection and prevention in financial industry.
- Financial analysis, e.g., learning from current and past price behaviours of a product in order to decide whether the product should be sold, bought or held.

In addition to these, classification methods have many other application areas as well.

Most classification algorithms involve the reduction of the data dimensionality for the following reasons:

- In many data analysis applications, collections of data are acquired in a high dimensional ambient space; however, the intrinsic dimensionality of the data is much lower. Dimensionality is related to the minimum number of features needed to describe any data sample in its space. If we wish to represent a point in a real line, only a number will be enough to describe it; therefore, the

dimensionality is one. If we wish to represent a point in a planar surface, we need two coordinates; therefore the dimensionality is two.

- If the count of features is much higher compared to the number of classes, the problem called the "curse of dimensionality" occurs. This idiom was put into words by Bellman in 1961 and can be explained as follows. For example, if each data sample has one dimension and there are 3 different classes, the space in which the one dimensional data samples lie is divided into three regions. The amount of data in each region is measured. If each data sample has two dimension and there are 3 different classes, the two dimensional space is divided into 9 regions and again we consider the amount of data in each region. If each data sample has three dimensions and there are 3 different classes, the three dimensional space is divided into 27 regions. If the amount of data is small and the dimension is high, most of the regions cannot have any data and no inference can be done in these regions. This situation is called the “curse of dimensionality”.

- Some features may be irrelevant or redundant and it is not desired that these irrelevant features affect the designed model.

- A large dimensionality increases the complexity of the learning.

Due to these reasons, many learning algorithms apply dimensionality reduction, i.e., reduce the number of features, before learning a model. However, reducing the number of features cannot be done arbitrarily or useful information may be lost. The important thing is to eliminate unnecessary features while preserving the amount of useful information.

In the recent years, many research effort has focalized on the reduction of the dimensionality of data. Dimensionaliy reduction in a manner consistent with the intrinsic geometry of data is also called manifold learning.

A "manifold" can be intuitively described as a surface of any shape in layman’s terms. The manifold concept is a generalization of surfaces in D-dimensional space. For example, a 0-dimensional manifold is a point, a 1-dimensional manifold is a curve, a 2-dimensional manifold is a surface and similarly there are manifolds of much higher

dimensions too. Manifold learning moves with the opinion that the dataset lies along a low-dimensional manifold embedded in a high-dimensional space. The objective is to uncover the low-dimensional manifold structure in a high-dimensional data set. The low-dimensional space reflects the underlying parameters and the high-dimensional space is the feature space. For example, a single camera takes a video stream of an object from varying view angles. The images have many pixels. For instance, if an image is 100×100, there are 10000 pixels. When the images are vectorized, each data sample or image vector lies in R^{10000}. However, the camera has only two degrees of freedom : tilting the x axis, and the y axis. So the images at consecutive frames are very similar and their difference can be described by the 2-degrees-of-freedom motion of camera. In other words, the images lie on a 2-dimensional surface embedded in a 10000-dimensional space. In such a setting, there are too many unnecessary features that might reduce the speed of the system.

At first, unsupervised manifold learning methods that aim to preserve and capture the geometric structure of the data set have been developed. Unsupervised learning methods do not exploit the class label information. After that, supervised manifold learning methods have been developed, which aim to enhance the separation between training samples from different classes by taking advantage of the class label information while respecting the geometric structure of data.

The intention of linear dimensionality reduction methods is to compute a low dimensional linear mapping of the originally high dimensional data $x \in \mathbb{R}^d$, where x is a data specimen residing in a high-dimensional space. Then, in order to project x into an r-dimensional space, linear dimensionality reduction methods find a linear projection matrix T that is a $d \times r$ matrix. Linear dimensionality reduction methods have different ways to find the linear projection matrix. Consequently, the lower dimensional representation of x is found as follows:

$$y = T^T x \tag{1.1}$$

However, linear dimensionality reduction methods tend to be unsuccessful to perceive the nonlinear structure in the data. Figure 1.1 [1] provides an example to demonstrate this. The projections given by PCA, LLE [30] and IsoMap [34] are compared in

this figure. PCA is a popular linear dimensionality reduction technique. It measures the similarity between data samples by the Euclidean distance. However, Euclidean distance sometimes causes PCA algorithm to fail in manifolds like in Figure 1.1. In particular, in a classification problem manifold-modeled data, the Euclidean distance of samples from different classes may be small. LLE and IsoMap are popular non-linear dimensionality reduction techniques. As it is seen in the figure, LLE and IsoMap can successfully unroll the original manifold. In addition, IsoMap gets rid of the disadvantage of the Euclidean distance by using the geodesic distance.

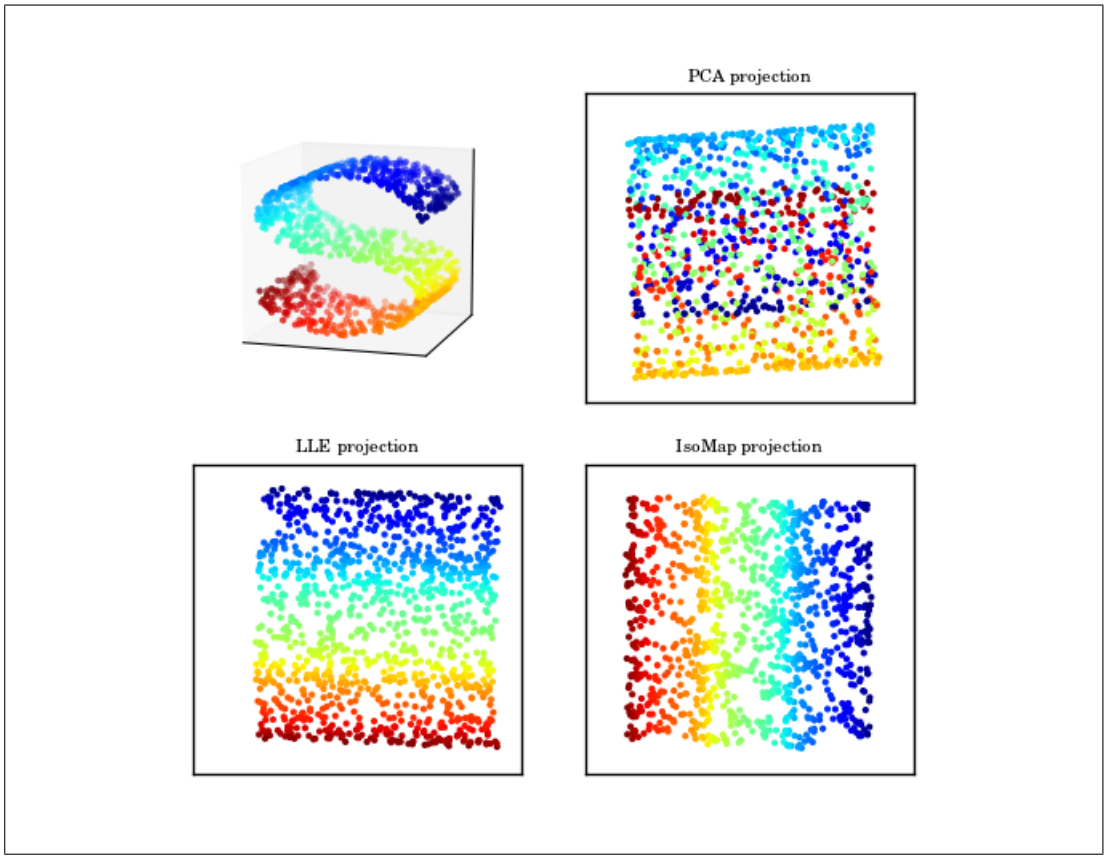


Figure 1.1: Dimension Reduction of Swiss Roll Data by PCA, ISOMAP and LLE [1]

Fisher Discriminant Analysis (FDA) and many similar techniques that aim to maximize the between-class scatter and minimize the within-class scatter show poor performance for intricate data distributions. Another disadvantage of this kind of linear methods is that the embedding dimension is limited to the number of classes. This situation is caused from the rank deficiency of the between class scatter matrix.

Nonlinear dimensionality reduction methods such as [29] have greater flexibility in the learnt representation. However, two critical issues arise concerning supervised dimensionality reduction methods: First, most nonlinear methods compute a point-wise mapping only for the initially available data samples. In order to generalize them to initially unavailable points, an interpolation needs to be done, which is called the out-of-sample extension of the embedding. Second, existing dimensionality reduction methods focus on the properties of the computed embedding only as far as the training samples are concerned: Existing algorithms mostly aim to increase the between-class separation and preserve the local structure, however, only for the training data. Meanwhile, the important question is how well these algorithms generalize to test data. This question is even more critical for nonlinear dimensionality reduction methods, as the classification performance obtained on test data will not only depend on the properties of the embedding of the training data, but also on the properties of the interpolator used for extending the embedding to the whole ambient space. Several methods have been suggested to solve the out-of-sample extension problem, such as unsupervised generalizations with smooth functions [11], [28], [13], [26] or semi-supervised interpolators [38]. These methods intend to generalize an already computed embedding to new data and are constrained by the initially prescribed coordinates for training data. Meanwhile, the best strategy for achieving satisfactory generalization to test data would not consist in learning the embedding and the interpolation sequentially, but rather in learning them in a joint and coherent manner.

In this thesis, we have developed a nonlinear supervised manifold learning method for classification where the embeddings of training data are learned and optimized in a joint way along with the interpolator that extends the embedding to the whole ambient space. A distinctive property of our method is the fact that it explicitly aims to have good generalization to test data in the learning objective. In order to achieve this, we build on the previous work [37] where a theoretical analysis of supervised manifold learning is proposed. The theoretical results in [37] show that for good classification performance, the separation between different classes in the embedding of training data needs to be sufficiently high, while at the same time the interpolation function that extends the embedding to test data must be sufficiently regular. For good generalization to initially unavailable test samples, a compromise needs to be found

between these two important criteria.

In this work, we adopt radial basis function interpolators for the generalization of the embedding, and learn the embedding of the training data and the parameters of the interpolator, i.e., the coefficients and the scale parameter of the interpolation function, at the same time with a joint optimization algorithm. The analysis in [37] characterizes the regularity of an interpolator via its Lipschitz regularity. We first derive an upper bound on the Lipschitz constant of the interpolator in terms of the parameters of the embedding. Then, relying on the theoretical analysis in [37], we propose to optimize an objective function that maximizes the separation between different classes and preserves the local geometry of training samples, while at the same time minimizing an upper bound on the Lipschitz constant of the RBF interpolator. We propose an alternating iterative optimization scheme that first updates the embedding coordinates, and then the interpolator parameters in each iteration. We test the classification achievement of the proposed method on several real data sets and show that it outperforms the supervised manifold learning methods in comparison and traditional classifiers.

1.2 Thesis Outline

The thesis is organized as follows. In Chapter 2, some unsupervised and supervised learning algorithms we have obtained through literature review are summarized. In Chapter 3, we formulate the supervised manifold learning problem and present the proposed algorithm. In Chapter 4, we evaluate our method with experiments on several face and object data sets. Finally, we conclude in Chapter 5.

CHAPTER 2

RELATED WORK

In this chapter we present an overview of the literature on manifold learning. In Section 2.1, we discuss unsupervised manifold learning methods. Next, in Section 2.2, supervised manifold learning methods are overviewed.

2.1 Unsupervised Manifold Learning Methods

Unsupervised manifold learning is a learning process which is performed by unlabeled data. We only have input data, and there is no corresponding true output variables, namely class labels are not available. Hence, unsupervised manifold learning algorithms try to explore the unknown structure of the dataset without the class information of the samples.

In this section, some well-known linear and nonlinear unsupervised manifold learning techniques will be overviewed. Principal Component Analysis and Locality Preserving Projection [20] are popular linear unsupervised dimensionality reduction techniques. Laplacian Eigenmaps [9], Locally Linear Embedding, Isomap [34] are among the well-known nonlinear dimensionality reduction methods.

When we discuss unsupervised manifold learning algorithms, the following notation will be used. Let the original high-dimensional data matrix be $X = [x_1\ x_2\ ...\ x_N] \in R^{dxN}$, mean of all data samples in X be μ, the data dimension after dimensionality reduction be r and the output data matrix be $Y = [y_1\ y_2\ ...\ y_N] \in R^{rxN}$.

Principal Component Analysis

Principal component analysis (PCA) is a linear unsupervised dimensionality reduction method that is frequently used in pattern recognition. PCA is used for dimensionality reduction, and estimation and visualization of a dataset. The purpose of PCA is to find r uncorrelated principal components related to the data set. Direction on which the dataset shows maximum variation is chosen as the first principal component. Then among the directions orthogonal to the first principal component, the one along which the dataset shows maximum variation is chosen as the second principal component. Continuing in this way, the direction ortogonal to the previously found principal components along which the dataset shows maximum variation is chosen as the new principal component.

The principal components are computed by first obtaining the covariance matrix of the data samples:

$$S = \sum_{j=1}^{N} (x_j - \mu)(x_j - \mu)^T \tag{2.1}$$

Then, the eigenvectors of S are calculated. Finally, the r eigenvectors associated with the r largest eigenvalues give us the principal component vectors. The new low dimensional coordinates of the initially high dimensional data samples are then given by the projections of the samples onto the principal components.

Laplacian Eigenmaps

PCA is interested in global statistical characteristics of a dataset and does not care about the conservation of the local structure of the dataset. Meanwhile, in some applications, one may wish to maintain the geometric structure of the data in the embedding. Laplacian Eigenmaps algorithm (LE) [9] is a nonlinear dimensionality reduction algorithm that aims to find a low dimensional representation for a dataset that is embedded in a high-dimensional ambient space. Unlike PCA, LE is able to capture the nonlinear structure of the data. LE consists of the following steps:

• The neighborhood graph is built. If the data sample x_j is among the ϵ-nearest

neighbors of x_k or x_k is among the ϵ-nearest neighbors of x_j, nodes j and k are connected by an edge.

• Connected edges are weighted and weight matrix W is constructed.

$$W_{j,k} = \begin{cases} \exp(-\|x_j - x_k\|^2/\beta) & \text{if nodes } j \text{ and } k \text{ are connected} \\ 0 & \text{if nodes } j \text{ and } k \text{ are not connected} \end{cases} \tag{2.2}$$

where β is a heat kernel parameter and is generally tuned as the mean of squared distances between all sample pairs.

• The following objective function is minimized, which intend to maintain the local structure of the dataset by bringing the neighboring samples in the original space close to each other in the new space.

$$\min \sum_{jk} (y_j - y_k)^2 W_{jk} \tag{2.3}$$

The above objective function can be rewritten as

$$\begin{aligned} \sum_{jk} (y_j - y_k)^2 W_{jk} &= \sum_{jk} (y_j^2 + y_k^2 - 2y_j y_k) W_{jk} \\ &= \sum_j y_j^2 D_{jj} + \sum_k y_k^2 D_{kk} - 2 \sum_{jk} y_j y_k W_{jk} = 2y^T L y \end{aligned} \tag{2.4}$$

where D is the diagonal weight matrix, whose inputs are found by summing the columns of W, so that $D_{jj} = \sum_k W_{jk}$. The Laplacian matrix described as $L = D - W$.

• Lastly, the following generalized eigenvector problem is solved:

$$Lf = \lambda D f \tag{2.5}$$

Let the eigenvalues found in this problem be $[0 = \lambda_0 \leq \lambda_1 \leq ..., \leq \lambda_r]$ and the corresponding eigenvectors be $[f_0, f_1, ..., f_r]$. f_0 is discarded because it corresponds to the eigenvalue 0 and usually gives an almost constant vector. Finally, the first r eigenvectors, $[f_1, ..., f_r]$, gives the output data Y. The purpose of selecting the eigenvectors corresponding to the smallest eigenvalues in LE is the following: The LE

method seeks to embed neighboring samples in the original space to nearby samples in the new domain. The new coordinates of the data then correspond to functions that have the slowest possible variation on the original data graph, which are given by the eigenvectors $[f_1, ..., f_r]$.

Locality Preserving Projection

Another algorithm that gives importance to maintaining the local neighboring structure of the dataset is Locality Preserving Projection (LPP) [20]. This algorithm is a linear approximation of the nonlinear Laplacian Eigenmaps. It is a faster technique, thanks to this linear approximation. Nonlinear techniques like Laplacian Eigenmaps, Locally Linear Embedding [30] compute a pointwise mapping only for the training samples that is initially available, and their generalization to new test is called the out-of-sample problem. Meanwhile, since it learns a linear projection, LPP can be directly applied to new test points. The following algorithmic steps describe the LPP method:

• The adjacency graph and the weight matrix are constructed as in the Laplacian Eigenmaps method.

• Eigenvalues and eigenvectors of the following generalized eigenvector problem are found.

$$XLX^Te = \lambda XDX^Te \tag{2.6}$$

where D and L are calculated as in the Laplacian Eigenmaps method. This reformulates the same problem as in the Laplacian Eigenmaps method, however, under the linear projection constraint $f = X^Te$. The first r eigenvectors corresponding to smallest eigenvalues give the linear transformation matrix T. New test points are mapped by this transformation matrix, namely:

$$Y = T^TX \tag{2.7}$$

ISOMAP

The purpose of ISOMAP [34] is to embed the data samples into a much lower dimensional space while preserving the geodesic distances of the data samples on the data manifold. Most of the classical techniques use Euclidean distance. Meanwhile, in settings where the data is known to lie on a manifold, the geodesic distance, which is the closest distance on the manifold, may be a more meaningful dissimilarity measure than the Euclidean distance (see Figure 2.1 for an illustration of the geodesic distance and the Euclidean distance). Furthermore, the classical linear techniques like PCA are successful only if the intrinsic geometry of the data set conforms to a subspace model. Otherwise, these algorithms will fail to explore the true underlying structure of the original high-dimensional manifold. The ISOMAP algorithm works as follows:

• In the first stage of the ISOMAP, the neighborhood graph G is built. Each point is connected to its ϵ-nearest neighbors by edges and these edges are weighted with $d_x(j,k)$ which is the distance between points j and k.

• In the second stage of the algorithm, the shortest geodesic distances $d_G(j,k)$ between all sample pairs on the manifold are estimated and these distance values form the matrix $D_G = \{d_G(j,k)\}$. The geodesic distances are computed as follows. If points j and k are linked by an edge, $d_G(j,k)$ will be equated to $d_x(j,k)$. Otherwise, $d_G(j,k) = \infty$. Then $d_G(j,k)$ is replaced by $\min(d_G(j,k), d_G(j,m) + d_G(m,k))$ for the values of $m = 1, 2, ..., N$ one by one.

• In the final stage, the new low-dimensional coordinates y_j of the samples x_j are computed by minimizing the following cost function:

$$E(Y) = \|\xi(D_G) - \xi(D_Y)\|_F \tag{2.8}$$

where D_Y is formed by Euclidean distances of output samples y_j and y_k. The aim of this objective function is to form the lower dimensional embedding Ysuch that the estimated intrinsic geometry of the original manifold is maintained. In order to boost the efficiency of the optimization, the ξ operator converts the distances in matrices D_G and D_Y to inner products and is defined as follows

$$\tau(D) = -HSH/2 \tag{2.9}$$

where $S_{jk} = D_{jk}^2$ and $H = I - (1/N)\mathbf{1}\mathbf{1}^T$ is the "centering matrix".

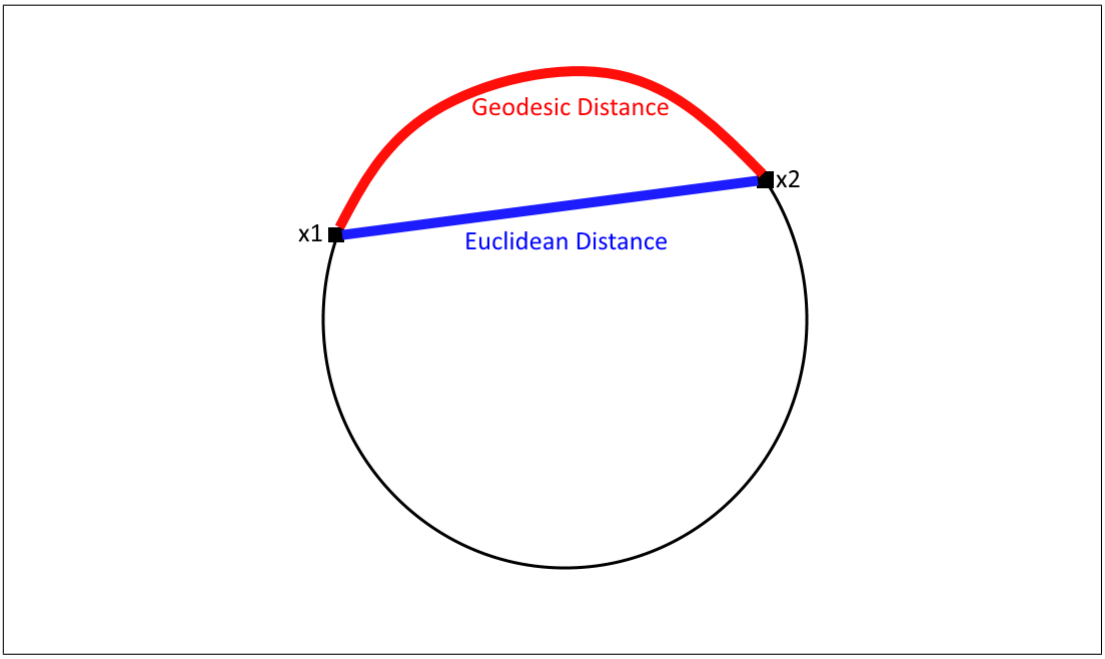


Figure 2.1: Geodesic and Euclidean Distances

Locally Linear Embedding

Since most databases have a nonlinear structure, nonlinear methods have been needed. Locally Linear Embedding (LLE) [30] is a popular nonlinear dimensionality reduction algorithm. LLE intends to learn the global structure of the nonlinear manifold by reconstructing every sample by its neighbors linearly. This algorithm comprises the following three steps:

• The ϵ-nearest neighbors are found for every sample in the dataset X. The Euclidean distance is commonly used as a measure of distance between the samples.

• The best reconstruction of each x_j is found with its ϵ-nearest neighbors. The reconstruction weights build the reconstruction matrix W. The goal in this step is to find the matrix W that minimizes reconstruction error. The reconstruction error is

calculated as

$$E(W) = \sum_{j=1}^{n} \|x_j - \sum_{k=1}^{\epsilon} W_{jk} x_{jk}\|^2 \tag{2.10}$$

subject to two constraints

$$\sum_{k=1}^{n} W_{jk} = 1 \tag{2.11}$$

and $W_{jk} = 0$ if x_k is not among the ϵ-nearest neighbors of x_j.

• After calculating the weight matrix W, output points are calculated by minimizing the following objective function that preserves the neighborhood characteristics :

$$E(y) = \sum_{j=1}^{n} \|y_j - \sum_{k=1}^{\epsilon} W_{jk} y_{jk}\|^2 \tag{2.12}$$

subject to two constraints :

$$\sum_{j=1}^{n} y_j = 1 \tag{2.13}$$

and

$$\sum_{j=1}^{n} y_j y_j^T / n = 1 \tag{2.14}$$

Let $R = (I - W)^T (I - W)$. Then Y is found by computing the $r + 1$ eigenvectors of R corresponding to the $r + 1$ smallest eigenvalues under these constraints. The first eigenvector whose eigenvalue is close to zero is discarded. The remaining r eigenvectors give the coordinates Y of the embedding. However, the mapping of new test samples to the low-dimensional domain is not straightforward, namely the "out of sample" problem occurs in this method.

2.2 Supervised Manifold Learning Methods

Supervised manifold learning is a learning process that uses labeled observations. The algorithm learns a classification rule from the labels of the training data. In supervised manifold learning, a dimensionality reducing mapping is learnt over the training data. In other words, a mapping is built from labelled training data to desired outputs. The purpose is to approximate the mapping function so well that when you have new input data, you can predict the output variables correctly for that data.

Generally, supervised learning follows the following procedure. First, a training dataset is constituted by collecting information concerned with the application at hand. For better learning, the quantity of the training data is important. In addition, there should be sufficiently many samples from each class. If we have 100 articles from class A and 1 article from class B, "overfitting" will occur. Namely, the learning algorithm will not be able to estimate the labels of the samples that belong to class B. Secondly, input features for the training data and the corresponding output values (labels) are determined. Thirdly, the most suitable training algorithm should be choosen. Finally, once a model is built, the class labels of test data can be estimated via the learnt model.

The vast majority of supervised dimensionality reduction methods rely on linear projections, and the methods computing a continuous supervised nonlinear embedding are less common. The generalization of the embedding of a given set of training samples to the whole space via continuous interpolation functions is known as the out-of-sample problem. The out-of-sample problem is of critical importance especially for nonlinear supervised manifold learning methods computing a pointwise embedding at only training samples. While out-of-sample extensions via the Nyström method [11], locally linear representations [17], or smooth interpolators such as polynomials [28] are commonly employed in unsupervised manifold learning, fewer works have studied the interpolation problem within a formulation specifically suited to supervised manifold learning [38], [24].

In this section, several linear and nonlinear supervised learning techniques will be explained with their implementation details. When we discuss supervised manifold learning algorithms, the following notation will be used. Let the matrix which contains all training samples be $X = [x_1\ x_2\ ...\ x_N] \in R^{dxN}$, the class label of the sample x_j be $l(x_j)$, the training data matrix that belongs to class-j be X_j, the number of classes be C, the number of training samples in class-j be n_j, the new dimension after dimensionality reduction be r, the mean of all training samples be μ and the mean of all training samples in the class-j be μ_j. Let the nearest neighbors of a sample x_j in the same class be denoted as $N_w(x_j)$; and the nearest neighbors of x_j in the other classes be $N_b(x_j)$. $N_w(x_j)$ and $N_b(x_j)$ are defined as follows:

$$N_w(x_j) = \{x_k | l(x_k) = l(x_j), \text{if } x_j \text{ is one of the } \epsilon\text{-nearest neighbors of } x_k \text{ or } x_k \text{ is one of the } \epsilon\text{-nearest neighbors of } x_j\} \tag{2.15}$$

$$N_b(x_j) = \{x_k | l(x_k) \neq l(x_j), \text{if } x_j \text{ is one of the } \epsilon\text{-nearest neighbors of } x_k \text{ or } x_k \text{ is one of the } \epsilon\text{-nearest neighbors of } x_j\} \tag{2.16}$$

Fisher Linear Discriminant Analysis

Fisher Linear Discriminant Analysis (FDA) is a famous linear transformation method which is one of the earliest works aiming supervised dimensionality reduction. It shrinks the dimensionality of data by learning a projection so that the between-class separation is increased while the within-class compactness is enhanced. In order to do this, first, the within-class scatter and the between-class scatter matrices are found:

$$S_w = \sum_{k=1}^{l} \sum_{j:y_j=k} (x_j - \mu_k)(x_j - \mu_k)^T \tag{2.17}$$

and

$$S_b = \sum_{k=1}^{l} n_k(\mu_k - \mu)(\mu_k - \mu)^T \tag{2.18}$$

In order to maximize the between-class scatter and to minimize the within-class scatter, the following objective function is used:

$$J(e) = \frac{e^T S_B e}{e^T S_W e} \tag{2.19}$$

Then the following eigenvector problem needs to be solved to maximize this objective function:

$$S_b e = \lambda S_w e \tag{2.20}$$

Assume that $\{e_j\}_{i=1}^{r}$ are the generalized eigenvectors corresponding to the generalized eigenvalues $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_r$. Lastly, the transformation matrix, T_{FDA}, is formed by these eigenvectors:

$$T_{FDA} = (e_1|e_2|...|e_r) \tag{2.21}$$

Mapped samples are simply found by the multiplication of the transformation matrix with the original high dimensional samples.

Although methods like FDA are easy to implement, some disadvantages arise. One of the important disadvantages is the rank defficiency of the between-class scatter matrix. This limits the maximum embedding dimension r to the number of classes in the original training data. Because of the fact that these methods are based on global statistics, such as the mean, and do not attach importance to the local structure of the data, their accuracy rate in multimodal data where samples in a class may come from from several clusters, is generally low.

Furthermore, it is known that linear dimensionality reduction techniques may perform badly for nonlinearly distributed complicated datasets. In order to adapt FDA to this type of datasets, the "kernel trick" is applied to FDA and it is called "Kernel Discriminant Analysis (KDA)". First, the input data samples $\{x_i\}_{i=1}^{N}$ are mapped to a higher dimensional space via a nonlinear mapping function $\phi(.)$. After this, the scatter matrices and the objective functions of FDA will be written with $\phi(x)$ instead of x :

$$S_W^{\phi} = \sum_{j=1}^{C} \sum_{k=1}^{n_j} (\phi(x_k^j) - \mu_j^{\phi})(\phi(x_k^j) - \mu_j^{\phi})^T \tag{2.22}$$

where n_j is number of samples in class-j and x_k^j is the j^{th} sample in class-j. The between-class scatter matrix becomes :

$$S_B^{\phi} = \sum_{j=1}^{C} (\mu_j^{\phi} - \mu^{\phi})(\mu_j^{\phi} - \mu^{\phi})^T \tag{2.23}$$

where

$$\mu_j^{\phi} = (1/n_j) \sum_{k=1}^{n_j} \phi(x_k^j) \tag{2.24}$$

and

$$\mu^{\phi} = (1/N) \sum_{k=1}^{N} \phi(x_k) \tag{2.25}$$

Then the objective function that will be maximized becomes the following:

$$Q(w) = \frac{(w^T S_B^{\phi} w)}{(w^T S_W^{\phi} w)} \tag{2.26}$$

Local Feature Discriminant Analysis

When the data is multimodal or has an intricate geometry, maintaining the local structure is important. LPP can be successful in multimodal data because it preserves the local structure of the data when embedding them into lower dimensions. However LPP is an unsupervised method. In order to combine the idea of FDA and the idea of LPP, Local Fisher Discriminant Analysis (LFDA) [33] is suggested. LFDA is also a linear dimensionality reduction algorithm. It maximizes the between-class scatter, while preserving the local structure at the same time.

Let n_l be the number of samples in class l. The local within-class scatter and the local between-class scatter matrices are calculated as follows :

$$\tilde{S}^{(w)} = (1/2) \sum_{j,k=1}^{n} \tilde{W}_{jk}^{(w)} (x_j - x_k)(x_j - x_k)^T \tag{2.27}$$

$$\tilde{S}^{(b)} = (1/2) \sum_{j,k=1}^{n} \tilde{W}_{jk}^{(b)} (x_j - x_k)(x_j - x_k)^T \tag{2.28}$$

where

$$\tilde{W}_{j,k}^{(w)} = \begin{cases} A_{j,k}/n_l & \text{if } y_j = y_k = l \\ 0 & \text{if } y_j \neq y_k \end{cases} \tag{2.29}$$

$$\tilde{W}_{j,k}^{(b)} = \begin{cases} \dfrac{A_{j,k}}{(1/n - 1/n_l)} & \text{if } y_j = y_k = l \\ 1/n & \text{if } y_j \neq y_k \end{cases} \tag{2.30}$$

Sample pairs from the same class are weighted by $A_{j,k} = exp(-\|x_j - x_k\|^2/\sigma)$. In this way, far apart sample pairs have less influence on $\tilde{S}^{(w)}$ and $\tilde{S}^{(b)}$. Hence, in contrast to LDA, distant sample pairs and nearby sample pairs are weighted differently in the within-class scatter objective. This takes into account the local data geometry in the learning. Sample pairs from different classes are not weighted. Then the transformation matrix of LFDA is found from the following optimization problem

$$T_{LFDA} = \arg\min_{T \in R^{dxr}} [tr((T^T \tilde{S}^{(w)} T)^{-1} T^T \tilde{S}^{(b)} T)] \tag{2.31}$$

Locality Sensitive Discriminant Analysis

Another method that is based on both the local structure of the data and the discriminant information is Locality Sensitive Discriminant Analysis (LSDA) [12]. The within-class graph and the between-class graph are constructed in this method too. Then, the within-class and between-class weight matrices are calculated from these graphs as follows:

$$W_{w,jk} = \begin{cases} 1 & \text{if } x_k \in N_w(x_j) \text{ or } x_j \in N_w(x_k) \\ 0 & \text{otherwise} \end{cases} \tag{2.32}$$

$$W_{b,jk} = \begin{cases} 1 & \text{if } x_k \in N_b(x_j) \text{ or } x_j \in N_b(x_k) \\ 0 & \text{otherwise} \end{cases} \tag{2.33}$$

Two objective functions that are formed by these weight matrices are optimized. One of them is optimized to maintain the local structure and the other one is optimized to benefit from the discriminant information of the data. These objective functions are given respectively by

$$\begin{aligned} \min \sum_{jk} (y_j - y_k)^2 W_{w,jk} &= \min \sum_{jk} (e^T x_j - e^T x_k)^2 W_{w,jk} \\ &= \min(\sum_j e^T x_j D_{w,jj} x_j^T e - \sum_{jk} e^T x_j W_{w,jk} x_k^T e) \\ &= \min(e^T X D_w X^T e - e^T X W_w X^T e) \end{aligned} \tag{2.34}$$

where $D_{w,jj} = \sum_k W_{w,jk}$ is a diagonal matrix and provides a natural measure on the data points. If $D_{w,jj}$ is large, then it alludes that the class containing x_j has a high density around x_j and this means that x_j is more important. Thus, the following constraint is imposed:

$$y^T D_w y = e^T X D_w X^T e = 1 \tag{2.35}$$

The objective function 2.34 becomes:

$$\min_e 1 - e^T X W_w X^T e = 1 \tag{2.36}$$

$$\begin{aligned} \max \sum_{jk} (y_j - y_k)^2 W_{b,jk} &= \max \sum_{jk} (e^T x_j - e^T x_k)^2 W_{b,jk} \\ &= \max(\sum_j e^T x_j D_{b,jj} x_j^T e - \sum_{jk} e^T x_j W_{b,jk} x_k^T e) \\ &= \max(e^T X D_b X^T e - e^T X W_b X^T e) \\ &= \max e^T X L_b X^T e \end{aligned} \tag{2.37}$$

Finally, these objective functions are combined as follows:

$$\arg\max_e e^T X(\beta L_b + (1-\beta) W_w) X^T e \;\text{ s.t. }\; e^T X D_w X^T e = 1 \tag{2.38}$$

where β is a constant that is between 0 and 1.

However, if the data manifold is nonlinear, LSDA tends to fail because it is a linear learning algorithm. In order to deal with this negative outcome of linearity, kernel trick can be applied to LSDA.

Local Discriminant Embedding

Local Discriminant Embedding (LDE) [14] is another linear manifold learning algorithm. The purpose is to decrease the distance between samples of the same class and

increase the separation between samples of different classes in the low dimensional space of embedding. The algorithm consists of three steps:

• For all data samples, two neighborhood graphs are constructed. Let the first graph be G and the second graph be G'. In the first graph G, there is an edge between points x_j and x_k if x_j and x_k are in the same class and x_k is one of x_j's ϵ-nearest neighbors. In the second graph G', there is an edge between points x_j and x_k if x_j and x_k are in different classes and x_k is one of x_j's ϵ-nearest neighbors.

• Two affinity weight matrices are calculated for two graphs that are built in the first step. The first affinity weight matrix that belongs to the first neighborhood graph G is calculated as follows:

$$W_{jk} = \begin{cases} \exp[-\|x_j - x_k\|^2/\beta] & \text{if there is an edge between } x_j \text{ and } x_k \text{ in } G \\ 0 & \text{otherwise} \end{cases} \tag{2.39}$$

The other affinity weight matrix W' of G' is calculated as follows:

$$W'_{jk} = \begin{cases} \exp[-\|x_j - x_k\|^2/\beta] & \text{if there is an edge between } x_j \text{ and } x_k \text{ in } G' \\ 0 & \text{otherwise} \end{cases} \tag{2.40}$$

• The generalized eigenvectors $[v_1, v_2, ..., v_r]$ that correspond to the r largest eigenvalues and give a linear transformation matrix are found from the following eigenvector problem :

$$X(D' - W')X^T v = \lambda X(D - W)X^T v \tag{2.41}$$

where D and D' are diagonal matrices with diagonal elements $d_{jj} = \sum_k w_{jk}$ and $d'_{jj} = \sum_k w'_{jk}$.

Since the classification performance of linear methods is limited, especially in non-linearly distributed complex manifolds, the "kernel trick" can be applied to this linear method too to enhance the performance. The input data is mapped to higher dimensional space via nonlinear kernel functions. Let $\phi : R^n \to F$ be a nonlinear mapping

that transforms the input data $\{x_j\}_{j=1}^N$ to higher dimensional feature space F. The algorithmic steps of LDE will be applied to $\{\phi(x_j)|\phi(x_j) \in F\}$ instead of $\{x_j\}_{j=1}^N$. Hence, the generalized eigenvalue problem of LDE becomes :

$$K(D' - W')K^T z = \lambda K(D - W)K^T z \tag{2.42}$$

where K is a kernel matrix with $K_{jk} = k(x_j, x_k) = \phi(x_j)^T \phi(x_k)$. After this, the transformation matrix is learnt and the embedding of new test samples are computed from the eigenvectors of this problem.

Stable Orthogonal Local Discriminant Embedding

It is known that PCA, FDA and LDA methods pay attention only to the global structure of data. Many manifold learning methods consider preservation of the within-class similarity and compactness but ignore the within-class variation. Supervised Optimal Local Discriminant Embedding(SOLDE) [18] takes into account both the within-class similarity and the within-class diversity in order to achieve generalization and stability. An orthogonality constraint is imposed for the vectors of the transformation matrix.

• The weight matrix S_w is built as follows:

$$S_{w,jk} = \begin{cases} \exp(-\|x_j - x_k\|^2/\beta) & \text{if } x_k \in N_w(x_j) \text{ or } x_j \in N_w(x_k) \\ 0 & \text{otherwise} \end{cases} \tag{2.43}$$

and the following objective function that imposes within-class compactness is minimized:

$$\sum_{jk}(y_j - y_k)^2 S_{jk} = 2T^T X L_s X^T T \tag{2.44}$$

• Weight matrix H is built as follows:

$$H_{jk} = \begin{cases} 1 - \exp(-\|x_j - x_k\|^2/\beta) & \text{if } x_k \in N_w(x_j) \text{ or } x_j \in N_w(x_k) \\ 0 & \text{otherwise} \end{cases} \tag{2.45}$$

and the following objective function that represents the within-class variation is maximized:

$$\sum_{jk}(y_j - y_k)^2 H_{jk} = 2T^T X L_h X^T T \tag{2.46}$$

• The weight matrix F is built as follows:

$$F_{jk} = \begin{cases} \exp(-\|x_j - x_k\|^2/\beta) & \text{if } x_k \in N_b(x_j) \text{ or } x_j \in N_b(x_k) \\ 0 & \text{otherwise} \end{cases} \tag{2.47}$$

and the following objective function that represents the between-class separation is maximized:

$$\sum_{jk}(y_j - y_k)^2 F_{jk} = 2T^T X L_f X^T T \tag{2.48}$$

All of these objective functions are combined into a single objective function:

$$T^* = \arg\max \frac{T^T X L_f X^T T}{(T^T X L_s X^T T - T^T X L_h X^T T)} \tag{2.49}$$

However, in the objective function, intraclass similarity $T^T X L_s X^T T$ and intraclass diversity $T^T X L_h X^T T$ are weighted equally, whereas the task of $T^T X L_h X^T T$ is to avoid the over-fitting problem caused by $T^T X L_s X^T T$. In order to obtain a stable and compact intraclass representation, the component $T^T X L_s X^T T$ should be weighted larger and the component $T^T X L_h X^T T$ should be weighted smaller. Thus, the optimal objective function should be

$$\begin{aligned} T^* &= \arg\max \frac{T^T X L_f X^T T}{(T^T X (bL_s - (1-b)L_h) X^T T)} \\ &= \arg\max \frac{T^T X L_f X^T T}{(T^T X L_d X^T T)} \end{aligned} \tag{2.50}$$

where $L_d = \alpha L_s - (1-\alpha)L_h$, and α is a constant that is set as 0.9 in [18]. Then the eigenvectors of the following eigenvector problem gives the projection vector.

$$XL_fX^TT = \lambda XL_dX^TT \tag{2.51}$$

Supervised Optimal Locality Preserving Projection (SOLPP)

It is known that LPP can preserve the local neighborhood structure but it is an unsupervised technique. Unlike Laplacian Eigenmap, Locally Linear Embedding and Isomap algorithms, LPP is a linear dimensionality reduction algorithm and easy to implement. Upon this, Wong and Zhao proposed the Supervised Optimal Locality Preserving Projection (SOLPP) [39] , which uses the class label information too. The features extracted by this technique are statistically uncorrelated and orthogonal. This is why this method is called "optimal". When the extracted features are correlated, there will be redundant information and this reduce the performance. Unfortunately LPP has this disadvantage. In the SOLPP method, because of the availability of the class label information, the similarity matrix is defined as :

$$W_{jk} = \begin{cases} \exp(-\|x_j - x_k\|^2/\beta) & \text{if } x_j \text{ and } x_k \text{ are in the same class} \\ 0 & \text{otherwise} \end{cases} \tag{2.52}$$

However when this similarity matrix is used, the neighborhood graph of the training data can be disjointed and this contradicts the idea of LPP that aims to preserve the local structure of the original manifold. Thus, similarity matrix W of SOLPP is constructed as follows:

Let the d dimensional input training data be $\{x_j, l_j\}_{j=1}^N$ in which $l_j \in 1, 2, ..., c$ is the class label of x_j, and $p(j)(j = 1, 2, ..., c)$ be the class prior probability of the j^{th} class. Then, the similarity matrix is defined as

$$W_{jk} = \begin{cases} p(l_j)^2 \exp(\frac{-\|x_j - x_k\|^2}{\beta})(1 + \exp(\frac{-\|x_j - x_k\|^2}{\beta})) & \text{if } x_k \in N_w(x_j) \\ & \text{or } x_j \in N_w(x_k) \\ p(l_j)p(l_k) \exp(\frac{-\|x_j - x_k\|^2}{\beta})(1 - \exp(\frac{-\|x_j - x_k\|^2}{\beta})) & \text{if } x_k \in N_b(x_j) \\ & \text{or } x_j \in N_b(x_k) \\ 0 & \text{otherwise} \end{cases} \tag{2.53}$$

where the class label prior probalities makes the algorithm more suitable to classification applications when the number of training samples in different classes are unequal. The following objective function is minimized :

$$J(\phi) = \min(tr(\phi^T X L X^T \phi)) \tag{2.54}$$

under the following constraints, which provide the orthogonal and uncorrelated features to this method:

$\phi^T \phi = I$, $\phi_j^T X D X^T \phi_k = 0$ and $\phi_k^T X X^T \phi_j = 0$ $(j \neq k)$

Two-Stage Multiple Kernel Learning

Since most data collections have a high dimensional structure and nonlinear mappings give better results for nonlinearly distributed data, the kernel trick is used in order to convert linear methods to nonlinear. The kernel extensions of many famous dimensionality reduction methods such as PCA, LDA, ICA exist [32], [7], [6]. The construction of continuous functions via smooth kernels is also quite common in Reproducing Kernel Hilbert Space (RKHS) methods [10], [5]; however, these methods differ from supervised manifold learning methods in that the learnt mapping often represents class labels of data samples rather than their coordinates in a lower-dimensional domain of embedding as in manifold learning. The choice of the kernel type and parameters can be critical in kernel methods. Several previous works in the semi-supervised learning literature have addressed the learning of kernels by combin-

ing known kernels [16], [4]. A two-stage multiple kernel learning method is recently proposed in [22] for supervised dimensionality reduction, which finds a nonlinear mapping by the perspective of the multiple kernel learning.

The approach in multiple kernel learning (MKL) methods is to construct a new kernel function benefiting from several known kernel functions $k_m(.,.)$ instead of using one kernel. In this way, a more suitable kernel can be found. Two-stage Multiple Kernel Learning (TSMKL) [22] is a recent MKL method. There are two steps in TSMKL. In the first step, a new kernel function is built as a linear combination of valid kernels as follows:

$$\phi(x,z) = \sum_{m=1}^{M} \beta_m \phi_m(x,z) \tag{2.55}$$

where the weights $[\beta_1, ..., \beta_m]$ are determined according to an appropriate criterion. The "Maximizing Fisher Criterion", "maximizing homoscedasticity criterion[40]" and the "maximizing between-class distance criterion" could be chosen. In the second step, KDA is applied in order to calculate a nonlinear mapping which maximizes between-class distances and minimizes within-class distances at the same time.

Supervised Versions of Locally Linear Embedding

The major disadvantage of LLE is the "out of sample problem". LLE does not give us a transformation that projects new data points onto the embedded space. In addition, LLE does not exploit the class label information; namely, LLE is an unsupervised method. In order to profit from class label information, The Supervised Locally Linear Embedding (SLLE) [42] is proposed. The only difference of the SLLE from the LLE is the calculation of the distances between the samples. In the LLE, the distances between the samples are calculated by the Euclidean distance and no class information is used. In the SLLE, the distance between the samples are calculated as follows:

$$\delta' = \delta + \tau \max(\delta)\kappa_{jk}, \tau \subset [0,1] \tag{2.56}$$

where δ is the Euclidean distance that is calculated without utilizing the class label information and $\max(\delta)$ is the largest Euclidean distance between the classes. If X_j and X_k are in different classes, $\kappa_{jk} = 1$; and $\kappa_{jk} = 0$ otherwise. The purpose is to increase the distance between the samples from different classes more. τ is the control parameter that determines the amount of class label information to be added. SLLE can be converted into the unsupervised LLE by making $\tau = 0$. The Step 2 and Step 3 of SLLE is the same as the Step 2 and Step 3 of LLE.

Enhanced Supervised Locally Linear Embedding (ESLLE) [42] is proposed by modifying the distance in SLLE. The new distance between the samples is calculated as follows:

$$\delta' = \begin{cases} \sqrt{(exp(\delta^2/\beta)) + \tau} & \text{if } l_j \neq l_k \\ \sqrt{(1 - exp(-\delta^2/\beta))} & \text{if } l_j = l_k \end{cases} \tag{2.57}$$

where l_j is the class label of X_j and β is a positive constant.

However these two methods do not solve the "out of sample" problem. In order to overcome the out of sample problem of LLE, the Neighborhood Preserving Embedding (NPE) is proposed. NPE is also an unsupervised dimensionality reduction algorithm. NPE can preserve the local neighborhood structures on the original data manifold. An optimal transformation matrix A is found by minimizing the following cost function:

$$A = \arg\min_A \sum_{j=1}^{n} ||y_j - \sum_{j=1}^{n} W_{jk} y_k||^2 = \arg\min_A tr(A^T X M X^T A) \tag{2.58}$$

subject to $A^T X X^T A = I$, where $M = (I - W)^T (I - W)$. The Lagrange multipliers technique takes this cost function to the following generalized eigenvector problem:

$$XMX^T A = \lambda X X^T A \tag{2.59}$$

Finally output samples Y are found by $Y = AX$.

LLE considers that each sample can be reconstructed by its ϵ-nearest neighbors so there is one weight matrix. In the Neighborhood Preserving Discriminant Embedding (NPDE) [25] algorithm, there are two reconstruction weight matrices. The first

one is a within-class reconstruction matrix that is found by reconstructing each sample with its ϵ-nearest neighbors from the same class. The second one a between-class reconstruction matrix that is found by reconstructing each sample with its ϵ-nearest neighbors from the other classes. Then, two cost functions are formulated as in LLE in order to preserve the local structure with the aid of the within-class and the between-class reconstruction matrices. An optimization problem that minimizes the cost function that is related with the within-class reconstruction matrix while maximizing the cost function that is related with the between-class reconstruction matrix is solved.

Supervised Laplacian Eigenmaps

Laplacian Eigenmaps (LE) preserves the geometric structure of the data but does not benefit from discriminant information of data. The Supervised Laplacian Eigenmaps (SUPLAP) algorithm [29] exploits discriminant information too. Geometrical and discriminant information are represented by two different graphs. The within-class graph G_w and the between-class graph G_b. The within-class weight matrix W_w and the between class weight matrix W_b are calculated by these graphs respectively. Then, two objective functions are defined and the optimization of these objective functions gives the embedding. The algorithm is implemented by the following steps.

• The average similarity is calculated for each sample:

$$AS(x_j) = (1/N) \sum_{k=1}^{N} \exp(-\|x_j - x_k\|^2/\beta) \tag{2.60}$$

• The nearest neighbors of x_j from the same class and the other classes are determined as

$$N_w(x_j) = \{x_k | l(x_k) = l(x_j), \exp(-\|x_j - x_k\|^2/\beta) > AS(x_j)\} \tag{2.61}$$

$$N_b(x_j) = \{x_k | l(x_k) \neq l(x_j), \exp(-\|x_j - x_k\|^2/\beta) > AS(x_j)\} \tag{2.62}$$

where $l(x_k)$ denotes the class label of the sample x_k.

• The weight matrices W_w and W_b are built:

$$W_{w,jk} = \begin{cases} \exp(-\|x_j - x_k\|^2/\beta) & \text{if } x_k \in N_w(x_j) \text{ or } x_j \in N_w(x_k) \\ 0 & \text{otherwise} \end{cases} \tag{2.63}$$

$$W_{b,jk} = \begin{cases} \exp(-\|x_j - x_k\|^2/\beta) & \text{if } x_k \in N_b(x_j) \text{ or } x_j \in N_b(x_k) \\ 0 & \text{otherwise} \end{cases} \tag{2.64}$$

• The following objective functions are optimized:

$$\min 1/2 \sum_{jk} \|y_j - y_k\|^2 W_{w,jk} \tag{2.65}$$

$$\max 1/2 \sum_{jk} \|y_j - y_k\|^2 W_{b,jk} \tag{2.66}$$

These functions can be rewritten as:

$$\min tr(Y^T L_w Y) \tag{2.67}$$

$$\max tr(Y^T L_b Y) \tag{2.68}$$

where L_w and L_b are the Laplacian matrices of W_w and W_b respectively. These objective functions are combined into a single objective function:

$$\arg\max_Y \{\gamma tr(Y^T L_b Y) + (1-\gamma) tr(Y^T W_w Y)\} \text{ s.t. } Y^T D_w Y = I \tag{2.69}$$

$$\arg\max_Y tr(Y^T(\gamma L_b + (1-\gamma) W_w) Y) \text{ s.t. } Y^T D_w Y = I \tag{2.70}$$

where $\gamma \in [0, 1]$ is a balance parameter and if we denote $B = \gamma L_b + (1-\gamma) W_w$, the following eigenvalue problem gives the solution of the optimization problem:

$$By = \lambda D_w y \tag{2.71}$$

Let the eigenvalues found in this problem be $[\lambda_1 \geq \lambda_2 \geq ..., \geq \lambda_k]$ and the corresponding eigenvectors be $[y_1, y_2, ..., y_k]$. The first r eigenvectors, $[y_1, ..., y_r]$, give the embedding Y of the training data.

Hybrid Manifold Embedding

Unlike many algorithms that give a linear mapping function, the goal of Hybrid Manifold Embedding (HyME) [21] is to find a general nonlinear mapping function. This nonlinear mapping function is found by a learning procedure with two layers. In the first layer, the original data is divided into subsets. Each subset has minimum nonlinearity. This procedure is called "geodesic clustering (GC)". In the second layer, supervised dimension reduction is applied to each subset. This procedure is called "Locally Conjugate Discriminant Projection (LCDP)". Only one mapping function is found by combining these procedures. Therefore, both the local and the global manifold structure is learned.

Datasets in real world can be very complex. Only one global model may be inadequate. The GC procedure divides the original dataset into the subsets that have minimum nonlinearity. In this way, the learning capability of the model is enhanced. The nonlinearity of a subset is measured by the maximum geodesic distance of that subset. Let P be a partition of the dataset, C be the number of subsets, X_c be cth subset and $dist_G(x_j, x_k)$ be the geodesic distance between datasamples x_j and x_k. The division of the original dataset into the subsets is achieved by the objective function of GC that is defined as follows:

$$P = \arg\min_{P} \max_{c=1,...,C} \max_{j,k:x_j,x_k \in X_C} dist_G(x_j, x_k) \tag{2.72}$$

After the clustering of the original dataset, "Locally Conjugate Discriminant Projection (LCDP)" is applied. Assume that X_c is the datamatrix of subset c, videlicet $X_c = [x_1, ..., X_{n_c}]$ where n_c is the number of data points in X_c. LCDP learns a $D \times d$ transformation matrix $W_c = [w_{c_1}, ..., w_{c_d}]$ that maximizes the discriminant scatter and preserves the local geometry in the new space. Then a discriminant scatter matrix ${S_c}^d$

and a locality scatter matrix ${S_c}^l$ are defined by using the class label informations of each data sample as follows:

$$ {S_c}^d = \sum_{j,k} ({A_c}^d)_{jk}(y_j - y_k)(y_j - y_k)^T = \sum_{j,k} ({A_c}^d)_{jk} {W_c}^T (x_j - x_k)^T (x_j - x_k) W_c \tag{2.73} $$

$$ {S_c}^l = \sum_{j,k} ({A_c}^l)_{jk}(y_j - y_k)(y_j - y_k)^T = \sum_{j,k} ({A_c}^l)_{jk} {W_c}^T (x_j - x_k)^T (x_j - x_k) W_c \tag{2.74} $$

where ${A_C}^d$ and ${A_C}^l$ are discriminant coefficient matrix and locality coefficient matrix respectively. If x_j is in the K-nearest neighbors of x_k and class labels of x_j and x_k are the same, $({A_C}^l)_{jk} = e^{-\|x_j - x_k\|^2/2\sigma}$, otherwise $({A_C}^l)_{jk} = 0$. If x_j is in the ϵ-nearest neighbors of x_k and class labels of x_j and x_k are different, $({A_C}^d)_{jk} = e^{-\|x_j - x_k\|^2/2\sigma}$, otherwise $({A_C}^d)_{jk} = 0$.

The following optimization problem gives the transformation matrix for a subset C. The transformation matrix is found separately for each subsets. The transformation matrix is found separately for each cluster as follows.

$$ \mathbf{W}_c = \arg\max_{W_c} \mathrm{tr}((S_c^l)^\dagger S_c^d) \tag{2.75} $$

where $(S_c^l)^\dagger$ is the Moore-Penrose pseudoinverse of ${S_c}^l$. Then these transformation matrices are combined and finally the combination gives us a nonlinear mapping function.

Local Feature Discriminant Projection

Local Feature Discriminant Projection (LFDP) [41] is a supervised dimensionality reduction method. One of the main ideas proposed in [41] is the Differential Scatter Discriminant Criterion (DSDC), which overcomes the matrix singularity problem. In classical methods such as LDA, because of the matrix singularity problem, the embedding dimension is restricted to the number of classes. A general orthogonalization procedure is also proposed in [41] that makes the subspace more compact and less redundant.

Not only the dimension is reduced but also the discriminant ability of the local features is increased. The Image to class (I2C) distance is found for every sample by using the class label information.

The I2C distance from an image x_j to class c is defined as:

$$D_{x_j}^c = (1/d) \sum_{k=1}^{d} \|x_{jk} - x_{jk}^c\|^2 \tag{2.76}$$

where x_{jk} is k-th feature of the image x_j and x_{jk}^c is the nearest neighbor of x_{jk} is class c. After the I2C distances are found, the DSDC is applied to the I2C distances that are found for all samples.

The purpose is to find a matrix $w \in \mathbb{R}^{Dxd}$ to project the original local features x_{jk} to a lower-dimensional but more discriminative space $\mathbb{R}^d$. The projected I2C distance of the image X_j becomes

$$\hat{D}_{X_j}^c = tr(w^T \Delta X_{jc}^T \Delta X_{jc} w) \tag{2.77}$$

where

$$\Delta X_{jc} = (1/\sqrt{m_j})[(x_{j1} - x_{j1}^c), \ldots, (x_{jm_j} - x_{jm_j}^c)]^T \in \mathbb{R}^{m_j \times D} \tag{2.78}$$

The projected I2C distances of the image X_j be

$$d_j = (\hat{D}_{X_j}^1, \ldots, \hat{D}_{X_j}^C) \tag{2.79}$$

$$J = \sum_{c=1}^{C} n_c \|\mu_c - \mu\|^2 - \lambda \sum_{c=1}^{C} \sum_{d_k \in classc} \|d_k - \mu_c\|^2 \tag{2.80}$$

where λ is a tuning parameter.

$$\mu_c = (1/n_c) \sum_{d_k \in classc} d_k \tag{2.81}$$

$$\mu = (1/N) \sum_{k=1}^{N} d_k \tag{2.82}$$

The intention of the objective function of DSCD is to minimize the within-class variance and to maximize the between-class variance. However, because of the quadratic structure of the objective function, the objective function cannot be directly converted to an eigenvalue problem. Therefore, the gradient descent algorithm on sphere is applied to minimize the objective function. In addition a general orthogonalization procedure is applied in order to make the embedding more compact and less redundant.

Dimensionality Reduction by Minimizing Nearest-Neighborhood Classification Error

"Dimensionality reduction by minimizing nearest-neighborhood classification error (DRMNNCE)" [36], is a supervised linear dimensionality reduction method. However there are no within-class and between-class scatters in this procedure, in contrast to conventional linear dimensionality reduction methods. An optimization procedure is applied in order to minimize the estimation error of the nearest neighbor classifier in the embedding. A linear projection and a small set of prototypes that support the class boundaries are learned.

The following sigmoid function with slope β, centered at $z = 1$ is used in order to define the 1-NN error rate:

$$S_\beta(z) = 1/(1 + e^{\beta(1-z)}) \tag{2.83}$$

$$S'_\beta(z) = \beta e^{\beta(1-z)}/(1 + e^{\beta(1-z)})^2 \tag{2.84}$$

The aim is to learn a projection base $B \in R^{DxE}$ by minimizing the error rate of the nearest neighbor classifier on a training set $X = \{x_1, ..., x_N\} \subset R^D$ with C classes.

In order to estimate the nearest neighbor classification error, a set of labeled prototypes $P = \{p_1, ..., p_M\} \subset R^D$ which is different from and much smaller than the training set X is defined. There should be at least one prototype per class, i.e. $P \not\subset X$, $C \ll M \ll N$.

Let $d(\tilde{x}, \tilde{p_{\in}})$ be the Euclidean distance between the training sample x and its same class nearest prototype in the target space ; and let $d(\tilde{x}, \tilde{p_{\notin}})$ be the Euclidean distance between the training sample x and its different class nearest prototype in the target space. By using the reference prototypes, the 1-NN error rate can be written as

$$J_x(B, P) = (1/N) \sum_{\forall x \in X} S_\beta(R_x) \tag{2.85}$$

where

$$R_x = d(\tilde{x}, \tilde{p_{\in}}) / d(\tilde{x}, \tilde{p_{\notin}}) \tag{2.86}$$

The sigmoid function is used rather than the step function in the 1-NN error rate definition because the contribution of each sample to the overall error J_x becomes more important or less important depending on the quotient of the distances. The gradients of J_x are derived as follows:

$$\begin{aligned} \nabla_B J_x = (1/N) \sum_{\forall x \in X} (S'_\beta(R_x) R_x / d(\tilde{x}, \tilde{p_{\in}})) \nabla_B d(\tilde{x}, \tilde{p_{\in}}) \\ -(1/N) \sum_{\forall x \in X} (S'_\beta(R_x) R_x / d(\tilde{x}, \tilde{p_{\notin}})) \nabla_B d(\tilde{x}, \tilde{p_{\notin}}) \end{aligned} \tag{2.87}$$

$$\begin{aligned} \nabla_{p_m} J_x = (1/N) \sum_{\forall x \in X : \tilde{p_m} = \tilde{p_{\in}}} (S'_\beta(R_x) R_x / d(\tilde{x}, \tilde{p_{\in}})) \nabla_{p_m} d(\tilde{x}, \tilde{p_{\in}}) \\ -(1/N) \sum_{\forall x \in X : \tilde{p_m} = \tilde{p_{\notin}}} (S'_\beta(R_x) R_x / d(\tilde{x}, \tilde{p_{\notin}})) \nabla_{p_m} d(\tilde{x}, \tilde{p_{\notin}}) \end{aligned} \tag{2.88}$$

where

$$\nabla_B d(\tilde{x}, \tilde{p}) = 2(x - p)(\tilde{x} - \tilde{p}) \tag{2.89}$$

$$\nabla_p d(\tilde{x}, \tilde{p}) = -2B(\tilde{x} - \tilde{p}) \tag{2.90}$$

In order to simplify the gradient equations the following factors can be defined.

$$F_{\in} = \frac{S'_\beta(R_x) R_x}{d(\tilde{x}, \tilde{p_{\in}})} \tag{2.91}$$

$$F_{\not\in} = \frac{S'_{\beta}(R_x)R_x}{d(\tilde{x}, \tilde{p_{\not\in}})} \tag{2.92}$$

Learning Discriminant Projections and Prototypes (LDPP) [36] is applied to get the following expressions:

$$\nabla_B J_x = XG^T + PH^T \tag{2.93}$$

$$\nabla_P J_x = BH \tag{2.94}$$

When the Euclidean distance is used, the n-th and m-th columns of the factor matrices G and H are:

$$g_n = (2/N)F_{\in_n}(\tilde{x}_n - \tilde{p}_{\in_n}) - (2/N)F_{\not\in_n}(\tilde{x}_n - \tilde{p}_{\not\in_n}) \tag{2.95}$$

$$h_m = (-2/N) \sum_{\forall x \in X : \tilde{p_m} = \tilde{p_{\in}}} F_{\in_n}(\tilde{x}_n - \tilde{p}_{\in_n}) + (2/N) \sum_{\forall x \in X : \tilde{p_m} = \tilde{p_{\not\in}}} F_{\not\in_n}(\tilde{x}_n - \tilde{p}_{\not\in_n}) \tag{2.96}$$

By using the following gradient descent update equations, optimization is performed.

$$B^{(t+1)} = B^{(t)} - \gamma \bigtriangledown_B J_x \tag{2.97}$$

$$P^{(t+1)} = P^{(t)} - \eta \bigtriangledown_P J_x \tag{2.98}$$

Finally the Gram-Schmidt procedure is applied to B in order to do orthonormal projection.

Two-Dimensional Discriminant Locality Preserving Projection Based on L1-Norm Maximization

Two-Dimensional Discriminant Locality Preserving Projection Based on L1-Norm Maximization(2DDLPP-L1) [15] is an improved LPP based method. Unlike other

methods that consider each image in the dataset as a vector and use L2-norm, 2DDLPP-L1 projects an image matrix onto a low-dimensional subspace directly without vectorizing it, which may destroy the structure information of input images, lead to dimensional disaster and increase the instability of matrix calculation. Since L2-norm magnifies noise and outliers and L1-norm is more robust to noise and outliers; L1-norm is preferred in this method.

Before explaining this method, we briefly describe the "Discriminant Locality Preserving Projection Based on L2-norm Maximization (DLPP-L2)", "Two-Dimensional Discriminant Locality Preserving Projection Based on L2-norm Maximization (2DDLPP-L2)", "Discriminant Locality Preserving Projection Based on L1-norm Maximization (DLPP-L1)" methods.

Discriminant Locality Preserving Projection Based on L2-norm Maximization (DLPP-L2) [25], converts an image to a vector as in other methods and then finds an optimal linear projection matrix $T = [t_1\ t_2\ ...\ t_r] \in \mathbb{R}^{dxr}$ by maximizing the following objective function based on L2-norm:

$$J(t) = \sum_{j,k=1}^{c} B_{jk}(t^T\mu_j - t^T\mu_k)^2 / \sum_{i=1}^{c}\sum_{j,k=1}^{N_i} W_{ij}^i(t^T x_j^k - t^T x_k^k)^2 \tag{2.99}$$

where $B_{jk} = \exp(-\|\mu_j - \mu_k\|^2/\sigma_b)$ is a weight between the mean vectors of the class-j and class-k, $W_{jk}^i = \exp(-\|x_j^i - x_k^i\|^2/\sigma_w)$ is a weight between two samples x_j and x_k in the class-i, and σ_b and σ_w are scale paramaters.

Two-Dimensional Discriminant Locality Preserving Projection Based on L2-norm Maximization (2DDLPP-L2) projects image matrices without vectorizing them by maximizing the following objective function based on L2-norm:

$$J(t) = \sum_{j,k=1}^{c} B_{jk}\|t^T\bar{A}_j - t^T\bar{A}_k\|^2 / \sum_{i=1}^{c}\sum_{j,k=1}^{N_i} W_{jk}^i\|t^T A_j^i - t^T A_k^i\|^2 \tag{2.100}$$

where A_j^i is the image matrix of *j*-th training sample in class-i, $\overline{A_j}$ is the mean of the image matrices in class-j, $B_{jk} = \exp(-\|\overline{A_j} - \overline{A_k}\|_F^2/\sigma_b)$ is a weight between the two mean matrices of the class-j and class-k, $W_{jk}^i = \exp(-\|A_j - A_k\|_F^2/\sigma_w)$ is a weight

between two matrices A_j and A_k in the class-i and σ_b and σ_w are scale paramaters.

Discriminant Locality Preserving Projection Based on L1-norm Maximization (DLPP-L1) converts an image to a vector and finds an optimal linear projection matrix $T = [t_1 t_2 ... t_r] \in \mathbb{R}^{dxr}$ by maximizing the following objective function based on L1-norm:

$$J(t) = \sum_{j,k=1}^{c} B_{jk} |t^T \mu_j - t^T \mu_k| / \sum_{i=1}^{c} \sum_{j,k=1}^{N_i} W_{jk}^i |t^T x_j^i - t^T x_k^i|^2 \tag{2.101}$$

Two-Dimensional Discriminant Locality Preserving Projection Based on L1-norm Maximization (2DDLPP-L1) optimizes the following objective function based on L1-norm:

$$J(t) = \sum_{j,k=1}^{c} B_{jk} \|(\bar{A}_j - \bar{A}_k)t\|_1 / \sum_{i=1}^{c} \sum_{j,k=1}^{N_i} W_{jk}^i \|(A_j^i - A_k^i)t\|_1 \tag{2.102}$$

,which preserves the image structure during learning while achieving robustness to noise and outliers via the usage of the L1-norm.

CHAPTER 3

PROPOSED METHOD

In this chapter, we present the proposed method for supervised dimensionality reduction. We first give an overview of the theoretical results that motivate our work in Section 3.1. We then formulate the manifold learning problem in Section 3.2 and define an optimization problem based on these theoretical perspectives. We then describe our algorithm in Section 3.3

3.1 Theoretical Perspectives

Nonlinear dimensionality reduction methods minimizing objectives as in (2.69) often yield embeddings where training samples from different classes are linearly separable, and the local neighborhoods on the same manifold are preserved as imposed by the within-class graph Laplacian matrix. On the other hand, what is critical is how well these embeddings generalize to new test data; i.e., when a test sample of unknown class label is mapped to the low-dimensional domain of embedding via an interpolator or an out-of-sample extension method, we are interested in how likely the sample is to be correctly classified. This depends both on the coordinates of the embedding for the training samples and the interpolator used to generalize the embedding to the whole ambient space. In the previous work [37], this problem is theoretically studied. We now overview some results from [37].

The classification problem is analyzed in [37] in a setting where each data sample in the training set $X = \{x_i\}_{i=1}^{N}$ is assumed to belong to one of the classes $\{1, 2, \ldots, M\}$ and the samples of each class m are distributed according to the probability measure

ν_m. Let $\mathcal{M}_m$ denote the support of the probability measure ν_m. Denoting as $B_\delta(x)$ an open ball of radius δ around a point x

$$B_\delta(x) = \{u \in \mathbb{R}^n : \|x - u\| < \delta\},$$

the following definition introduces the smallest possible measure for a ball $B_\delta(x)$ of radius δ centered around a point in the support $\mathcal{M}_m$ of the m-th class.

$$\eta_{m,\delta} := \inf_{x \in \mathcal{M}_m} \nu_m(B_\delta(x))$$

Next, we recall the definition of Lipschitz continuity for a function f.

Definition 1. *We say that a function $f : \mathbb{R}^n \to \mathbb{R}^d$ is Lipschitz continuous with constant $L > 0$ if for any $u, v \in \mathbb{R}^n$*

$$\|f(u) - f(v)\| \leqslant L \|u - v\|.$$

The analysis considers a supervised manifold learning algorithm that computes the embedding $y_i \in \mathbb{R}^d$ of each training sample $x_i \in \mathbb{R}^n$. Then a test sample x of unknown class label is first mapped to $\mathbb{R}^d$ via an interpolation function $f : \mathbb{R}^n \to \mathbb{R}^d$. The following main result from [37] gives a bound on the classification error, when the estimate $\hat{C}(x)$ of the class label $C(x)$ of x is estimated via the nearest-neighbor classification in R^d as $\hat{C}(x) = C(x_i)$, where

$$i = \arg\min_j \|y_j - f(x)\|.$$

Theorem 1. *Let $X = \{x_i\}_{i=1}^N \subset \mathbb{R}^n$ be a set of training samples such that each x_i is drawn i.i.d. from one of the probability measures $\{\nu_m\}_{m=1}^M$, with ν_m denoting the probability measure of the m-th class. Let $Y = \{y_i\}_{i=1}^N$ be an embedding of X in $\mathbb{R}^d$ such that*

$$\|y_i - y_j\| < D_\delta, \text{ if } \|x_i - x_j\| \leqslant \delta \text{ and } C(x_i) = C(x_j)$$
$$\|y_i - y_j\| > \gamma, \text{ if } C(x_i) \neq C(x_j).$$

For given $\epsilon > 0$ and $\delta > 0$, let $f : \mathbb{R}^n \to \mathbb{R}^d$ be a Lipschitz-continuous interpolation function with constant L, which maps each x_i to $f(x_i) = y_i$, such that

$$L\delta + \sqrt{d}\epsilon + D_{2\delta} \leqslant \frac{\gamma}{2}. \tag{3.1}$$

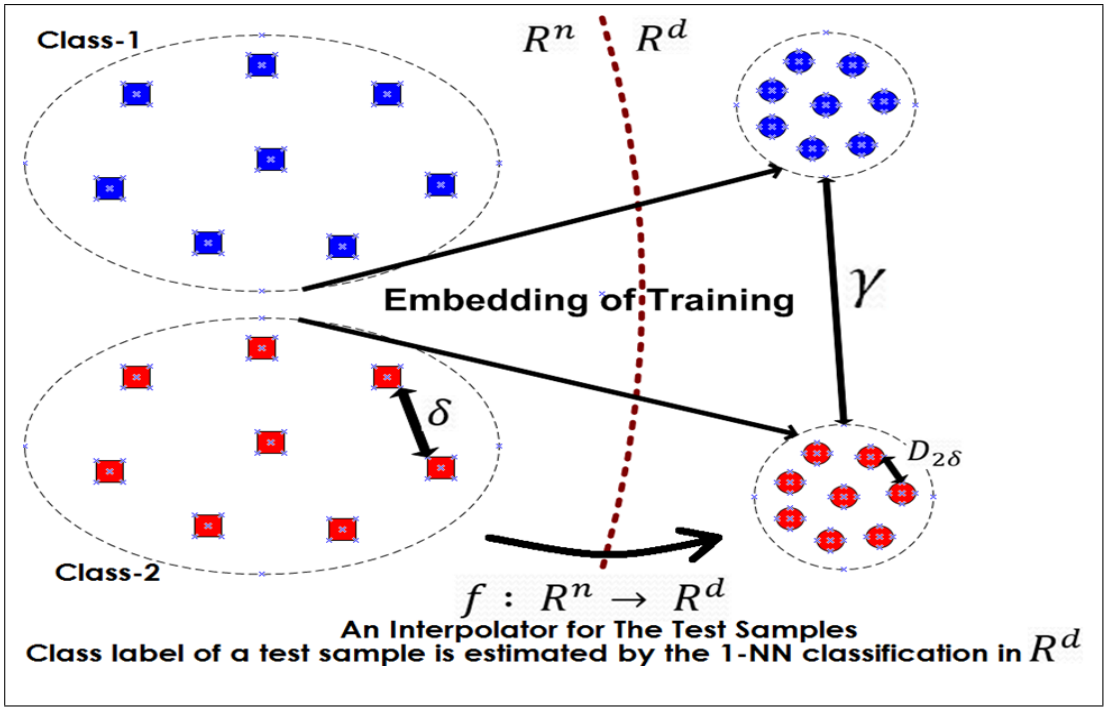


Figure 3.1: An illustration for Theorem 1

Consider a test sample x randomly drawn according to the probability measure ν_m of class m. For any $Q > 0$, if X contains at least N_m training samples from the m-th class drawn i.i.d. from ν_m such that

$$N_m > \frac{Q}{\eta_{m,\delta}}$$

then the probability of correctly classifying x with nearest-neighbor classification in $\mathbb{R}^d$ is lower bounded as

$$P\left(\hat{C}(x) = m\right) \geqslant 1 - \exp\left(-\frac{2\left(N_m\,\eta_{m,\delta} - Q\right)^2}{N_m}\right) - 2d\exp\left(-\frac{Q\,\epsilon^2}{2L^2\delta^2}\right). \quad (3.2)$$

Theorem 1 considers an embedding such that nearby training samples from the same class are mapped to nearby coordinates, while training samples from different classes are separated by a distance of at least γ in the low-dimensional domain of embedding. The parameter γ can be considered as the separation margin of the embedding. Then for such an embedding, the condition in (3.1) assumes an interpolator f that is sufficiently regular compared to the separation margin γ. An illustration of the parameters of the theorem is given in Figure 3.1. Finally, a probabilistic classification guarantee is given for this setting in (3.2), which states that the misclassification probability decreases exponentially with the number of samples. An extension of this result is also

presented in [37] which studies the performance of classification with a linear classifier in the low-dimensional domain. When a linear classifier is used in the domain of embedding, a necessary condition very similar to (3.1) is obtained that establishes a very similar relation between the interpolator regularity and the separation margin, and a similar bound on the misclassification error is obtained.

While most supervised manifold learning methods in the literature focus on achieving a large separation between the training samples from different classes in the embedding, the condition (3.1) in the above theoretical analysis points to a critical compromise that must be sought in supervised dimensionality reduction: Achieving high separation between different classes in the training set does not necessarily mean that the classifier will generalize well to test samples. The presence of a sufficiently regular interpolator is furthermore needed, so that the Lipschitz constant L of the interpolator remains below a threshold determined by the separation margin γ of the embedding. From this perspective, depending on the data distribution, increasing the separation too much has the risk of forcing the interpolator to be too irregular, which may in turn cause condition (3.1) to fail. What we propose to do in this thesis is to learn the embedding $\{y_i\}_{i=1}^N$ together with the interpolator f in view of the condition (3.1), which is detailed in the next section.

3.2 Problem Formulation

Given training points $X = \{x_i\}_{i=1}^N \subset \mathbb{R}^n$ from M classes, our purpose is to learn an embedding of data $Y = \{y_i\}_{i=1}^N \subset \mathbb{R}^d$ together with a continuous interpolation function $f : \mathbb{R}^n \to \mathbb{R}^d$, such that $f(x_i) = y_i$. The interpolator f will then be used to classify new test points x by mapping x to the low-dimensional domain $\mathbb{R}^d$ as $f(x)$, so that examining $f(x) \in \mathbb{R}^d$ with respect to the embedding Y of the training points with known class labels provides an estimate of the class label of x.

Our method relies on the theoretical results presented in Section 3.1. Recall from Theorem 1 that, a necessary condition to obtain good generalization performance is

$$L\delta + \sqrt{d}\epsilon + D_{2\delta} \leq \frac{\gamma}{2}.$$

In the sequel, we formulate a manifold learning problem in view of this condition,

whose purpose is to make the the Lipschitz constant L of the interpolator and the distance $D_{2\delta}$ between neighboring points from the same class as small as possible, while making the separation γ between different classes as large as possible, in order to increase the chances that the above condition be met.

Let $f(x) = [f^1(x) \ \dots \ f^d(x)] \in \mathbb{R}^d$, where $f^k(x)$ is the k-th dimension of $f(x)$, with $f^k : \mathbb{R}^n \to \mathbb{R}$. We propose to choose the function f a radial basis function (RBF) interpolator as RBF interpolators, which are a well-studied family of functions [8], [27] with many desirable properties such as smoothness and adjustable spread around anchor points. Hence, each component f^k of f is of the form

$$f^k(x) = \sum_{i=1}^{N} c_i^k \phi(\|x - x_i\|) \tag{3.3}$$

where $\phi : \mathbb{R} \to \mathbb{R}^+$ is an RBF kernel, c_i^k are the coefficients, and x_i are the kernel centers. A common choice for the RBF kernel is the Gaussian kernel $\phi(r) = e^{-r^2/\sigma^2}$, which we also adopt in this work. Under this setting, we now examine our three entities of interest, namely the regularity of the interpolator, the distance between neighboring points from the same class and the separation between different classes.

Interpolator regularity. We begin with deriving a Lipschitz constant for f in terms of the function parameters. We first derive the Lipschitz constant L_ϕ of the RBF kernel, i.e., we seek a constant L_ϕ such that

$$|\phi(u) - \phi(v)| \leqslant L_\phi |u - v|$$

for all $u, v \in \mathbb{R}$. It is easy to show that the derivative of $\phi(r) = e^{-r^2/\sigma^2}$ takes is maximum magnitude at $r = \sigma/\sqrt{2}$, so that for all r,

$$\left| \frac{d\phi(r)}{dr} \right| \leqslant \left| \left. \frac{d\phi(t)}{dt} \right|_{t=\sigma/\sqrt{2}} \right| = \sqrt{2} e^{-\frac{1}{2}} \sigma^{-1}.$$

This upper bound on the derivative magnitude provides the Lipschitz constant $L_\phi = \sqrt{2} e^{-\frac{1}{2}} \sigma^{-1}$ for the Gaussian kernel. Then we derive the Lipschitz constant of $f(x)$ as follows. First, observe that

$$\begin{aligned} |f^k(u) - f^k(v)| &= \left| \sum_{i=1}^{N} c_i^k \big(\phi(\|u - x_i\|) - \phi(\|v - x_i\|) \big) \right| \\ &\leqslant \sum_{i=1}^{N} |c_i^k| \ |\phi(\|u - x_i\|) - \phi(\|v - x_i\|)| \end{aligned}$$

where the second term inside the sum can be upper bounded as

$$|\phi(\|u - x_i\|) - \phi(\|v - x_i\|)| \leqslant L_\phi \big| \|u - x_i\| - \|v - x_i\| \big| \leqslant L_\phi \|u - v\|.$$

This gives in the above equation

$$|f^k(u) - f^k(v)| \leqslant L_\phi \sum_{i=1}^{N} |c_i^k| \|u - v\| \leqslant L_\phi \|c^k\|_1 \|u - v\| \leqslant \sqrt{N} L_\phi \, \|c^k\| \|u - v\|$$

where $c^k = [c_1^k \dots c_N^k]^T$ denotes the coefficient vector of the function f^k and $\| \cdot \|_1$ is the ℓ_1-norm of a vector. Then we have

$$\begin{aligned} \|f(u) - f(v)\| &= \sqrt{\sum_{k=1}^{d} |f^k(u) - f^k(v)|^2} \leqslant \sqrt{\sum_{k=1}^{d} N L_\phi^2 \|c^k\|^2 \|u - v\|^2} \\ &= \sqrt{N} L_\phi \|u - v\| \sqrt{\sum_{k=1}^{d} \|c^k\|^2} = \sqrt{N} L_\phi \|C\|_F \|u - v\| \end{aligned}$$

where $\| \cdot \|_F$ denotes the Frobenius norm of a matrix and C is the matrix consisting of the RBF coefficients

$$C = \begin{pmatrix} c_1^1 & . & . & c_1^d \\ . & . & . & . \\ c_N^1 & . & . & c_N^d \end{pmatrix}.$$

Hence, we get

$$\|f(u) - f(v)\| \leqslant L \|u - v\|$$

where $L := \sqrt{N} L_\phi \|C\|_F$ is the Lipschitz constant of $f(x)$. Then, in order to minimize the Lipschitz constant of $f(x)$, we need to minimize the terms L_ϕ and $\|C\|_F$. From the form (3.3) of the interpolator components and the fact that the interpolator values at training points must correspond to the coordinates of the embedding $y_i = f(x_i)$, we get the relation

$$\Psi C = Y$$

where Ψ is the matrix consisting of the values of the RBF kernels

$$\Psi = \begin{pmatrix} \phi(\|x_1 - x_1\|) & \phi(\|x_1 - x_2\|) & \cdots & \phi(\|x_1 - x_N\|) \\ \cdots & & & \cdots \\ \phi(\|x_N - x_1\|) & \phi(\|x_N - x_2\|) & \cdots & \phi(\|x_N - x_N\|) \end{pmatrix}$$

and Y is an $N \times d$ matrix consisting of the coordinates of the embeddings of the training samples

$$Y = [y_1 \; y_2 \; \dots \; y_N]^T.$$

Then the coefficient matrix is given by $C = \Psi^{-1}Y$, so that

$$\|C\|_F^2 = \|\Psi^{-1}Y\|_F^2 = \mathrm{tr}(Y^T\Psi^{-2}Y). \tag{3.4}$$

In order to keep the Lipschitz constant $L = \sqrt{N}L_\phi\|C\|_F$ of the interpolator small in the learnt embedding, we need to keep both the Lipschitz constant L_ϕ of the Gaussian kernel and the norm $\|C\|_F$ of the coefficient matrix small. Using the expression of $\|C\|_F^2$ in (3.4) and recalling that $L_\phi = \sqrt{2}e^{-\frac{1}{2}}\sigma^{-1}$, we thus propose to minimize the following objective for controlling the interpolator regularity

$$\min\ \mathrm{tr}(Y^T\Psi^{-2}Y) + \frac{\mu}{\sigma^2} \tag{3.5}$$

where μ is a weight parameter. The objective is chosen proportionally to the squares of the terms $\|C\|_F$ and L_ϕ instead of themselves, due to the convenience of the analytical expression obtained for $\|C\|_F^2$ in (3.4).

Distance between neighboring points from the same class. Recall from Theorem 1 that the condition (3.1) required for good classification performance necessitates the term $D_{2\delta}$ to be sufficiently small, where D_δ is an upper bound on the distance between the embeddings of nearby samples, i.e., $\|y_i - y_j\| < D_\delta$ whenever $\|x_i - x_j\| \leqslant \delta$. It is not easy to study the distance $\|y_i - y_j\|$ in relation with the ambient space distance $\|x_i - x_j\|$ for each pair of samples x_i, x_j. Nevertheless, we adopt a constructive solution here and relax this problem to the minimization of the distance between the embeddings of nearby points from the same class. The total distance between the embeddings of neighboring points from the same class, weighted by the edge weights, is given by

$$\sum_{x_i,x_j:\ C(x_i)=C(x_j)} \|y_i - y_j\|^2 w_{ij} = \mathrm{tr}(Y^T L_w Y)$$

where W_w is the within-class weight matrix containing edge weights w_{ij} only between neighboring samples from the same class, and $L_w = D_w - W_w$ is the corresponding within-class Laplacian matrix. Hence, the objective

$$\min\ \mathrm{tr}(Y^T L_w Y) \tag{3.6}$$

used in several previous works as discussed in Chapter 2 is an appropriate choice for our purpose.

Separation between samples from different classes. The last entity to be examined in view of the condition (3.1) is the separation margin γ. In order to satisfy the condition (3.1) more easily, the separation between the samples between different classes must be sufficiently high. Although the margin γ denotes a lower bound for the distance $\|y_i - y_j\|$ between any pair of samples from different classes in Theorem 1, the examination of the minimum value of $\|y_i - y_j\|$ for all pairs of samples is a relatively hard problem. We propose to relax this and evaluate the total distance between the embeddings of different-class samples in this study. Hence, in order to increase the separation margin γ, we propose to maximize

$$\sum_{C(x_i)\neq C(x_j)} \|y_i - y_j\|^2 = \mathrm{tr}(Y^T L_b Y)$$

where W_b is a between-class weight matrix given by

$$W_b(i,j) = \begin{cases} 1 & \text{if } C(x_i) \neq C(x_j) \\ 0 & \text{if } C(x_i) = C(x_j) \end{cases}$$

and $L_b = D_b - W_b$ is the corresponding between-class Laplacian matrix. Thus, the maximization of the separation margin is represented by the objective function

$$\max \mathrm{tr}(Y^T L_b Y). \tag{3.7}$$

Overall formulation. Now, bringing together the objective functions presented in (3.5), (3.6), and (3.7), we propose to solve the following optimization problem for learning an embedding Y together with its corresponding interpolator:

$$\min_{Y,\sigma} \mathrm{tr}(Y^T L_w Y) - \mu_1 \mathrm{tr}(Y^T L_b Y) + \mu_2 \mathrm{tr}(Y^T \Psi^{-2} Y) + \mu_3/\sigma^2, \text{ s.t. } Y^T Y = I \tag{3.8}$$

Here μ_1, μ_2, and μ_3 are positive weights that balance the different terms in the objective function, and the normalization condition $Y^T Y = I$ is imposed in order to prevent solutions with arbitrarily small embedding coordinates that trivially minimize the objective.

3.3 The algorithm

We minimize the proposed objective function (3.8) with an alternating, iterative optimization algorithm. In each iteration, we first fix the scale parameter σ and optimize the embedding coordinates Y, which is then followed by fixing Y and optimizing σ.

Optimization of Y. When the scale parameter σ is fixed, the minimization of the objective (3.8) is equivalent to the following optimization problem

$$\begin{aligned} Y^* &= \arg\min_{Y} \ \mathrm{tr}(Y^T L_w Y) - \mu_1 \mathrm{tr}(Y^T L_b Y) + \mu_2 \mathrm{tr}(Y^T \Psi^{-2} Y) \text{ s.t. } Y^T Y = I \\ &= \arg\min_{Y} \ \mathrm{tr}\left(Y^T (L_w - \mu_1 L_b + \mu_2 \Psi^{-2}) Y\right) \text{ s.t. } Y^T Y = I. \end{aligned} \tag{3.9}$$

The solution to this problem is given by the $N \times d$ matrix Y^* whose k-th column consists of the eigenvector of the matrix

$$A = L_w - \mu_1 L_b + \mu_2 \Psi^{-2} \tag{3.10}$$

that corresponds to its k-th smallest eigenvalue, for $k = 1, \ldots, d$. Note that as A is a symmetric matrix, its eigenvectors are orthonormal, hence the learnt embedding satisfies the condition $Y^T Y = I$.

Optimization of σ. Note that the dependence of the objective function (3.8) on the scale parameter σ is through its third term $\mu_2 \mathrm{tr}(Y^T \Psi^{-2} Y)$ and fourth term μ_3/σ^2. Hence, when the embedding Y is fixed, the optimization of the objective can be achieved by solving

$$\sigma^* = \arg\min_{\sigma} \ \mu_2 \mathrm{tr}(Y^T \Psi^{-2} Y) + \frac{\mu_3}{\sigma^2}. \tag{3.11}$$

The objective in (3.11) is not a convex function of σ in general. Nevertheless, a useful observation is the following: As the entries of the matrix Ψ consist of the RBF kernel terms

$$\phi(\|x_i - x_j\|) = \exp\left(-\frac{\|x_i - x_j\|^2}{\sigma^2}\right)$$

the matrix Ψ and its inverse Ψ^{-1} have poor conditioning when σ takes arbitrarily large values. Hence, the first term $\mathrm{tr}(Y^T \Psi^{-2} Y)$ of in (3.11) increases with increasing large values of σ. On the other hand, the term σ^{-2} approaches infinity as σ approaches

Algorithm 1 Nonlinear Supervised Smooth Embedding (NSSE)

1: **Input:**

$X = \{x_i\}_{i=1}^N \subset \mathbb{R}^n$: Training samples with known class labels

d: Embedding dimension

μ_1, μ_2, μ_3: Weight parameters

2: **Initialization:** Set kernel scale σ^* to a typical positive value.

3: **repeat**

4: Fix $\sigma = \sigma^*$ and optimize Y by solving

$Y^* = \arg\min_Y \operatorname{tr}\left(Y^T(L_w - \mu_1 L_b + \mu_2 \Psi^{-2})Y\right)$ s.t. $Y^T Y = I$

5: Fix $Y = Y^*$ and optimize σ by solving

$\sigma^* = \arg\min_\sigma \ \mu_2 \operatorname{tr}(Y^T \Psi^{-2} Y) + \mu_3 \sigma^{-2}$

6: **until** Objective function in (3.8) is stabilized

7: Compute interpolator coefficients as $C = \Psi^{-1} Y$.

8: **Output**:

$Y = \{y_i\}_{i=1}^N \subset \mathbb{R}^d$: Embedding of training samples

$f : \mathbb{R}^n \to \mathbb{R}^d$: Interpolation function

0. These observations imply that that there exists a positive kernel scale $\sigma^* > 0$ that minimizes the objective (3.11). As the problem (3.11) requires the optimization of a single parameter σ and has a differentiable objective, the optimal value σ^* can be computed in practice via a basic descent-type optimization algorithm or with exhaustive search within a typical range of σ values.

These steps for the alternating optimization of Y and σ are applied successively until the stabilization of the objective function. Note that if μ_1 is chosen sufficiently small to make the matrix A in (3.10) positive semi definite, the overall objective function (3.8) is positive. In this case, since both of the alternating optimization steps in (3.9) and (3.11) bring updates that cannot increase the objective function in each iteration, being bounded from below, the objective function is guaranteed to converge.

Once the embedding Y of the training points and the kernel scale parameter σ are computed in this way, the interpolation function f is simply obtained as in (3.3) by computing the coefficients c_i^k as $C = \Psi^{-1} Y$. We call the proposed method Nonlinear Supervised Smooth Embedding (NSSE) and give its description in Algorithm 1.

3.4 Complexity of the Proposed Method

We now analyze the computational complexity of the NSSE method. The proposed algorithm is composed of three main stages, which are the initialization stage (calculation of the L_w and L_b matrices), the main loop between steps 3 and 6 of Algorithm 1, and the finalization stage in step 7.

In the initialization step, the complexity of the computation of L_w and L_b is mainly determined by the complexity of computing the within-class and between-class weight matrices W_w and W_b, which is of $O(nN^2)$.

We next consider the main loop of the algorithm. The matrix Ψ in step 4 can be calculated with complexity $O(nN^2)$ and it is inverted with complexity $O(N^3)$ to obtain Ψ^{-1}. As a result, the computation of Ψ^{-2} is of complexity $O(nN^2) + O(N^3)$. In order to find Y^* in step 4, the eigenvectors of $L_w - \mu_1 L_b + \mu_2 \Psi^{-2}$ should be found, which is of complexity $O(N^3)$. Consequently, the total complexity of step 4 is $O(nN^2) + O(N^3)$. In step 5, the expression $\mu_2 \mathrm{tr}(Y^T \Psi^{-2} Y) + \mu_3 \sigma^{-2}$ must be computed repeatedly to find σ^*, which is of complexity $O(N^3)$. Hence, the complexity of the main loop of the algorithm is found as $O(nN^2) + O(N^3)$.

In step 7, the complexity of the calculation of Ψ^{-1} is $O(N^3)$, and the matrix product $\Psi^{-1}Y$ is of complexity $O(dN^2)$. We may assume $d \ll N$, which then gives the complexity of step 7 as of $O(N^3)$. Combining this with the previous stages, the overall complexity of the algorithm is found as $O(nN^2) + O(N^3)$.

CHAPTER 4

EXPERIMENTAL RESULTS

In this chapter, we evaluate the performance of the proposed NSSE method on six real data sets. We first describe the used datasets, then study the behavior of the NSSE algorithm throughout the iterations, and then compare the performance of NSSE with that of other supervised manifold learning algorithms and traditional classifiers.

4.1 Datasets and Experimentation Settings

We experiment on the data sets listed below. Some sample images from one class of each data set are presented in Figure 4.1.

Yale Face Database. The data set consists of 2242 greyscale face images of 38 different subjects, where each subject has 59 images [19]. All images are taken from a single viewpoint with variations in the lighting angles and lighting rates.

COIL-20 Database. The Columbia Object Image Library database consists of 1440 grayscale images of 20 different objects, where each object has 72 images captured by rotation increments of 5 degrees [23].

ORL Database. The database consists of a total of 400 images, with 10 images of each one of the 40 subjects taken in an upright, frontal position [31]. The images contain variations in the the lighting, facial expressions and facial details such as glasses.

FEI Database. The FEI database is a Brazilian face database containing a total of 2800 images, with 14 images for each one of the 200 subjects taken in an upright

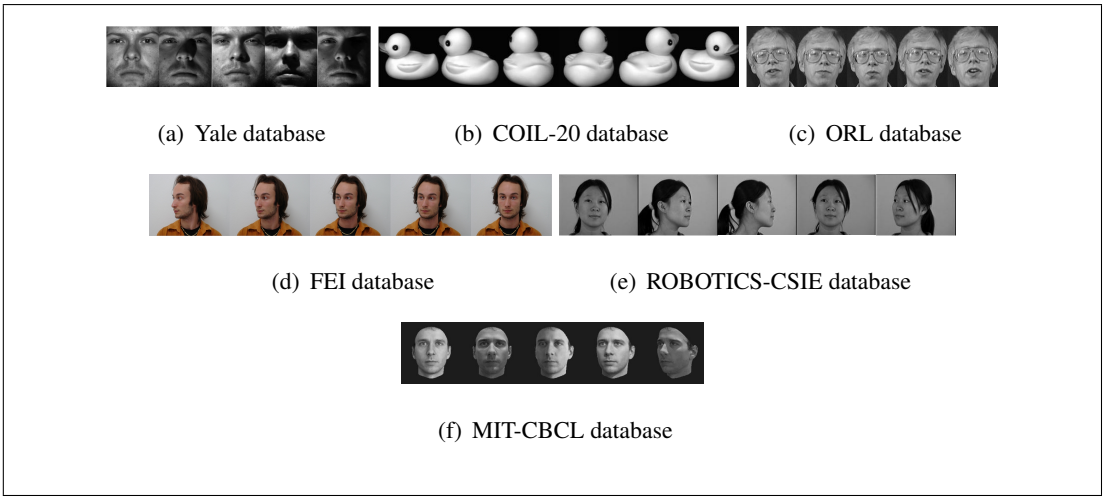


Figure 4.1: Sample images from one class of the used databases

frontal position with profile rotation of up to about 180 degrees and scale variation of about 10% [35]. We experiment on 50 classes from this database.

ROBOTICS-CSIE Database. The database contains a total of 3330 grayscale face images of 90 subjects, with 37 images for each subject captured under rotation increments of 5 degrees [3]. We experiment on 40 classes from this database.

MIT-CBCL Database. The database contains face images of 10 subjects [2]. We experiment on a total of 5240 images, with 524 images per subject captured under rotations of up to 30 degrees and varying illumination conditions.

We experiment on greyscale versions of the images resized to around 25×25 pixels. All experiments are conducted in a supervised setup, by randomly separating the images into a training set and a test set in each repetition of the experiment. In all experiments, the proposed NSSE algorithm is evaluated in a setting where the training images are used to learn a continuous embedding into a low-dimensional domain. The test images are then mapped to the domain of embedding via the learnt interpolator and their class labels are estimated via nearest neighbor classification in the low-dimensional domain. The graph edge weights are set with a Gaussian kernel. In all experiments, the weight parameters μ_1, μ_2, and μ_3 of NSSE are set with cross-validation. According to results of the cross-validation, in order to obtain good results, μ values should be selected around the values given in Table 4.1.

Table 4.1: μ values found from cross-validation

Databases	μ_1	μ_2	μ_3
YALE	40	0.0001	4
COIL-20	3	0.01	4
MIT-CBCL	400	0.0003	1.2
FEI	400	0.0005	2
ORL	900	0.005	0.3
ROBOTICS-CSIE	400	0.008	3

4.2 Study of the iterative optimization procedure

In this first experiment, we study the iterative optimization procedure employed in the proposed method. As discussed in Section 3.3, the NSSE algorithm follows an alternating optimization scheme by minimizing the objective function in (3.8) first with respect to the embedding Y of the training samples, and then the scale parameter σ of the RBF kernels.

The results given in Figure 4.2 are obtained on the FEI face data set, where an embedding into a $d = 10$ dimensional domain is computed using a total of 100 training samples. Figure 4.2(a) shows the variation of the objective function in (3.8) throughout the iterations. Although the proposed alternating optimization procedure is not theoretically guaranteed to find the global optimum of the objective, it is observed from the figure that the proposed scheme can effectively minimize the objective function, which converges in a small number of iterations. The misclassification rates of the test images in percentage are reported in Figure 4.2(b) obtained with the embeddings and interpolators computed in each iteration. The results show that the progressive update of the continuous embedding throughout the iterations improves the classification performance. The comparison of the plots in Figures 4.2(a) and 4.2(b) reveals that the variations of the objective function and the misclassification rate throughout the iterations are quite similar. This suggests that the choice of the objective function in (3.8), motivated by theoretical bounds, indeed matches the actual classification error. Figure 4.2(c) shows the evolution of the RBF kernel scale parameter σ throughout

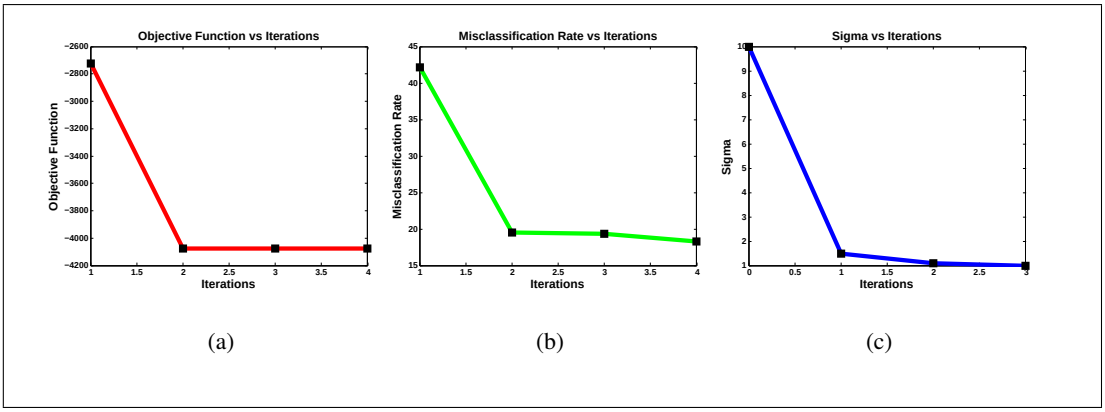


Figure 4.2: Algorithm performance throughout the iterative optimization procedure when initialized with a high RBF kernel scale. (a) Convergence of the objective function (b) Stabilization of the misclassification rate (c) Stabilization of the RBF kernel scale σ

the iterations. The RBF kernel scale σ is deliberately initialized with a too high value in this experiment in order to study the effect of the initial conditions on the algorithm performance. Despite the initialization of σ with a too large value, the iterative minimization of the objective gradually pulls the kernel scale towards a favorable value that improves the classification performance.

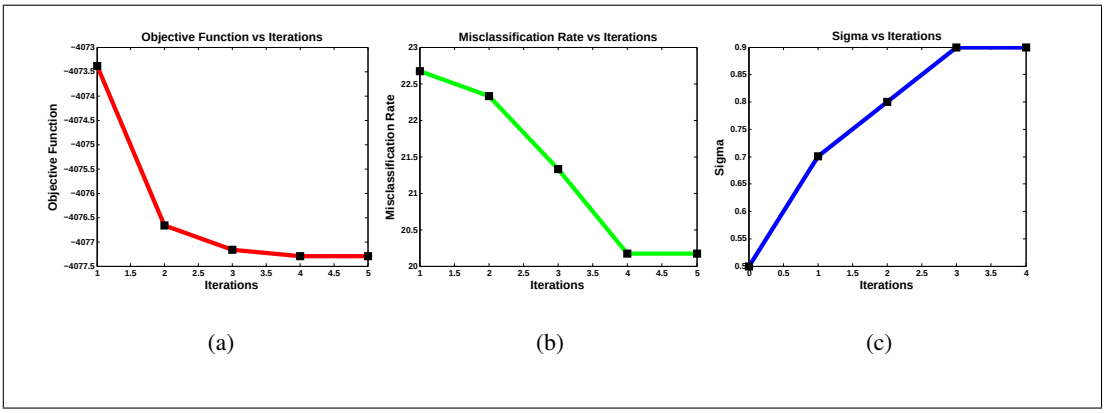


Figure 4.3: Algorithm performance throughout the iterative optimization procedure when initialized with a low RBF kernel scale. (a) Convergence of the objective function (b) Stabilization of the misclassification rate (c) Stabilization of the RBF kernel scale σ

The same experiment is repeated in Figure 4.3, by initializing the RBF kernel scale this time with a small value. It is observed that the RBF scale σ is effectively optimized throughout the iterations towards a larger value, which gradually decreases

the objective function and improves the classification accuracy. These results suggest that the algorithm performance is not affected much by the initialization of the RBF kernel scale. We have obtained similar results on the other data sets and under different choices of the parameters such as the number of training samples, which we skip here for brevity.

4.3 Variation of the classification performance with the embedding dimension

We now study the classification performance of the proposed algorithm in relation with the dimension d of the embedding. The proposed NSSE method is compared with the algorithms listed below.

(SUPLAP) The Supervised Laplacian Eigenmaps method proposed in [29] computes a nonlinear low-dimensional embedding of the training samples by minimizing the objective in (2.69). We extend the embedding of the training samples given by the SUPLAP method to the whole space via an RBF interpolator of the same form as in NSSE. We then embed the test samples into the low-dimensional domain with this interpolation function.

(LFDA) The Local Fisher Discriminant Analysis method proposed in [33] is a supervised manifold learning algorithm that computes a linear embedding by optimizing a Fisher-type cost with additional locality preservation objectives.

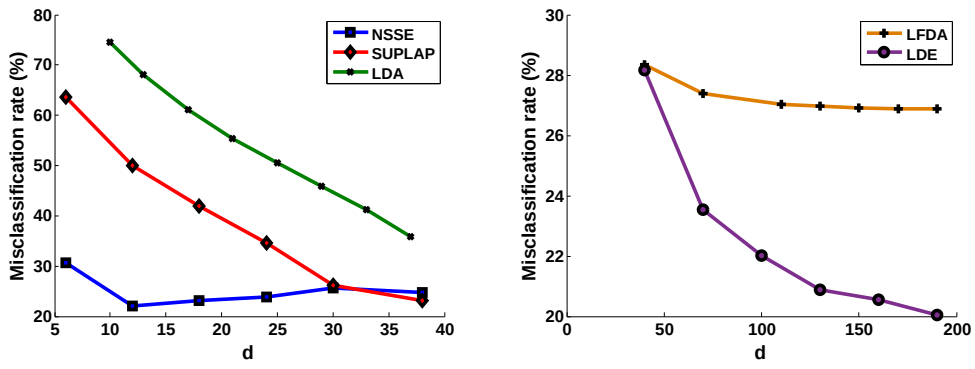
(LDE) The Local Discriminant Embedding method [14] is a manifold learning method that optimizes a similar objective as in the SUPLAP method; however, learns a linear projection.

(LDA) Linear Discriminant Analysis is a classical dimensionality reduction technique that maximizes the between-class scatter while minimizing the within-class scatter.

These algorithms are described in more detail in Chapter 2. The dimensionality reduction methods are applied on training samples to compute a d-dimensional embedding, which is then used to classify test samples via nearest neighbor classification in the domain of embedding. The algorithms are evaluated for a range of d values. The parameters of the other methods in comparison are adjusted to attain their best

performance.

The variation of the misclassification rates of test samples in percentage with the dimension d of the embedding is presented in Figures 4.4 - 4.13, for the Yale, COIL-20, MIT-CBCL, FEI, ORL and the ROBOTICS-CSIE databases. The number of training images used in the computation of the embeddings are indicated in the figure captions for each experiment. The results are the average of 20 random realizations of the experiments with different training and test sets. Most of the tested methods are based on solving a generalized eigenvalue problem and the rank of the involved matrices may be different for each method depending on the number of training samples and the number of classes. Hence, the maximum possible dimension of the embedding may vary between different methods, as well as the best range of dimensions where the methods perform well. For this reason, the results on each data set are grouped into two figures with different d ranges for better visual clarity.

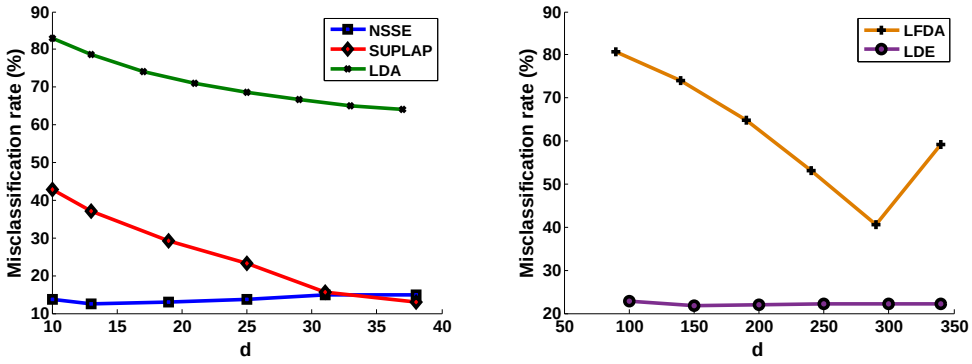


(a) NSSE, SUPLAP and LDA

(b) LFDA and LDE

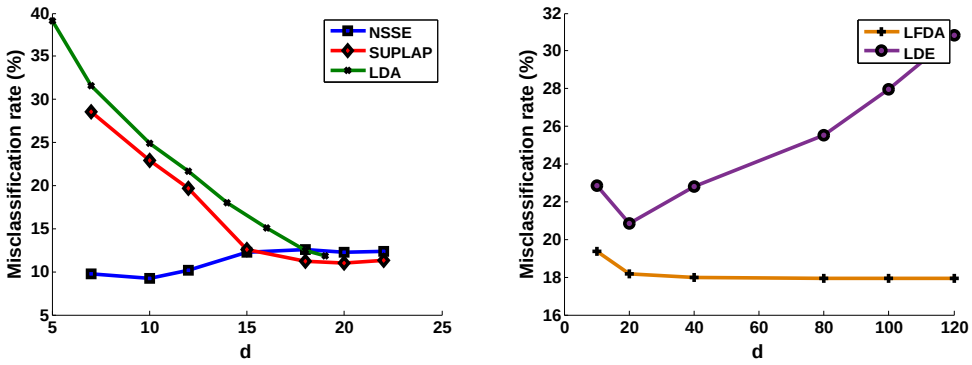
Figure 4.4: Variation of the misclassification rate with the embedding dimension in Yale dataset, with 6 training samples per class

The results in Figures 4.4-4.13 show that the classification righteousness of the proposed NSSE algorithm compares quite favorably to those of the other methods, as NSSE often yields the smallest misclassification rate. The misclassification rate of LDA is observed to decrease monotonically with the dimension d and its best performance is attained when d reaches the number of classes. The LDE and LFDA algorithms exhibit their best performances at much higher dimensions compared to the other algorithms. The error rates of these algorithms may decrease as the embed-



(a) NSSE, SUPLAP and LDA (b) LFDA and LDE

Figure 4.5: Variation of the misclassification rate with the embedding dimension in Yale dataset, with 10 training samples per class

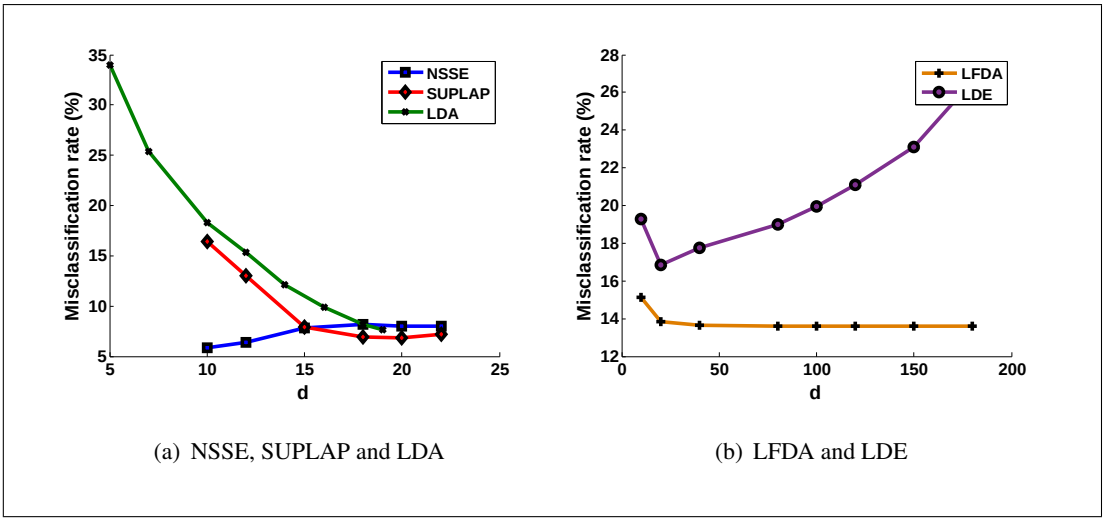


(a) NSSE, SUPLAP and LDA (b) LFDA and LDE

Figure 4.6: Variation of the misclassification rate with the embedding dimension in Coil20 dataset, with 7 training samples per class

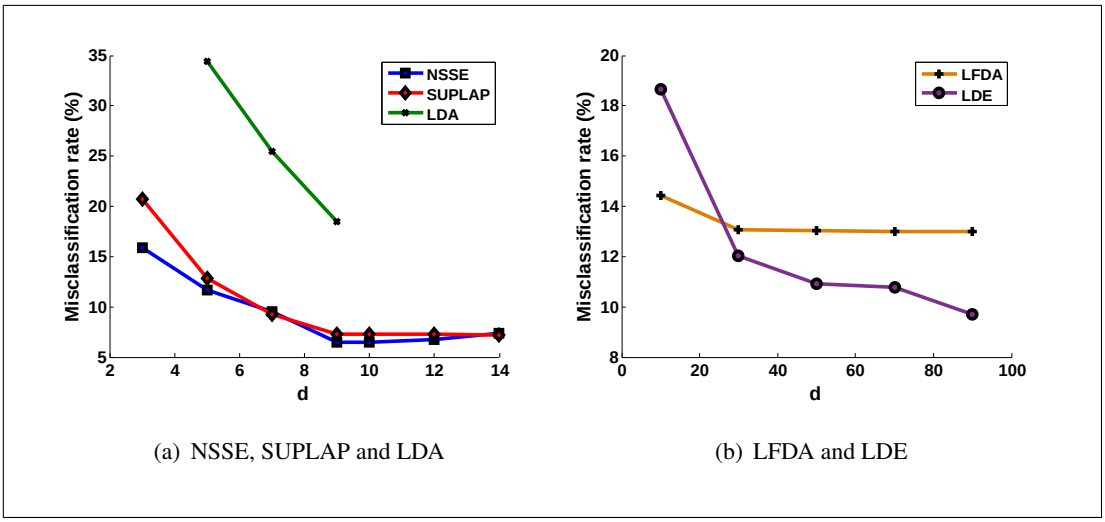
ding dimension increases; however, in some datasets a local optimum for d can also be observed.

Among all methods, the nonlinear NSSE and SUPLAP methods often perform better than the linear LDA, LFDA, and LDE methods. This shows that the flexibility of nonlinear methods when learning an embedding is likely to bring an advantage in computing better representations for data. It is then interesting to compare the performances of the two nonlinear methods; NSSE and SUPLAP. The SUPLAP algorithm attains its best performance when the dimension d of the embedding is close



(a) NSSE, SUPLAP and LDA (b) LFDA and LDE

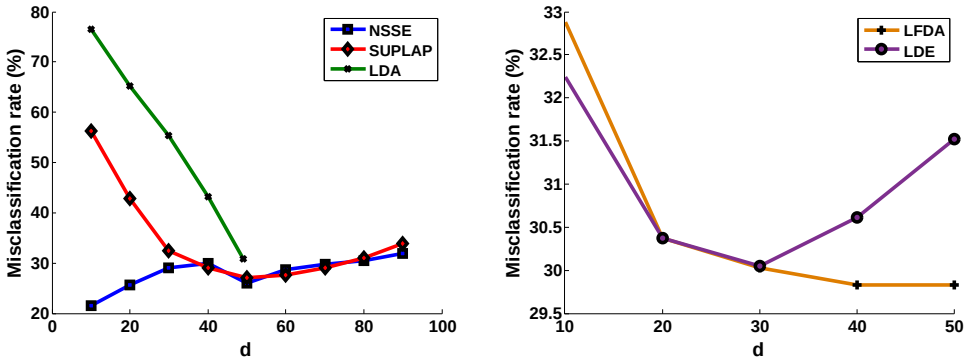
Figure 4.7: Variation of the misclassification rate with the embedding dimension in Coil20 dataset, with 10 training samples per class



(a) NSSE, SUPLAP and LDA (b) LFDA and LDE

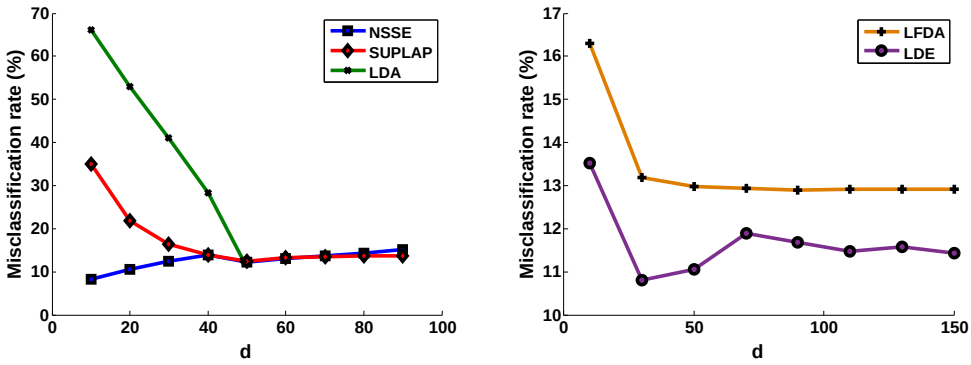
Figure 4.8: Variation of the misclassification rate with the embedding dimension in MITCBCL dataset, with 10 training samples per class

to the number of classes, while the optimum value of d for the proposed NSSE algorithm is smaller in most data sets. Interestingly, the optimal dimension of NSSE is much smaller than that of SUPLAP in data sets with a low intrinsic dimension such as COIL-20, FEI, and ROBOTICS-CSIE, which are generated by the variation of only one or two camera angle parameters. Similarly, in data sets of larger intrinsic dimension such as MIT-CBCL due to several pose and lighting parameters, the optimal dimension of NSSE is higher and closer to that of SUPLAP. This may suggest that the embedding computed with NSSE tries to capture the intrinsic geometry of



(a) NSSE, SUPLAP and LDA (b) LFDA and LDE

Figure 4.9: Variation of the misclassification rate with the embedding dimension in FEI dataset, with 2 training samples per class

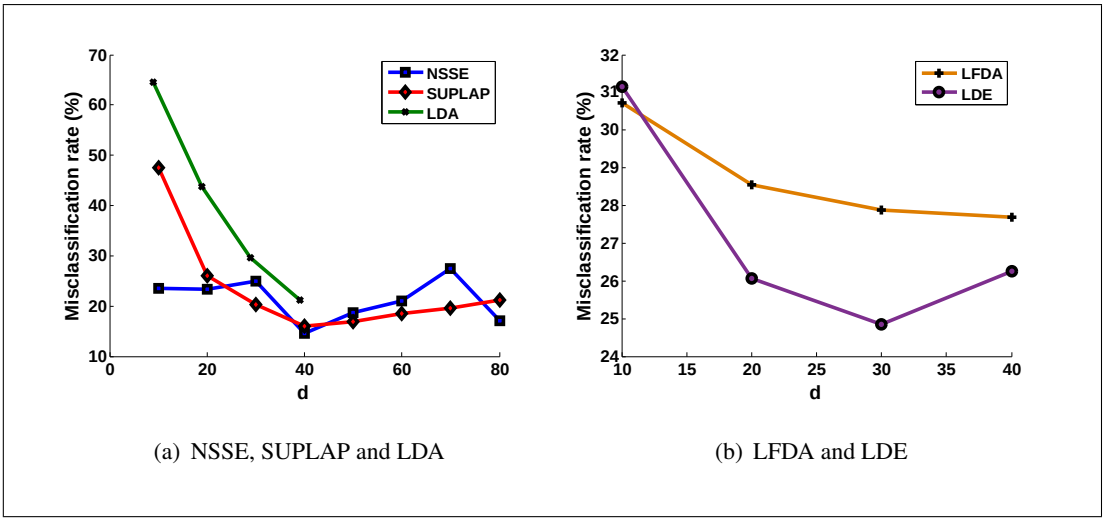


(a) NSSE, SUPLAP and LDA (b) LFDA and LDE

Figure 4.10: Variation of the misclassification rate with the embedding dimension in FEI dataset, with 4 training samples per class

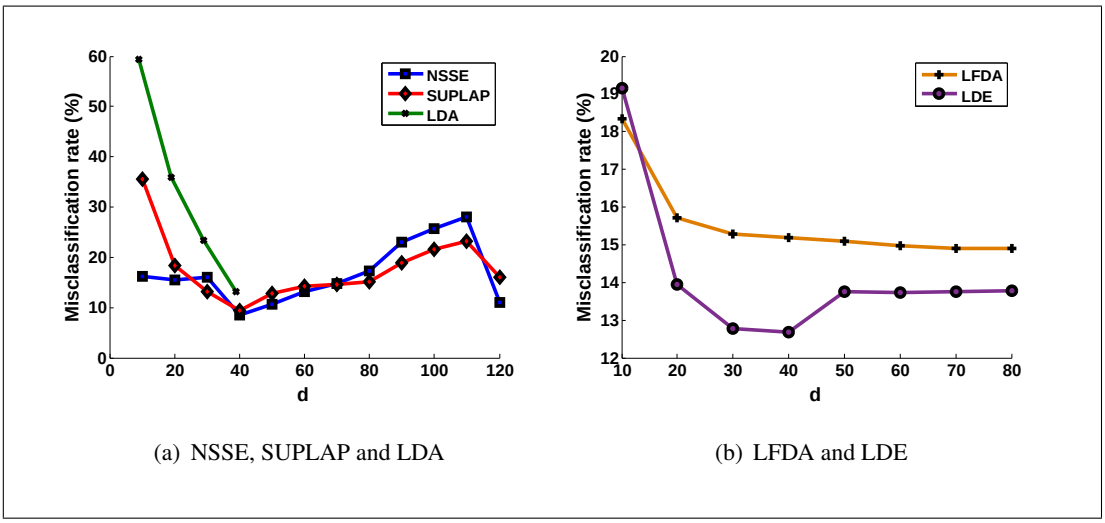
data and provides a better representation when the embedding dimension is chosen proportionally to the intrinsic dimension of data.

The reduction of the embedding dimension is desirable especially regarding the complexity of the classification of test samples in a practical application. Another advantage of NSSE over SUPLAP is that NSSE is less sensitive to the choice of the dimension, as the misclassification performance is less affected for non-optimal values of d. Such benefits of the proposed NSSE algorithm mainly result from the fact that the Lipschitz continuity of the interpolator is imposed in the learning objective.



(a) NSSE, SUPLAP and LDA (b) LFDA and LDE

Figure 4.11: Variation of the misclassification rate with the embedding dimension in ORL dataset, with 2 training samples per class

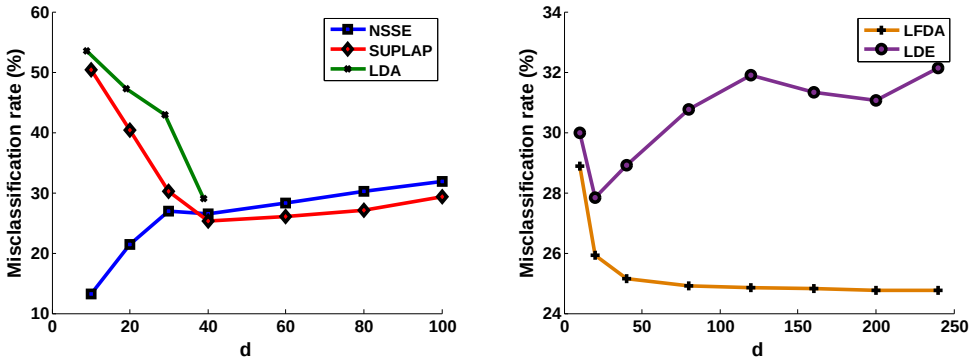


(a) NSSE, SUPLAP and LDA (b) LFDA and LDE

Figure 4.12: Variation of the misclassification rate with the embedding dimension in ORL dataset, with 3 training samples per class

Consequently, the training samples are embedded more evenly in the low-dimensional space so as to allow the construction of a regular interpolator, which in return reduces the required number of dimensions or the sensitivity to the non-optimal choice of d.

In fact, Figure 4.14 provides a visual comparison of the embeddings obtained with the NSSE and the SUPLAP algorithms. Panels (a) and (b) show the two-dimensional embeddings of 70 training samples from 10 classes of the ROBOTICS-CSIE data set, respectively with the NSSE and the SUPLAP methods. The embeddings of training samples look similar between the two methods, although different classes are more



(a) NSSE, SUPLAP and LDA

(b) LFDA and LDE

Figure 4.13: Variation of the misclassification rate with the embedding dimension in ROBOTICS-CSIE dataset, with 7 training samples per class

regularly spaced in NSSE. The performance difference between these two methods becomes much clearer when the embeddings of the test samples in panels (c) and (d) are observed. Even at this very small embedding dimension of 2, the NSSE method separates test samples from different classes much more successfully than SUPLAP, which is due to the inclusion of the interpolator parameters in the learning objective in order to attain good generalization performance.

The embedding of training and test data onto a 2-dimensional space is provided also in Figure 4.15 but a previously unseen class is also embedded beside the test data at this time. Panels (a) and (b) are deduced from 70 training samples from 10 classes of the ROBOTICS-CSIE data set. A previously unseen new class is embedded with the learnt interpolator and the 2-dimensional embedding of the new class is shown as black filled diamond shape in Panel (b). Although the proposed method is not developed for previously unseen classes, the learnt interpolator takes the untrained class to a place that can be considered reasonable in the 2-dimensional embedding space.

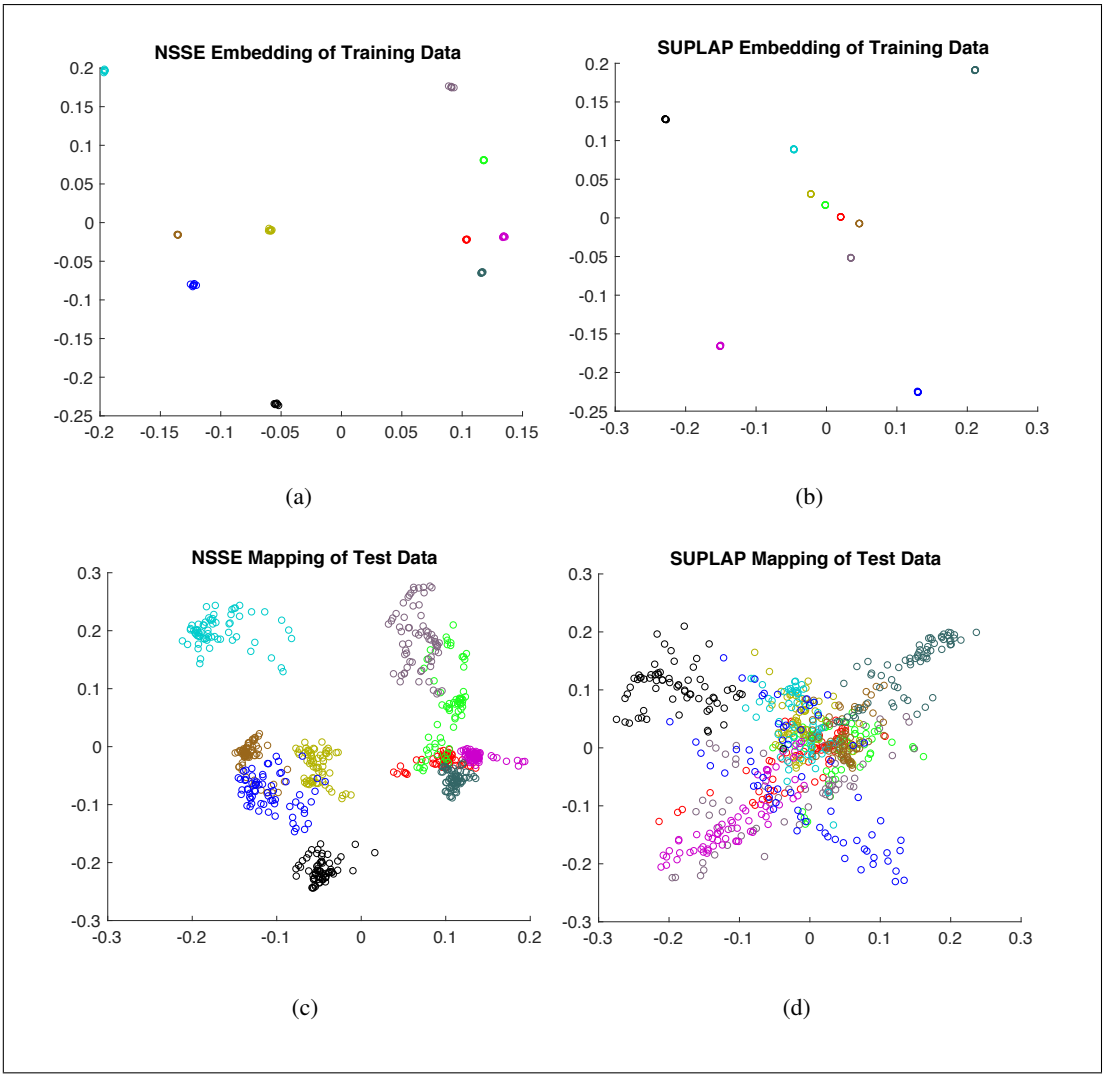


Figure 4.14: Visual comparison of the embeddings given by the NSSE and SUPLAP algorithms

4.4 Overall comparison with baseline classifiers and supervised dimensionality reduction methods

We now provide an overall comparison of the proposed NSSE method with baseline classifiers and other supervised manifold learning methods. We compare NSSE with the supervised manifold learning algorithms discussed in Section 4.3, as well as the SVM and the nearest neighbor (NN) classifiers in the original domain. The embedding dimensions and other algorithm parameters of the manifold learning methods are set to their optimal values yielding the best performance. The classification errors over test samples are studied by varying the training/test ratio and the results are av-

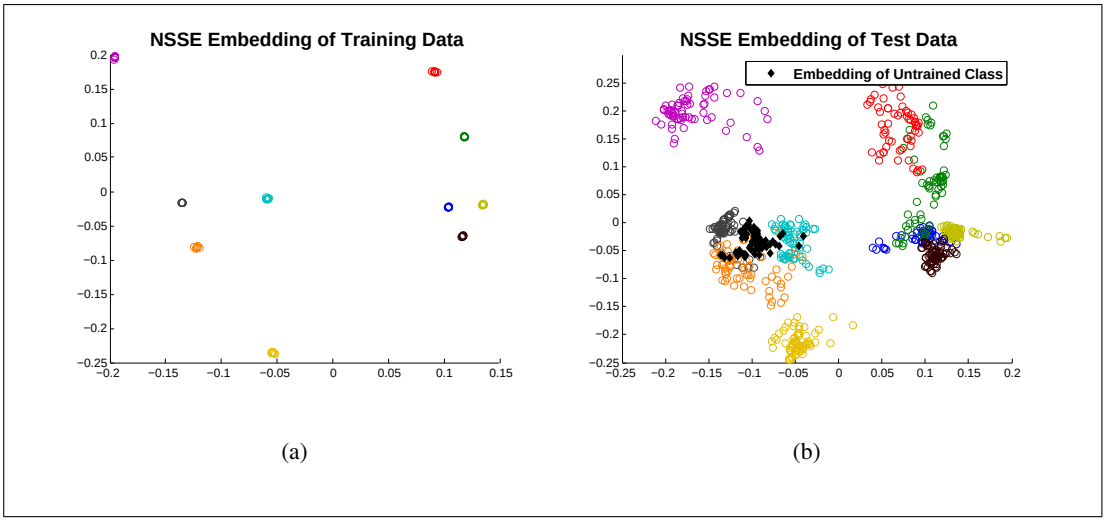


Figure 4.15: Embedding of training and test samples (including an untrained class)

eraged over 20 realizations of the experiments under different random choices of the training and test samples.

The misclassification rates of test samples in percentage are presented for the compared methods for different training data sizes in Tables 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, respectively for the Yale, COIL-20, ORL, FEI, ROBOTICS-CSIE, and MIT-CBCL data sets. The leftmost columns of the tables show the number of training samples per class used for learning the classifiers. Experiments are conducted over a suitable range of number of training samples for each data set, considering the total number of samples in the data set. The smallest classification error of each experiment is shown in bold.

The proposed NSSE method is observed to often outperform the other methods in Tables 4.2-4.7. The performances of the algorithms improve as the number of training samples increases as expected, while the algorithm closest in performance to NSSE in almost all experiments is the nonlinear supervised manifold learning algorithm SUPLAP. On the other hand, the linear manifold learning algorithms LFDA, LDA, and LDE exhibit variable performance depending on the data set. Their comparison to the baseline SVM and NN classifiers shows that these linear methods may get outperformed by baseline classifiers especially when the number of samples is sufficiently high, while the nonlinear NSSE and SUPLAP methods often yield better performance than these baseline classifiers. The proposed NSSE often outperforms the SUPLAP

method, with the performance gap being more significant especially when the number of training samples is limited. This can be explained with the fact that the lack of training samples is likely to lead to degenerate embeddings in nonlinear methods computing a pointwise embedding as in SUPLAP, while the regularization term enforcing the regularity of the interpolator in NSSE proves effective for the prevention of such degeneracies and ensuring the preservation of the overall geometric structure of data in the embedding.

Table 4.2: Misclassification rates (%) of compared methods on Yale database

# Training / class	NSSE	SUPLAP	SVM	NN	LFDA	LDA	LDE
6	22.09	23.10	29.43	63.48	26.88	35.85	**20.04**
10	**12.59**	12.92	15.16	52.89	40.62	63.97	21.78
15	**7.52**	7.95	9.09	43.57	10.77	57.86	7.95
20	**5.02**	5.60	6.14	37.51	7.42	52.79	5.15
30	**2.56**	2.56	2.98	30.12	3.22	46.43	3.04

Table 4.3: Misclassification rates (%) of compared methods on COIL-20 database

# Training / class	NSSE	SUPLAP	SVM	NN	LFDA	LDA	LDE
7	**9.18**	10.97	10.38	13.89	17.93	11.84	20.86
10	**5.88**	6.81	6.92	10.21	13.59	7.68	16.84
15	**3.26**	3.84	4.59	6.87	11.32	4.22	14.00
20	**1.50**	2.03	3.23	4.51	9.53	2.28	12.64
30	0.81	**0.80**	2.27	2.30	7.08	0.99	13.27

Table 4.4: Misclassification rates (%) of compared methods on ORL database

# Training / class	NSSE	SUPLAP	SVM	NN	LFDA	LDA	LDE
2	**14.63**	16.04	19.73	19.34	27.69	21.17	24.91
3	**8.54**	9.48	10.70	12.96	14.89	13.13	12.73
5	**3.90**	5.31	4.35	6.91	8.10	7.73	7.05

Table 4.5: Misclassification rates (%) of compared methods on FEI database

# Training / class	NSSE	SUPLAP	SVM	NN	LFDA	LDA	LDE
2	**21.45**	27.06	35.37	32.13	29.82	30.92	30.05
4	**8.28**	12.46	12.85	19.45	12.90	12.56	10.80
7	**5.06**	6.41	9.09	10.85	9.74	5.40	7.76

Table 4.6: Misclassification rates (%) of compared methods on ROBOTICS-CSIE database

# Training / class	NSSE	SUPLAP	SVM	NN	LFDA	LDA	LDE
7	**13.55**	27.22	23.96	34.46	24.87	29.42	25.13
14	**4.38**	11.73	8.77	17.80	11.96	14.15	9.73
21	**2.83**	6.51	4.76	10.09	6.99	10.56	5.88

Table 4.7: Misclassification rates (%) of compared methods on MIT-CBCL database

# Training / class	NSSE	SUPLAP	SVM	NN	LFDA	LDA	LDE
10	**6.48**	7.31	9.90	14.42	12.32	18.43	9.69
20	**2.56**	3.38	4.18	5.64	8.36	8.38	6.01
40	**0.85**	1.21	1.52	1.45	5.29	3.18	2.97

CHAPTER 5

CONCLUSION

In this thesis, we have proposed a nonlinear supervised manifold learning method that learns an embedding of the training data jointly with a smooth RBF interpolation function that extends the embedding to the whole space. The embedding and the interpolator parameters are jointly optimized with the purpose of good generalization to initially unavailable data, based on recent theoretical results on the performance of supervised manifold learning algorithms. In particular, the embedding and the RBF paramaters are learnt such that the interpolator has sufficiently good Lipschitz regularity while the samples from different classes are separated as much as possible.

We have tested the performance of the proposed NSSE method by three different experiments conducted on six real data sets. In the first experiment, the behavior of the proposed method in the iterative optimization procedure is observed. The objective function shows a rapid decline from an initially high value until convergence. Thanks to this rapid decline, the objective function is stabilized in only a few iterations. For each iteration, the misclassification rate falls in parallel with the objective function. This indicates that the proposed objective function can serve the desired purpose successfully. We have also observed the behavior of the RBF kernel scale σ for each iteration. Even if the RBF kernel scale σ is initially selected to be too high or too low, it moves toward a more favorable value with iterations. The RBF kernel scale σ reaches its final value when the objective function converges. As a result, as the objective function converges, the classification error is reduced and the RBF kernel scale automatically reaches a good value.

In the second experiment, the classification performance of the proposed algorithm

according to the embedding dimension is investigated. The proposed NSSE method in this context is compared with the SUPLAP, LFDA, LDE, LDA algorithms described in Chapter 2. In cases where the embedding dimension is too low, the error rates of the other algorithms except for the NSSE are much higher than the NSSE. When the embedding dimension is increased, the error rate of the LDA decreases monotonically. However, the maximum limit of the embedding dimension of the LDA is up to the number of classes. Best performances of the LDE and the LFDA algorithms depend on the dataset. The misclassification rates of these algorithms may decrease as the embedding dimension increases; however, in some datasets a local optimum for the embedding dimension can also be observed. The SUPLAP shows the best performance at a dimension close to the number of classes. The optimal embedding dimension of our proposed algorithm is observed to be highly related to the number of degrees of freedom, i.e., the intrinsic dimension, of the data set. For example the optimum dimension of the NSSE is low in the COIL-20, FEI and ROBOTICS-CSIE datasets which have low degrees of freedom and the optimum dimension of the NSSE is higher in the MIT-CBCL dataset which have higher degrees of freedom. This shows that our algorithm is better fit to the intrinsic geometry of the original data manifold.

Another advantage of the NSSE over the other algorithms is its robostness against nonoptimal selection of the embedding dimension which does not affect the performance of the NSSE very much. However other algorithms are affected significantly with the change of the embedding dimension. The proposed NSSE algorithm achieves this through imposing the Lipschitz continuity of the interpolator in the learning objective function. In all these methods, the nonlinear NSSE and SUPLAP algorithms seem to perform better than LDA, LDE and LFDA which are linear methods. This shows that the flexibility of nonlinear methods when learning an embedding is likely to bring an advantage in computing better representations for data.

Figure 4.14 gives a visual comparison of NSSE and SUPLAP. Embedding of training samples seems to be successful in both methods although embedding of training samples is distributed more regularly for the NSSE method. However, if we would look at the generalization of the embedding to the initially unavailable test samples, the NSSE does this task much better than the SUPLAP as it can be seen in the figure.

In our last experiment, the best performances of algorithms in all datasets have been compared. SVM and Nearest Neighbor (NN) classifiers applied in the original data domain are also included in this comparison. We have observed that the proposed NSSE often outperformed the other algorithms in comparison. Especially when the number of available training samples is insufficient in comparison with the number of classes such as in the FEI and ROBOTICS-CSIE datasets, the performance gap of the NSSE over the other methods becomes more significant. This shows that explicitly including the generalization performance in the learning via the interpolator regularity brings robustness in the classification performance for scarcely sampled data sets. From a general perspective, these comparative experiments suggest that in settings where the training data is limited, employing priors on the data model in the learning, such as the low-dimensionality of the data as in this thesis, may be preferable to learning generic classifiers that assume no data priors. This contrasts with the favorableness of generic classifiers with rich models, e.g. deep learning methods, when vast amounts of training data are available.

In the proposed method, several algorithm parameters need to be tuned, such as the kernel scale and weight parameters. To begin with, as we have previously observed, the choice of the RBF kernel scale does not affect the performance of the proposed method as it is optimized throughout the learning process. However, there is no method for selecting the weight parameters in the objective function. The weight parameters can be found by the cross validation process to be performed with the training data that is initially available. For the calculation of the withing-class Laplacian matrix L_w in the proposed algorithm, the within-class weight matrix W_w must be calculated first. Since the sensitivity of the proposed method to the heat kernel is low, the heat kernel parameter is set to a typical positive value in our experiments. In addition, all samples in the same class are treated as neighbors in the matrix W_w. One of the future works on this algorithm might be to develop a method to find the optimal value of these weight parameters without dealing with the cross validation process.

Conventional linear dimensionality reduction methods such as FDA suffer from difficulties if the samples in a class are multimodal, that is, if the same data class is made up of separate clusters. Because such methods assume that samples in the same class have a continuous and Gaussian distribution. Unfortunately, this is not always the

case in reality. In such cases, the algorithm used should give importance to preserving the local structure of the original data. Non-linear methods, for example Locally Linear Embedding (LLE), Laplacian Eigenmaps (LE), Isomap, Supervised Laplacian Eigenmaps (SUPLAP) and our method, are more successful in preserving the original data structure. We already know that non-linear methods are more flexible in representing the original structure of the data set. However, these aforementioned linear and nonlinear algorithms and our algorithm build only a single graph that considers the data structure as a whole. No extra prevention has been taken against the risk of being multimodal. Each class in the data sets in our experiments creates a continuous manifold in space. In the case that the data set is multimodal, for our algorithm, perhaps the construction of more than one graph for each class can improve the algorithm performance. And also, the embedding of these graphs is another problem that needs to be solved.

In the experiments we have done, we knew the class label information in each of the data sets. However, one can also consider settings where the class label information of some data samples is not available. In this case, we would need to adapt our algorithm for semi-supervised learning to take advantage of unlabeled data.

Another important point is that the training and test data are not always collected under the same capturing conditions. This can cause training and test data to have different distributions in the data space. However, both the training and the test data have the same distribution in our experiments. Another future direction of our work is its extension to overcome this problem.

Another extension that can be done for the proposed method may be an adaptation for zero-shot learning problems. Learning style of the interpolator may be modified such that previously unseen classes are also embedded on reasonable places in the lower dimensional embedding space.

To sum up, this thesis study shows that nonlinear dimensionality reduction may yield promising results in machine learning problems where the data is known to have a low-dimensional structure. The idea of explicitly taking the generalization performance of algorithms into account in the learning can potentially be extended to other settings as well in the future.

REFERENCES

[1] Dimension reduction of swiss roll data by PCA, ISOMAP and LLE. Available: http://www.astroml.org/book_figures/chapter7/fig_S_manifold_PCA.html.

[2] MIT-CBCL face recognition database. Available: http://cbcl.mit.edu/software-datasets/heisele/facerecognition-database.html.

[3] Robotics CSIE database for face detection. Available: http://robotics.csie.ncku.edu.tw/Databases/FaceDetect_PoseEstimate.htm.

[4] A. Argyriou, M. Herbster, and M. Pontil. Combining graph laplacians for semi-supervised learning. In *Advances in Neural Information Processing Systems 18*, pages 67–74, 2005.

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

[6] F. R. Bach and M. I. I. Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48, 2002.

[7] G. Baudat and F. Anouar. Generalized discriminant analysis using a kernel approach. *Neural Computation*, 12(10):2385–2404, 2000.

[8] B. J. C. Baxter. *The interpolation theory of radial basis functions*. PhD thesis, Cambridge University, Trinity College, 1992.

[9] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, June 2003.

[10] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006.

[11] Y. Bengio, J. F. Paiement, P. Vincent, O. Delalleau, N. Le Roux, and M. Ouimet. Out-of-sample extensions for LLE, ISOMAP, MDS, Eigenmaps, and Spectral Clustering. In *Adv. Neural Inf. Process. Syst.*, pages 177–184. MIT Press, 2004.

[12] D. Cai, X. He, K. Zhou, J. Han, and H. Bao. Locality sensitive discriminant analysis. In *IJCAI 2007, Proceedings of the 20th International Joint Conference*

on Artificial Intelligence, Hyderabad, India, January 6-12, 2007, pages 708–713, 2007.

[13] G. H. Chen, C. Wachinger, and P. Golland. Sparse projections of medical images onto manifolds. In *Proc. Information Processing in Medical Imaging - 23rd International Conference*, pages 292–303, 2013.

[14] H. Chen, H. Chang, and T. Liu. Local discriminant embedding and its variants. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*, pages 846–853, 2005.

[15] S. Chen, J. Wang, C. Liu, and B. Luo. Two-dimensional discriminant locality preserving projection based on l1-norm maximization. *Pattern Recognition Letters*, 87:147–154, 2017.

[16] G. Dai and D. Y. Yeung. Kernel selection for semi-supervised kernel machines. In *Prof. 24th Int. Conf. Machine Learning*, pages 185–192, 2007.

[17] F. Dornaika and B. Raduncanu. Out-of-sample embedding for manifold learning applied to face recognition. In *IEEE Conf. Computer Vision and Pattern Recognition, CVPR Workshop*, pages 862–868, 2013.

[18] Q. Gao, J. Ma, H. Zhang, X. Gao, and Y. Liu. Stable orthogonal local discriminant embedding for linear dimensionality reduction. *IEEE Trans. Image Processing*, 22(7):2521–2531, 2013.

[19] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 23(6):643–660, 2001.

[20] X. He and P. Niyogi. Locality Preserving Projections. In *Advances in Neural Information Processing Systems 16*, Cambridge, MA, 2004. MIT Press.

[21] Y. Liu, Y. Liu, K. C. C. Chan, and K. A. Hua. Hybrid manifold embedding. *IEEE Trans. Neural Netw. Learning Syst.*, 25(12):2295–2302, 2014.

[22] A. Nazarpour and P. Adibi. Two-stage multiple kernel learning for supervised dimensionality reduction. *Pattern Recognition*, 48(5):1854–1862, 2015.

[23] S. A. Nene, S. K. Nayar, and H. Murase. Columbia Object Image Library (COIL-20). Technical report, Feb 1996.

[24] C. Orsenigo and C. Vercellis. Kernel ridge regression for out-of-sample mapping in supervised manifold learning. *Expert Syst. Appl.*, 39(9):7757–7762, 2012.

[25] Y. Pang, A. T. B. Jin, and F. S. Abas. Neighbourhood preserving discriminant embedding in face recognition. *J. Visual Communication and Image Representation*, 20(8):532–542, 2009.

[26] B. Peherstorfer, D. Pflüger, and H. J. Bungartz. A sparse-grid-based out-of-sample extension for dimensionality reduction and clustering with laplacian eigenmaps. In *AI 2011: Proc. Advances in Artificial Intelligence - 24th Australasian Joint Conference*, pages 112–121, 2011.

[27] C. Piret. *Analytical and Numerical Advances in radial basis functions*. PhD thesis, University of Colorado, 2007.

[28] H. Qiao, P. Zhang, D. Wang, and B. Zhang. An explicit nonlinear mapping for manifold learning. *IEEE T. Cybernetics*, 43(1):51–63, 2013.

[29] B. Raducanu and F. Dornaika. A supervised non-linear dimensionality reduction approach for manifold learning. *Pattern Recognition*, 45(6):2432–2444, 2012.

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

[31] F. Samaria and A. Harter. Parameterisation of a stochastic model for human face identification. In *Proc. Second IEEE Workshop on Applications of Computer Vision*, pages 138–142, 1994.

[32] B. Schölkopf, A. J. Smola, and K. R. Müller. Kernel principal component analysis. In *7th Int. Conf. Artificial Neural Networks*, pages 583–588, 1997.

[33] M. Sugiyama. Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis. *Journal of Machine Learning Research*, 8:1027–1061, 2007.

[34] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, December 2000.

[35] C. E. Thomaz and G. A. Giraldi. A new ranking method for principal components analysis and its application to face image analysis. *Image Vision Comput.*, 28(6):902–913, 2010.

[36] M. Villegas and R. Paredes. Dimensionality reduction by minimizing nearest-neighbor classification error. *Pattern Recognition Letters*, 32(4):633–639, 2011.

[37] E. Vural and C. Guillemot. A study of the classification of low-dimensional data with supervised manifold learning. *Submitted for publication*. . [Online]. Available: https://arxiv.org/abs/1507.05880.

[38] E. Vural and C. Guillemot. Out-of-sample generalizations for supervised manifold learning for classification. *IEEE Transactions on Image Processing*, 25(3):1410–1424, March 2016.

[39] W. Wong and H. Zhao. Supervised optimal locality preserving projection. *Pattern Recognition*, 45:186 – 197, 2012.

[40] D. You, O. C. Hamsici, and A. M. Martinez. Kernel optimization in discriminant analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33:631–638, 2011.

[41] M. Yu, L. Shao, X. Zhen, and X. He. Local feature discriminant projection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(9):1908–1914, 2016.

[42] S. Zhang. Enhanced supervised locally linear embedding. *Pattern Recognition Letters*, 30(13):1208–1218, 2009.