Universidade Federal de Pernambuco
Centro de Informática

Programa de Pós-graduação em Ciência da Computação

Two-dimensional Extensions of Semi-supervised Dimensionality Reduction Methods

Lailson Bandeira de Moraes

Dissertação de Mestrado

Recife, Pernambuco, Brasil
August 19, 2013

Universidade Federal de Pernambuco
Centro de Informática

Lailson Bandeira de Moraes

Two-dimensional Extensions of Semi-supervised Dimensionality Reduction Methods

Trabalho apresentado ao Programa de Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Orientador: *George Darmiton da Cunha Cavalcanti*

Recife, Pernambuco, Brasil
August 19, 2013

Catalogação na fonte
Bibliotecária Jane Souto Maior, CRB4-571

Moraes, Lailson Bandeira de
Two-dimensional extensions of semi-supervised dimensionality reduction methods/ Lailson Bandeira de Moraes. - Recife: O Autor, 2013.
74 f.: il., fig., tab.

Orientador: George Darmiton da Cunha Cavalcanti.
Dissertação (mestrado) - Universidade Federal de Pernambuco. CIn, Ciência da Computação, 2013.

Inclui referências.

1. Ciência da Computação. 2. Inteligência artificial. I. Cavalcanti, George Darmiton da Cunha (orientador). II. Título.

004 CDD (23. ed.) MEI2013 – 137

Dissertação de Mestrado apresentada por **Lailson Bandeira de Moraes** à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título **"Two-dimensional Extensions of Semi-supervised Dimensionality Reduction Methods"** orientada pelo Prof. George Darmiton da Cunha Cavalcanti e aprovada pela Banca Examinadora formada pelos professores:

_____

Prof. Carlos Alexandre Barros de Mello
Centro de Informática / UFPE

_____

Prof. Tiago Alessandro Espinola Ferreira
Departamento de Estatística e Informática / UFRPE

_____

Prof. George Darmiton da Cunha Cavalcanti
Centro de Informática / UFPE

Visto e permitida a impressão.
Recife, 19 de agosto de 2013

_____

Profa. Edna Natividade da Silva Barros
Coordenadora da Pós-Graduação em Ciência da Computação do
Centro de Informática da Universidade Federal de Pernambuco.

Aos meus pais.

Agradecimentos

Agradeço a Deus, que nos deu um universo tão vasto e maravilhoso e, ao mesmo tempo, nos presenteou com inteligência, curiosidade e força para explorá-lo.

Agradeço à minha mãe, Inez, pelo amor incondicional e pelo suporte diário, que foram absolutamente fundamentais para que eu chegasse até aqui. Agradeço ao meu pai, Pedro, que eu tenho certeza que continua cuidando de mim, onde quer que ele esteja.

Agradeço ao meu orientador, George, por ter acreditado em mim desde muito cedo e por estar sempre disponível para me ajudar. Agradeço aos professores Carlos e Thiago por terem gentilmente aceitado participar da banca que avaliou este trabalho e pelas valiosas sugestões para melhorá-lo.

Agradeço à minha namorada, Naianne, pelo carinho, pela compreensão e pelo apoio emocional, cruciais para que eu pudesse chegar ao fim desta jornada.

Agradeço aos meus amigos Guilherme e Lucas, pelos quais eu tenho grande admiração, pela força e apoio que tenho recebido nos últimos anos.

Agradeço à minha amiga Joyce, que mesmo longe está sempre presente na minha vida, alegrando os meus dias.

Agradeço, enfim, a todos os meus amigos, que estando perto ou distante, me fazem ser uma pessoa melhor e contribuem, das mais diversas formas, para o meu sucesso.

To improve is to change; to be perfect is to change often.
—WINSTON CHURCHILL

Abstract

An important pre-processing step in machine learning systems is dimensionality reduction, which aims to produce compact representations of high-dimensional patterns. In computer vision applications, these patterns are typically images, that are represented by two-dimensional matrices. However, traditional dimensionality reduction techniques were designed to work only with vectors, what makes them a suboptimal choice for processing two-dimensional data. Another problem with traditional approaches for dimensionality reduction is that they operate either on a fully unsupervised or fully supervised way, what limits their efficiency in scenarios where supervised information is available only for a subset of the data. These situations are increasingly common because in many modern applications it is easy to produce raw data, but it is usually difficult to label it. In this study, we propose three dimensionality reduction methods that can overcome these limitations: Two-dimensional Semi-supervised Dimensionality Reduction (2D-SSDR), Two-dimensional Discriminant Principal Component Analysis (2D-DPCA), and Two-dimensional Semi-supervised Local Fisher Discriminant Analysis (2D-SELF). They work directly with two-dimensional data and can also take advantage of supervised information even if it is available only for a small part of the dataset. In addition, a fully supervised method, the Two-dimensional Local Fisher Discriminant Analysis (2D-LFDA), is proposed too. The methods are defined in terms of a two-dimensional framework, which was created in this study as well. The framework is capable of generally describing scatter-based methods for dimensionality reduction and can be used for deriving other two-dimensional methods in the future. Experimental results showed that, as expected, the novel methods are faster and more stable than the existing ones. Furthermore, 2D-SSDR, 2D-SELF, and 2D-LFDA achieved competitive classification accuracies most of the time when compared to the traditional methods. Therefore, these three techniques can be seen as viable alternatives to existing dimensionality reduction methods.

Keywords: computer vision, dimensionality reduction, feature extraction, semi-supervised learning, tensor discriminant analysis

Resumo

Um estágio importante de pré-processamento em sistemas de aprendizagem de máquina é a redução de dimensionalidade, que tem como objetivo produzir representações compactas de padrões de alta dimensionalidade. Em aplicações de visão computacional, estes padrões são tipicamente imagens, que são representadas por matrizes bi-dimensionais. Entretanto, técnicas tradicionais para redução de dimensionalidade foram projetadas para lidar apenas com vetores, o que as torna opções inadequadas para processar dados bi-dimensionais. Outro problema com as abordagens tradicionais para redução de dimensionalidade é que elas operam apenas de forma totalmente não-supervisionada ou totalmente supervisionada, o que limita sua eficiência em cenários onde dados supervisionados estão disponíveis apenas para um subconjunto das amostras. Estas situações são cada vez mais comuns por que em várias aplicações modernas é fácil produzir dados brutos, mas é geralmente difícil rotulá-los. Neste estudo, propomos três métodos para redução de dimensionalidade capazes de contornar estas limitações: *Two-dimensional Semi-supervised Dimensionality Reduction* (2D-SSDR), *Two-dimensional Discriminant Principal Component Analysis* (2D-DPCA), e *Two-dimensional Semi-supervised Local Fisher Discriminant Analysis* (2D-SELF). Eles operam diretamente com dados bi-dimensionais e também podem explorar informação supervisionada, mesmo que ela esteja disponível apenas para uma pequena parte das amostras. Adicionalmente, um método completamente supervisionado, o *Two-dimensional Local Fisher Discriminant Analysis* (2D-LFDA) é proposto também. Os métodos são definidos nos termos de um *framework* bi-dimensional, que foi igualmente criado neste estudo. O *framework* é capaz de descrever métodos para redução de dimensionalidade baseados em dispersão de forma geral e pode ser usado para derivar outras técnicas bi-dimensionais no futuro. Resultados experimentais mostraram que, como esperado, os novos métodos são mais rápidos e estáveis que as técnicas existentes. Além disto, 2D-SSDR, 2D-SELF, e 2D-LFDA obtiveram taxas de erro competitivas na maior parte das vezes quando comparadas aos métodos tradicionais. Desta forma, estas três técnicas podem ser vistas como alternativas viáveis aos métodos existentes para redução de dimensionalidade.

Palavras-chave: visão computacional, redução de dimensionalidade, extração de características, aprendizagem semi-supervisionada, análise tensorial de discriminantes

List of Figures

List of Tables

List of Abbreviations

PCA	Principal Component Analysis
FDA	Fisher Discriminant Analysis
LPP	Locality Preserving Projections
LFDA	Local Fisher Discriminant Analysis
SSDR	Semi-supervised Discriminant Analysis
DPCA	Discriminant Principal Component Analysis
SELF	Semi-supervised Local Fisher Discriminant Analysis
2D-PCA	Two-dimensional Principal Component Analysis
2D-FDA	Two-dimensional Fisher Discriminant Analysis
2D-LPP	Two-dimensional Locality Preserving Projections
2D-LFDA	Two-dimensional Local Fisher Discriminant Analysis
2D-SSDR	Two-dimensional Semi-supervised Discriminant Analysis
2D-DPCA	Two-dimensional Discriminant Principal Component Analysis
2D-SELF	Two-dimensional Semi-supervised Local Fisher Discriminant Analysis
SSS	Small Sample Size
USP	Undersample Problem

Contents

CHAPTER 1

Introduction

With the advances in imaging technologies, nowadays it is possible to capture a large amount of images and videos with minimal effort. For instance, numerous high-resolution images are produced everyday by medical imaging systems for disease diagnosis. Satellites and aerial devices are constantly taking detailed, multi-spectral pictures from the Earth. Surveillance systems can record a multitude of video hours in a single day. And millions of photographs and videos captured by mobile devices are being continuously published on the Internet. Imaging data is plentiful today and it should keep growing in a fast pace.

In this scenario, a central area is *computer vision*, a field that provides automated means for dealing with such enormous quantity of images and videos. Computer vision combines multiple disciplines such as image processing, pattern recognition and machine learning in order to create systems that are capable of extracting useful information from visual data. In these systems, images are usually represented as matrices of *picture elements* (pixels). A challenging aspect of this type of data is that it is high-dimensional because even small images contain a large number of pixels. Fortunately, most applications do not need to use all these pixels for two reasons. First, pixels within an image are highly correlated due to their spatial arrangement. Thus, images contain a lot of redundancies that can be eliminated. Second, for many problems, some regions of the image are not relevant at all. For example, consider a recognition system that receives a facial picture and must identify the person who is depicted in it. For this system, the pixels in the background of the scene are not useful and should be discarded. Furthermore, even the pixels that correspond to the face are not all necessary because they contain redundant information. The system can select the pixels that are more important for identifying faces or combine them in some way to create new dimensions that make the identification job easier. This process is known as *dimensionality reduction* and it is a key pre-processing task in computer vision.

Dimensionality reduction is an active research area and there are many well-known dimensionality reduction methods. However, the majority of these methods work only with one-dimensional vectors. For this reason, to reduce the dimensionality of two-dimensional images, they must be first transformed into vectors. The issue is that important information can be lost in the transforming process because the structure of the data is greatly changed: pixels that were neighbors in the original image can be far apart in the vectorized version. Also, because images typically contain a large number of pixels, the generated vectors have a large number of elements, making the dimensionality reduction time-consuming or even impossible when the dimensional-

ity is too high. To overcome these limitations, *two-dimensional dimensionality reduction methods* were developed. Instead of dealing with vectors, these methods use the image directly, what makes them much faster to compute. In addition, they are generally more stable and work better when there are few sample images available. Due to these advantages, two-dimensional methods have been extensively developed and used in the last years.

Methods for dimensionality reduction are traditionally grouped into two general categories, depending whether they make use of discriminative knowledge or not. *Unsupervised methods* rely only on the information contained in the samples to reduce their dimensionality, without using any external information. In contrast, *supervised methods* make use of some discriminative information (in the form of labels associated to each sample, for example) to guide the reduction process in a way that best separates samples that belongs to different groups. The literature shows that classification systems based on supervised dimensionality reduction methods often perform better than the ones that use unsupervised methods (Theodoridis and Koutroumbas, 2009). This is not a surprise, since supervised methods have access to important discriminative data and therefore can perform a better data separation.

However, in many practical applications, labels are available only for a subset of the data. This may be because the labeling process is expensive or simply because the dataset is too large to be manually inspected. In these situations, neither unsupervised nor supervised methods are an optimal choice. On one hand, if an unsupervised dimensionality reduction method is used, the system can take advantage of the data contained in all samples, but the useful discriminative information cannot be used. On the other hand, if a supervised algorithm is chosen, the labels can be incorporated in the dimensionality reduction process, but the samples that do not have labels should be discarded. The complication is that both aspects are important. The unlabeled samples allow to better characterize the underling data distribution whereas the discriminative information allows to better segregate samples from different groups, what helps the classification system. Hence, the ideal alternative would make use of these two aspects at the same time. This is what *semi-supervised methods* do. They are capable of combining supervised and unsupervised information in order to improve the dimensionality reduction process.

In this chapter we introduce some fundamental concepts. Section 1.1 formally defines the dimensionality reduction problem and discuss related concepts. Section 1.2 describes two-dimensional dimensionality reduction and its advantages compared to the conventional techniques. Section 1.3 discusses the semi-supervised learning paradigm. Section 1.4 presents the goals of this work. Finally, Section 1.5 summarizes the content of the upcoming chapters.

1.1 Dimensionality Reduction

Dimensionality reduction is the process of finding low-dimensional representations for high-dimensional *data patterns* (also called of *samples*, *examples* or *observations*). The

fundamental motivation for dimensionality reduction is that patterns with a large number of *dimensions* (also known as *features*, *variables* or *attributes*) often contain a lot of redundant or unnecessary information that can be eliminated. In this way, the central goal of dimensionality reduction is to produce a compact representation for high-dimensional samples such that most of their "intrinsic information" is preserved (Sugiyama et al., 2010). From another perspective, dimensionality reduction methods aim to find meaningful low-dimensional structures hidden in a high-dimensional space (Tenenbaum et al., 2000).

The generic problem of dimensionality reduction is defined as follows. Given a set $\mathbf{x}_1, \ldots, \mathbf{x}_n$ of n patterns contained in a vector space $\mathbb{V}$, find a set $\mathbf{x}'_1, \ldots, \mathbf{x}'_n$ of corresponding patterns that are contained in another vector space $\mathbb{V}'$ such that this new space has a lower number of dimensions than $\mathbb{V}$ and $\mathbf{x}'_i$ represents $\mathbf{x}_i$ in some way. Throughout the next chapters, more concrete definitions will be provided for particular dimensionality reduction strategies.

Working with the compact representation of the patterns is helpful because data processing algorithms typically have their time and space complexity affected by the dimensionality of the input data. Thus, reducing the number of data dimensions means that less computation and memory will be required to process it. Also, most data algorithms are susceptible to the well-known problem of the *curse of dimensionality*, which refers to the severe degradation in effectiveness as the number of features increases (Bishop, 2006). Another argument for dimensionality reduction is that it is easier to understand the data when it can be explained with fewer features. As a consequence, the extraction of useful knowledge from the patterns is simplified and more accurate models can be derived with less effort. Finally, representing the data with fewer dimensions makes possible to plot and visually analyze it in order to better grasp its geometric structure or detect outliers, for example (Alpaydin, 2010).

There are two main approaches for reducing dimensionality: feature selection and feature extraction. *Feature selection* methods operate by picking the features that are more important for a given goal and discarding the others. Conversely, *feature extraction* (also called *feature generation* and *feature transformation*) methods combine the existing features to create new ones, which are devised to be more representative for the problem in question (Liu and Motoda, 1998). The precise definition of what "more important" or "more representative" mean is given by each specific method. In practice, feature extraction is much more used than feature selection to reduce the dimensionality of images because the single pixels that correspond to the raw dimensions of an image are not representative enough. Instead, the interesting features of image are normally contained in groups of pixels. For this reason, feature extraction is a much more suitable approach for images. Hence, in this work we are concerned only with feature extraction. Additionally, from now on the terms dimensionality reduction and feature extraction are used interchangeably.

Dimensionality reduction can be performed on a broad range of data types, from speech signals (Errity and McKenna, 2007) to textual documents (Jain et al., 2004) and gait patterns (Tao et al., 2007). It also has been successfully employed in many

tasks, such as data representation (Belkin and Niyogi, 2003), compression (Karni and Gotsman, 2004), visualization (Wang et al., 2004) and even regression (Jolliffe, 2002). Nonetheless, here we are interested only in dimensionality reduction of images within the context of classification systems.

1.2 Two-dimensional Data Analysis

Two-dimensional data analysis (also known as *matrix-based data analysis* [1]) is a sub-area of data analysis concerned with inputs that have a matrix form, such as images. It can be seen as a special case of multilinear or tensor data analysis because matrices are equivalent to tensors of second rank in multilinear algebra (Vasilescu and Terzopoulos, 2003). For this reason, two-dimensional methods are sometimes referred to as *tensor-based methods*.

When employed for dimensionality reduction of two-dimensional data, matrix-based methods bring important advantages over the established vector-based approaches. The main problem with vector-based methods is that, because they work only with data in one-dimensional form, they require the two-dimensional data to be converted to vectors in advance. However, treating two-dimensional patterns as vectors give raise to significant drawbacks. First, when the patterns are vectorized, their underlying spatial structure is destroyed (Nie et al., 2009). As a consequence, the data can be deeply spoiled since structural information that could be useful is thrown away in the transforming procedure.

Second, the generated vectors have a high number of dimensions, causing the dimensionality reduction process to be slow and even unfeasible in some cases. This happens because when a matrix is transformed into a vector, each element of the matrix becomes a dimension of that vector. The complication is that matrices, due to their two-dimensional nature, have a very large number of elements. For example, a matrix of size 112×96 have 10.752 elements (and in many real-world applications patterns may have a much larger size). Processing vectors with such great number of features is time- and space-consuming. Moreover, when the dimensionality is too high, the algorithm may require so much processing or memory resources that it is not viable in practice.

And finally, in many cases the number of dimensions is much higher than the quantity of patterns. In these cases, it is difficult to perform dimensionality reduction accurately because the number of parameters to estimate exceeds the number of available samples. This hurts the effectiveness of the methods up to a point where the reduced patterns may not make sense at all. This is known as the *small sample size* (SSS) or *undersample problem* (USP) in the literature and it is a significant limitation of

[1] In fact, "two-dimensional" is the predominant denomination in the literature. However, in the context of this work, the term is ambiguous because *dimension* may be confused with *number of features* (that is, a two-dimensional sample may be thought to be a sample with two features, which it is not the case). For this reason, we prefer the term *matrix-based*. Despite of that, the names of existing methods—such as *Two-dimensional Principal Component Analysis*—are used as originally defined.

the traditional vector-based methods (Fukunaga, 1990; Tao et al., 2007).

Matrix-based methods were developed to overcome all these problems. They can take advantage of the structural information contained in the samples because they work directly with the data in its original matrix form. They are faster to compute and require less memory resources. And they are much less affected by the SSS problem because they need less samples to run in an accurate way.

Methods for analyzing non-vectorial data are known for a long time. They were extensively developed in the second half of the last century mainly by remote sensing applications, which generate hyperspectral imaging data with lots of dimensions. However, it was only recently that these methods were formalized and studied as the field of tensor data analysis, using the established discipline of multilinear algebra as theoretical base. The sub-area of tensor data analysis concerned with dimensionality reduction is multilinear subspace learning. It is a new field of study (developed in the last few years) and has been growing fast, specially motivated by the increasing number of high-dimensional data available nowadays (Lu et al., 2011).

In a similar way, the idea of directly using two-dimensional images in the dimensionality reduction process is not new, but it attracted the attention of a wider audience only in the last decade. One of the earliest works that operate in this way was proposed by Liu et al. (1993). They perform an optimal discriminant analysis in order to extract projection vectors from a set of image matrices. Shashua and Levin (2001) also treat images as matrices. They consider the collection of all matrices as a third-order tensor and search for the best low-rank approximation of its tensor-rank. In the middle of the 2000s, many two-dimensional extensions of existing vector-based methods were proposed. Yang et al. (2004) came up with the Two-dimensional Principal Component Analysis (2D-PCA), a matrix-based version of the Principal Component Analysis (PCA). Ye et al. (2004); Kong et al. (2005); Yang et al. (2005); Li and Yuan (2005); Xiong et al. (2005) proposed two-dimensional variations of the Linear Discriminant Analysis (LDA). And Chen et al. (2007) derived the Two-dimensional Locality Preserving Projections (2D-LPP), a two-dimensional version of the Locality Preserving Projections (LPP). More recently, Cavalcanti et al. (2013) proposed the Weighted Modular Image Principal Component Analysis (MIMPCA), a modular version of two-dimensional PCA developed specifically for face recognition. The method aims to minimize the distortions caused by variations in illumination and head pose.

1.3 Semi-supervised Learning

Semi-supervised learning is a machine learning paradigm that is capable of exploiting both unsupervised and supervised data. As Zhu and Goldberg (2009) put, "semi-supervised learning is somewhere between unsupervised and supervised learning." In fact, most semi-supervised strategies are based on extending either unsupervised or supervised techniques to include additional information typical to other learning paradigm. Semi-supervised learning can be applied in many different settings, but it is commonly used for classification tasks.

A central concern in semi-supervised learning is how unsupervised data can be useful to improve the learning process. From the classification perspective, the question is whether unsupervised data can really make classification systems more accurate. The literature is abundant of works with positive answers to this question and the justification is that unsupervised data makes it possible to estimate decision boundaries more reliably (Chapelle et al., 2006). The example in Figure 1.1 illustrates this fact [2]. Looking only to the two labeled samples in the chart (a), we would intuitively categorize the ? point as pertaining to the class +. Moreover, we would think that the dashed line is a good choice for the decision boundary. However, when we analyze the complete picture in the chart (b), we realize that the two labeled samples are certainly not the most representative prototypes for the classes. In the face of that, we would be tempted to reconsider our decision and also change the classification boundary. This simple example shows how the geometry of unlabeled data may radically change our intuition about decision boundaries. In addition, the example points to a fundamental prerequisite in semi-supervised learning: that the distribution of the unlabeled samples should be representative of the underlying population and also relevant to the problem in question [3].

Semi-supervised learning is most useful in situations where there are far more unsupervised than supervised data. This happens because in many real applications collecting raw data is much easier than labeling it. The labels may be difficult to obtain because they require human annotators, special devices, or expensive and slow experiments. For example, in speech recognition it is possible to record hours of speech without effort, but accurate transcription by human expert annotators can be extremely time consuming. Godfrey et al. (1992) reports that it took as long as 400 hours to transcribe 1 hour of speech at the phonetic level for the Switchboard telephone conversational speech data. In bioinformatics protein sequences can be acquired in a short amount of time, but it can take months of expensive laboratory work to determine the three-dimensional structure of a single sample (Zhu and Goldberg, 2009). Spam filtering is another example. An e-mail inbox can contain hundreds of thousands of messages but the user have to read them to reliably determine whether they are legit or not. And the list goes on. There are many areas where unsupervised data is fairly easy to collect but the labeling process is costly. Sometimes, it can be made less cumbersome with the help of annotation tools (Spiro et al., 2010) or even by embedding the labeling routine into computer games for motivating the users to produce more labels (von Ahn and Dabbish, 2004). Still, in the majority of the cases it is simply not possible to label the whole dataset.

The typical form of expressing *supervised information* (also known as *discriminative information*) is by associating *labels* to the samples. The labels specify to what *group* or

[2] Example adapted from Theodoridis and Koutroumbas (2009).

[3] Actually, semi-supervised learning make other assumptions too. For example, the *smoothness assumption* states that "if two points in a high-density region are close, then so should be the corresponding outputs" and the *cluster assumption* states that "if points are in the same cluster, they are likely to be of the same class". Nonetheless, the assumptions made by semi-supervised learning and their details are beyond the scope of this work. To know more about them, consult Chapelle et al. (2006).

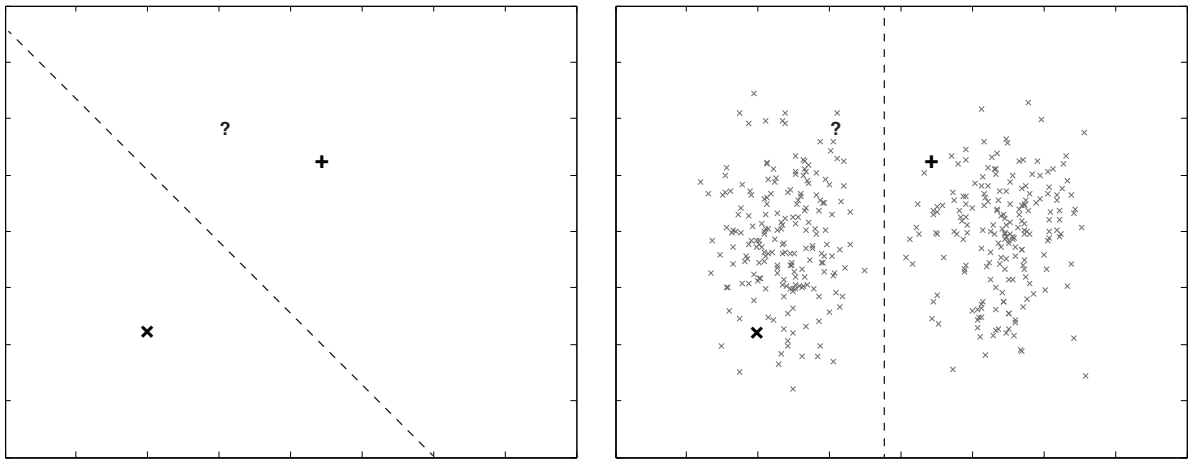


Figure 1.1 Example of how the unsupervised data can help to characterize the distribution of the classes. The chart (a) contains three patterns: one from the class +, other from the class ×, and another with an unknown class, showed as ?. Based only on the two samples with known labels, it is reasonable to classify the pattern ? as belonging to the class +. Furthermore, the dashed line shown in the chart seems to be a sensible choice for a decision boundary between the classes. However, when unlabeled data is added in the chart (b), the situation changes. Now it seems more likely that the ? pattern belongs to the × class. Also, the unlabeled data reveals that the previous decision boundary was a bad choice for the problem; the new dashed line seems to be a more accurate option.

class the samples belong to. Another form of discriminative information are *equivalence constraints* (or *pairwise constraints*). They link two samples and indicate that they pertain to the same class (*must-link constraints*) or to different classes (*cannot-link constraints*), even though the identity of the class is unknown. Bar-Hillel et al. (2005) argues that there are scenarios where obtaining pairwise constraints is much cheaper than obtaining labels and that in some cases constraints can be even generated automatically. Note that constraints can be easily derived from labels and for this reason labeled data can be used in all places where constraints are expected. It is also important to observe that, since labels are the most common format of discriminative information, the term *labeled data* is often used as a synonym for *supervised data*.

A topic that is closely related to semi-supervised learning is *transduction* (sometimes called *transductive learning* or *transductive inference*). The idea behind transduction is to learn only the necessary to classify a given set of unlabeled samples. In contrast, the traditional *inductive inference* aims to generalize the particular knowledge contained in the supervised data into a predictive model, capable of classifying unknown examples that will appear in the future. Clearly, induction is much harder than transduction because it involves the prediction of unknown data. This is especially true when the supervised data is scarce (see more about the small sample size problem in the previous section). If the application just needs to predict the labels

of the data points it has now, the transductive approach is likely to give more accurate results because it incorporates the unsupervised data in the learning process. For this reason, transductive inference is usually associated with semi-supervised learning. However, the concepts are distinct; some semi-supervised learning algorithms are transductive, but there are a lot of inductive semi-supervised methods too.

Although semi-supervised learning has attracted a lot of interest recently, the idea of combining unsupervised and supervised data in machine learning is not new. Chapelle et al. (2006) reports that as early as in the 1960s this concept was already exploited by self-learning methods (Scudder, 1965; Fralick, 1967; Agrawala, 1970). In the following years, semi-supervised learning was occasionally employed in many settings, such as mixture models (Hosmer Jr, 1973), discriminant analysis (O'Neill, 1978; McLachlan and Ganesalingam, 1982), co-training (Blum and Mitchell, 1998) and constrained clustering (Wagstaff et al., 2001). Finally, the interest in semi-supervised learning increased in the 2000s, mostly due to applications in bioinformatics, computer vision, natural language processing and text classification. It is interesting to note that the term "semi-supervised" as it is known today appeared in the literature only in 1992, when Merz et al. used it for the first time to describe the use of unsupervised and supervised data in machine learning. The expression was employed previously, but in a different context (Chapelle et al., 2006).

1.4 Objectives

The main goal of this study is to propose dimensionality reduction methods that can overcome the limitations of the established techniques in computer vision applications. In these applications, it is common to have an enormous quantity of large two-dimensional samples, but supervised information is often available only for a subset of them. Motivated by this observation, we aim to create techniques that can efficiently process two-dimensional data and, at the same time, also take advantage of the important but limited supervised information that may be available. To do this, we intend to investigate existing matrix-based and semi-supervised methods for dimensionality reduction in order to understand how they work and how their useful aspects can be combined. Finally, another objective of this study is to evaluate these new methods with two public image databases and compare their performance with the results of state-of-art techniques.

1.5 Outline

Chapter 2 considers vector-based methods. It formally defines the dimensionality reduction problem and notation and describes seven existing vector-based methods: Principal Component Analysis (PCA), Fisher Discriminant Analysis (FDA), Locality Preserving Projections (LPP), Local Fisher Discriminant Analysis (LFDA), Semi-supervised Dimensionality Reduction (SSDR), Discriminant Principal Component Analysis

(DPCA), and Semi-supervised Local Discriminant Analysis (SELF). Then, it presents the vector scatter-based framework for dimensionality reduction and shows how the described methods are expressed in terms of the framework. Finally, the chapter discusses some practical aspects involved in the computational implementation of the framework.

Chapter 3 shows the definition and notation for the matrix-based dimensionality reduction problem and introduces three existing matrix-based methods in their original form: Two-dimensional Principal Component Analysis (2D-PCA), Two-dimensional Fisher Discriminant Analysis (2D-FDA), and Two-dimensional Locality Preserving Projections (2D-LPP).

Chapter 4 presents the proposed matrix scatter-based framework for dimensionality reduction, similar to the vector-based framework discussed previously. It also shows how the existing two-dimensional methods can be described within the new framework. Following that, four novel methods for dimensionality reduction are proposed: Two-dimensional Local Fisher Discriminant Analysis (2D-LFDA), Two-dimensional Semi-supervised Dimensionality Reduction (2D-SSDR), Two-dimensional Discriminant Principal Component Analysis (2D-DPCA) and Two-dimensional Semi-supervised Local Discriminant Analysis (2D-SELF). To conclude, the chapter examines implementation issues of the matrix-based framework.

Chapter 5 describes the experiments and analyzes the results. The chapter details the methodology and the characteristics of the tasks and databases used in the experiments. After that, it shows the result tables and discusses the obtained results, comparing the methods performance in each case.

Chapter 6 finishes this dissertation and reviews what was discussed in the previous chapters. The chapter summarizes the contributions made by this study and considers the directions that can be followed in future works. Appendix A demonstrates how the existing vector- and matrix-based methods were converted to the corresponding frameworks.

CHAPTER 2

Vector-based Dimensionality Reduction Methods

Most of the existing dimensionality reduction methods expect data samples to be *vectors*. These methods are known as *vector-based methods*. Vectors represent samples as ordered lists of values and, within a sample, each value corresponds to the measurement of a feature. Because the samples are represented by vectors, they are contained in a vector space. The dimensions of this space correspond to the sample features.

A typical way to perform dimensionality reduction is by mapping the samples into a new vector space that has fewer dimensions than the original one. Methods that operate like this are called *mapping* or *projective methods*. The central task of projective methods is to find the best low-dimensional space to project the samples. The rationale employed to define what "best" means is what differentiates the various existing projective methods for dimensionality reduction.

The operation of mapping vectors from one space into another is called *transformation*. When a transformation is performed only with linear operations, it is said to be linear. *Linear transformations* are widely used because their properties are well defined and they can be completely characterized by a *transformation matrix*. Therefore, in this context, finding a projection space is equivalent to finding a transformation matrix. Projective methods based on linear transformations are called *linear projective methods*.

In this chapter, we discuss some vector-based linear projective methods. Section 2.1 formally defines the general problem and notation. Section 2.2 presents the Principal Component Analysis (PCA), a popular unsupervised method. Section 2.3 describes the Fisher Discriminant Analysis (FDA), a supervised method that is also extensively employed for dimensionality reduction. Section 2.4 analyzes a more recent unsupervised method, the Locality Preserving Projections (LPP), which exploits the local arrangement of the samples. Section 2.5 considers the Local Fisher Discriminant Analysis (LFDA), a combination of FDA and LPP. Sections 2.6, 2.7 and 2.8 discuss three semi-supervised methods: the Semi-supervised Dimensionality Reduction (SSDR), the Discriminant Principal Component Analysis (DPCA) and the Semi-supervised Local Fisher Discriminant Analysis (SELF), respectively. Finally, Section 2.9 shows a generalized pairwise form for all these methods.

2.1 Definition and Notation

Let $\mathbf{x} \in \mathbb{R}^m$ be a m-dimensional column vector that represents a sample and let $\mathbf{X} \in \mathbb{R}^{m \times n}$ be the matrix of all n samples

$$\mathbf{X} = (\mathbf{x}_1 | \mathbf{x}_2 | \cdots | \mathbf{x}_n). \tag{2.1}$$

Let $\mathbf{p} \in \mathbb{R}^r (1 \leq r \leq m)$ be a low-dimensional representation of the high-dimensional sample $\mathbf{x} \in \mathbb{R}^m$, where r is the dimensionality of the reduced space. In linear projective methods, $\mathbf{p}$ is a projection of $\mathbf{x}$ in another vector space. This space is defined by a transformation matrix $\mathbf{T} \in \mathbb{R}^{m \times r}$. Therefore, the representation $\mathbf{p}$ of the sample $\mathbf{x}$ is obtained as

$$\mathbf{p} = \mathbf{T}^\top \mathbf{x}, \tag{2.2}$$

where $^\top$ denotes the transpose of a matrix.

The *Euclidean distance* between two vectors $\mathbf{x_1} = (x_{11}, \ldots, x_{1m})$ and $\mathbf{x_2} = (x_{21}, \ldots, x_{2m})$ is defined as

$$d(\mathbf{x}_1, \mathbf{x}_2) = \sqrt{(x_{11} - x_{21})^2 + \cdots + (x_{1m} - x_{2m})^2}. \tag{2.3}$$

When the original sample is not a vector, it must be transformed into one. For example, a two-dimensional image that is represented by a matrix is vectorized by placing its columns one after another. Thus, an image composed by $l \times m$ pixels generates a vector with lm dimensions.

2.2 Principal Component Analysis (PCA)

Principal Component Analysis (PCA), also known as *Karhunen–Loève expansion*, is one of the most used unsupervised methods for data representation and dimensionality reduction. Its origin traces back to the work developed by Pearson (1901), but it was only in 1991 that the technique was really popularized in the pattern recognition field, when Turk and Pentland presented the well-known Eigenfaces method for face recognition. Since then, PCA has been extensively investigated and applied in computer vision problems (Jolliffe, 2002).

Let $\bar{\mathbf{x}}$ be the mean of all samples and $\mathbf{S}_t$ be the *total scatter matrix*, defined as

$$\bar{\mathbf{x}} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i, \tag{2.4}$$

$$\mathbf{S}_t = \sum_{i=1}^{n} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^\top. \tag{2.5}$$

PCA seeks a new space that maximizes the scatter of the projected samples. The optimal transformation matrix $\mathbf{T}_{\text{PCA}}$ that leads to this space is obtained by the equation

$$\mathbf{T}_{\text{PCA}} = \underset{\mathbf{T} \in \mathbb{R}^{m \times r}}{\arg\max} \left[\det \left(\mathbf{T}^\top \mathbf{S}_t \mathbf{T} \right) \right]. \tag{2.6}$$

A solution for this optimization problem is given by the eigenvectors $\{\boldsymbol{\phi}_k\}_{k=1}^r$ of the following eigendecomposition problem (Fukunaga, 1990)

$$\mathbf{S}_t \boldsymbol{\phi} = \lambda \boldsymbol{\phi}, \tag{2.7}$$

assuming that the eigenvectors are sorted in descending order according to their associated eigenvalues $\{\lambda_k\}_{k=1}^r$.

2.3 Fisher Discriminant Analysis (FDA)

Fisher Discriminant Analysis (FDA), sometimes called *Linear Discriminant Analysis* (LDA) [1], is another popular dimensionality reduction technique (Fisher, 1936; Fukunaga, 1990). Because it is supervised, FDA is generally more suitable for classification tasks than unsupervised methods. FDA is one of the simplest forms of introducing discriminatory knowledge into the feature extraction.

Let n' be the number of labeled samples and n'_l be the number of labeled samples of the class l, such that $n' = \sum_{l=1}^c n'_l$. In addition, let $\mathbf{S}_b$ be the *between-class scatter matrix* and $\mathbf{S}_w$ be the *within-class scatter matrix*, defined as

$$\mathbf{S}_b = \sum_{l=1}^{c} n'_l (\bar{\mathbf{x}}_l - \bar{\mathbf{x}})(\bar{\mathbf{x}}_l - \bar{\mathbf{x}})^\top, \tag{2.8}$$

$$\mathbf{S}_w = \sum_{l=1}^{c} \sum_{i:y_i=l} (\mathbf{x}_i - \bar{\mathbf{x}}_l)(\mathbf{x}_i - \bar{\mathbf{x}}_l)^\top, \tag{2.9}$$

where $\sum_{i:y_i=l}$ indicates the summation over the samples with class l and $\bar{\mathbf{x}}_l$ is the mean of samples of the class l such that

$$\bar{\mathbf{x}}_l = \frac{1}{n'_l} \sum_{i:y_i=l} \mathbf{x}_i. \tag{2.10}$$

FDA aims to maximize the between-scatter and to minimize the within-scatter in the projection space. Thus, the FDA transformation matrix $\mathbf{T}_{\text{FDA}}$ is defined as

$$\mathbf{T}_{\text{FDA}} = \underset{\mathbf{T} \in \mathbb{R}^{m \times r}}{\arg\max} \left[\frac{\det \left(\mathbf{T}^\top \mathbf{S}_b \mathbf{T} \right)}{\det \left(\mathbf{T}^\top \mathbf{S}_w \mathbf{T} \right)} \right]. \tag{2.11}$$

A solution for this maximization problem is given by the generalized eigenvectors $\{\boldsymbol{\phi}_k\}_{k=1}^r$ of the following generalized eigendecomposition problem

[1] Actually, FDA is just a particular kind of LDA that uses the Fisher criterion as base. Another example of linear discriminant analysis is the Bhattacharyya LDA (Ordowski and Meyer, 2004).

$$\mathbf{S}_b\boldsymbol{\phi} = \lambda\mathbf{S}_w\boldsymbol{\phi}, \tag{2.12}$$

assuming that the eigenvectors are sorted in descending order according to their associated eigenvalues $\{\lambda_k\}_{k=1}^{r}$.

2.4 Locality Preserving Projections (LPP)

Locality Preserving Projections (LPP) is a more recent technique for unsupervised dimensionality reduction, proposed by He and Niyogi (2004). It incorporates neighborhood information into the dimensionality reduction process in order to preserve the the local structure of the samples in the projected space.

Let $\mathbf{A}$ be an *affinity matrix* with size $n \times n$. Each element $A_{i,j}$ represents the affinity between the samples $\mathbf{x}_i$ and $\mathbf{x}_j$ and it is usually assumed that $0 \leq A_{i,j} \leq 1$. The affinity value $A_{i,j}$ should be large if $\mathbf{x}_i$ and $\mathbf{x}_j$ are close and small otherwise. Note that *close* does not necessarily means near in a spatial sense; it can be defined as any meaningful relation between the samples (like the perceptual similarity for natural signals or the hyperlink structures for web documents, for instance). The affinity matrix can be defined in several different ways. In this work, we use the *local scaling heuristic* (Zelnik-Manor and Perona, 2005) [2], defined as

$$A_{i,j} = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma_i\sigma_j}\right), \tag{2.13}$$

where $\|\cdot\|$ denotes the Euclidean norm and the parameter σ_i represents the local scaling around $\mathbf{x}_i$, given by

$$\sigma_i = \|\mathbf{x}_i - \mathbf{x}_j^{(k)}\|, \tag{2.14}$$

with $\mathbf{x}_i^{(k)}$ being the k-th nearest neighbor of $\mathbf{x}_i$. Sugiyama (2007) determined that $k = 7$ is a useful choice.

Let $\mathbf{D}$ be the diagonal matrix whose entries are column sums of $\mathbf{A}$ such that $D_{i,i} = \sum_j A_{i,j}$ and $\mathbf{L} = \mathbf{D} - \mathbf{A}$ be the *Laplacian matrix*. The LPP transformation matrix $\mathbf{T}_{\text{LPP}}$ is defined as

$$\mathbf{T}_{\text{LPP}} = \underset{\mathbf{T}\in\mathbb{R}^{m\times r}}{\arg\min}\left[\frac{\det\left(\mathbf{T}^\top\mathbf{X}\mathbf{L}\mathbf{X}^\top\mathbf{T}\right)}{\det\left(\mathbf{T}^\top\mathbf{X}\mathbf{D}\mathbf{X}^\top\mathbf{T}\right)}\right]. \tag{2.15}$$

A solution for this problem is given by the generalized eigenvectors $\{\boldsymbol{\phi}_k\}_{k=1}^{r}$ of the following generalized eigendecomposition problem

$$\mathbf{X}\mathbf{L}\mathbf{X}^\top\boldsymbol{\phi} = \lambda\mathbf{X}\mathbf{D}\mathbf{X}^\top\boldsymbol{\phi}, \tag{2.16}$$

[2] Sugiyama (2007) discusses other typical ways of defining the affinity matrix.

assuming that the eigenvectors are sorted in *ascending* order according to their associated eigenvalues $\{\lambda_k\}_{k=1}^{r}$. Note that, contrary to the methods shown so far, the LPP transformation matrix is defined as a minimization problem. That is the reason that the ordering direction of the eigenvalues is different this time.

2.5 Local Fisher Discriminant Analysis (LFDA)

Local Fisher Discriminant Analysis (LFDA) is a fully supervised combination of LPP and FDA proposed by Sugiyama (2007). It uses LPP for adding local information to the dimensionality reduction process in order to overcome the weakness of the original FDA against within-class multimodality or outliers.

Let $\mathbf{S}_{lb}$ be the *local between-class scatter matrix* and $\mathbf{S}_{lw}$ be the *local within-class scatter matrix*, defined as

$$\mathbf{S}_{lb} = \frac{1}{2}\sum_{i,j=1}^{n} W_{i,j}^{(lb)}(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top, \quad W_{i,j}^{(lb)} = \begin{cases} A_{i,j}(1/n' - 1/n'_{y_i}) & \text{if } y_i = y_j \\ 1/n' & \text{if } y_i \neq y_j \end{cases}, \tag{2.17}$$

$$\mathbf{S}_{lw} = \frac{1}{2}\sum_{i,j=1}^{n} W_{i,j}^{(lw)}(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top, \text{ with } W_{i,j}^{(lw)} = \begin{cases} A_{i,j}/1/n'_{y_i} & \text{if } y_i = y_j \\ 0 & \text{if } y_i \neq y_j \end{cases}, \tag{2.18}$$

where $A_{i,j}$ is the affinity value between the samples $\mathbf{x}_i$ and $\mathbf{x}_j$, as defined by the equation (2.13), y_i is the class of the sample $\mathbf{x}_i$ and n'_{y_i} is the number of samples of the class y_i. LFDA was designed to keep nearby samples that belongs to the same class close together and to keep samples of different classes far apart. Therefore, it seeks to maximize the localized version of between-scatter matrix and to minimize the local within-scatter. The transformation matrix $\mathbf{T}_{\text{LFDA}}$ is defined as

$$\mathbf{T}_{\text{LFDA}} = \underset{\mathbf{T}\in\mathbb{R}^{m\times r}}{\arg\max}\left[\frac{\det(\mathbf{T}^\top \mathbf{S}_{lb}\mathbf{T})}{\det(\mathbf{T}^\top \mathbf{S}_{lw}\mathbf{T})}\right]. \tag{2.19}$$

Like FDA, a solution for this maximization problem is given by the generalized eigenvectors $\{\boldsymbol{\phi}_k\}_{k=1}^{r}$ of the following generalized eigendecomposition problem

$$\mathbf{S}_{lb}\boldsymbol{\phi} = \lambda \mathbf{S}_{lw}\boldsymbol{\phi}, \tag{2.20}$$

assuming that the eigenvectors are sorted in descending order according to their associated eigenvalues $\{\lambda_k\}_{k=1}^{r}$.

2.6 Semi-supervised Dimensionality Reduction (SSDR)

Semi-supervised Dimensionality Reduction (SSDR) is a semi-supervised method proposed by Zhang et al. (2007). It can handle unsupervised data and also take advantage of

must-link and cannot-link constraints to guide the dimensionality reduction operation.

Let $S_{i,j}$ be the weight between the samples $\mathbf{x}_i$ and $\mathbf{x}_j$ such that

$$S_{i,j} = \begin{cases} 1/n^2 + \alpha/n_c & \text{if } (\mathbf{x}_i, \mathbf{x}_j) \in \mathbf{C}^{(l)} \\ 1/n^2 - \beta/n_m & \text{if } (\mathbf{x}_i, \mathbf{x}_j) \in \mathbf{M}^{(l)} \\ 1/n^2 & \text{otherwise} \end{cases}, \tag{2.21}$$

where $\mathbf{C}^{(l)}$ and $\mathbf{M}^{(l)}$ are respectively the cannot-link and must-link constraints sets, n_c and n_m are the number of cannot-link and must-link constraints, and α and β are scaling parameters that balance the contributions of each constraint type.

SSDR seeks a projection space that maximizes the distance between samples that belong to different classes (or, equivalently, samples involved in cannot-link constraints) and that minimizes the distance between samples that belong to the same class (or, equivalently, samples involved in must-link constraints). For unlabeled samples, SSDR just uses the same criteria from PCA.

Let $\mathbf{D}$ be the diagonal matrix whose entries are column sums of $\mathbf{S}$ such that $D_{i,i} = \sum_j S_{i,j}$ and $\mathbf{L} = \mathbf{D} - \mathbf{A}$ be the Laplacian matrix in spectral graph theory. The optimal SSDR transformation matrix $\mathbf{T}_{\text{SSDR}}$ is defined as

$$\mathbf{T}_{\text{SSDR}} = \underset{\mathbf{T} \in \mathbb{R}^{m \times r}}{\arg\max} \Big[\det(\mathbf{T}^\top \mathbf{X} \mathbf{L} \mathbf{X}^\top \mathbf{T}) \Big]. \tag{2.22}$$

A solution for this problem is given by the eigenvectors $\{\boldsymbol{\phi}_k\}_{k=1}^r$ of the following eigendecomposition problem

$$\mathbf{X} \mathbf{L} \mathbf{X}^\top \boldsymbol{\phi} = \lambda \boldsymbol{\phi}, \tag{2.23}$$

assuming that the eigenvectors are sorted in descending order according to their associated eigenvalues $\{\lambda_k\}_{k=1}^r$.

2.7 Discriminant Principal Component Analysis (DPCA)

Discriminant Principal Component Analysis (DPCA) is a semi-supervised method for dimensionality reduction created by Sun and Zhang (2009). It adds labels and pairwise constraints to PCA in order to boost its discriminative power. DPCA is closely related to SSDR. Both operate in a similar way but the definition of their weights differs. Moreover, DPCA is defined in function of PCA whereas SSDR is defined over the Laplacian matrix.

Let $\mathbf{S}'_b$ and $\mathbf{S}'_w$ be respectively the *generalized between-scatter matrix* and *generalized within-scatter matrix*, defined as

$$\mathbf{S}'_b = \frac{1}{|\mathbf{C}^{(l)}|} \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathbf{C}} (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top \tag{2.24}$$

$$\mathbf{S}'_w = \frac{1}{|\mathbf{M}^{(l)}|} \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathbf{M}} (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top \tag{2.25}$$

where $\mathbf{C}^{(l)}$ and $\mathbf{M}^{(l)}$ are respectively the cannot-link and must-link constraints sets and $|\cdot|$ denotes the cardinality of a set. Note that these scatter matrices are generalized versions of the ones shown in Section 2.3. This version is adapted to handle pairwise constraints.

DPCA intends to maximize the distance in the projection space between samples from different classes and to minimize the distance between samples of the same class. In addition, it uses the PCA rationale for unlabeled samples. In this way, it creates the scatter matrix $\mathbf{S}_d$, such that

$$\mathbf{S}_d = \mathbf{S}'_b - \eta \mathbf{S}'_w + \lambda \mathbf{S}_t \tag{2.26}$$

where $\mathbf{S}_t$ is the total scatter matrix (defined by the equation 2.5) and η and λ are regularizing coefficients that balance the contributions of each term. The optimal DPCA transformation matrix $\mathbf{T}_{\text{DPCA}}$ is defined as

$$\mathbf{T}_{\text{DPCA}} = \underset{\mathbf{T} \in \mathbb{R}^{m \times r}}{\arg\max} \Big[\det(\mathbf{T}^\top \mathbf{S}_d \mathbf{T}) \Big]. \tag{2.27}$$

A solution for this problem is given by the eigenvectors $\{\boldsymbol{\phi}_k\}_{k=1}^r$ of the following eigendecomposition problem

$$\mathbf{S}_d \boldsymbol{\phi} = \lambda \boldsymbol{\phi}, \tag{2.28}$$

assuming that the eigenvectors are sorted in descending order according to their associated eigenvalues $\{\lambda_k\}_{k=1}^r$.

2.8 Semi-supervised Local Fisher Discriminant Analysis (SELF)

Semi-supervised Local Fisher Discriminant Analysis (SELF) is a method for dimensionality reduction proposed by Sugiyama et al. (2010). It combines PCA and LFDA in a single procedure, making both methods work in a complementary manner: PCA can exploit the global structure of unsupervised data whereas LFDA can take advantage of the discriminative information brought by the labeled samples. SELF was designed to work only with explicit labels, but it can be easily extended to include pairwise constraints too.

Let $\mathbf{S}_{rlb}$ be the *regularized local between-class scatter matrix* and $\mathbf{S}_{rlw}$ be the *regularized local within-class scatter matrix*, defined as

$$\mathbf{S}_{rlb} = (1 - \beta)\mathbf{S}_{lb} + \beta \mathbf{S}_t, \tag{2.29}$$

$$\mathbf{S}_{rlw} = (1 - \beta)\mathbf{S}_{lw} + \beta \mathbf{I}_m, \tag{2.30}$$

where $0 \leq \beta \leq 1$ is a trade-off parameter that controls the influence of the matrices, $\mathbf{S}_{lb}$ and $\mathbf{S}_{lw}$ are respectively the local versions of the between- and within-scatter matrices described by the equations (2.17) and (2.18), and $\mathbf{I}_m$ is the identity matrix in $\mathbb{R}^{m \times m}$. Observe that PCA and LFDA are special cases of SELF (when $\beta = 1$ and $\beta = 0$, respectively). The transformation matrix $\mathbf{T}_{\text{SELF}}$ is defined as

$$\mathbf{T}_{\text{SELF}} = \underset{\mathbf{T} \in \mathbb{R}^{m \times r}}{\arg\max} \left[\frac{\det(\mathbf{T}^\top \mathbf{S}_{rlb} \mathbf{T})}{\det(\mathbf{T}^\top \mathbf{S}_{rlw} \mathbf{T})} \right]. \tag{2.31}$$

A solution for this maximization problem is given by the generalized eigenvectors $\{\boldsymbol{\phi}_k\}_{k=1}^r$ of the following generalized eigendecomposition problem

$$\mathbf{S}_{rlb} \boldsymbol{\phi} = \lambda \mathbf{S}_{rlw} \boldsymbol{\phi}, \tag{2.32}$$

assuming that the eigenvectors are sorted in descending order according to their associated eigenvalues $\{\lambda_k\}_{k=1}^r$.

2.9 The Vector Scatter-Based Framework

The methods discussed in this chapter can be generalized to a *vector scatter-based dimensionality reduction framework*. All of them are based on scatter matrices, which can be expressed in a pairwise form (Belkin and Niyogi, 2003; Sugiyama, 2007) such as

$$\mathbf{S} = \frac{1}{2} \sum_{i,j=1}^{n} W_{i,j} (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top, \tag{2.33}$$

where $\mathbf{W}$ is an $n \times n$ symmetric matrix of weights that is defined according to each method. The element $W_{i,j}$ represents the weight between the samples $\mathbf{x}_i$ and $\mathbf{x}_j$.

Then, to find the optimal projection matrix $\mathbf{T}_{\text{OPT}}$, methods proceed to an optimization problem in the form [3]

$$\mathbf{T}_{\text{OPT}} = \underset{\mathbf{T} \in \mathbb{R}^{m \times r}}{\arg\max} \left[\frac{\det(\mathbf{T}^\top \mathbf{B} \mathbf{T})}{\det(\mathbf{T}^\top \mathbf{C} \mathbf{T})} \right], \tag{2.34}$$

where $\det(\cdot)$ denotes the determinant of a matrix. Roughly, $\mathbf{B}$ corresponds to the characteristic we want to increase in the projection space (between-class scatter, for

[3] There are other ways to obtain the same solution $\mathbf{T}_{\text{OPT}}$ (Fukunaga, 1990), such as:

$$\mathbf{T}_{\text{OPT}} = \underset{\mathbf{T} \in \mathbb{R}^{m \times r}}{\arg\max} [\text{tr}(\mathbf{T}^\top \mathbf{B} \mathbf{T} (\mathbf{T}^\top \mathbf{C} \mathbf{T})^{-1})],$$

$$\mathbf{T}_{\text{OPT}} = \underset{\mathbf{T} \in \mathbb{R}^{m \times r}}{\arg\max} [\text{tr}(\mathbf{T}^\top \mathbf{B} \mathbf{T})] \text{ subject to } \mathbf{T}^\top \mathbf{C} \mathbf{T} = \mathbf{I}_r,$$

where $\text{tr}(\cdot)$ denotes the trace of a matrix and $\mathbf{I}_r$ is the identity matrix in $\mathbb{R}^{r \times r}$.

instance) and $\mathbf{C}$ corresponds to the characteristic we want to decrease (within-class scatter, for instance).

Let $\{\boldsymbol{\phi}_k\}_{k=1}^{r}$ be the generalized eigenvectors associated with the eigenvalues $\{\lambda_k\}_{k=1}^{r}$ of the following generalized eigendecomposition problem

$$\mathbf{B}\boldsymbol{\phi} = \lambda\mathbf{C}\boldsymbol{\phi}. \tag{2.35}$$

Assuming that the generalized eigenvalues are sorted in descending order as $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_r$ and that the generalized eigenvectors are normalized as $\boldsymbol{\phi}_k^\top \mathbf{C}\boldsymbol{\phi}_k = 1$ for $k = 1,2,\ldots,r$, then a solution $\mathbf{T}_{\text{OPT}}$ is analytically given as (Fukunaga, 1990)

$$(\boldsymbol{\phi}_1|\boldsymbol{\phi}_2|\ldots|\boldsymbol{\phi}_r). \tag{2.36}$$

Sugiyama (2007) argues that, although the equation (2.34) is invariant under linear transformations, the *distance metric* in the projection space is arbitrary. He observed empirically that the following adjustment is useful to improve the distance metric:

$$\mathbf{T}_{\text{OPT}} = (\sqrt{\lambda_1}\boldsymbol{\phi}_1, \sqrt{\lambda_2}\boldsymbol{\phi}_2, \ldots, \sqrt{\lambda_r}\boldsymbol{\phi}_r). \tag{2.37}$$

Thus, the minor eigenvectors are deemphasized according to the square root of the eigenvalues. In this work, we performed experiments with and without these weights for all methods to compare their performance.

2.9.1 Defining Methods within the Framework

This section shows how the matrices $\mathbf{B}$ and $\mathbf{C}$ are defined for each method. Analyzing methods within this framework is useful to compare their similarities and differences. Moreover, it helps us to understand how the methods are related. The framework also simplifies the computational implementation, because the source code can be written once for the general form and then accommodate the variations required by each method. Appendix A demonstrates how the following weights were obtained.

Principal Component Analysis (PCA)

The optimal projection matrix $\mathbf{T}_{\text{PCA}}$ is given by the equations (2.35) and (2.37) with

$$\mathbf{B} = \mathbf{S}_t \quad \text{and} \quad \mathbf{C} = \mathbf{I}_m, \tag{2.38}$$

$$\mathbf{S}_t = \frac{1}{2}\sum_{i,j=1}^{n} W_{i,j}^{(t)}(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top, \text{ with } W_{i,j}^{(t)} = \frac{1}{n} \tag{2.39}$$

where $\mathbf{I}_m$ is the identity matrix in $\mathbb{R}^{m\times m}$ and n is the total number of samples.

Fisher Discriminant Analysis (FDA)

The optimal projection matrix $\mathbf{T}_{\text{FDA}}$ is given by the equations (2.35) and (2.37) with

$$\mathbf{B} = \mathbf{S}_b \quad \text{and} \quad \mathbf{C} = \mathbf{S}_w, \tag{2.40}$$

$$\mathbf{S}_b = \frac{1}{2}\sum_{i,j=1}^{n} W_{i,j}^{(b)}(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top, \text{ with } W_{i,j}^{(b)} = \begin{cases} 1/n' - 1/n'_{y_i} & \text{if } y_i = y_j \\ 1/n' & \text{if } y_i \neq y_j \end{cases}, \tag{2.41}$$

$$\mathbf{S}_w = \frac{1}{2}\sum_{i,j=1}^{n} W_{i,j}^{(w)}(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top, \text{ with } W_{i,j}^{(w)} = \begin{cases} 1/n'_{y_i} & \text{if } y_i = y_j \\ 0 & \text{if } y_i \neq y_j \end{cases}, \tag{2.42}$$

where n' is the number of labeled samples, y_i is the class of the sample $\mathbf{x}_i$ and n'_{y_i} is the number of labeled samples with label y_i.

Locality Preserving Projections (LPP)

LPP was originally formulated as a minimization problem. To fit it in the vector scatter-based framework, we should consider an inverted version of LPP (iLPP). Note that, whereas the rationale of LPP and iLPP are equivalent, the results are not guaranteed to be equal. The optimal projection matrix $\mathbf{T}_{\text{iLPP}}$ is given by the equations (2.35) and (2.37) with

$$\mathbf{B} = \mathbf{S}_n \quad \text{and} \quad \mathbf{C} = \mathbf{S}_l, \tag{2.43}$$

$$\mathbf{S}_n = \sum_{i=1}^{n} D_{i,i}^{(n)} \mathbf{x}_i \mathbf{x}_i^\top, \text{ with } D_{i,i}^{(n)} = \sum_{j=1}^{n} A_{i,j}, \tag{2.44}$$

$$\mathbf{S}_l = \frac{1}{2}\sum_{i,j=1}^{n} W_{i,j}^{(l)}(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top, \text{ with } W_{i,j}^{(l)} = A_{i,j}, \tag{2.45}$$

where $\mathbf{A}_{i,j}$ is the affinity between the samples $\mathbf{x}_i$ and $\mathbf{x}_j$ (as defined in Section 2.4).

Local Fisher Discriminant Analysis (LFDA)

The optimal projection matrix $\mathbf{T}_{\text{LFDA}}$ is given by the equations (2.35) and (2.37) with

$$\mathbf{B} = \mathbf{S}_{lb} \quad \text{and} \quad \mathbf{C} = \mathbf{S}_{lw}, \tag{2.46}$$

$$\mathbf{S}_{lb} = \frac{1}{2}\sum_{i,j=1}^{n} W_{i,j}^{(lb)}(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top, \; W_{i,j}^{(lb)} = \begin{cases} A_{i,j}(1/n' - 1/n'_{y_i}) & \text{if } y_i = y_j \\ 1/n' & \text{if } y_i \neq y_j \end{cases}, \tag{2.47}$$

$$\mathbf{S}_{lw} = \frac{1}{2}\sum_{i,j=1}^{n} W_{i,j}^{(lw)}(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top, \text{ with } W_{i,j}^{(lw)} = \begin{cases} A_{i,j}/n'_{y_i} & \text{if } y_i = y_j \\ 0 & \text{if } y_i \neq y_j \end{cases}, \tag{2.48}$$

here $\mathbf{A}_{i,j}$ is the affinity between the samples $\mathbf{x}_i$ and $\mathbf{x}_j$ (as defined in Section 2.4), n' is the number of labeled samples, y_i is the class of the sample $\mathbf{x}_i$ and n'_{y_i} is the number of labeled samples with label y_i.

Semi-supervised Dimensionality Reduction (SSDR)

The optimal projection matrix $\mathbf{T}_{\text{SSDR}}$ is given by the equations (2.35) and (2.37) with

$$\mathbf{B} = \mathbf{S}_{ssdr} \quad \text{and} \quad \mathbf{C} = \mathbf{I}_m, \tag{2.49}$$

$$\mathbf{S}_{ssdr} = \frac{1}{2}\sum_{i,j=1}^{n} W_{i,j}^{(ssdr)}(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top, \; W_{i,j}^{(ssdr)} = \begin{cases} 1/n^2 + \alpha/n_c & \text{if } (\mathbf{x}_i, \mathbf{x}_j) \in \mathbf{C}^{(l)} \\ 1/n^2 - \beta/n_m & \text{if } (\mathbf{x}_i, \mathbf{x}_j) \in \mathbf{M}^{(l)} \\ 1/n^2 & \text{otherwise} \end{cases}, \tag{2.50}$$

where $\mathbf{I}_m$ is the identity matrix in $\mathbb{R}^{m\times m}$, $\mathbf{C}^{(l)}$ and $\mathbf{M}^{(l)}$ are, respectively, the cannot-link and must-link constraints sets, n_c and n_m are, respectively, the numbers of cannot-link and must-link constraints, n is the total number of samples, and α and β are user-defined parameters.

Discriminant Principal Component Analysis (DPCA)

The optimal projection matrix $\mathbf{T}_{\text{DPCA}}$ is given by the equations (2.35) and (2.37) with

$$\mathbf{B} = \mathbf{S}_{dpca} \quad \text{and} \quad \mathbf{C} = \mathbf{I}_m, \tag{2.51}$$

$$\mathbf{S}_{dpca} = \frac{1}{2}\sum_{i,j=1}^{n} W_{i,j}^{(dpca)}(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top, \; W_{i,j}^{(dpca)} = \begin{cases} \lambda/n^2 + 2/n_c & \text{if } (\mathbf{x}_i, \mathbf{x}_j) \in \mathbf{C}^{(l)} \\ \lambda/n^2 - 2\eta/n_m & \text{if } (\mathbf{x}_i, \mathbf{x}_j) \in \mathbf{M}^{(l)} \\ \lambda/n^2 & \text{otherwise} \end{cases}, \tag{2.52}$$

where $\mathbf{I}_m$ is the identity matrix in $\mathbb{R}^{m\times m}$, $\mathbf{C}^{(l)}$ and $\mathbf{M}^{(l)}$ are, respectively, the cannot-link and must-link constraints sets, n_c and n_m are, respectively, the numbers of cannot-link and must-link constraints, n is the total number of samples, and λ and η are user-defined parameters.

Semi-supervised Local Fished Discriminant Analysis (SELF)

The optimal projection matrix $\mathbf{T}_{\text{SELF}}$ is given by the equations (2.35) and (2.37) with

$$\mathbf{B} = \mathbf{S}_{rlb} \quad \text{and} \quad \mathbf{C} = \mathbf{S}_{rlw}, \tag{2.53}$$

$$\mathbf{S}_{rlb} = (1-\beta)\mathbf{S}_{lb} + \beta\mathbf{S}_t, \tag{2.54}$$

$$\mathbf{S}_{rlw} = (1-\beta)\mathbf{S}_{lw} + \beta\mathbf{I}_m, \tag{2.55}$$

where $\mathbf{S}_{lb}$ and $\mathbf{S}_{lw}$ are the local between- and within-scatter matrices as defined by LFDA, $\mathbf{S}_t$ is the total covariance matrix as defined by PCA, $\mathbf{I}_m$ is the identity matrix in $\mathbb{R}^{m\times m}$, and β is an user-defined parameter.

2.9.2 Implementation Notes

The calculation of the scatter matrix $\mathbf{S}$ in the pairwise form is computationally expensive, because it involves many multiplications of high-dimensional vectors. However, this matrix can be expressed in another form, more efficient to compute (Sugiyama, 2007).

Let $\mathbf{D}$ be the $n \times n$ diagonal matrix $D_{i,i} = \sum_{j=1}^{n} W_{i,j}$ and let $\mathbf{L} = \mathbf{D} - \mathbf{W}$. Then the matrix $\mathbf{S}$ can be expressed in terms of $\mathbf{L}$ as

$$\mathbf{S} = \mathbf{X}\mathbf{L}\mathbf{X}^\top, \tag{2.56}$$

which has a much lower computational cost than the pairwise version.

If we interpret $\mathbf{W}$ as a weight matrix for a graph with n nodes, $\mathbf{L}$ can be regarded as the *graph Laplacian matrix* in spectral graph theory (Chung, 1997).

CHAPTER 3

Matrix-based Dimensionality Reduction Methods

Matrix-based dimensionality reduction methods are methods that operate directly with matrices. They are also called *two-dimensional* or *tensor-based methods*. These methods were specifically developed for dimensionality reduction of patterns that have a matrix structure, such as images.

When a conventional vector-based method is used to reduce the dimensionality of two-dimensional samples, they first must be transformed into vectors (see more in Section 2.1). When the samples are images, the resulting vectors usually have a high number of dimensions, what leads to a corresponding high-sized scatter matrix. This scatter matrix is difficult to evaluate accurately due to its large size and the relatively small numbers of training samples. This is known as the *small sample size* problem: as the number of dimensions increase, much more samples are needed to correctly calculate the scatter matrix (Fukunaga, 1990).

The vectorizing step is not necessary for matrix-based methods because they already expect inputs to be in their native matrix form. This difference implies in some important advantages over the vector-based approach. First, the spatial structure of each sample is preserved, what means that there is more information available for feature extraction. Second, scatter matrices that are computed directly from two-dimensional samples have considerably lower dimensionality and therefore are more reliable and faster to manipulate. Third, because scatter matrices are smaller, the small sample size problem is alleviated, meaning that fewer samples are needed.

For these reasons, matrix-based methods often perform better than equivalent vector-based methods, especially in classification tasks, where dimensionality reduction is employed as a preprocessing stage. The literature shows that in many cases the matrix-based approach results in better classification rates (Yang et al., 2004; Xiong et al., 2005; Yang et al., 2005). Also, since the feature extraction process is much faster in these methods, they are suitable for on-line learning scenarios, where new information is constantly being incorporated to the model.

In this chapter, we present three existing matrix-based methods. Section 3.1 defines the general problem of matrix-based dimensionality reduction in a formal way. Sections 3.2, 3.3, and 3.4 discuss Two-dimensional Principal Component Analysis (2D-PCA), Two-dimensional Fisher Discriminant Analysis (2D-FDA), and Two-dimensional Locality Preserving Projections (2D-LPP), respectively, the matrix-based versions of PCA, FDA, and LPP.

3.1 Definition and Notation

Let $\mathbf{M} \in \mathbb{R}^{l \times m}$ be an $l \times m$ matrix that represents a sample and let $\mathbf{P} \in \mathbb{R}^{l \times r}$ be a low-dimensional, compact representation of the sample $\mathbf{M}$ in a reduced space with dimensionality r.

In linear projective methods, $\mathbf{P}$ is a projection of the matrix $\mathbf{M}$ in another vector space, which is defined by a transformation matrix $\mathbf{T} \in \mathbb{R}^{m \times r}$. The compact representation $\mathbf{P}$ of the sample $\mathbf{M}$ is obtained as

$$\mathbf{P} = \mathbf{M}\mathbf{T}. \tag{3.1}$$

Let $\tilde{\mathbf{X}} \in \mathbb{R}^{ln \times m}$ be the $ln \times m$ matrix of all n samples. This matrix is constructed by stacking the samples such as

$$\tilde{\mathbf{X}} = \begin{bmatrix} \mathbf{M}_1 \\ \mathbf{M}_2 \\ \vdots \\ \mathbf{M}_n \end{bmatrix}. \tag{3.2}$$

The *Euclidean distance* between two matrices $\mathbf{M}_1 = (\mathbf{m}_{11}|\mathbf{m}_{12}|\cdots|\mathbf{m}_{1m})$ and $\mathbf{M}_2 = (\mathbf{m}_{21}|\mathbf{m}_{22}|\cdots|\mathbf{m}_{2m})$ is defined as the sum of the distances between its columns, that is

$$D(\mathbf{M}_1, \mathbf{M}_2) = \sum_{k=1}^{m} d(\mathbf{m}_{1k}, \mathbf{m}_{2k}), \tag{3.3}$$

where $\mathbf{m}_{jk}$ (for $k = 1, 2, \ldots, m$) is the k-th column of the matrix $\mathbf{M}_j$ and $d(\mathbf{m}_1, \mathbf{m}_2)$ denotes the Euclidean distance between column vectors $\mathbf{m}_1$ and $\mathbf{m}_2$, as defined in Section 2.1.

3.2 Two-dimensional Principal Component Analysis (2D-PCA)

Two-dimensional Principal Component Analysis (2D-PCA) is a matrix-based variant of PCA proposed by Yang et al. (2004) [1]. As the name suggests, it is very similar to the original PCA. In fact, it only changes the way the total scatter matrix is computed, adapting it to deal directly with matrices.

Let $\tilde{\mathbf{S}}_t$ be the new total scatter matrix, defined as

$$\tilde{\mathbf{S}}_t = \sum_{i=1}^{n} (\mathbf{M}_i - \overline{\mathbf{M}})^\top (\mathbf{M}_i - \overline{\mathbf{M}}), \tag{3.4}$$

where $\overline{\mathbf{M}}$ is the mean of all samples, such that $\overline{\mathbf{M}} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{M}_i$. The optimal transformation matrix $\mathbf{T}_{\text{2D-PCA}}$ is defined in the same manner that the original PCA discussed in Section 2.2, just replacing $\mathbf{S}_t$ by $\tilde{\mathbf{S}}_t$.

[1] Actually, the method was first described in Yang and yu Yang (2002), but it was called IMPCA. Later, Yang et al. (2004) provided a better analysis of the technique and renamed it to 2D-PCA.

3.3 Two-dimensional Fisher Discriminant Analysis (2D-FDA)

Two-dimensional Fisher Discriminant Analysis (2D-FDA) is a matrix-based version of FDA described by Xiong et al. (2005) [2]. It works like the original FDA, only adjusting the scatter matrices to use two-dimensional patterns.

Let $\tilde{\mathbf{S}}_b$ and $\tilde{\mathbf{S}}_w$ be, respectively, the new between-class scatter matrix and the new within-class scatter matrix, computed as

$$\tilde{\mathbf{S}}_b = \sum_{l=1}^{c} n'_l (\overline{\mathbf{M}}_l - \overline{\mathbf{M}})^\top (\overline{\mathbf{M}}_l - \overline{\mathbf{M}}), \tag{3.5}$$

$$\tilde{\mathbf{S}}_w = \sum_{l=1}^{c} \sum_{i:y_i=l} (\mathbf{M}_i - \overline{\mathbf{M}}_l)^\top (\mathbf{M}_i - \overline{\mathbf{M}}_l), \tag{3.6}$$

where n' is the number of labeled samples, n'_l is the number of labeled samples of the class l, $\sum_{i:y_i=l}$ indicates the summation over the samples with class l and $\overline{\mathbf{M}}_l$ is the mean of samples of the class l. The optimal transformation matrix $\mathbf{T}_{\text{2D-FDA}}$ has the same definition of the original FDA discussed in Section 2.3, but replacing $\mathbf{S}_b$ and $\mathbf{S}_w$ by $\tilde{\mathbf{S}}_b$ and $\tilde{\mathbf{S}}_w$, respectively.

3.4 Two-dimensional Locality Preserving Projections (2D-LPP)

Two-dimensional Locality Preserving Projections (2D-LPP) is a matrix-based extension of LPP proposed by Chen et al. (2007). Like the previous techniques, LPP is analogous to the original method. It just slightly modifies the affinity matrix and the definition of the optimal transformation matrix to work with two-dimensional patterns.

Let $\tilde{\mathbf{A}}$ be the new affinity matrix such that

$$\tilde{A}_{i,j} = \exp\left(-\frac{\|\mathbf{M}_i - \mathbf{M}_j\|^2}{\sigma_i \sigma_j}\right), \tag{3.7}$$

where $\|\cdot\|$ denotes the sum of the Euclidean norm of the matrix rows and the parameter σ_i represents the local scaling around $\mathbf{M}_i$, as defined in Section 2.4.

Let $\tilde{\mathbf{D}}$ be the diagonal matrix whose entries are column sums of $\tilde{\mathbf{A}}$ such that $\tilde{D}_{i,i} = \sum_j \tilde{A}_{i,j}$ and $\tilde{\mathbf{L}} = \tilde{\mathbf{D}} - \tilde{\mathbf{A}}$ be the new Laplacian matrix. The optimal transformation matrix $\mathbf{T}_{\text{2D-LPP}}$ is defined as

$$\mathbf{T}_{\text{2D-LPP}} = \underset{\mathbf{T}\in\mathbb{R}^{m\times r}}{\arg\min} \left[\frac{\det(\mathbf{T}^\top \tilde{\mathbf{X}}^\top (\tilde{\mathbf{L}} \otimes \mathbf{I_m}) \tilde{\mathbf{X}} \mathbf{T})}{\det(\mathbf{T}^\top \tilde{\mathbf{X}}^\top (\tilde{\mathbf{D}} \otimes \mathbf{I_m}) \tilde{\mathbf{X}} \mathbf{T})}\right], \tag{3.8}$$

where $\otimes$ denotes the Kronecker product and $\mathbf{I}_m$ is the identity matrix in $\mathbb{R}^{m\times m}$.

[2] In reality, the original name of the method is *Two-dimensional Fisher Linear Discriminant* (2D-FLD), but here we use the equivalent name 2D-FDA.

A solution for this problem is given by the generalized eigenvectors $\{\boldsymbol{\phi}_k\}_{k=1}^{r}$ of the following generalized eigendecomposition problem

$$\tilde{\mathbf{X}}^{\top}(\tilde{\mathbf{L}} \otimes \mathbf{I_m})\tilde{\mathbf{X}}\boldsymbol{\phi} = \lambda \tilde{\mathbf{X}}^{\top}(\tilde{\mathbf{D}} \otimes \mathbf{I_m})\tilde{\mathbf{X}}\boldsymbol{\phi}, \tag{3.9}$$

assuming that the eigenvectors are sorted in ascending order according to their associated eigenvalues $\{\lambda_k\}_{k=1}^{r}$.

CHAPTER 4

The Matrix Scatter-based Dimensionality Reduction Framework

In this chapter we show that, like the vector-based counterparts, the matrix-based methods discussed in the previous chapter can also be generalized to a *matrix scatter-based dimensionality framework*.

We already discussed that analyzing methods within a framework is useful to compare them and to understand how they are related (see more in Section 2.9.1). However, in this case there is one additional, even more important benefit. The framework enables us to easily create novel matrix-based extensions for existing vector-based methods.

In this work, we create matrix-based (or, equivalently, two dimensional) versions of the supervised method LFDA and of the semi-supervised methods SSDR, DPCA and SELF. These versions combine the advantages of the matrix-based approach with the particular characteristics of each method. Section 4.1 describes the matrix-based framework. Section 4.2 shows how 2D-PCA, 2D-FDA and 2D-LPP are defined within the matrix-based framework. Section 4.3 describe the proposed matrix-based methods Two-dimensional Local Fisher Discriminant Analysis (2D-LFDA), Two-dimensional Semi-supervised Dimensionality Reduction (2D-SSDR), Two-dimensional Discriminant Principal Component Analysis (2D-DPCA) and Two-dimensional Semi-supervised Local Fisher Discriminant Analysis (2D-SELF) in the terms of the framework.

4.1 The Matrix Scatter-based Framework

All scatter matrices of the matrix-based methods discussed in the previous chapter can be expressed in the following pairwise form

$$\tilde{\mathbf{S}} = \frac{1}{2} \sum_{i,j=1}^{n} W_{i,j} (\mathbf{M}_i - \mathbf{M}_j)^\top (\mathbf{M}_i - \mathbf{M}_j), \tag{4.1}$$

where $\mathbf{M}_i$ are $l \times m$ matrices that represent samples, as defined in Section 3.1, and $\mathbf{W}$ is an $n \times n$ symmetric matrix that is defined according to each method. The element $W_{i,j}$ represents the weight between the samples $\mathbf{M}_i$ and $\mathbf{M}_j$.

Note that this equation is similar to the pairwise form used by the vector scatter-based dimensionality reduction framework, described in Section 2.9. In fact, the frameworks are analogous, in the same way the matrix-based methods are analogous to the

corresponding vector-based methods. Both use a maximization problem to find $\mathbf{T}_{\text{OPT}}$, the optimal projection matrix, defined as

$$\mathbf{T}_{\text{OPT}} = \underset{T \in \mathbb{R}^{m \times r}}{\arg\max} \left[\frac{\det(\mathbf{T}^\top \mathbf{B} \mathbf{T})}{\det(\mathbf{T}^\top \mathbf{C} \mathbf{T})} \right], \tag{4.2}$$

where $\det(\cdot)$ denotes the determinant of a matrix. Again, $\mathbf{B}$ corresponds to the characteristic we want to increase in the projection space (between-class scatter, for instance) whereas $\mathbf{C}$ corresponds to the characteristic we want to decrease (within-class scatter, for instance).

Let $\{\boldsymbol{\phi}_k\}_{k=1}^r$ be the generalized eigenvectors associated with the eigenvalues $\{\lambda_k\}_{k=1}^r$ of the following generalized eigenvalue problem

$$\mathbf{B}\boldsymbol{\phi} = \lambda \mathbf{C}\boldsymbol{\phi}. \tag{4.3}$$

As discussed before, assuming that the generalized eigenvalues are sorted in descending order as $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_r$ and that the generalized eigenvectors are normalized as $\boldsymbol{\phi}_k^\top \mathbf{C} \boldsymbol{\phi}_k = 1$ for $k = 1, 2, \ldots, r$, then a solution $\mathbf{T}_{\text{OPT}}$ is analytically given as (Fukunaga, 1990)

$$(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_r). \tag{4.4}$$

We observed empirically that the heuristic proposed by Sugiyama (2007) is also useful for matrix-based methods. He found out that the distance metric in the projection space can be improved by deemphasizing the minor eigenvectors according to the square root of their corresponding eigenvalues. For this reason, we adjust the solution such as

$$\mathbf{T}_{\text{OPT}} = (\sqrt{\lambda_1}\boldsymbol{\phi}_1, \sqrt{\lambda_2}\boldsymbol{\phi}_2, \ldots, \sqrt{\lambda_r}\boldsymbol{\phi}_r). \tag{4.5}$$

In this work, we performed experiments to evaluate the efficiency of this weighting scheme.

4.2 Defining Methods within the Framework

In this section, we show how the matrix-based methods discussed in the previous chapter are expressed in the matrix scatter-based framework. The matrices $\mathbf{B}$ and $\mathbf{C}$ are defined for each method. Appendix A demonstrates how we derived these equations from the original definitions.

4.2.1 Two-dimensional Principal Component Analysis (2D-PCA)

The optimal projection matrix $\mathbf{T}_{\text{2D-PCA}}$ is given by the equations (4.3) and (4.5) with

$$\mathbf{B} = \tilde{\mathbf{S}}_t \quad \text{and} \quad \mathbf{C} = \mathbf{I}_m, \tag{4.6}$$

$$\tilde{\mathbf{S}}_t = \frac{1}{2}\sum_{i,j=1}^{n} \tilde{W}_{i,j}^{(t)} (\mathbf{M}_i - \mathbf{M}_j)^\top (\mathbf{M}_i - \mathbf{M}_j), \text{ with } \tilde{W}_{i,j}^{(t)} = \frac{1}{n} \tag{4.7}$$

where $\mathbf{I}_m$ is the identity matrix in $\mathbb{R}^{m \times m}$ and n is the total number of samples. Note that the weights $\tilde{W}_{i,j}^{(t)}$ are the same obtained for the pairwise form of PCA.

4.2.2 Two-dimensional Fisher Discriminant Analysis (2D-FDA)

The optimal projection matrix $\mathbf{T}_{\text{2D-FDA}}$ is given by the equations (4.3) and (4.5) with

$$\mathbf{B} = \tilde{\mathbf{S}}_b \quad \text{and} \quad \mathbf{C} = \tilde{\mathbf{S}}_w, \tag{4.8}$$

$$\tilde{\mathbf{S}}_b = \frac{1}{2}\sum_{i,j=1}^{n} \tilde{W}_{i,j}^{(b)} (\mathbf{M}_i - \mathbf{M}_j)^\top (\mathbf{M}_i - \mathbf{M}_j), \text{ with } \tilde{W}_{i,j}^{(b)} = \begin{cases} 1/n' - 1/n'_{y_i} & \text{if } y_i = y_j \\ 1/n' & \text{if } y_i \neq y_j \end{cases}, \tag{4.9}$$

$$\tilde{\mathbf{S}}_w = \frac{1}{2}\sum_{i,j=1}^{n} \tilde{W}_{i,j}^{(w)} (\mathbf{M}_i - \mathbf{M}_j)^\top (\mathbf{M}_i - \mathbf{M}_j), \text{ with } \tilde{W}_{i,j}^{(w)} = \begin{cases} 1/n'_{y_i} & \text{if } y_i = y_j \\ 0 & \text{if } y_i \neq y_j \end{cases}, \tag{4.10}$$

where n' is the number of labeled samples, y_i is the class of the sample $\mathbf{M}_i$ and n'_{y_i} is the number of samples of the class y_i. Note that the weights $\tilde{W}_{i,j}^{(b)}$ and $\tilde{W}_{i,j}^{(w)}$ are the same obtained for the pairwise form of FDA.

4.2.3 Two-dimensional Locality Preserving Projections (2D-LPP)

Like the original LPP, the 2D-LPP was formulated as a minimization problem. However, since the matrix-based framework is based on a maximization problem, we should consider an inverted version of 2D-LPP (2D-iLPP).

The optimal projection matrix $\mathbf{T}_{\text{2D-iLPP}}$ is given by the equations (4.3) and (4.5) with

$$\mathbf{B} = \tilde{\mathbf{S}}_n \quad \text{and} \quad \mathbf{C} = \tilde{\mathbf{S}}_l, \tag{4.11}$$

$$\tilde{\mathbf{S}}_n = \sum_{i=1}^{n} \tilde{D}_{i,i}^{(n)} \mathbf{M}_i^\top \mathbf{M}_i, \text{ with } \tilde{D}_{i,i}^{(n)} = \sum_{j=1}^{n} \tilde{A}_{i,j}, \tag{4.12}$$

$$\tilde{\mathbf{S}}_l = \frac{1}{2}\sum_{i,j=1}^{n} \tilde{W}_{i,j}^{(l)} (\mathbf{M}_i - \mathbf{M}_j)^\top (\mathbf{M}_i - \mathbf{M}_j), \text{ with } \tilde{W}_{i,j}^{(l)} = \tilde{A}_{i,j}, \tag{4.13}$$

where $\tilde{A}_{i,j}$ is the affinity matrix between the samples $\mathbf{M}_i$ and $\mathbf{M}_j$ (as defined in Section 3.4). Note that the weights $\tilde{W}_{i,j}^{(l)}$ are equivalent to the ones obtained for the pairwise form of iLPP.

4.3 Two-dimensional Extensions of LFDA, SSDR, DPCA and SELF

As noted in the previous section, the weights obtained for the pairwise form of 2D-PCA, 2D-FDA, and 2D-LPP are exactly the same obtained for the pairwise form of PCA, FDA, and LPP, respectively.

Although expected, this fact has interesting implications. First, it reinforces the notion that the discussed matrix-based methods are equivalent to the corresponding vector-based methods. But, more importantly, it suggests that novel two-dimensional methods can be created just by porting the weights in the pairwise equation from the vector-based framework to the matrix-based framework. Based on this observation, we define two-dimensional extensions for LFDA, SSDR, DPCA, and SELF in this section.

4.3.1 Two-dimensional Local Fisher Discriminant Analysis (2D-LFDA)

The *Two-dimensional Local Fisher Discriminant Analysis* (2D-LFDA) is the matrix-based extension of LFDA. It aims to keep nearby samples that belongs to the same class close together and to keep samples in different classes far apart. That is, it seeks to maximize a localized version of the between-scatter matrix and to minimize the local within-scatter.

By definition, 2D-LFDA is expressed in the matrix-based framework and uses the same weights from the pairwise equation of LFDA. The optimal projection matrix $\mathbf{T}_{\text{2D-LFDA}}$ is given by the equations (4.3) and (4.5) with

$$\mathbf{B} = \tilde{\mathbf{S}}_{lb} \quad \text{and} \quad \mathbf{C} = \tilde{\mathbf{S}}_{lw}, \tag{4.14}$$

$$\tilde{\mathbf{S}}_{lb} = \frac{1}{2}\sum_{i,j=1}^{n} \tilde{W}_{i,j}^{(lb)} (\mathbf{M}_i - \mathbf{M}_j)^\top (\mathbf{M}_i - \mathbf{M}_j), \;\; \tilde{W}_{i,j}^{(lb)} = \begin{cases} \tilde{A}_{i,j}(1/n' - 1/n'_{y_i}) & \text{if } y_i = y_j \\ 1/n' & \text{if } y_i \neq y_j \end{cases}, \tag{4.15}$$

$$\tilde{\mathbf{S}}_{lw} = \frac{1}{2}\sum_{i,j=1}^{n} \tilde{W}_{i,j}^{(lw)} (\mathbf{M}_i - \mathbf{M}_j)^\top (\mathbf{M}_i - \mathbf{M}_j), \text{ with } \tilde{W}_{i,j}^{(lw)} = \begin{cases} \tilde{A}_{i,j}/n'_{y_i} & \text{if } y_i = y_j \\ 0 & \text{if } y_i \neq y_j \end{cases}, \tag{4.16}$$

where $\tilde{A}_{i,j}$ is the affinity matrix between the samples $\mathbf{M}_i$ and $\mathbf{M}_j$ (as defined in Section 3.4), n' is the number of labeled samples, y_i is the class of the sample $\mathbf{M}_i$ and n'_{y_i} is the number of samples of the class y_i.

4.3.2 Two-dimensional Semi-supervised Dimensionality Reduction (2D-SSDR)

The *Two-dimensional Semi-supervised Dimensionality Reduction* (2D-SSDR) is the matrix-based version of SSDR. It seeks a projection space that maximizes the distance between

samples that belong to different classes (or, equivalently, samples involved in cannot-link constraints) and that minimizes the distance between samples that belong to the same class (or, equivalently, samples involved in must-link constraints). For unlabeled samples, 2D-SSDR just uses the same criteria from 2D-PCA.

By definition, 2D-SSDR is expressed in the matrix-based framework and uses the same weights from the pairwise equation of SSDR. The optimal projection matrix $\mathbf{T}_{\text{2D-SSDR}}$ is given by the equations (4.3) and (4.5) with

$$\mathbf{B} = \tilde{\mathbf{S}}_{ssdr} \quad \text{and} \quad \mathbf{C} = \mathbf{I}_m, \tag{4.17}$$

$$\tilde{\mathbf{S}}_{ssdr} = \frac{1}{2}\sum_{i,j=1}^{n} \tilde{W}_{i,j}^{(ssdr)} (\mathbf{M}_i - \mathbf{M}_j)^\top (\mathbf{M}_i - \mathbf{M}_j), \;\; \tilde{W}_{i,j}^{(ssdr)} = \begin{cases} 1/n^2 + \alpha/n_c & \text{if } (\mathbf{M}_i, \mathbf{M}_j) \in \mathbf{C}^{(l)} \\ 1/n^2 - \beta/n_m & \text{if } (\mathbf{M}_i, \mathbf{M}_j) \in \mathbf{M}^{(l)} \\ 1/n^2 & \text{otherwise} \end{cases}, \tag{4.18}$$

where $\mathbf{I}_m$ is the identity matrix in $\mathbb{R}^{m \times m}$, $\mathbf{C}^{(l)}$ and $\mathbf{M}^{(l)}$ are, respectively, the cannot-link and must-link constraints sets, n_c and n_m are, respectively, the numbers of cannot-link and must-link constraints, n is the total number of samples, and α and β are user-defined parameters.

4.3.3 Two-dimensional Discriminant Principal Component Analysis (2D-DPCA)

The *Two-dimensional Discriminant Principal Component Analysis* (2D-DPCA) is the matrix-based variant of DPCA. It is similar to 2D-SSDR because it also intends to maximize the distance in the projection space between samples from different classes and to minimize the distance between samples of the same class. Additionally, it uses the 2D-PCA rationale for unlabeled samples too. However, 2D-DPCA defines the weights in a different way.

By definition, 2D-DPCA is expressed in the matrix-based framework and uses the same weights from the pairwise equation of DPCA. The optimal projection matrix $\mathbf{T}_{\text{2D-DPCA}}$ is given by the equations (4.3) and (4.5) with

$$\mathbf{B} = \tilde{\mathbf{S}}_{dpca} \quad \text{and} \quad \mathbf{C} = \mathbf{I}_m, \tag{4.19}$$

$$\tilde{\mathbf{S}}_{dpca} = \frac{1}{2}\sum_{i,j=1}^{n} \tilde{W}_{i,j}^{(dpca)} (\mathbf{M}_i - \mathbf{M}_j)^\top (\mathbf{M}_i - \mathbf{M}_j), \;\; \tilde{W}_{i,j}^{(dpca)} = \begin{cases} \lambda/n^2 + 2/n_c & \text{if } (\mathbf{M}_i, \mathbf{M}_j) \in \mathbf{C}^{(l)} \\ \lambda/n^2 - 2\eta/n_m & \text{if } (\mathbf{M}_i, \mathbf{M}_j) \in \mathbf{M}^{(l)} \\ \lambda/n^2 & \text{otherwise} \end{cases}, \tag{4.20}$$

where $\mathbf{I}_m$ is the identity matrix in $\mathbb{R}^{m \times m}$, $\mathbf{C}^{(l)}$ and $\mathbf{M}^{(l)}$ are, respectively, the cannot-link and must-link constraints sets, n_c and n_m are, respectively, the numbers of cannot-link and must-link constraints, n is the total number of samples, and λ and η are user-defined parameters.

4.3.4 Two-dimensional Semi-supervised Local Fisher Discriminant Analysis (2D-SELF)

The *Two-dimensional Semi-supervised Local Fisher Discriminant Analysis* (2D-SELF) is the matrix-based version of SELF. It combines 2D-LFDA and 2D-PCA. Its matrices are expressed by the 2D-LFDA between- and within-scatter matrices regularized, respectively, by the covariance matrix from 2D-PCA and by the identity matrix.

By definition, 2D-SELF is expressed in the matrix-based framework and uses the same weights from the pairwise equation of SELF. The optimal projection matrix $\mathbf{T}_{\text{2D-SELF}}$ is defined by the equations (4.3) and (4.5) with

$$\mathbf{B} = \tilde{\mathbf{S}}_{rlb} \quad \text{and} \quad \mathbf{C} = \tilde{\mathbf{S}}_{rlw}, \tag{4.21}$$

$$\tilde{\mathbf{S}}_{rlb} = (1-\beta)\tilde{\mathbf{S}}_{lb} + \beta\tilde{\mathbf{S}}_{t}, \tag{4.22}$$

$$\tilde{\mathbf{S}}_{rlw} = (1-\beta)\tilde{\mathbf{S}}_{lw} + \beta\mathbf{I}_{m}, \tag{4.23}$$

where $\tilde{\mathbf{S}}_{lb}$ and $\tilde{\mathbf{S}}_{lw}$ are the local between- and within-scatter matrices as defined by 2D-LFDA, $\tilde{\mathbf{S}}_{t}$ is the total covariance matrix as defined by 2D-PCA, $\mathbf{I}_m$ is the identity matrix in $\mathbb{R}^{m\times m}$, and β is an user-defined parameter.

4.4 Implementation Notes

The scatter matrices in the pairwise form can be represented in a compact way, as

$$\begin{aligned}
\tilde{\mathbf{S}} &= \frac{1}{2}\sum_{i,j=1}^{n} W_{i,j}(\mathbf{M}_i - \mathbf{M}_j)^\top(\mathbf{M}_i - \mathbf{M}_j) && \text{(4.24a)}\\
&= \sum_{i,j=1}^{n} W_{i,j}\frac{1}{2}\left(\mathbf{M}_i^\top\mathbf{M}_i - \mathbf{M}_i^\top\mathbf{M}_j - \mathbf{M}_j^\top\mathbf{M}_i + \mathbf{M}_j^\top\mathbf{M}_j\right) && \text{(4.24b)}\\
&= \sum_{i,j=1}^{n}\left(W_{i,j}\mathbf{M}_i^\top\mathbf{M}_i - W_{i,j}\mathbf{M}_i^\top\mathbf{M}_j\right) && \text{(4.24c)}\\
&= \sum_{i,j=1}^{n}\mathbf{M}_i^\top(W_{i,j}\mathbf{I}_l)\mathbf{M}_i - \sum_{i,j=1}^{n}\mathbf{M}_i^\top(W_{i,j}\mathbf{I}_l)\mathbf{M}_j && \text{(4.24d)}\\
&= \tilde{\mathbf{X}}^\top\left[(\tilde{\mathbf{D}} - \tilde{\mathbf{W}})\otimes\mathbf{I}_l\right]\tilde{\mathbf{X}} && \text{(4.24e)}\\
&= \tilde{\mathbf{X}}^\top(\tilde{\mathbf{L}}\otimes\mathbf{I}_l)\tilde{\mathbf{X}}, && \text{(4.24f)}
\end{aligned}$$

where $\otimes$ denotes the Kronecker product, $\tilde{\mathbf{X}}$ is the $ln \times m$ matrix with all samples (as defined in Section 3.1), $\tilde{\mathbf{D}}$ is diagonal matrix where $\tilde{D}_{i,i} = \sum_{j=1}^{n} W_{i,j}$ and $\mathbf{I}_l$ is the identity matrix in $\mathbb{R}^{l\times l}$.

In Section 2.9.2, we observed that the vector-based framework also has a compact representation for its pairwise equation. Then, we argued that this representation is more efficient to compute. However, this is not the case here. In the matrix-based framework, the computation of the compact form is much more expensive than the computation of the pairwise form.

This difference is caused by the Kronecker product. This product creates very large matrices, what makes the computation of the compact form memory-intensive. The problem could be alleviated by making some matrices sparse, but the memory consumption would still be higher than in the pairwise version.

Additionally, the operations performed inside the summation have different complexities. In the the vector-based pairwise form, each iteration of the sum involves the product of two lm-dimensional vectors, producing $lm \times lm$ matrices. In the matrix-based pairwise form, the iterations perform the product of two $l \times m$ matrices, what produces smaller matrices, with size $m \times m$. For these reasons, the pairwise form is the recommended way to implement matrix-based methods.

CHAPTER 5

Experiments and Analysis

We evaluated the proposed methods 2D-LFDA, 2D-SSDR, 2D-DPCA, 2D-SELF and the other ones discussed in the previous chapters in the context of image classification. For this purpose, two face databases were used: ORL and FEI. The ORL database was used to examine the methods performance when images have varying pose and scaling in addition to occlusive accessories, such as glasses. The FEI database was used to evaluate the methods performance with images of different genders and facial expressions.

The experiments were carried out in the *transductive setting* (Vapnik and Kotz, 2006), which uses all the available patterns—labeled and unlabeled—both for training and testing. The goal of the transductive approach is to create a model to classify only the unlabeled samples we have now. This contrasts with the traditional generalization approach, which aims to build a model capable of classifying any new pattern that might appear. The transductive setting is a standard choice in works related to semi-supervised learning (Chapelle et al., 2006).

In this chapter, we describe how the experiments were conducted and compare the results for distinct methods. We also analyze how the methods performance varies with different numbers of labeled samples, principal components and image sizes. Section 5.1 describes the general methodology used in the experiments. Section 5.2 discusses the experiments with the ORL database in the context of two tasks (face recognition and glasses detection). Section 5.3 presents the experiments with the FEI database in the context of two another tasks (gender and smile detection). Finally, Section 5.4 discusses the results in general and summarizes the conclusions.

5.1 Methodology

The datasets were divided into two groups: the labeled and unlabeled sets, which respectively correspond to the training and testing sets. The training set was used to create the classifying system whereas the testing set was employed to evaluate its performance.

All the experiments started with the dimensionality reduction step. To do this, the discussed methods were used to extract the features of the samples. The principal component vectors (PCs) produced by each method were used to project all the samples into a new, reduced feature space. Then, a classifier was built only with the labeled reduced samples. This classifier was used to predict the labels of the unla-

beled reduced samples. Finally, the predictions were compared with the actual labels to compute misclassification rates for each method.

In order to avoid bias caused by any particular classification technique, we chose the *nearest neighbor classifier* (1-NN) for all experiments. This is one of the simplest classifiers and it has been shown to perform reasonably well across many datasets and methods, specially in semi-supervised contexts (Sugiyama et al., 2010; Chapelle et al., 2006).

As explained in Chapter 2, the vector-based methods (PCA, iLPP, FDA, LFDA, SSDR, DPCA, and SELF) work only with vectors. For these methods, the images were transformed into one-dimensional column vectors before feature extraction. The matrix-based methods (2D-PCA, 2D-iLPP, 2D-FDA, 2D-LFDA, 2D-SSDR, 2D-DPCA, and 2D-SELF) used the two-dimensional images directly as input.

Because the experiments were performed in the transductive setting, the training and testing sets were handled differently for each learning paradigm in the dimensionality reduction step. For the unsupervised methods (PCA, iLPP, 2D-PCA, and 2D-iLPP), the sets were merged and all the samples were used for feature extraction, without labels. For the supervised techniques (FDA, LFDA, 2D-FDA, and 2D-LFDA), only the training samples were used as input, accompanied by their respective labels. And for the semi-supervised methods (SSDR, DPCA, SELF, 2D-SSDR, 2D-DPCA, and 2D-SELF), all the samples were used along with the labels available in the training set.

Hence, in the experiments we simulated a scenario where there are many samples but only some of them are labeled. On one hand, unsupervised methods can use all samples, but no labels. On the other hand, supervised methods can use labels, but are limited only to the information provided by the (few) labeled samples. In contrast, semi-supervised methods can overcome these limitations by using the information from all samples while also taking advantage of the discriminative power provided by the available labels.

It is known that the choice of the training set can affect the system performance in a great extent (Mitchell, 1997). To assure that the results are not tied to a particular training set, all experiments were repeated 12 times, with different splits of labeled and unlabeled samples. The samples were put randomly in each set. The labeled sets contain the same number of samples for each class. To enable a fair comparison, the same splits were used for all methods. The misclassification rates shown in the results tables are averages between the repetitions.

Another important aspect to consider is how many labeled samples are necessary to achieve a reasonable accuracy. To investigate this, we performed experiments with training sets of different sizes. For the face recognition task, experiments were executed with 1, 2, 3, 4, and 5 labeled samples of each class. For the other tasks, experiments were executed with 1, 5, 10, 25, 50, and 100 labeled samples of each class. In the result tables, the number of samples is indicated by n. Note that this variation is relevant even for unsupervised techniques because although the labels are ignored in the dimensionality reduction phase, they are used to create the classifier.

We also evaluated how the system performance is affected by the number of se-

lected PCs. Intuitively, more PCs should improve the accuracy since there is more information available for the classifier. In practice, however, this is not always true because the additional information may not be useful for classification (Duda et al., 2001). Actually, we observed that in many situations more PCs yielded higher misclassification rates. Generally, it is not possible to determine the optimal number of PCs in advance and, even if that was possible, this number would be different for each method and we could not compare them in a meaningful way. So, as proposed by Sugiyama et al. (2010), we decided to compare the methods performance by using the average misclassification rate over the reduced dimensions.

Finally, we wanted to determine whether the weighting scheme described in Chapters 2 and 3 is really useful to improve the system accuracy. For this extent, we performed experiments with *plain* and *weighted* versions of the methods. In general we observed that, contrary to Sugiyama et al. (2010) findings, the versions without weights produced better misclassification rates for vector-based methods. In contrast, the weighted versions performed better for matrix-based methods. Hence, the tables show results obtained with plain vector methods and weighted matrix methods.

For methods that require parameters, we chose the same values used in the works where they were proposed. For SSDR and 2D-SSDR, $\alpha = 1$ and $\beta = 20$; for DPCA and 2D-DPCA, $\eta = 10$ and $\lambda = 0.1$; and for SELF and 2D-SELF, $\beta = 0.5$. Because LFDA generated some singular scatter matrices, we treated SELF with $\beta = 0.001$ as a slightly regularized version of LFDA. The experiments were run on the Matlab enviroment (version 2012b) over the Mac OS X platform. We released the source code used in the experiments as free software [1].

5.2 Experiments on the ORL Database

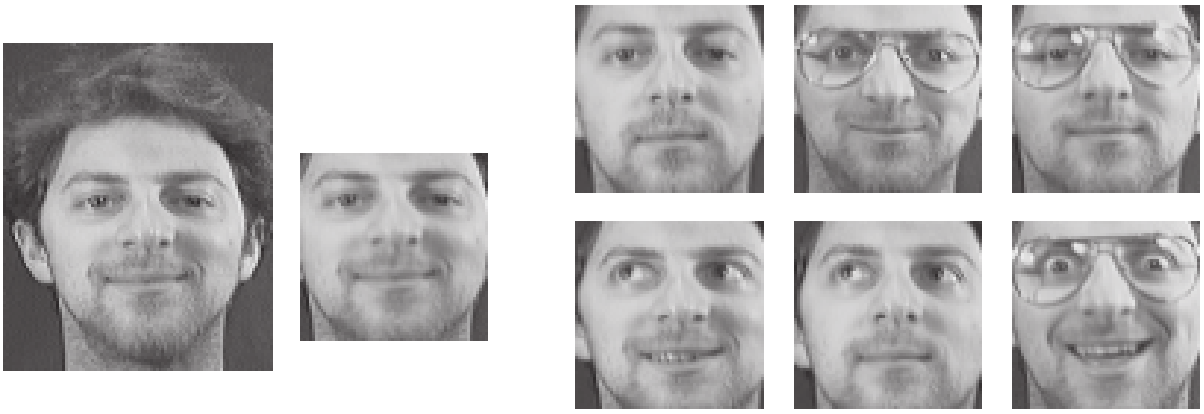
The ORL Database of Faces (from AT&T Laboratories Cambridge) [2] (Samaria and Harter, 1994) contains images from 40 individuals, each having 10 different images, summing up 400 images in total. For some subjects, the images were taken at different times. There are variations in facial expressions (open or closed eyes, smiling or non-smiling) and facial details (glasses or no glasses). The images were taken with a tolerance for some tilting and rotation of the face up to 20 degrees. There is also some variation in the scale up to 10 percent. All images are grayscale (256 levels) and of size 112×92 pixels.

To improve the overall performance and to simplify the computation in the experiments, we used a cropped version provided by Roweis [3]. In this version, part of the background was removed and the images were aligned to have the eyes in the same

[1] The source code of the experiments is available for download at http://github.com/lailsonbm/2d_semi_supervised

[2] The ORL Database of Faces is available for download at http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.

[3] The cropped version of the ORL database is available in Matlab format at http://www.cs.nyu.edu/~roweis/data.html.



(a) (b)

Figure 5.1 Example of images from the ORL database. The group (a), on the left, shows an original image and its respective cropped version. The group (b), on the right, shows another six cropped images from the same subject, with different facial expressions.

positions. Then, the images were resized to 64×64 pixels (4.096 dimensions in vector form). Figure 5.1 shows an uncropped image and its corresponding cropped version, with other cropped images of the same individual on the right.

5.2.1 Face Recognition Task

In the first experiment, we created a system for the face recognition task: given a face image, the classifier must determine what person is depicted in it. The database contains 40 individuals, that are represented by 10 images each. Thus, this is a classification problem with 40 classes and 10 samples per class.

The experiments were performed with 1, 2, 3, 4 and 5 labeled samples of each class, or, equivalently, 40, 80, 120, 160 and 200 labeled samples in total. This corresponds to 10%, 20%, 30%, 40% and 50% of the total samples, respectively.

The misclassification rates are listed in Table 5.1. As expected, most of the best rates were produced by the semi-supervised methods. Surprisingly, the majority of these best rates came from vector methods instead of matrix methods (a matrix method performed better only when $n = 1$). This observation could imply that two-dimensional methods have a bad performance in most situations, but that is not the case. When we compared the rates of two-dimensional methods with the rates of their one-dimensional counterparts, we noticed that 2D-PCA, 2D-iLPP, 2D-FDA, and 2D-SSDR achieved better results than the corresponding vector methods. It is also interesting to see that LFDA—a supervised method—produced some of the best rates. We acknowledge this behavior to its similarities with SELF.

It was clear that methods performed better with more labeled samples, except for FDA and 2D-DPCA. Both yielded poor results in general, especially when more la-

bels were present. It is interesting to note that the decrease in misclassification rates was not even. The difference was more pronounced from $n = 1$ to $n = 2$, where some rates decreased by half. From there, they dropped more slowly. This observation suggests that the relative power of discriminative information diminishes as more labeled examples are added.

5.2.2 Glasses Detection Task

In the glass detection task, the system should decide whether the person in the image is wearing glasses or not. The ORL database contains 119 images of subjects with glasses and 281 of subjects without glasses. Among the 40 individuals present in the database, some have pictures only with glasses, others have pictures only without glasses and few have both types, as the one shown in Figure 5.1.

The experiments were performed with 1, 5, 10, 25, 50 and 100 labeled samples of each class, or, equivalently, 2, 10, 50, 100 and 200 labeled samples in total. This corresponds to 0.5%, 2.5%, 5%, 12.5%, 25% and 50% of the total samples, respectively.

The misclassification rates for the glasses detection task are listed in Table 5.2. It has many similarities with the table of the previous experiment. LFDA, DPCA and SELF—three vector-based methods—reached the best rates most of the time, except when $n = 1$. In this case, the best rates are from two-dimensional methods once more. 2D-PCA, 2D-iLPP, 2D-FDA, and 2D-SSDR also showed a better or comparable performance than their analogous one-dimensional methods. Finally, the misclassification rates were generally lower when there were more labeled samples, except for FDA and 2D-DPCA, which again showed the worst performances.

Nonetheless, it is possible to notice some important differences too. When $n = 1$, the best rates are from 2D-FDA and 2D-LFDA, which are not semi-supervised, as we would expect. Actually, the matrix-based semi-supervised methods achieved best rates only for $n = 5$, when almost all methods had best performances as well. Another interesting fact is that the lowest rates increased from $n = 1$ to $n = 5$, that is, in this experiment some systems built with 2 labeled samples performed better than all systems built with 10 labeled samples.

METHOD	$n = 1$	$n = 2$	$n = 3$	$n = 4$	$n = 5$
PCA	44.19 (2.13)	29.99 (2.36)	22.67 (2.14)	17.19 (2.44)	13.22 (1.70)
iLPP	62.84 (2.25)	47.74 (2.42)	38.18 (2.31)	31.94 (2.65)	26.09 (2.24)
FDA	49.04 (1.68)	93.29 (1.72)	91.70 (1.90)	90.74 (2.51)	87.39 (4.14)
LFDA	45.43 (2.42)	**18.33** (1.73)	**10.90** (1.72)	**7.94** (1.42)	**6.43** (1.12)
SSDR	44.98 (2.21)	30.66 (2.29)	23.24 (2.19)	17.76 (2.43)	13.65 (1.71)
DPCA	45.25 (2.28)	21.07 (2.30)	**11.26** (1.90)	**8.73** (2.01)	**6.98** (1.57)
SELF	45.18 (2.29)	**18.16** (1.68)	**10.45** (1.50)	**7.54** (1.43)	**5.95** (1.53)
2D-PCA	40.56 (1.88)	25.97 (2.46)	19.08 (2.11)	13.88 (2.67)	9.70 (1.94)
2D-iLPP	46.73 (1.66)	33.96 (1.92)	24.68 (2.38)	21.29 (2.09)	15.93 (1.96)
2D-FDA	97.50 (0.00)	44.47 (2.58)	17.34 (2.48)	23.81 (2.94)	8.22 (1.72)
2D-LFDA	97.50 (0.00)	32.54 (2.67)	17.34 (2.48)	11.75 (3.10)	8.22 (1.72)
2D-SSDR	40.79 (1.97)	26.41 (2.57)	19.36 (2.11)	14.13 (2.57)	9.92 (1.96)
2D-DPCA	40.88 (2.01)	41.08 (14.7)	68.39 (12.6)	75.99 (1.82)	75.30 (1.81)
2D-SELF	**38.85** (1.85)	33.85 (3.24)	18.60 (2.51)	12.67 (3.21)	8.80 (1.62)

Table 5.1 Misclassification rates (%) for the Face Recognition task with the ORL database. The heading $n = k$ indicates the k numbers of labeled samples used in the experiment. For each k, the table shows the averaged rates over the reduced dimensions (up to 320 PCs for vector methods and 64 PCs for matrix methods, except for FDA, which used only the 39 first PCs) along with their standard deviations in parenthesis. The best mean and the means with no significant statistical difference are highlighted in bold (*t-test* with 5% of significance). All the rates are averages over 12 repetitions with different dataset splits. The feature extraction was performed with the plain version of vector methods and with the weighted version of matrix methods.

METHOD	$n = 1$	$n = 5$	$n = 10$	$n = 25$	$n = 50$	$n = 100$
PCA	44.24 (7.36)	**34.81** (7.10)	27.29 (5.95)	17.23 (2.85)	9.99 (2.58)	5.64 (1.59)
iLPP	42.27 (9.99)	45.86 (7.26)	40.11 (5.69)	31.16 (5.17)	26.33 (4.51)	20.46 (3.03)
FDA	43.76 (10.7)	44.62 (7.63)	48.62 (4.82)	49.95 (5.12)	50.56 (2.95)	51.25 (12.6)
LFDA	43.88 (7.59)	**34.13** (4.83)	**20.51** (6.27)	**9.05** (2.85)	**5.92** (2.06)	**3.17** (1.32)
SSDR	43.92 (7.59)	**34.26** (7.27)	26.76 (5.85)	16.77 (2.95)	9.50 (2.53)	5.42 (1.48)
DPCA	43.91 (7.60)	**34.65** (5.10)	**21.95** (6.27)	**10.33** (3.74)	**6.15** (1.74)	**3.24** (1.89)
SELF	43.97 (7.59)	**34.27** (4.92)	**20.60** (6.35)	**9.12** (2.86)	**5.80** (1.97)	**2.75** (1.66)
2D-PCA	43.46 (6.87)	**34.02** (7.29)	26.25 (6.21)	15.69 (2.95)	8.87 (2.95)	4.54 (1.44)
2D-iLPP	42.43 (9.33)	**31.38** (5.71)	**24.07** (4.62)	15.92 (2.43)	8.66 (2.34)	**3.86** (1.13)
2D-FDA	**29.65** (0.00)	**32.30** (9.74)	31.53 (11.2)	24.14 (6.83)	17.51 (3.25)	5.43 (1.51)
2D-LFDA	**29.65** (0.00)	**32.30** (9.74)	31.53 (11.2)	24.15 (6.84)	17.52 (3.26)	5.44 (1.51)
2D-SSDR	43.47 (6.81)	**33.84** (7.07)	26.57 (6.34)	15.68 (3.04)	8.87 (2.61)	4.57 (1.51)
2D-DPCA	43.68 (6.85)	**32.72** (10.5)	35.85 (13.9)	37.41 (12.8)	40.53 (11.7)	37.0 (8.29)
2D-SELF	43.48 (6.88)	**33.80** (9.21)	32.85 (10.4)	28.15 (8.52)	21.81 (4.19)	7.46 (1.82)

Table 5.2 Misclassification rates (%) for the Glasses Detection task with the ORL database. The heading $n = k$ indicates the k numbers of labeled samples used in the experiment. For each k, the table shows the averaged rates over the reduced dimensions (up to 320 PCs for vector methods and 64 PCs for matrix methods, except for FDA, which used only the first PC) along with their standard deviations in parenthesis. The best mean and the means with no significant statistical difference are highlighted in bold (*t-test* with 5% of significance). All the rates are averages over 12 repetitions with different dataset splits. The feature extraction was performed with the plain version of vector methods and with the weighted version of matrix methods.

5.3 Experiments on the FEI Face Database

The FEI face database [4] (Thomaz and Giraldi, 2010) contains images of 200 subjects with ages between 19 and 40 years. The images are colorful (RGB), have 640×480 pixels and were taken against a homogeneous background, with scaling varying up to 10% of the total images size. This database has many images for each subject, with variations in profile rotation, facial expression and illumination. However, in this work we used a reduced version of the database, which contains 2 images per subject, only in frontal pose. In this reduced version, the images were manually aligned to have eyes and noses positioned roughly in the same location. Then, the images were cropped to 360×260 pixels. The reduced database was created by the same group that developed the original FEI database and is provided along with it. Figure 5.2 shows an image from the database along with its cropped version. Figure 5.3 shows another images from the database. They were converted to grayscale before the experiments.

As discussed in the Section 5.1, bi-dimensional images have to be transformed into vectors before they are processed by vector-based methods. But, due to the relatively high resolution of the images in this database, the resulting vectors would be very large (93.600 dimensions) and it would take too much time to perform the eigendecomposition of the corresponding computed scatter matrices. For this reason, the images were resized to 25% of their original size (90×65 pixels or 5.850 dimensions in vector form) by using the bi-cubic interpolation algorithm (Gonzalez and Woods, 2011).

For the matrix-based methods, we performed experiments with both the resized and the original image sizes. Interestingly, we observed that the performance of the systems that used full-sized images was generally worse than the performance of systems with resized images (and much worse in some cases). It seems that most of the information lost in the resizing process is redundant or even detrimental for classification purposes. So, the resizing operation itself appears to be a form of dimensionality reduction. The tables in the following subsections show results only for experiments performed with the resized images.

5.3.1 Gender Detection Task

The gender detection task is a binary classification problem, in which the system must determine whether the image contains a female or a male subject. The reduced FEI database contains 200 images of female individuals and 200 of male individuals.

As in the previous case, the experiments were performed with 1, 5, 10, 25, 50 and 100 labeled samples of each class, or, equivalently, 2, 10, 50, 100 and 200 labeled samples in total. This corresponds to 0.5%, 2.5%, 5%, 12.5%, 25% and 50% of the total samples, respectively.

The misclassification rates for the gender detection task are listed in Table 5.3. At this time, almost all methods performed well when there were very few labeled samples ($n = 1$). For $n = 5$ and $n = 10$, the best results were produced by three vector-

[4] The FEI face database is available at http://fei.edu.br/~cet/facedatabase.html.

based methods (LFDA, DPCA, and SELF) and two matrix-based methods (2D-PCA and 2D-SSDR). However, when the number of labeled samples increased, only the three vector-based methods kept achieving the lowest rates. In general, all methods performed better with more labeled samples, except for FDA and 2D-DPCA. These results are consistent with was observed in the previous experiments.

One more time, the vector-based methods produced more best rates than the matrix-based methods. In spite of this, when we compare the results of two-dimensional and one-dimensional methods, we notice that 2D-PCA, 2D-iLPP, 2D-FDA, and 2D-SSDR have a comparable or even better performance than their one-dimensional counterparts. And, even in the cases where the result of a two-dimensional method is worse, the difference from the best rate is usually small and the method can still be useful, especially considering its advantages over vector-based approaches.

5.3.2 Smile Detection Task

In the smile detection task, the system should specify whether the subject in the image is smiling or not. The reduced FEI database has images of 200 subjects and each one of them has 2 images. In one image the subject is smiling whereas in the other image the subject has a neutral facial expression. Therefore, this is a binary classification task with 200 samples of each class.

Again, the experiments were performed with 1, 5, 10, 25, 50 and 100 labeled samples of each class, or, equivalently, 2, 10, 50, 100 and 200 labeled samples in total. This corresponds to 0.5%, 2.5%, 5%, 12.5%, 25% and 50% of the total samples, respectively.

The misclassification rates for the smile detection task are listed in Table 5.4. It shows some notable differences from the results we have obtained so far. In the previous experiments, the rates usually decreased as more labeled samples were given. However, in this experiment only SELF had monotonically decreasing rates for higher values of n. Specifically, almost all rates raised from $n = 50$ to $n = 100$. In addition, the results for matrix-based methods were particularly poor. When there were more labeled samples ($n >= 10$), the misclassification rates of two-dimensional methods were much higher than the best rates.

But there are similarities too. For $n = 1$, the best result is again from a matrix-based method (2D-iLPP). Interestingly, this method is unsupervised and not semi-supervised as we expected. Also, LFDA and SELF produced the best rates most of the time and DPCA had a good performance too. In fact, these three were the only techniques that delivered consistently useful results in this experiment. 2D-FDA and 2D-LFDA achieved best rates when $n = 5$, although they are still high in absolute terms (near 50%). Actually, misclassification rates higher than 50% appear with frequency in the results table. It was clear that the smile detection task was a difficult problem for most methods.

Figure 5.2 Original image from the FEI database (on the left) with its corresponding cropped version (on the right).

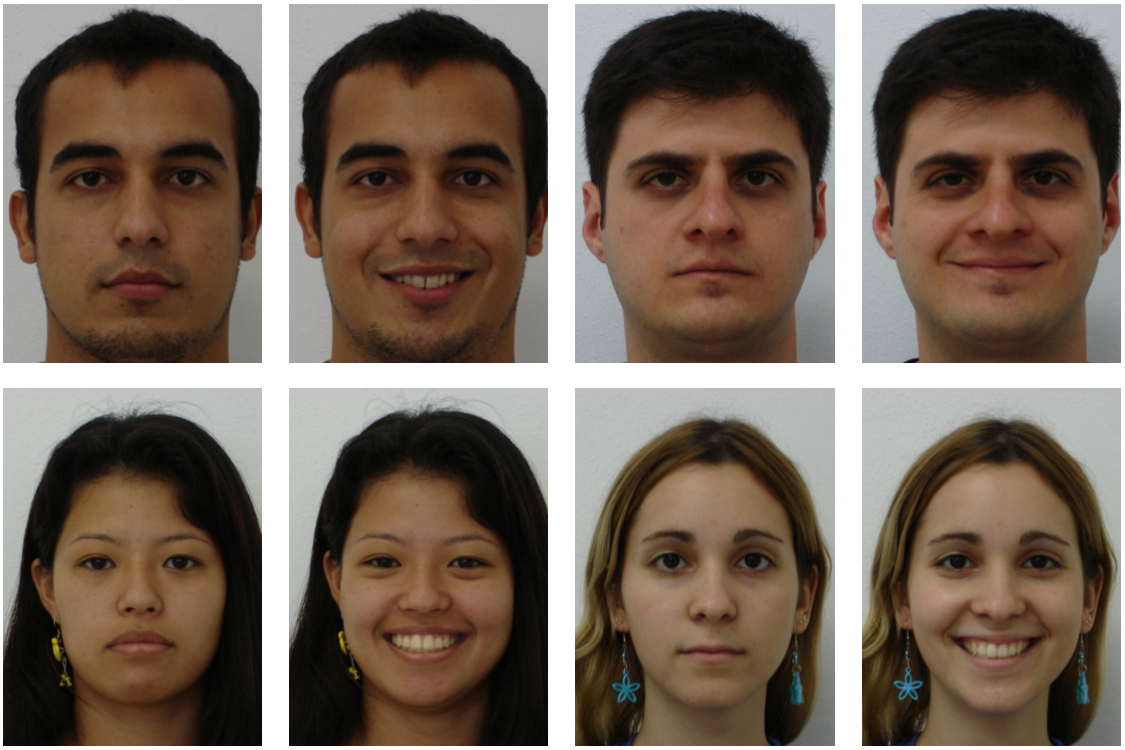


Figure 5.3 Example of images from the FEI database for four individuals. Each individual has two pictures, one with a neutral expression (left) and one smiling (right). The four images on the top depict male subjects and the four images in bottom depict female subjects.

METHOD	$n = 1$	$n = 5$	$n = 10$	$n = 25$	$n = 50$	$n = 100$
PCA	**22.07** (12.5)	**17.81** (5.35)	15.50 (4.15)	14.72 (3.58)	12.22 (2.10)	8.50 (2.06)
iLPP	**21.58** (9.48)	22.39 (12.1)	19.26 (6.13)	18.44 (5.85)	16.31 (1.66)	11.60 (2.33)
FDA	37.63 (14.4)	43.23 (5.73)	46.40 (5.91)	47.90 (4.25)	49.25 (4.53)	48.29 (4.45)
LFDA	**22.15** (12.4)	**16.09** (2.98)	**13.20** (3.21)	**11.29** (1.90)	8.86 (1.69)	**5.67** (1.54)
SSDR	**22.17** (12.6)	**17.84** (5.40)	15.60 (4.27)	14.80 (3.62)	12.37 (2.12)	8.58 (2.00)
DPCA	**22.15** (12.4)	**14.42** (2.64)	**12.41** (2.55)	**11.21** (2.18)	**7.57** (1.24)	**5.60** (1.57)
SELF	**22.17** (12.6)	**16.04** (2.92)	**13.18** (3.16)	**11.29** (1.90)	**8.72** (1.75)	**5.33** (1.76)
2D-PCA	**21.76** (12.0)	**18.05** (5.44)	**15.24** (4.47)	14.57 (3.24)	11.65 (1.63)	8.15 (1.93)
2D-iLPP	**26.77** (10.9)	22.50 (8.11)	19.02 (3.79)	17.67 (2.64)	14.94 (2.07)	9.24 (2.39)
2D-FDA	50.00 (0.00)	20.76 (5.99)	16.65 (4.25)	14.52 (2.62)	11.50 (2.42)	7.19 (1.95)
2D-LFDA	50.00 (0.00)	20.76 (5.99)	16.65 (4.25)	14.53 (2.62)	11.50 (2.42)	7.19 (1.95)
2D-SSDR	**22.29** (13.4)	**18.17** (5.57)	**15.31** (4.54)	14.45 (3.27)	11.67 (1.72)	8.07 (1.87)
2D-DPCA	**21.94** (11.4)	35.44 (7.90)	37.91 (3.33)	39.34 (2.44)	35.37 (2.13)	30.69 (1.59)
2D-SELF	**22.03** (11.9)	26.99 (7.40)	21.12 (6.14)	16.45 (3.94)	13.79 (2.05)	7.61 (1.55)

Table 5.3 Misclassification rates (%) for the Gender Detection task with the reduced FEI database. The heading $n = k$ indicates the k numbers of labeled samples used in the experiment. For each k, the table shows the averaged rates over the reduced dimensions (up to 320 PCs for vector methods and 65 PCs for matrix methods, except for FDA, which used only the first PC) along with their standard deviations in parenthesis. The best mean and the means with no significant statistical difference are highlighted in bold (*t-test* with 5% of significance). All the rates are averages over 12 repetitions with different dataset splits. The feature extraction was performed with the plain version of vector methods and with the weighted version of matrix methods.

METHOD	$n = 1$	$n = 5$	$n = 10$	$n = 25$	$n = 50$	$n = 100$
PCA	48.60 (1.31)	48.92 (1.63)	50.13 (1.55)	51.87 (1.91)	56.46 (1.81)	68.36 (1.19)
iLPP	49.46 (0.68)	49.57 (1.93)	50.44 (1.22)	53.54 (1.61)	58.04 (1.16)	68.36 (1.61)
FDA	50.29 (1.00)	48.80 (5.30)	47.92 (4.45)	49.81 (3.92)	51.78 (2.71)	48.04 (2.53)
LFDA	48.51 (1.43)	**42.80** (5.48)	**37.57** (5.26)	**20.10** (4.72)	**13.61** (1.64)	16.92 (2.18)
SSDR	48.55 (1.42)	48.66 (1.51)	50.03 (1.40)	51.89 (1.98)	56.50 (1.78)	68.34 (1.26)
DPCA	48.53 (1.43)	**44.86** (4.27)	**39.39** (5.69)	**23.84** (5.00)	16.28 (1.74)	19.50 (1.13)
SELF	48.55 (1.42)	**42.93** (5.29)	**37.51** (5.19)	**20.10** (4.74)	**13.14** (1.69)	**12.08** (1.72)
2D-PCA	48.67 (1.24)	49.38 (1.58)	50.55 (1.33)	52.29 (2.54)	57.71 (1.63)	68.40 (1.84)
2D-iLPP	**46.67** (1.80)	47.07 (2.21)	48.31 (2.08)	48.62 (1.76)	52.77 (2.12)	64.82 (2.02)
2D-FDA	50.00 (0.00)	**43.96** (5.17)	42.13 (4.16)	38.79 (4.37)	40.38 (4.01)	43.96 (4.30)
2D-LFDA	50.00 (0.00)	**43.97** (5.17)	42.13 (4.16)	38.79 (4.37)	40.39 (4.01)	43.99 (4.30)
2D-SSDR	48.54 (1.53)	49.26 (1.34)	50.40 (1.53)	52.23 (2.36)	57.58 (1.66)	68.58 (1.75)
2D-DPCA	48.77 (1.21)	46.92 (2.79)	46.85 (1.79)	48.16 (1.46)	50.86 (2.49)	54.90 (1.59)
2D-SELF	48.88 (1.09)	47.13 (3.17)	46.24 (2.87)	47.33 (2.10)	51.47 (2.39)	54.76 (2.51)

Table 5.4 Misclassification rates (%) for the Smile Detection task with the reduced FEI database. The heading $n = k$ indicates the k numbers of labeled samples used in the experiment. For each k, the table shows the averaged rates over the reduced dimensions (up to 320 PCs for vector methods and 65 PCs for matrix methods, except for FDA, which used only the first PC) along with their standard deviations in parenthesis. The best mean and the means with no significant statistical difference are highlighted in bold (*t-test* with 5% of significance). All the rates are averages over 12 repetitions with different dataset splits. The feature extraction was performed with the plain version of vector methods and with the weighted version of matrix methods.

5.4 Discussion

In this section, we discuss the general findings observed in the experiments. Although there were some minor variations in the results, in general the methods showed a consistent behavior through all the experiments. As expected, the systems performance was affected in a great extent by the number of available labels. We observed that the misclassification rates generally decreased when more labeled samples were used. The two notable exceptions were FDA and 2D-DPCA, which showed increasing rates in most situations. Interestingly, the influence of the labeled samples was not uniform. We noticed that adding more labeled samples seems to be more effective when the total number of labels is small. For example, in the last three experiments, adding 10 labeled samples (from $n = 5$ to $n = 10$) had a much greater impact in the classification accuracy than adding 100 labeled samples (from $n = 50$ to $n = 100$).

The vector-based methods LFDA, DPCA and SELF regularly reached the best performances in all experiments when $n >= 5$ (or $n >= 2$ in the face recognition experiment). But when $n = 1$, the best results were mostly from matrix-based methods. This finding suggests that the matrix-based methods are more useful when there are very few labeled samples. However, contrary to what we expected, the proposed new matrix-based methods (2D-LFDA, 2D-SSDR, 2D-DPCA, and 2D-SELF) did not achieved significantly better rates than the existing two-dimensional methods.

Still, when we compared the performance of the matrix-based methods with their vector-based counterparts, we observed that 2D-PCA, 2D-iLPP, 2D-FDA, and 2D-SSDR had better or at least equivalent performances than the corresponding vector methods. This observation is consistent with the findings of Yang et al. (2004); Chen et al. (2007); Xiong et al. (2005), that respectively reported better results for the two-dimensional versions of PCA, LPP and FDA. 2D-LFDA and 2D-SELF also obtained good rates, that in many cases were just slightly higher than the rates produced by the respective vector-based methods. Thus, 2D-SSDR, 2D-LFDA, and 2D-SELF can be seen as faster and more stable alternatives to SSDR, LFDA, and SELF, with all the other advantages that two-dimensional methods include.

On the other hand, 2D-DPCA obtained very poor results when compared to the other methods, at least with the parameters used in the experiments. We intend to investigate its behavior with other parameters in the future.

CHAPTER 6

Conclusions

In this work, we described seven existing vector-based methods for dimensionality reduction. Two of them operate in a fully unsupervised way (PCA and iLPP), two operate in a fully supervised way (FDA and LFDA), and three operate in a semi-supervised way (SSDR, DPCA, and SELF). We presented their original formulations and then analyzed these methods within a scatter vector-based framework. The framework describes all methods in a general form, highlighting their similarities and differences.

We also discussed three existing matrix-based methods: two unsupervised (2D-PCA and 2D-LPP) and one supervised (2D-FDA). We explained that matrix-based methods have many advantages over vector-based techniques for extracting features of two-dimensional data, such as images. For example, matrix-based methods can fully exploit the structural information present in the samples because they operate directly with their native matrix form. In contrast, vector-based methods require the samples to be transformed into vectors before processing them, changing their structure and discarding potentially useful information.

Another benefit of two-dimensional techniques is that they produce smaller scatter matrices. This fact has two important consequences. First, the computation of two-dimensional methods is much faster, what makes them a viable choice for on-line learning systems. Second, a reduced scatter-matrix alleviates the small sample size problem, what makes matrix-based methods more stable when the number of patterns available for training is small.

From the vector-based framework, we derived a scatter matrix-based framework and then showed how the previously discussed two-dimensional methods can be described in its terms. This new framework also enabled us to propose two-dimensional extensions for LFDA, SSDR, DPCA, and SELF. The new methods 2D-LFDA, 2D-SSDR, 2D-DPCA, and 2D-SELF apply the same ideas that were used to create the original techniques, but in a two-dimensional context.

To evaluate the methods performance in various scenarios, we carried out some experiments. All techniques were tested with two face databases, in four image classification tasks: face recognition and glasses detection tasks with the ORL database, and gender and smile detection tasks with the FEI database. In order to minimize a possible interference in the results caused by the choice of the learning method, we used the nearest neighbor classifier (1-NN), one of the simplest classifiers available.

We started by investigating whether the weighting scheme proposed by Sugiyama et al. (2010) is useful or not to improve the classifying accuracy. We found that, for the vector-based methods, the misclassification rates generated by the weighted versions

were actually worse than the rates generated by the non-weighted versions. In contrast, for the matrix-based methods, the weighted versions generally outperformed the non-weighted versions. For this reason, we applied the weighting scheme only to matrix-based methods.

Because the FEI database have relatively high-resolution images, we had to shrink its samples in order to compute the vector-based methods in a reasonable amount of time. On the other hand, matrix-based methods do not have this problem. They can easily deal with samples of large sizes, so it was possible to compare the performance of two-dimensional methods with both the resized and with the full-sized images. We expected that the systems that used the full-sized images would generate lower misclassification rates, because these images have more pixels and supposedly have more information. But the contrary happened. The lowest rates were achieved when the resized images were used. This led us to believe that most information in the full-sized images is redundant and that the resizing process itself acts as a form of feature extraction.

We also analyzed how the number of labeled samples affects the results of each method. For this purpose, we performed experiments with a varying number of labeled samples. As expected, the methods performed better when more labeled samples were used, except for FDA and 2D-DPCA. Furthermore, we observed that the influence of the discriminative information was not uniform. Adding more labeled samples had a greater effect in lowering the misclassification rates when there were few labels, but this impact diminished as the number of labeled samples increased.

Finally, we compared the performance of vector-based and matrix-based methods. Because they use the images directly and therefore can take advantage of more information, we expected that the matrix-based methods produced the absolute best rates. However, this happened only when there were one labeled sample of each class. When more labels were given, the majority of best rates were produced by vector-based methods, especially LFDA, DPCA and SELF, that consistently achieved most of the better rates through the experiments. This observation suggests that matrix-based methods are more useful when there are very few labeled samples available.

However, when we compared the results of two-dimensional methods with the results of their one-dimensional counterparts, we noticed that 2D-PCA, 2D-iLPP, 2D-FDA and 2D-SSDR achieved better or at least equivalent misclassification rates than the corresponding vector-based methods. Also, 2D-LFDA and 2D-SELF produced good rates, that sometimes were close to the best results. Therefore, 2D-SSDR, 2D-LFDA, and 2D-SELF can be seen as viable alternatives to SSDR, LFDA, and SELF, especially considering the advantages of two-dimensional methods, such as faster computation and more stability.

6.1 Contributions

The contributions of this work can be summarized as follows:

- Discussion of two-dimensional data analysis and semi-supervised learning, summarizing their motivations, applications and advantages;
- Description of the semi-supervised methods SSDR and DPCA within the existing vector-based framework;
- Development of a new matrix-based framework, capable of generally describing scatter-based two-dimensional methods for dimensionality reduction. The framework simplifies the comparison of distinct techniques, making it easy to analyze their similarities and differences. Furthermore, the framework allows the creation of new matrix-based techniques from existing vector-based techniques with minor effort;
- Analysis of practical aspects for efficient implementation of the vector- and matrix-based frameworks;
- Presentation of four novel matrix-based dimensionality reduction methods, one supervised (2D-LFDA) and three semi-supervised (2D-SSDR, 2D-DPCA, and 2D-SELF). The new methods are based on the two-dimensional framework;
- Evaluation of the performance of all vector- and matrix-based methods discussed in this work. Experiments were performed within different tasks and image databases and in varying conditions regarding the weighting scheme and numbers of labeled samples and principal components;
- Distribution of the complete source code used in the experiments in the form of open source software;
- Combination of two relevant and distinct areas (tensor-based analysis and semi-supervised learning), creating new dimensionality reduction methods capable of dealing two current problems: high-dimensional and partially labeled data.

6.2 Future Works

Future works include the possibility of incorporating other existing one-dimensional methods—semi-supervised or not—to the vector-based framework. This would make possible to create two-dimensional versions of these methods with minimal effort. Also, we plan to test the methods that require parameters more extensively. We believe that the performance of 2D-SSDR, 2D-DPCA, and 2D-SELF can be improved by carefully choosing their input parameters.

In addition, it is known that the performance of any given method can vary considerably with different databases. For instance, Chapelle et al. (2006) conducted systematic experiments for comparing various semi-supervised methods. The results showed that whereas a method can perform very well for a particular database, it can also perform very poorly for other kinds of data. For this reason, in future works we intend

to test the methods with more image databases and compare how the performance of each one is affected. We also consider to use another types of two-dimensional data, such as hyper-spectral images.

Finally, an important criticism for the original matrix-based methods is that they need much more coefficients than vector-based methods to represent the data. Moreover, they operate only in a single, arbitrary row direction. To address these problems, $(2D)^2$-PCA (Zhang and Zhou, 2005) and $(2D)^2$-FLD (Nagabhushan et al., 2006) were proposed. They simultaneously consider the row and column directions, what makes them to require less coefficients for data representation. In following works, we aim to investigate the possibility of describing $(2D)^2$ methods within a general framework and porting other existing matrix-based methods to it.

APPENDIX A

Derivation of Pairwise Equations

A.1 Vector-based methods

A.1.1 Principal Component Analysis (PCA)

The pairwise form of the total scatter matrix $\mathbf{S}_t$ is obtained as follows. Let $\mathbf{x}_i \in \mathbb{R}^m$ be the m-dimensional column vectors that represents the n samples and let $\overline{\mathbf{x}}$ be the mean of all samples (as defined in Section 2.2). Then

$$\mathbf{S}_t = \sum_{i=1}^{n} (\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})^\top \tag{A.1a}$$

$$= \sum_{i=1}^{n} (\mathbf{x}_i \mathbf{x}_i^\top + \overline{\mathbf{x}}\overline{\mathbf{x}}^\top - \mathbf{x}_i \overline{\mathbf{x}}^\top - \overline{\mathbf{x}} \mathbf{x}_i^\top) \tag{A.1b}$$

$$= \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^\top + n\overline{\mathbf{x}}\overline{\mathbf{x}}^\top - \sum_{i=1}^{n} \mathbf{x}_i \overline{\mathbf{x}}^\top - \sum_{i=1}^{n} \overline{\mathbf{x}} \mathbf{x}_i^\top \tag{A.1c}$$

$$= \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^\top + \frac{n}{n^2} \sum_{i=1}^{n} \mathbf{x}_i \sum_{i=1}^{n} \mathbf{x}_i^\top - \frac{1}{n} \sum_{i=1}^{n} \left(\mathbf{x}_i \sum_{j=1}^{n} \mathbf{x}_j^\top \right) - \frac{1}{n} \sum_{i=1}^{n} \left[\left(\sum_{j=1}^{n} \mathbf{x}_j \right) \mathbf{x}_i^\top \right] \tag{A.1d}$$

$$= \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^\top + \frac{1}{n} \sum_{i,j=1}^{n} \mathbf{x}_i \mathbf{x}_j^\top - \frac{1}{n} \sum_{i,j=1}^{n} \mathbf{x}_i \mathbf{x}_j^\top - \frac{1}{n} \sum_{i,j=1}^{n} \mathbf{x}_i \mathbf{x}_j^\top \tag{A.1e}$$

$$= \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^\top - \frac{1}{n} \sum_{i,j=1}^{n} \mathbf{x}_i \mathbf{x}_j^\top \tag{A.1f}$$

$$= \sum_{i=1}^{n} \left[\left(\sum_{j=1}^{n} \frac{1}{n} \right) \mathbf{x}_i \mathbf{x}_i^\top \right] - \sum_{i,j=1}^{n} \frac{1}{n} \mathbf{x}_i \mathbf{x}_j^\top \tag{A.1g}$$

$$= \sum_{i,j=1}^{n} \frac{1}{n} \mathbf{x}_i \mathbf{x}_i^\top - \sum_{i,j=1}^{n} \frac{1}{n} \mathbf{x}_i \mathbf{x}_j^\top \tag{A.1h}$$

$$= \frac{1}{2} \sum_{i,j=1}^{n} \frac{1}{n} (\mathbf{x}_i \mathbf{x}_i^\top + \mathbf{x}_i \mathbf{x}_j^\top - \mathbf{x}_j \mathbf{x}_i^\top + \mathbf{x}_j \mathbf{x}_j^\top) \tag{A.1i}$$

$$= \frac{1}{2} \sum_{i,j=1}^{n} W_{i,j}^{(t)} (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top, \text{ with } W_{i,j}^{(t)} = 1/n. \tag{A.1j}$$

A.1.2 Fisher Discriminant Analysis (FDA)

The pairwise form of the within-scatter matrix $\mathbf{S}_w$ is derived as follows. Let $\mathbf{x}_i \in \mathbb{R}^m$ be the m-dimensional column vectors that represents the n' labeled samples , $\overline{\mathbf{x}_l}$ be the mean of the n'_l samples with class l and c be number of classes. Then

$$\mathbf{S}_w = \sum_{l=1}^{c} \sum_{i:y_i=l} (\mathbf{x}_i - \overline{\mathbf{x}}_l)(\mathbf{x}_i - \overline{\mathbf{x}}_l)^\top \tag{A.2a}$$

$$= \sum_{l=1}^{c} \sum_{i:y_i=l} \left(\mathbf{x}_i - \frac{1}{n'_l} \sum_{j:y_j=l} \mathbf{x}_j\right) \left(\mathbf{x}_i - \frac{1}{n'_l} \sum_{j:y_j=l} \mathbf{x}_j\right)^\top \tag{A.2b}$$

$$= \sum_{l=1}^{c} \sum_{i:y_i=l} \left(\mathbf{x}_i\mathbf{x}_i^\top - \mathbf{x}_i \frac{1}{n'_l} \sum_{j:y_j=l} \mathbf{x}_j^\top - \frac{1}{n'_l} \sum_{j:y_j=l} \mathbf{x}_j\mathbf{x}_i^\top + \frac{1}{n'^2_l} \sum_{j:y_j=l} \mathbf{x}_j \sum_{j:y_j=l} \mathbf{x}_j^\top\right) \tag{A.2c}$$

$$= \sum_{l=1}^{c} \sum_{i:y_i=l} \mathbf{x}_i\mathbf{x}_i^\top + \sum_{l=1}^{c} \frac{1}{n_l} \sum_{i:y_i=l} \left(- \sum_{j:y_j=l} \mathbf{x}_i\mathbf{x}_j^\top - \sum_{j:y_j=l} \mathbf{x}_j\mathbf{x}_i^\top + \frac{1}{n'_l} \sum_{j,k:y_j=y_k=l} \mathbf{x}_j\mathbf{x}_k^\top\right) \tag{A.2d}$$

$$= \sum_{i=1}^{n} \mathbf{x}_i\mathbf{x}_i^\top + \sum_{l=1}^{c} \frac{1}{n'_l} \left(- \sum_{i,j:y_i=y_j=l} \mathbf{x}_i\mathbf{x}_j^\top - \sum_{i,j:y_i=y_j=l} \mathbf{x}_j\mathbf{x}_i^\top + n'_l \frac{1}{n'_l} \sum_{i,j:y_i=y_j=l} \mathbf{x}_i\mathbf{x}_j^\top\right) \tag{A.2e}$$

$$= \sum_{i=1}^{n} \mathbf{x}_i\mathbf{x}_i^\top - \sum_{l=1}^{c} \left(\frac{1}{n'_l} \sum_{i,j:y_i=y_j=l} \mathbf{x}_j\mathbf{x}_i^\top\right) \tag{A.2f}$$

$$= \sum_{i=1}^{n} \left(\sum_{j=1}^{n} W_{i,j}^{(w)}\right) \mathbf{x}_i\mathbf{x}_i^\top - \sum_{i,j=1}^{n} W_{i,j}^{(w)} \mathbf{x}_j\mathbf{x}_i^\top \tag{A.2g}$$

$$= \sum_{i,j=1}^{n} W_{i,j}^{(w)} \frac{1}{2} (\mathbf{x}_i\mathbf{x}_i^\top - \mathbf{x}_i\mathbf{x}_j^\top - \mathbf{x}_j\mathbf{x}_i^\top + \mathbf{x}_j\mathbf{x}_j^\top) \tag{A.2h}$$

$$= \frac{1}{2} \sum_{i,j=1}^{n} W_{i,j}^{(w)} (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top, \text{ with } W_{i,j}^{(w)} = \begin{cases} 1/n'_{y_i} & \text{if } y_i = y_j \\ 0 & \text{if } y_i \neq y_j \end{cases}, \tag{A.2i}$$

where y_i is the class of the sample $\mathbf{x}_i$ and n_{y_i} is the number of samples with class y_i.

It can be proved that the total scatter matrix $\mathbf{S}_t$ is the sum of the between- and within-scatter matrices $\mathbf{S}_b$ and $\mathbf{S}_w$ (both defined in Section 2.3):

$$\mathbf{S}_t = \sum_{i=1}^{n} (\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})^\top \tag{A.3a}$$

$$= \sum_{l=1}^{c} \sum_{i:y_i=l} (\mathbf{x}_i - \bar{\mathbf{x}}_l + \bar{\mathbf{x}}_l - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}}_l + \bar{\mathbf{x}}_l - \bar{\mathbf{x}})^\top \tag{A.3b}$$

$$= \sum_{l=1}^{c} \sum_{i:y_i=l} (\mathbf{x}_i - \bar{\mathbf{x}}_l)(\mathbf{x}_i - \bar{\mathbf{x}}_l)^\top + \sum_{l=1}^{c} \sum_{i:y_i=l} (\bar{\mathbf{x}}_l - \bar{\mathbf{x}})(\bar{\mathbf{x}}_l - \bar{\mathbf{x}})^\top \tag{A.3c}$$

$$= \mathbf{S}_w + \sum_{l=1}^{c} n'_l (\bar{\mathbf{x}}_l - \bar{\mathbf{x}})(\bar{\mathbf{x}}_l - \bar{\mathbf{x}})^\top \tag{A.3d}$$

$$= \mathbf{S}_w + \mathbf{S}_b. \tag{A.3e}$$

We take advantage of this fact to derive the pairwise form of the between-scatter matrix $\mathbf{S}_b$ as follows

$$\mathbf{S}_b = \mathbf{S}_t - \mathbf{S}_w \tag{A.4a}$$

$$= \frac{1}{2} \sum_{i,j=1}^{n} W_{i,j}^{(t)} (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top - \frac{1}{2} \sum_{i,j=1}^{n} W_{i,j}^{(w)} (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top \tag{A.4b}$$

$$= \frac{1}{2} \sum_{i,j=1}^{n} (W_{i,j}^{(t)} - W_{i,j}^{(w)})(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top \tag{A.4c}$$

$$= \frac{1}{2} \sum_{i,j=1}^{n} W_{i,j}^{(b)} (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top, \text{ with } W_{i,j}^{(b)} = \begin{cases} 1/n' - 1/n'_{y_i} & \text{if } y_i = y_j \\ 1/n' & \text{if } y_i \neq y_j \end{cases}. \tag{A.4d}$$

A.1.3 Locality Preserving Projections (LPP)

The pairwise form of the local scatter matrix $\mathbf{S}_l$ is obtained as follows. Let $\mathbf{x}_i \in \mathbb{R}^m$ be the m-dimensional column vectors that represents the n samples and $\mathbf{X}$ be the matrix of all samples (as defined in Section 2.1). Then

$$\mathbf{S}_l = \mathbf{X}\mathbf{L}\mathbf{X}^\top \tag{A.5a}$$

$$= \mathbf{X}(\mathbf{D} - \mathbf{A})\mathbf{X}^\top \tag{A.5b}$$

$$= \sum_{i=1}^{n} D_{i,i} \mathbf{x}_i \mathbf{x}_i^\top - \sum_{i,j=1}^{n} A_{i,j} \mathbf{x}_i \mathbf{x}_j^\top \tag{A.5c}$$

$$= \sum_{i=1}^{n} \left(\sum_{j=1}^{n} A_{i,j} \right) \mathbf{x}_i \mathbf{x}_i^\top - \sum_{i,j=1}^{n} A_{i,j} \mathbf{x}_i \mathbf{x}_j^\top \tag{A.5d}$$

$$= \sum_{i,j=1}^{n} A_{i,j} \mathbf{x}_i \mathbf{x}_i^\top - \sum_{i,j=1}^{n} A_{i,j} \mathbf{x}_i \mathbf{x}_j^\top \tag{A.5e}$$

$$= \sum_{i,j=1}^{n} A_{i,j} (\mathbf{x}_i \mathbf{x}_i^\top - \mathbf{x}_i \mathbf{x}_j^\top) \tag{A.5f}$$

$$= \frac{1}{2}\sum_{i,j=1}^{n} A_{i,j}(\mathbf{x}_i\mathbf{x}_i^\top - \mathbf{x}_i\mathbf{x}_j^\top - \mathbf{x}_j\mathbf{x}_i^\top + \mathbf{x}_j\mathbf{x}_j^\top) \tag{A.5g}$$

$$= \frac{1}{2}\sum_{i,j=1}^{n} W_{i,j}^{(l)}(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top, \text{ with } W_{i,j}^{(l)} = A_{i,j}, \tag{A.5h}$$

where $\mathbf{A}$ is the affinity matrix (as defined in Section 2.4) and $\mathbf{D}$ is the diagonal matrix whose entries are column sums of $\mathbf{A}$ such that $D_{i,i} = \sum_j A_{i,j}$.

The iterative version of the normalization matrix $\mathbf{S}_n$ is straightforward:

$$\mathbf{S}_n = \mathbf{X}\mathbf{D}\mathbf{X}^\top \tag{A.6a}$$

$$= (D_{1,1}\mathbf{x}_1\mathbf{x}_1^\top + D_{2,2}\mathbf{x}_2\mathbf{x}_2^\top + \cdots + D_{n,n}\mathbf{x}_n\mathbf{x}_n^\top) \tag{A.6b}$$

$$= \sum_{i=1}^{n} D_{i,i}\mathbf{x}_i\mathbf{x}_i^\top. \tag{A.6c}$$

A.1.4 Local Fisher Discriminant Analysis (LFDA)

The scatter matrices of LFDA were already defined in the pairwise form.

A.1.5 Semi-supervised Dimensionality Reduction (SSDR)

The pairwise form of the SSDR scatter matrix $\mathbf{S}_{ssdr}$ is derived in the same way that the local scatter matrix $\mathbf{S}_{ssdr}$ of LPP, just replacing $A_{i,j}$ by $S_{i,j}$.

A.1.6 Discriminant Principal Component Analysis (DPCA)

The scatter matrix of DPCA $\mathbf{S}_d$ is defined by the equation (2.26) in function of the generalized between-scatter matrix $\mathbf{S}'_b$, the generalized within-scatter matrix $\mathbf{S}'_w$, and the total scatter matrix $\mathbf{S}_t$. Then, the pairwise form of $\mathbf{S}_d$ can be easily obtained as

$$\mathbf{S}_d = \mathbf{S}'_b - \eta\mathbf{S}'_w + \lambda\mathbf{S}_t \tag{A.7a}$$

$$= \frac{1}{2}\sum_{i,j=1}^{n} (W_{i,j}^{(b')} - \eta W_{i,j}^{(w')} + \lambda W_{i,j}^{(t)})(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top \tag{A.7b}$$

$$= \frac{1}{2}\sum_{i,j=1}^{n} W_{i,j}^{(d)}(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top, \quad W_{i,j}^{(d)} = \begin{cases} \lambda/n^2 + 2/n_c & \text{if } (\mathbf{x}_i, \mathbf{x}_j) \in \mathbf{C}^{(l)} \\ \lambda/n^2 - 2\eta/n_m & \text{if } (\mathbf{x}_i, \mathbf{x}_j) \in \mathbf{M}^{(l)} \\ \lambda/n^2 & \text{otherwise} \end{cases}, \tag{A.7c}$$

where $\mathbf{C}^{(l)}$ and $\mathbf{M}^{(l)}$ are, respectively, the cannot-link and must-link constraints sets, n_c and n_m are, respectively, the numbers of cannot-link and must-link constraints and n is the total number of samples.

A.1.7 Semi-supervised Local Fisher Discriminant Analysis (SELF)

The scatter matrices of SELF were defined in function of LFDA and PCA. The original definition of LFDA was already in the pairwise form and the pairwise form of the total scatter matrix of PCA was shown previously in this appendix.

A.2 Matrix-based methods

A.2.1 Two-dimensional Principal Component Analysis (2D-PCA)

The pairwise form of the total scatter matrix $\tilde{\mathbf{S}}_t$ is obtained as follows. Let $\mathbf{M}_i \in \mathbb{R}^{l \times m}$ be $l \times m$ matrices that represent the n samples and let $\overline{\mathbf{M}}$ be the mean of all samples (as defined in Section 3.2). Then

$$\tilde{\mathbf{S}}_t = \sum_{i=1}^{n} (\mathbf{M}_i - \overline{\mathbf{M}})^\top (\mathbf{M}_i - \overline{\mathbf{M}}) \tag{A.8a}$$

$$= \sum_{i=1}^{n} (\mathbf{M}_i^\top \mathbf{M}_i - \mathbf{M}_i^\top \overline{\mathbf{M}} - \overline{\mathbf{M}}^\top \mathbf{M}_i + \overline{\mathbf{M}}^\top \overline{\mathbf{M}}) \tag{A.8b}$$

$$= \sum_{i=1}^{n} \mathbf{M}_i^\top \mathbf{M}_i - \sum_{i=1}^{n} \mathbf{M}_i^\top \overline{\mathbf{M}} - \sum_{i=1}^{n} \overline{\mathbf{M}}^\top \mathbf{M}_i + n\overline{\mathbf{M}}^\top \overline{\mathbf{M}} \tag{A.8c}$$

$$= \sum_{i=1}^{n} \mathbf{M}_i^\top \mathbf{M}_i - \sum_{i=1}^{n} \left(\mathbf{M}_i^\top \frac{1}{n} \sum_{j=1}^{n} \mathbf{M}_j \right) - \sum_{i=1}^{n} \left(\frac{1}{n} \sum_{j=1}^{n} \mathbf{M}_j^\top \mathbf{M}_i \right) + n \frac{1}{n} \sum_{i=1}^{n} \mathbf{M}_i^\top \frac{1}{n} \sum_{i=1}^{n} \mathbf{M}_i \tag{A.8d}$$

$$= \sum_{i=1}^{n} \mathbf{M}_i^\top \mathbf{M}_i - \frac{1}{n} \sum_{i=1}^{n} \left(\mathbf{M}_i^\top \sum_{j=1}^{n} \mathbf{M}_j \right) - \frac{1}{n} \sum_{i=1}^{n} \mathbf{M}_i^\top \sum_{i=1}^{n} \mathbf{M}_i + \frac{1}{n} \sum_{i=1}^{n} \mathbf{M}_i^\top \sum_{i=1}^{n} \mathbf{M}_i \tag{A.8e}$$

$$= \sum_{i=1}^{n} \mathbf{M}_i^\top \mathbf{M}_i - \frac{1}{n} \sum_{i,j=1}^{n} \mathbf{M}_i^\top \mathbf{M}_j \tag{A.8f}$$

$$= \sum_{i=1}^{n} \left(\sum_{j=1}^{n} \frac{1}{n} \right) \mathbf{M}_i^\top \mathbf{M}_i - \frac{1}{n} \sum_{i,j=1}^{n} \mathbf{M}_i^\top \mathbf{M}_j \tag{A.8g}$$

$$= \frac{1}{n} \sum_{i,j=1}^{n} \mathbf{M}_i^\top \mathbf{M}_i - \frac{1}{n} \sum_{i,j=1}^{n} \mathbf{M}_i^\top \mathbf{M}_j \tag{A.8h}$$

$$= \frac{1}{n} \sum_{i,j=1}^{n} \left(\mathbf{M}_i^\top \mathbf{M}_i - \mathbf{M}_i^\top \mathbf{M}_j \right) \tag{A.8i}$$

$$= \frac{1}{n} \sum_{i,j=1}^{n} \frac{1}{2} \left(\mathbf{M}_i^\top \mathbf{M}_i - \mathbf{M}_i^\top \mathbf{M}_j - \mathbf{M}_j^\top \mathbf{M}_i + \mathbf{M}_j^\top \mathbf{M}_j \right) \tag{A.8j}$$

$$= \frac{1}{2} \sum_{i,j=1}^{n} \tilde{W}_{i,j}^{(t)} (\mathbf{M}_i - \mathbf{M}_j)^\top (\mathbf{M}_i - \mathbf{M}_j), \text{ with } \tilde{W}_{i,j}^{(t)} = 1/n. \tag{A.8k}$$

A.2.2 Two-dimensional Fisher Discriminant Analysis (2D-FDA)

The pairwise form of the within-scatter matrix $\tilde{\mathbf{S}}_w$ is derived as follows. Let $\mathbf{M}_i \in \mathbb{R}^{l \times m}$ be $l \times m$ matrices that represent the n' labeled samples, $\overline{\mathbf{M}_l}$ be the mean of the n'_l samples with class l and c be number of classes. Then

$$\tilde{\mathbf{S}}_w = \sum_{l=1}^{c} \sum_{i:y_i=l} (\mathbf{M}_i - \overline{\mathbf{M}_l})^\top (\mathbf{M}_i - \overline{\mathbf{M}_l}) \tag{A.9a}$$

$$= \sum_{l=1}^{c} \sum_{i:y_i=l} \left(\mathbf{M}_i - \frac{1}{n'_l} \sum_{j:y_j=l} \mathbf{M}_j \right)^\top \left(\mathbf{M}_i - \frac{1}{n'_l} \sum_{j:y_j=l} \mathbf{M}_j \right) \tag{A.9b}$$

$$= \sum_{l=1}^{c} \sum_{i:y_i=l} \left(\mathbf{M}_i^\top \mathbf{M}_i - \mathbf{M}_i^\top \frac{1}{n'_l} \sum_{j:y_j=l} \mathbf{M}_j - \frac{1}{n'_l} \sum_{j:y_j=l} \mathbf{M}_j^\top \mathbf{M}_i + \frac{1}{n'^2_l} \sum_{j:y_j=l} \mathbf{M}_j^\top \sum_{j:y_j=l} \mathbf{M}_j \right) \tag{A.9c}$$

$$= \sum_{l=1}^{c} \sum_{i:y_i=l} \mathbf{M}_i^\top \mathbf{M}_i + \sum_{l=1}^{c} \frac{1}{n_l} \sum_{i:y_i=l} \left(- \sum_{j:y_j=l} \mathbf{M}_i^\top \mathbf{M}_j - \sum_{j:y_j=l} \mathbf{M}_j^\top \mathbf{M}_i + \frac{1}{n'_l} \sum_{j,k:y_j=y_k=l} \mathbf{M}_j^\top \mathbf{M}_k \right) \tag{A.9d}$$

$$= \sum_{i=1}^{n} \mathbf{M}_i^\top \mathbf{M}_i + \sum_{l=1}^{c} \frac{1}{n'_l} \left(- \sum_{i,j:y_i=y_j=l} \mathbf{M}_i^\top \mathbf{M}_j - \sum_{i,j:y_i=y_j=l} \mathbf{M}_j^\top \mathbf{M}_i + n'_l \frac{1}{n'_l} \sum_{i,j:y_i=y_j=l} \mathbf{M}_i^\top \mathbf{M}_j \right) \tag{A.9e}$$

$$= \sum_{i=1}^{n} \mathbf{M}_i^\top \mathbf{M}_i - \sum_{l=1}^{c} \left(\frac{1}{n'_l} \sum_{i,j:y_i=y_j=l} \mathbf{M}_j^\top \mathbf{M}_i \right) \tag{A.9f}$$

$$= \sum_{i=1}^{n} \left(\sum_{j=1}^{n} \tilde{W}_{i,j}^{(w)} \right) \mathbf{M}_i^\top \mathbf{M}_i - \sum_{i,j=1}^{n} \tilde{W}_{i,j}^{(w)} \mathbf{M}_j^\top \mathbf{M}_i \tag{A.9g}$$

$$= \sum_{i,j=1}^{n} \tilde{W}_{i,j}^{(w)} \frac{1}{2} (\mathbf{M}_i^\top \mathbf{M}_i - \mathbf{M}_i^\top \mathbf{M}_j - \mathbf{M}_j^\top \mathbf{M}_i + \mathbf{M}_j^\top \mathbf{M}_j) \tag{A.9h}$$

$$= \frac{1}{2} \sum_{i,j=1}^{n} \tilde{W}_{i,j}^{(w)} (\mathbf{M}_i - \mathbf{M}_j)^\top (\mathbf{M}_i - \mathbf{M}_j), \text{ with } \tilde{W}_{i,j}^{(w)} = \begin{cases} 1/n'_{y_i} & \text{if } y_i = y_j \\ 0 & \text{if } y_i \neq y_j \end{cases}, \tag{A.9i}$$

where y_i is the class of the sample $\mathbf{M}_i$ and n_{y_i} is the number of samples with class y_i.

Like the vector total scatter matrix $\mathbf{S}_t$, the total scatter matrix $\tilde{\mathbf{S}}_t$ is the sum of the within-scatter and between-scatter matrices $\tilde{\mathbf{S}}_w$ and $\tilde{\mathbf{S}}_b$. We use this fact to obtain the pairwise form of the matrix $\tilde{\mathbf{S}}_b$ as follows

$$\begin{aligned}
\tilde{\mathbf{S}}_b &= \tilde{\mathbf{S}}_t - \tilde{\mathbf{S}}_w && \text{(A.10a)}\\
&= \frac{1}{2}\sum_{i,j=1}^{n} \tilde{W}_{i,j}^{(t)} (\mathbf{M}_i - \mathbf{M}_j)^\top (\mathbf{M}_i - \mathbf{M}_j) - \frac{1}{2}\sum_{i,j=1}^{n} \tilde{W}_{i,j}^{(w)} (\mathbf{M}_i - \mathbf{M}_j)^\top (\mathbf{M}_i - \mathbf{M}_j) && \text{(A.10b)}\\
&= \frac{1}{2}\sum_{i,j=1}^{n} (\tilde{W}_{i,j}^{(t)} - \tilde{W}_{i,j}^{(w)}) (\mathbf{M}_i - \mathbf{M}_j)^\top (\mathbf{M}_i - \mathbf{M}_j) && \text{(A.10c)}\\
&= \frac{1}{2}\sum_{i,j=1}^{n} \tilde{W}_{i,j}^{(b)} (\mathbf{M}_i - \mathbf{M}_j)^\top (\mathbf{M}_i - \mathbf{M}_j), \text{ with } \tilde{W}_{i,j}^{(b)} = \begin{cases} 1/n' - 1/n'_{y_i} & \text{if } y_i = y_j \\ 1/n' & \text{if } y_i \neq y_j \end{cases}. && \text{(A.10d)}
\end{aligned}$$

A.2.3 Two-dimensional Locality Preserving Projections (2D-LPP)

The pairwise form of the local scatter matrix $\tilde{\mathbf{S}}_l$ is obtained as follows. Let $\mathbf{M}_i \in \mathbb{R}^{l \times m}$ be $l \times m$ matrices that represent the n samples, $\tilde{\mathbf{X}}$ be the matrix of all samples (as defined in Section 3.1) and $\mathbf{I}_l$ be the identity matrix in $\mathbb{R}^{l \times l}$. Then

$$\begin{aligned}
\tilde{\mathbf{S}}_l &= \tilde{\mathbf{X}}^\top (\tilde{\mathbf{L}} \otimes \mathbf{I}_l) \tilde{\mathbf{X}} && \text{(A.11a)}\\
&= \tilde{\mathbf{X}}^\top \left[(\tilde{\mathbf{D}} - \tilde{\mathbf{W}}^{(l)}) \otimes \mathbf{I}_l \right] \tilde{\mathbf{X}} && \text{(A.11b)}\\
&= \sum_{i,j=1}^{n} \mathbf{M}_i^\top (\tilde{W}_{i,j}^{(l)} \mathbf{I}_l) \mathbf{M}_i - \sum_{i,j=1}^{n} \mathbf{M}_i^\top (\tilde{W}_{i,j}^{(l)} \mathbf{I}_l) \mathbf{M}_j && \text{(A.11c)}\\
&= \sum_{i,j=1}^{n} \left(\tilde{W}_{i,j}^{(l)} \mathbf{M}_i^\top \mathbf{M}_i - \tilde{W}_{i,j}^{(l)} \mathbf{M}_i^\top \mathbf{M}_j \right) && \text{(A.11d)}\\
&= \sum_{i,j=1}^{n} \tilde{W}_{i,j}^{(l)} \left(\mathbf{M}_i^\top \mathbf{M}_i - \mathbf{M}_i^\top \mathbf{M}_j \right) && \text{(A.11e)}\\
&= \sum_{i,j=1}^{n} \tilde{W}_{i,j}^{(l)} \frac{1}{2} \left(\mathbf{M}_i^\top \mathbf{M}_i - \mathbf{M}_i^\top \mathbf{M}_j - \mathbf{M}_j^\top \mathbf{M}_i + \mathbf{M}_j^\top \mathbf{M}_j \right) && \text{(A.11f)}\\
&= \frac{1}{2}\sum_{i,j=1}^{n} \tilde{W}_{i,j}^{(l)} (\mathbf{M}_i - \mathbf{M}_j)^\top (\mathbf{M}_i - \mathbf{M}_j), \text{ with } \tilde{W}_{i,j}^{(l)} = \tilde{A}_{i,j}, && \text{(A.11g)}
\end{aligned}$$

where $\otimes$ denotes the Kronecker product and $\tilde{\mathbf{A}}$ is the affinity matrix (as defined in Section 3.4) and $\tilde{\mathbf{D}}$ is the diagonal matrix whose entries are column sums of $\tilde{\mathbf{A}}$ such that $\tilde{D}_{i,i} = \sum_j \tilde{A}_{i,j}$.

The iterative version of the image normalization matrix $\tilde{\mathbf{S}}_n$ comes directly from the definition of the Kronecker product, such as

$$\tilde{\mathbf{S}}_n = \tilde{\mathbf{X}}^\top (\tilde{\mathbf{D}} \otimes \mathbf{I}_l) \tilde{\mathbf{X}} \tag{A.12a}$$

$$= \sum_{i=1}^{n} \tilde{D}_{i,i} \mathbf{M}_i^\top \mathbf{M}_i. \tag{A.12b}$$

References

Agrawala, A. (1970). Learning with a probabilistic teacher. *IEEE Transactions on Information Theory*, 16(4):373–379. 1.3

Alpaydin, E. (2010). *Introduction to Machine Learning*. Adaptive Computation and Machine Learning. The MIT Press, Cambridge, Massachusetts, USA, 2nd edition. 1.1

Bar-Hillel, A., Hertz, T., Shental, N., and Weinshall, D. (2005). Learning a mahalanobis metric from equivalence constraints. *Journal of Machine Learning Research*, 6(6):937–965. 1.3

Belkin, M. and Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396. 1.1, 2.9

Bishop, C. (2006). *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, New York, USA. 1.1

Blum, A. and Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory*, pages 92–100, New York, USA. ACM. 1.3

Cavalcanti, G. D., Ren, T. I., and Pereira, J. F. (2013). Weighted Modular Image Principal Component Analysis for face recognition. *Expert Systems with Applications*, 40(12):4971–4977. 1.2

Chapelle, O., Schölkopf, B., and Zien, A. (2006). *Semi-Supervised Learning*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, Massachusetts, USA. 1.3, 3, 1.3, 5, 5.1, 6.2

Chen, S., Zhao, H., Kong, M., and Luo, B. (2007). 2D-LPP: A two-dimensional extension of locality preserving projections. *Neurocomputing*, 70(4):912–921. 1.2, 3.4, 5.4

Chung, F. R. K. (1997). *Spectral Graph Theory*. Number 92 in CBMS Regional Conference Series in Mathematics. American Mathematical Society. 2.9.2

Duda, R. O., Hart, P. E., and Stork, D. G. (2001). *Pattern Classification*. Wiley, 2nd edition. 5.1

Errity, A. and McKenna, J. (2007). A comparative study of linear and nonlinear dimensionality reduction for speaker identification. In *Proceedings of the 15th International Conference on Digital Signal Processing (ICDSP)*, pages 587–590. 1.1

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188. 2.3

Fralick, S. (1967). Learning to recognize patterns without a teacher. *IEEE Transactions on Information Theory*, 13(1):57–64. 1.3

Fukunaga, K. (1990). *Introduction to Statistical Pattern Recognition*. Computer Science & Scientific Computing. Academic Press, 2nd edition. 1.2, 2.2, 2.3, 3, 2.9, 3, 4.1

Godfrey, J. J., Holliman, E. C., and McDaniel, J. (1992). Switchboard: Telephone speech corpus for research and development. In *Proceedings of the 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1, pages 517–520. IEEE. 1.3

Gonzalez, R. C. and Woods, R. E. (2011). *Digital Image Processing*. Prentice Hall, 3rd edition. 5.3

He, X. and Niyogi, P. (2004). Locality preserving projections. In *Advances in Neural Information Processing Systems 16*, volume 16, pages 153–160, Cambridge, Massachusetts, USA. MIT Press. 2.4

Hosmer Jr, D. W. (1973). A comparison of iterative maximum likelihood estimates of the parameters of a mixture of two normal distributions under three different types of sample. *Biometrics*, 29(4):761–770. 1.3

Jain, G., Ginwala, A., and Aslandogan, Y. (2004). An approach to text classification using dimensionality reduction and combination of classifiers. In *Proceedings of the 2004 IEEE International Conference on Information Reuse and Integration (IRI)*, pages 564–569. 1.1

Jolliffe, I. T. (2002). *Principal component analysis*. Springer series in statistics. Springer, 2nd edition. 1.1, 2.2

Karni, Z. and Gotsman, C. (2004). Compression of soft-body animation sequences. *Computers & Graphics*, 28(1):25–34. 1.1

Kong, H., Teoh, E. K., Wang, J. G., and Venkateswarlu, R. (2005). Two Dimensional Fisher Discriminant Analysis: Forget About Small Sample Size Problem. In *Proceedings of the 2005 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 2, pages 761–764. 1.2

Li, M. and Yuan, B. (2005). 2D-LDA: A statistical linear discriminant analysis for image matrix. *Pattern Recognition Letters*, 26(5):527–532. 1.2

Liu, H. and Motoda, H. (1998). *Feature Extraction, Construction and Selection: A Data Mining Perspective*. Kluwer International Series in Engineering and Computer Science. Kluwer Academic. 1.1

Liu, K., Cheng, Y.-Q., and Yang, J.-Y. (1993). Algebraic feature extraction for image recognition based on an optimal discriminant criterion. *Pattern Recognition*, 26(6):903–911. 1.2

Lu, H., Plataniotis, K. N., and Venetsanopoulos, A. N. (2011). A survey of multilinear subspace learning for tensor data. *Pattern Recognition*, 44(7):1540–1551. 1.2

McLachlan, G. J. (1977). Estimating the linear discriminant function from initial samples containing a small number of unclassified observations. *Journal of the American Statistical Association*, 72(358):403–406.

McLachlan, G. J. and Ganesalingam, S. (1982). Updating a discriminant function on the basis of unclassified data. *Communications in Statistics – Simulation and Computation*, 11(6):753–767. 1.3

Merz, C. J., St.Clair, D. C., and Bond, W. E. (1992). Semi-supervised adaptive resonance theory (smart2). In *Proceedings of the 1992 International Joint Conference on Neural Networks (IJCNN)*, volume 3, pages 851–856. 1.3

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill, Boston, Massachusetts, USA. 5.1

Nagabhushan, P., Guru, D., and Shekar, B. (2006). $(2D)^2$ FLD: An efficient approach for appearance based object recognition. *Neurocomputing*, 69(7–9):934–940. 6.2

Nie, F., Xiang, S., Song, Y., and Zhang, C. (2009). Extracting the optimal dimensionality for local tensor discriminant analysis. *Pattern Recognition*, 42(1):105–114. 1.2

O'Neill, T. J. (1978). Normal discrimination with unclassified observations. *Journal of the American Statistical Association*, 73(364):821–826. 1.3

Ordowski, M. and Meyer, G. G. (2004). Geometric linear discriminant analysis for pattern recognition. *Pattern Recognition*, 37(3):421–428. 1

Pearson, K. (1901). LIII. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine Series 6*, 2(11):559–572. 2.2

Samaria, F. S. and Harter, A. C. (1994). Parameterisation of a stochastic model for human face identification. In *Proceedings of the Second IEEE Workshop on Applications of Computer Vision (WACV)*, pages 138–142. 5.2

Scudder, H., I. (1965). Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11(3):363–371. 1.3

Shashua, A. and Levin, A. (2001). Linear image coding for regression and classification using the tensor-rank principle. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages I–42. IEEE. 1.2

Sirovich, L. and Kirby, M. (1987). Low-dimensional procedure for the characterization of human faces. *Journal of the Optical Society of America A*, 4(3):519–524.

Spiro, I., Taylor, G., Williams, G., and Bregler, C. (2010). Hands by hand: Crowd-sourced motion tracking for gesture annotation. In *Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 17–24. 1.3

Sugiyama, M. (2007). Dimensionality Reduction of Multimodal Labeled Data by Local Fisher Discriminant Analysis. *The Journal of Machine Learning Research*, 8:1027–1061. 2.4, 2, 2.5, 2.9, 2.9, 2.9.2, 4.1

Sugiyama, M., Idé, T., Nakajima, S., and Sese, J. (2010). Semi-supervised local Fisher discriminant analysis for dimensionality reduction. *Machine Learning*, 78(1-2):35–61. 1.1, 2.8, 5.1, 6

Sun, D. and Zhang, D. (2009). A new discriminant principal component analysis method with partial supervision. *Neural Processing Letters*, 30(2):103–112. 2.7

Tao, D., Li, X., Wu, X., and Maybank, S. J. (2007). General Tensor Discriminant Analysis and Gabor Features for Gait Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(10):1700–1715. 1.1, 1.2

Tenenbaum, J. B., Silva, V. d., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323. 1.1

Theodoridis, S. and Koutroumbas, K. (2009). *Pattern Recognition*. Academic Press, 4th edition. 1, 2

Thomaz, C. E. and Giraldi, G. A. (2010). A new ranking method for principal components analysis and its application to face image analysis. *Image and Vision Computing*, 28(6):902–913. 5.3

Turk, M. and Pentland, A. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–89. 2.2

Vapnik, V. and Kotz, S. (2006). *Estimation of Dependences Based on Empirical Data*. Information Science and Statistics. Springer, 2nd edition. 5

Vasilescu, M. A. O. and Terzopoulos, D. (2003). Multilinear subspace analysis of image ensembles. In *Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages II–93. IEEE. 1.2

von Ahn, L. and Dabbish, L. (2004). Labeling images with a computer game. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, pages 319–326, New York, USA. ACM. 1.3

Wagstaff, K., Cardie, C., Rogers, S., and Schrödl, S. (2001). Constrained k-means clustering with background knowledge. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML)*, volume 1, pages 577–584. 1.3

Wang, X. Z., Medasani, S., Marhoon, F., and Albazzaz, H. (2004). Multidimensional visualization of principal component scores for process historical data analysis. *Industrial & Engineering Chemistry Research*, 43(22):7036–7048. 1.1

Xiong, H., Swamy, M., and Ahmad, M. O. (2005). Two-dimensional FLD for face recognition. *Pattern Recognition*, 38(7):1121–1124. 1.2, 3, 3.3, 5.4

Yang, J. and yu Yang, J. (2002). From image vector to matrix: a straightforward image projection technique——IMPCA vs. PCA. *Pattern Recognition*, 35(9):1997–1999. 1

Yang, J., Zhang, D., Frangi, A. F., and Yang, J.-y. (2004). Two-dimensional PCA: a new approach to appearance-based face representation and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(1):131–137. 1.2, 3, 3.2, 1, 5.4

Yang, J., Zhang, D., Yong, X., and Yang, J.-y. (2005). Two-dimensional discriminant transform for face recognition. *Pattern recognition*, 38(7):1125–1129. 1.2, 3

Ye, J., Janardan, R., and Li, Q. (2004). Two-dimensional linear discriminant analysis. In *Advances in Neural Information Processing Systems*, pages 1569–1576. 1.2

Zelnik-Manor, L. and Perona, P. (2005). Self-tuning spectral clustering. In *Advances in Neural Information Processing Systems 17*, pages 1601–1608, Cambridge, Massachusetts, USA. MIT Press. 2.4

Zhang, D. and Zhou, Z.-H. (2005). $(2D)^2$ PCA: two-directional two-dimensional PCA for efficient face representation and recognition. *Neurocomputing*, 69(1):224–231. 6.2

Zhang, D., Zhou, Z.-H., and Chen, S. (2007). Semi-supervised dimensionality reduction. In *Proceedings of the 7th SIAM International Conference on Data Mining*, pages 1–6, Minneapolis, Minnesota, USA. 2.6

Zhu, X. and Goldberg, A. B. (2009). Introduction to Semi-Supervised Learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 3(1):1–130. 1.3, 1.3

This volume has been typeset in LaTeXwith the UFPEThesis class (www.cin.ufpe.br/~paguso/ufpethesis). For details about this document, click here.