Semi-Supervised Local Fisher Discriminant Analysis for Dimensionality Reduction*

Masashi Sugiyama
Department of Computer Science, Tokyo Institute of Technology
2-12-2 O-okayama, Meguro-ku, Tokyo 152-8552, Japan
sugi@cs.titech.ac.jp

Tsuyoshi Idé
IBM Research, Tokyo Research Laboratory
1623-14 Shimo-tsuruma, Yamato-shi, Kanagawa 242-8502, Japan
goodidea@jp.ibm.com

Shinichi Nakajima
Nikon Corporation
201-9 Oaza-Miizugahara, Kumagaya-shi, Saitama 360-8559, Japan
nakajima.s@nikon.co.jp

Jun Sese
Department of Information Science, Ochanomizu University
2-1-1 Otsuka, Bunkyo, Tokyo 112-8610, Japan
sesejun@is.ocha.ac.jp

Abstract

When only a small number of labeled samples are available, supervised dimensionality reduction methods tend to perform poorly because of overfitting. In such cases, unlabeled samples could be useful in improving the performance. In this paper, we propose a semi-supervised dimensionality reduction method which preserves the global structure of unlabeled samples in addition to separating labeled samples in different classes from each other. The proposed method, which we call *SEmi-supervised Local Fisher discriminant analysis* (SELF), has an analytic form of the globally optimal solution and it can be computed based on eigen-decomposition. We show the usefulness of SELF through experiments with benchmark and real-world document classification datasets.

[*] A preliminary version of this paper was previously published in Sugiyama et al. (2008). A MATLAB implementation of the proposed dimensionality reduction method *SELF* is available from 'http://sugiyama-www.cs.titech.ac.jp/~sugi/software/SELF'.

1 Introduction

The goal of dimensionality reduction is to obtain a low-dimensional representation of high-dimensional data samples while preserving most of the 'intrinsic information' contained in the original data (Roweis & Saul, 2000; Tenenbaum et al., 2000; Hinton & Salakhutdinov, 2006). If dimensionality reduction is carried out appropriately, the compact representation of the data can be used for various tasks such as visualization and classification.

In supervised learning scenarios where data samples are accompanied with class labels, *Fisher discriminant analysis* (FDA) (Fisher, 1936; Fukunaga, 1990) is a popular dimensionality reduction method. FDA seeks an embedding transformation such that the between-class scatter is maximized and the within-class scatter is minimized. FDA works very well if the samples in each class follow Gaussian distributions with a shared covariance structure. However, FDA tends to give undesired results if the samples in a class form several separate clusters or there are outliers (Fukunaga, 1990). To overcome this drawback, *Local FDA* (LFDA) has been proposed (Sugiyama, 2007). LFDA localizes the evaluation of the within-class scatter, and thus works well even when within-class multimodality or outliers exist. In addition, LFDA overcomes a critical limitation of the original FDA in dimensionality reduction—the dimension of the FDA embedding space should be less than the number of classes (Fukunaga, 1990), while LFDA does not suffer from this restriction in general. Moreover, LFDA was shown to compare favorably with other supervised dimensionality reduction methods through experiments (Sugiyama, 2007).

However, the performance of LFDA (and all other supervised dimensionality reduction methods) tends to be degraded when only a small number of labeled samples are available. Namely, the supervised dimensionality reduction methods tend to find embedding spaces which are overfitted to the labeled samples. In such cases, it is effective to make use of *unlabeled* samples that are often available abundantly—such a setup is called *semi-supervised* learning (Chapelle et al., 2006). Through extensive experiments, it was shown that *principal component analysis* (PCA) (Jolliffe, 1986), which is an unsupervised dimensionality reduction method for preserving the global data structure, works moderately well in semi-supervised learning scenarios (see e.g., Chapter 21 of Chapelle et al., 2006).

Although PCA was reported to work well, it may not be the best possible choice in the semi-supervised situation because of its unsupervised nature. In this paper, we propose an alternative semi-supervised dimensionality reduction method. Our basic idea is to smoothly bridge LFDA and PCA so that our reliance on the global structure of unlabeled samples and information brought by (a small number of) labeled samples can be controlled. We show experimentally that the proposed method, which we refer to as *semi-supervised LFDA* (SELF), compares favorably with other methods. Note that SELF maintains the same computational advantage of LFDA and PCA, i.e., a global solution can be analytically computed based on eigen-decomposition. Therefore, SELF is still computationally as efficient as LFDA and PCA.

The rest of this paper is organized as follows. In Section 2, the linear dimensionality reduction problem addressed in this paper is formulated and some mathematical facts used in the following sections are briefly summarized. In Section 3, existing supervised and unsupervised dimensionality reduction methods are reviewed in a systematic and unified manner. This unified view will be the foundation for developing our new method in the following section. Those who are familiar with the existing methods and interested in immediately looking at the new method may choose to skip the review materials provided in Section 3. In Section 4, we propose the new semi-supervised dimensionality reduction method SELF and show its properties. Section 5 is devoted to experiments showing the usefulness of the proposed approach. Finally, in Section 6, we conclude with a discussion on possible future directions.

2 Preliminaries

In this section, we formulate the linear dimensionality reduction problem and give some mathematical background.

2.1 Formulation

Let $\boldsymbol{x}_i \in \mathbb{R}^d$ $(i = 1, 2, \ldots, n)$ be d-dimensional sample vectors and let $\boldsymbol{X} \in \mathbb{R}^{d\times n}$ be the matrix of all samples:

$$\boldsymbol{X} := (\boldsymbol{x}_1|\boldsymbol{x}_2|\cdots|\boldsymbol{x}_n).$$

Let $\boldsymbol{z} \in \mathbb{R}^r$ $(1 \leq r \leq d)$ be a low-dimensional representation of a high-dimensional sample $\boldsymbol{x} \in \mathbb{R}^d$, where r is the dimensionality of the reduced space. For the moment, we focus on linear dimensionality reduction, i.e., using a transformation matrix $\boldsymbol{T} \in \mathbb{R}^{d\times r}$, an embedded representation $\boldsymbol{z}$ of the sample $\boldsymbol{x}$ is obtained as

$$\boldsymbol{z} = \boldsymbol{T}^\top \boldsymbol{x},$$

where $^\top$ denotes the transpose of a matrix or a vector. Later, we extend our discussion to cases where the mapping from $\boldsymbol{x}$ to $\boldsymbol{z}$ is non-linear.

2.2 Generalized Eigenvalue Problem

Many dimensionality reduction techniques developed so far involve an optimization problem of the following form:

$$\boldsymbol{T}^{(\mathrm{OPT})} := \underset{\boldsymbol{T}\in\mathbb{R}^{d\times r}}{\mathrm{argmax}} \left[\mathrm{tr}\big(\boldsymbol{T}^\top \boldsymbol{B}\boldsymbol{T}(\boldsymbol{T}^\top \boldsymbol{C}\boldsymbol{T})^{-1}\big)\right]. \tag{1}$$

Roughly speaking, $\boldsymbol{B}$ encodes the quantity that we want to increase (e.g., between-class separability), and $\boldsymbol{C}$ corresponds to the quantity that we want to decrease (e.g., within-class scatter). In the next section, we show how $\boldsymbol{B}$ and $\boldsymbol{C}$ are designed in some specific cases. Note that the same solution $\boldsymbol{T}^{(\mathrm{OPT})}$ can also be obtained as follows (see e.g., Fukunaga, 1990):

$$\boldsymbol{T}^{(\mathrm{OPT})} = \underset{\boldsymbol{T}\in\mathbb{R}^{d\times r}}{\mathrm{argmax}} \left[\mathrm{tr}\big(\boldsymbol{T}^\top \boldsymbol{B}\boldsymbol{T}\big)\right] \text{ subject to } \boldsymbol{T}^\top\boldsymbol{C}\boldsymbol{T} = \boldsymbol{I}_r, \tag{2}$$

$$\boldsymbol{T}^{(\mathrm{OPT})} = \underset{\boldsymbol{T}\in\mathbb{R}^{d\times r}}{\mathrm{argmax}} \left[\frac{\det\big(\boldsymbol{T}^\top \boldsymbol{B}\boldsymbol{T}\big)}{\det\big(\boldsymbol{T}^\top \boldsymbol{C}\boldsymbol{T}\big)}\right],$$

where $\boldsymbol{I}_r$ is the identity matrix on $\mathbb{R}^r$ and $\det(\cdot)$ denotes the determinant of a matrix.

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

$$\boldsymbol{B}\boldsymbol{\varphi} = \lambda\boldsymbol{C}\boldsymbol{\varphi}. \tag{3}$$

The generalized eigenvectors are $\boldsymbol{C}$-*orthogonal* (Bai et al., 2000), i.e., for $k \neq k'$,

$$\boldsymbol{\varphi}_k^\top \boldsymbol{C}\boldsymbol{\varphi}_{k'} = 0.$$

We assume that the generalized eigenvalues are sorted in descending order as

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d, \tag{4}$$

and the generalized eigenvectors are normalized as

$$\boldsymbol{\varphi}_k^\top \boldsymbol{C}\boldsymbol{\varphi}_k = 1 \quad \text{for } k = 1, 2, \ldots, d. \tag{5}$$

Note that this normalization is often carried out automatically by an eigen-solver. Then a solution $\boldsymbol{T}^{(\mathrm{OPT})}$ is analytically given as follows (e.g., Fukunaga, 1990):

$$(\boldsymbol{\varphi}_1|\boldsymbol{\varphi}_2|\cdots|\boldsymbol{\varphi}_r).$$

It can be confirmed that Eq.(1) is invariant under linear transformations (Fukunaga, 1990), i.e., for any r-dimensional invertible matrix $\boldsymbol{U}$, $\boldsymbol{T}^{(\mathrm{OPT})}\boldsymbol{U}$ is also a global solution. This implies that the *range* of the embedding space can be uniquely determined by Eq.(1), but the *metric* in the embedding space is arbitrary. A practically useful heuristic (e.g., Sugiyama, 2007) is to set

$$\boldsymbol{U} = \mathrm{diag}(\sqrt{\lambda_1}, \sqrt{\lambda_2}, \ldots, \sqrt{\lambda_r}), \tag{6}$$

where $\mathrm{diag}(a, b, \ldots, c)$ denotes the diagonal matrix with the diagonal elements $a, b, \ldots, c$ and we assume that the generalized eigenvalues $\{\lambda_k\}_{k=1}^d$ are non-negative. Then the solution is given as

$$\boldsymbol{T}^{(\mathrm{OPT})} = (\sqrt{\lambda_1}\boldsymbol{\varphi}_1|\sqrt{\lambda_2}\boldsymbol{\varphi}_2|\cdots|\sqrt{\lambda_r}\boldsymbol{\varphi}_r). \tag{7}$$

Thus, the minor eigenvectors are deemphasized according to the square root of the eigenvalues. We will use this weighted solution in this paper.

2.3 Pairwise Expression of Scatter Matrices

When addressing dimensionality reduction problems, we are often dealing with a matrix of the following *pairwise* form (Belkin & Niyogi, 2003; Sugiyama, 2007), since it is convenient to describe the relation between pairs of features regarding whether pairs are close together or far apart:

$$\boldsymbol{S} := \frac{1}{2}\sum_{i,j=1}^{n} W_{i,j}(\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top, \tag{8}$$

where $\boldsymbol{W}$ is some $n \times n$ matrix. Let $\boldsymbol{D}$ be the $n \times n$ diagonal matrix with

$$D_{i,i} := \sum_{j=1}^{n} W_{i,j},$$

and let $\boldsymbol{L}$ be

$$\boldsymbol{L} := \boldsymbol{D} - \boldsymbol{W}.$$

Then the matrix $\boldsymbol{S}$ can be expressed in terms of $\boldsymbol{L}$ as

$$\begin{aligned}\boldsymbol{S} &= \sum_{i,j=1}^{n} W_{i,j}\boldsymbol{x}_i\boldsymbol{x}_i^\top - \sum_{i,j=1}^{n} W_{i,j}\boldsymbol{x}_i\boldsymbol{x}_j^\top \\ &= \sum_{i=1}^{n} D_{i,i}\boldsymbol{x}_i\boldsymbol{x}_i^\top - \boldsymbol{X}\boldsymbol{W}\boldsymbol{X}^\top \\ &= \boldsymbol{X}\boldsymbol{L}\boldsymbol{X}^\top. \end{aligned} \tag{9}$$

If we regard $\boldsymbol{W}$ as a weight matrix for a graph with n nodes, $\boldsymbol{L}$ can be regarded as a *graph Laplacian* matrix in *spectral graph theory* (Chung, 1997). If $\boldsymbol{W}$ is symmetric and its elements are all non-negative, $\boldsymbol{L}$ is known to be positive semi-definite.

In the following, we frequently use the matrices $\boldsymbol{S}^{(\cdot)}$, $\boldsymbol{W}^{(\cdot)}$, $\boldsymbol{D}^{(\cdot)}$, and $\boldsymbol{L}^{(\cdot)}$. They are all defined as above.

3 Review of Existing Dimensionality Reduction Methods

In this section, we review the existing dimensionality reduction methods. Our review will be in terms of the pairwise expression (8) in a unified framework. This unified formulation facilitates the development of a new method in the following sections. Those who are familiar with existing methods of supervised and unsupervised dimensionality reduction and interested in immediately looking at the new method may skip this section and go directly to Section 4.

3.1 Principal Component Analysis (PCA)

A fundamental unsupervised dimensionality reduction method is *principal component analysis* (PCA) (Jolliffe, 1986), which iteratively finds the maximum-variance direction of the data points. Below, we formulate PCA in a slightly different manner based on the pairwise expression (8).

Let $\boldsymbol{S}^{(\mathrm{t})}$ be the *total scatter matrix*:

$$\boldsymbol{S}^{(\mathrm{t})} := \sum_{i=1}^{n} (\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^\top,$$

where $\boldsymbol{\mu}$ is the mean of all of the samples:

$$\boldsymbol{\mu} := \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i.$$

Note that $\boldsymbol{S}^{(\mathrm{t})}$ can be expressed in a pairwise form as

$$\begin{aligned}\boldsymbol{S}^{(\mathrm{t})} =& \sum_{i=1}^{n} \boldsymbol{x}_i\boldsymbol{x}_i^\top - n\boldsymbol{\mu}\boldsymbol{\mu}^\top \\ =& \frac{1}{n}\sum_{i,j=1}^{n} \boldsymbol{x}_i\boldsymbol{x}_i^\top - \frac{1}{n}\sum_{i,j=1}^{n} \boldsymbol{x}_i\boldsymbol{x}_j^\top \\ =& \frac{1}{2}\sum_{i,j=1}^{n} W_{i,j}^{(\mathrm{t})}(\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top,\end{aligned}$$

where $\boldsymbol{W}^{(\mathrm{t})}$ is the $n \times n$ matrix with

$$W_{i,j}^{(\mathrm{t})} := \frac{1}{n}. \tag{10}$$

The PCA transformation matrix $\boldsymbol{T}^{(\mathrm{PCA})}$ is defined as

$$\boldsymbol{T}^{(\mathrm{PCA})} := \underset{\boldsymbol{T}\in\mathbb{R}^{d\times r}}{\mathrm{argmax}} \left[\mathrm{tr}\left(\boldsymbol{T}^\top\boldsymbol{S}^{(\mathrm{t})}\boldsymbol{T}(\boldsymbol{T}^\top\boldsymbol{T})^{-1}\right)\right]. \tag{11}$$

If we use the equivalent formulation (2), we see that PCA seeks a transformation matrix $\boldsymbol{T}$ such that the scatter in the embedding space is maximized. A solution $\boldsymbol{T}^{(\mathrm{PCA})}$ is given by Eqs.(3) and (7) with

$$\boldsymbol{B} = \boldsymbol{S}^{(\mathrm{t})} \text{ and } \boldsymbol{C} = \boldsymbol{I}_d.$$

3.2 Locality-Preserving Projection (LPP)

Another useful unsupervised dimensionality reduction technique is *locality-preserving projection* (LPP) (He & Niyogi, 2004).

Let $\boldsymbol{A}$ be an *affinity matrix*, i.e., an $n \times n$ matrix with $A_{i,j}$ being the affinity between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. We assume that $A_{i,j} \in [0,1]$, where $A_{i,j}$ is large if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are 'close' and $A_{i,j}$ is small if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are 'far apart'. There are several different manners of defining $\boldsymbol{A}$, such as using the nearest neighbors (Roweis & Saul, 2000) or the heat kernel (Belkin & Niyogi, 2003). In this paper, we use the *local scaling heuristic* (Zelnik-Manor & Perona, 2005) as the definition of the affinity matrix $\boldsymbol{A}$, i.e.,

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

The parameter σ_i represents the local scaling around $\boldsymbol{x}_i$ defined by

$$\sigma_i := \|\boldsymbol{x}_i - \boldsymbol{x}_i^{(k)}\|,$$

where $\boldsymbol{x}_i^{(k)}$ is the k-th nearest neighbor of $\boldsymbol{x}_i$. A heuristic choice of $k = 7$ was shown to be useful through experiments (Zelnik-Manor & Perona, 2005; Sugiyama, 2007).

Let $\boldsymbol{S}^{(\mathrm{n})}$ and $\boldsymbol{S}^{(\mathrm{l})}$ be the *normalization matrix* and the *local scatter matrix* defined by

$$\boldsymbol{S}^{(\mathrm{n})} := \boldsymbol{X}\boldsymbol{D}^{(\mathrm{n})}\boldsymbol{X}^\top,$$
$$\boldsymbol{S}^{(\mathrm{l})} := \frac{1}{2}\sum_{i,j=1}^{n} W_{i,j}^{(\mathrm{l})}(\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top,$$

where $\boldsymbol{D}^{(\mathrm{n})}$ is the $n \times n$ diagonal matrix with

$$D_{i,i}^{(\mathrm{n})} := \frac{1}{n}\sum_{j=1}^{n} A_{i,j},$$

and $\boldsymbol{W}^{(\mathrm{l})}$ is the $n \times n$ matrix with

$$W_{i,j}^{(\mathrm{l})} := \frac{1}{n}A_{i,j}.$$

The LPP transformation matrix $\boldsymbol{T}^{(LPP)}$ is defined as

$$\boldsymbol{T}^{(LPP)} := \underset{\boldsymbol{T}\in\mathbb{R}^{d\times r}}{\operatorname{argmin}}\left[\mathrm{tr}\Big(\boldsymbol{T}^\top\boldsymbol{S}^{(\mathrm{l})}\boldsymbol{T}(\boldsymbol{T}^\top\boldsymbol{S}^{(\mathrm{n})}\boldsymbol{T})^{-1}\Big)\right].$$

Taking into account the equivalence between Eqs.(1) and (2), we see that LPP seeks a transformation matrix $\boldsymbol{T}$ such that *nearby* data pairs in the original space $\mathbb{R}^d$ are kept close in the embedding space $\mathbb{R}^r$ (with $(\boldsymbol{T}^\top\boldsymbol{S}^{(\mathrm{n})}\boldsymbol{T})^{-1}$ regarded as a normalization constraint). Thus, LPP tends to preserve the local structures of the data.

As shown above, LPP is formulated as a minimization problem. To make this consistent with the other methods reviewed here, let us consider an inverted version of LPP.

$$\boldsymbol{T}^{(\mathrm{iLPP})} := \underset{\boldsymbol{T}\in\mathbb{R}^{d\times r}}{\operatorname{argmax}}\left[\mathrm{tr}\Big(\boldsymbol{T}^\top\boldsymbol{S}^{(\mathrm{n})}\boldsymbol{T}(\boldsymbol{T}^\top\boldsymbol{S}^{(\mathrm{l})}\boldsymbol{T})^{-1}\Big)\right].$$

When $\boldsymbol{S}^{(\mathrm{n})}$ is an identity, the inverted LPP (iLPP) agrees with the original LPP according to Eq.(3); otherwise the iLPP solution may be different from that of the original LPP.

A solution $\boldsymbol{T}^{(\mathrm{iLPP})}$ is given by Eqs.(3) and (7) with

$$\boldsymbol{B} = \boldsymbol{S}^{(\mathrm{n})} \text{ and } \boldsymbol{C} = \boldsymbol{S}^{(\mathrm{l})}.$$

3.3 Fisher Discriminant Analysis (FDA) for Dimensionality Reduction

Fisher discriminant analysis (FDA) is a popular supervised dimensionality reduction technique (Fisher, 1936; Fukunaga, 1990). When discussing supervised learning problems, we suppose that we have n' labeled samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n'}$, where $y_i \in \{1, 2, \ldots, c\}$ is a class label associated with the sample $\boldsymbol{x}_i$ and c is the number of classes. Let n'_m be the number of labeled samples in class $m \in \{1, 2, \ldots, c\}$:

$$n' = \sum_{m=1}^{c} n'_m.$$

Let $\boldsymbol{S}^{(\mathrm{b})}$ and $\boldsymbol{S}^{(\mathrm{w})}$ be the *between-class scatter matrix* and the *within-class scatter matrix*:

$$\boldsymbol{S}^{(\mathrm{b})} := \sum_{m=1}^{c} n'_m (\boldsymbol{\mu}_m - \boldsymbol{\mu})(\boldsymbol{\mu}_m - \boldsymbol{\mu})^\top,$$

$$\boldsymbol{S}^{(\mathrm{w})} := \sum_{m=1}^{c} \sum_{i:y_i=m} (\boldsymbol{x}_i - \boldsymbol{\mu}_m)(\boldsymbol{x}_i - \boldsymbol{\mu}_m)^\top,$$

where $\sum_{i:y_i=m}$ indicates the summation over i such that $y_i = m$ and $\boldsymbol{\mu}_m$ is the mean of samples in class m:

$$\boldsymbol{\mu}_m := \frac{1}{n'_m} \sum_{i:y_i=m} \boldsymbol{x}_i.$$

The FDA transformation matrix $\boldsymbol{T}^{(\mathrm{FDA})}$ is defined as

$$\boldsymbol{T}^{(\mathrm{FDA})} := \underset{\boldsymbol{T} \in \mathbb{R}^{d \times r}}{\mathrm{argmax}} \left[\mathrm{tr}\Big(\boldsymbol{T}^\top \boldsymbol{S}^{(\mathrm{b})} \boldsymbol{T} (\boldsymbol{T}^\top \boldsymbol{S}^{(\mathrm{w})} \boldsymbol{T})^{-1} \Big) \right].$$

That is, FDA seeks a transformation matrix $\boldsymbol{T}$ such that the between-class scatter in the embedding space (i.e., $\boldsymbol{T}^\top \boldsymbol{S}^{(\mathrm{b})} \boldsymbol{T}$) is 'maximized' and the within-class scatter in the embedding space (i.e., $\boldsymbol{T}^\top \boldsymbol{S}^{(\mathrm{w})} \boldsymbol{T}$) is 'minimized'. A solution $\boldsymbol{T}^{(\mathrm{FDA})}$ is given by Eqs.(3) and (7) with

$$\boldsymbol{B} = \boldsymbol{S}^{(\mathrm{b})} \text{ and } \boldsymbol{C} = \boldsymbol{S}^{(\mathrm{w})}.$$

It is known (e.g., Fukunaga, 1990) that $\boldsymbol{S}^{(\mathrm{b})}$ and $\boldsymbol{S}^{(\mathrm{w})}$ are related to the total scatter matrix $\boldsymbol{S}^{(\mathrm{t})}$ as

$$\boldsymbol{S}^{(\mathrm{t})} = \boldsymbol{S}^{(\mathrm{b})} + \boldsymbol{S}^{(\mathrm{w})}. \tag{12}$$

This can also be confirmed from the fact that $\boldsymbol{S}^{(\mathrm{b})}$ and $\boldsymbol{S}^{(\mathrm{w})}$ are expressed in the pairwise form (8) with the following weight matrices (Sugiyama, 2007):

$$W_{i,j}^{(\mathrm{b})} := \begin{cases} 1/n' - 1/n'_{y_i} & \text{if } y_i = y_j, \\ 1/n' & \text{if } y_i \neq y_j, \end{cases}$$

$$W_{i,j}^{(\mathrm{w})} := \begin{cases} 1/n'_{y_i} & \text{if } y_i = y_j, \\ 0 & \text{if } y_i \neq y_j, \end{cases}$$

where n'_{y_i} denotes the number of labeled samples in class $y_i \in \{1, 2, \ldots, c\}$. In this case, we have

$$\boldsymbol{W}^{(\mathrm{t})} = \boldsymbol{W}^{(\mathrm{b})} + \boldsymbol{W}^{(\mathrm{w})},$$

since $W^{(\mathrm{t})}_{i,j} := 1/n'$ in the current setup (cf. Eq.(10)).

The between-class scatter matrix $\boldsymbol{S}^{(\mathrm{b})}$ has at most rank $c-1$ (Fukunaga, 1990). This implies that FDA allows us to obtain at most $c-1$ meaningful features (or equivalently the dimensionality r of the embedding space should be at most $c-1$), and the remaining features found by FDA are arbitrary in the null space of $\boldsymbol{S}^{(\mathrm{b})}$. This is an essential limitation of FDA in dimensionality reduction.

3.4 Local Fisher Discriminant Analysis (LFDA)

Local Fisher Discriminant Analysis (LFDA) is a supervised dimensionality reduction method (Sugiyama, 2007) which overcomes the weakness of the original FDA against within-class multimodality or outliers (Fukunaga, 1990).

Let $\boldsymbol{S}^{(\mathrm{lb})}$ and $\boldsymbol{S}^{(\mathrm{lw})}$ be the *local* between-class scatter matrix and the *local* within-class scatter matrix defined by

$$\boldsymbol{S}^{(\mathrm{lb})} := \frac{1}{2}\sum_{i,j=1}^{n'} W^{(\mathrm{lb})}_{i,j}(\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top,$$

$$\boldsymbol{S}^{(\mathrm{lw})} := \frac{1}{2}\sum_{i,j=1}^{n'} W^{(\mathrm{lw})}_{i,j}(\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top,$$

where $\boldsymbol{W}^{(\mathrm{lb})}$ and $\boldsymbol{W}^{(\mathrm{lw})}$ are the $n' \times n'$ matrices with

$$W^{(\mathrm{lb})}_{i,j} := \begin{cases} A_{i,j}(1/n' - 1/n'_{y_i}) & \text{if } y_i = y_j, \\ 1/n' & \text{if } y_i \neq y_j, \end{cases} \tag{13}$$

$$W^{(\mathrm{lw})}_{i,j} := \begin{cases} A_{i,j}/n'_{y_i} & \text{if } y_i = y_j, \\ 0 & \text{if } y_i \neq y_j. \end{cases} \tag{14}$$

$A_{i,j}$ is the affinity value between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ based on the local scaling heuristic (see Section 3.2). Note that the local scaling is computed in a classwise manner in LFDA since we want to preserve the within-class local structure (Sugiyama, 2007). This also contributes to reducing the computational cost for nearest neighbor search when computing the local scaling. The LFDA transformation matrix $\boldsymbol{T}^{(\mathrm{LFDA})}$ is defined as

$$\boldsymbol{T}^{(\mathrm{LFDA})} := \underset{\boldsymbol{T}\in\mathbb{R}^{d\times r}}{\mathrm{argmax}} \left[\mathrm{tr}\Big(\boldsymbol{T}^\top \boldsymbol{S}^{(\mathrm{lb})}\boldsymbol{T}(\boldsymbol{T}^\top \boldsymbol{S}^{(\mathrm{lw})}\boldsymbol{T})^{-1}\Big)\right].$$

In other words, LFDA seeks a transformation matrix $\boldsymbol{T}$ such that the local between-class scatter in the embedding space (i.e., $\boldsymbol{T}^\top\boldsymbol{S}^{(\mathrm{lb})}\boldsymbol{T}$) is 'maximized' and the local within-class scatter in the embedding space (i.e., $\boldsymbol{T}^\top\boldsymbol{S}^{(\mathrm{lw})}\boldsymbol{T}$) is 'minimized'.

In Eqs.(13) and (14), $A_{i,j}(1/n' - 1/n'_{y_i})$ is negative while $A_{i,j}/n'_{y_i}$ and $1/n'$ are positive. Thus LFDA imposes nearby data pairs in the same class to be close together and the data pairs in different classes to be far apart; far apart data pairs within the same class are not imposed to be close together. Samples in different classes are separated from each other irrespective of their affinity values. A solution $\boldsymbol{T}^{(\mathrm{LFDA})}$ is given by Eqs.(3) and (7) with

$$\boldsymbol{B} = \boldsymbol{S}^{(\mathrm{lb})} \text{ and } \boldsymbol{C} = \boldsymbol{S}^{(\mathrm{lw})}.$$

When $A_{i,j} = 1$ for all i, j (i.e., no locality), $\boldsymbol{S}^{(\mathrm{lw})}$ and $\boldsymbol{S}^{(\mathrm{lb})}$ are reduced to $\boldsymbol{S}^{(\mathrm{w})}$ and $\boldsymbol{S}^{(\mathrm{b})}$. Thus, LFDA can be regarded as a localized variant of FDA. The between-class scatter matrix $\boldsymbol{S}^{(\mathrm{b})}$ has at most rank $c-1$, while its local counterpart $\boldsymbol{S}^{(\mathrm{lb})}$ usually has full rank (given $n' \geq d$). Therefore, LFDA can be applied to dimensionality reduction into spaces of *any* dimension, which is also a significant advantage over the original FDA when the number of classes is small.

However, the performance of LFDA (and all other supervised dimensionality reduction methods) tends to be degraded if only a small number of labeled samples are available. The purpose of this paper is to give a new method that can overcome this weakness.

4 Semi-Supervised LFDA (SELF)

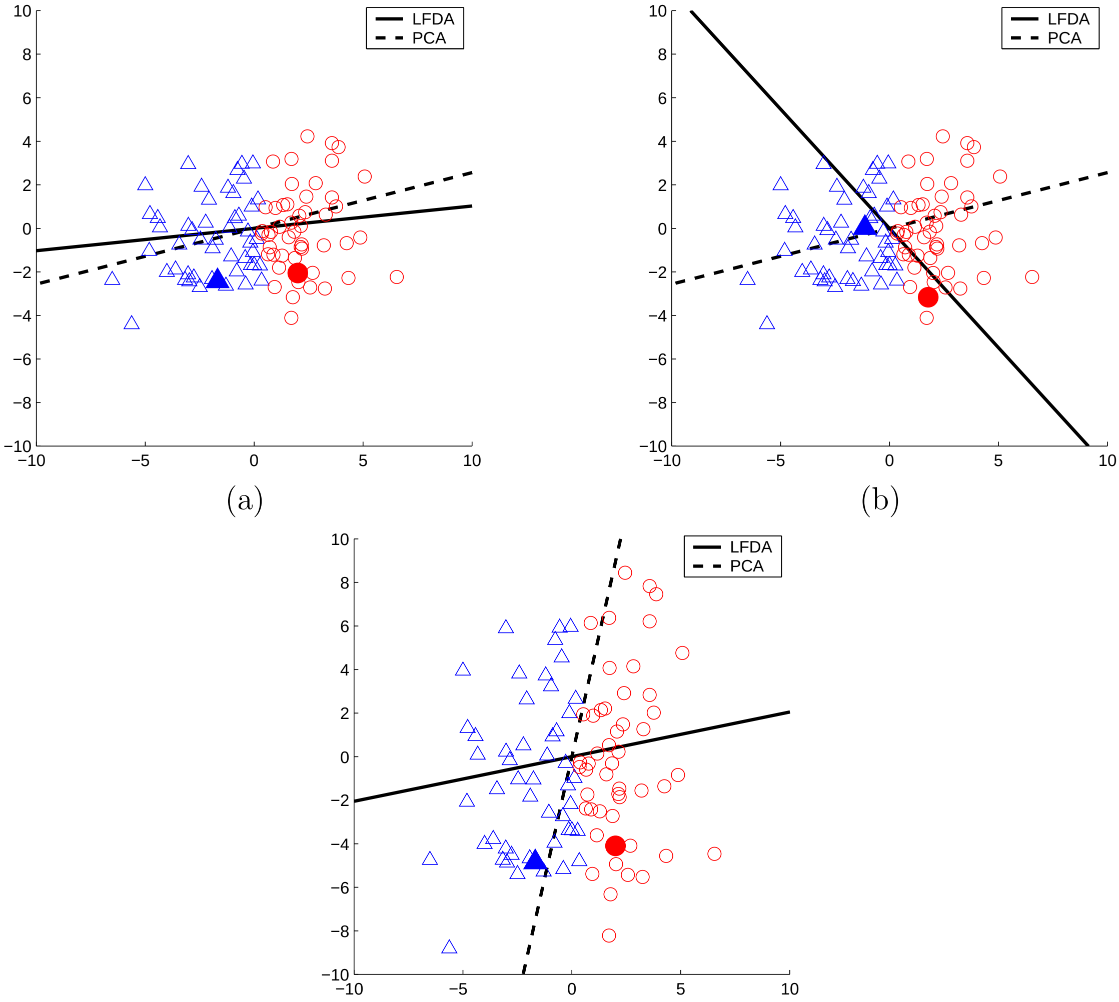
In this section, we propose a new dimensionality reduction method for semi-supervised learning scenarios. From here on, we consider the case where, among all of the samples $\{\boldsymbol{x}_i\}_{i=1}^{n}$, only $\{\boldsymbol{x}_i\}_{i=1}^{n'}$ $(1 \leq n' \leq n)$ are labeled and the rest are unlabeled.

4.1 Basic Idea

When only a small number of labeled samples are available, supervised dimensionality reduction methods tend to find the embedding spaces overfitted to the labeled samples. In such situations, the use of unlabeled samples can mitigate this problem—indeed, in Chapter 21 of Chapelle et al. (2006), it was shown through extensive experiments that PCA works well on the whole. Our experimental results in Section 5.1 also show that PCA is sometimes better than LFDA. This means that preserving the global structure of all of the samples in an unsupervised manner can be better than relying too much on class information provided by a small number of labeled samples.

Figure 1 depicts 2-dimensional 2-class examples. The circles and triangles denote the samples in positive and negative classes and the filled or unfilled symbols denote the labeled or unlabeled samples. The solid and dashed lines denote the 1-dimensional embedding spaces (onto which the data samples will be projected) found by LFDA and PCA, respectively.

For the dataset in Figure 1(a), both LFDA and PCA can find good embedding spaces which clearly separate unlabeled samples in different classes from each other. However, for the dataset in Figure 1(b), which contains the same sample points as (a) but in which the choice of the labeled samples is different, LFDA finds an embedding space that is



(a)

(b)

(c)

Figure 1: Illustrative examples of LFDA and PCA for toy datasets. The circles and triangles denote the samples in positive and negative classes and the filled or unfilled symbols denote the labeled or unlabeled samples. The solid and dashed lines denote the 1-dimensional embedding spaces (onto which the data samples will be projected) found by LFDA and PCA, respectively. The dataset is common to (a) and (b), but the choice of labeled samples is different. This only affects the LFDA solution because of its supervised nature; the PCA solution does not change because of its unsupervised nature. The choice of labeled samples is common to (a) and (c), but the vertical scaling of the data is doubled in (c). This affects both the LFDA and PCA solutions, but the PCA solution is more influenced because of its unsupervised nature.

overfitted to the labeled samples. Note that the choice of labeled samples only affects the LFDA solution—the PCA solution does not change because of its unsupervised nature. This illustrates a possible drawback of LFDA which relies strongly on a small number of labeled samples.

The dataset described in Figure 1(c) has the same choice of labeled samples as (a), but the vertical scaling of the data is doubled. Although this change of scales affects both the LFDA and PCA solutions, LFDA is not strongly influenced by the change of scales because of its supervised nature. In contrast, PCA is significantly influenced by the change of scales and does not work well for the dataset (c). This illustrates a possible weakness of PCA arising from its unsupervised nature.

The above result shows that LFDA and PCA have their own drawbacks. However, the above result also implies that LFDA and PCA can compensate for each other's weaknesses, i.e., LFDA can utilize label information, while PCA can avoid overfitting. Our experimental results with the benchmark datasets in Section 5.1 also show that LFDA and PCA tend to work in a complementary manner. Motivated by these facts, we propose to *bridge* LFDA and PCA so that our reliance on the global structure of unlabeled samples and class information brought by the labeled samples can be smoothly controlled. We refer to the proposed method as *semi-supervised LFDA* (SELF).

The embedding transformations of LFDA and PCA can be analytically computed through eigen-decomposition, as reviewed in the previous section. Based on this fact, we combine the eigenvalue problems of LFDA and PCA and solve them together. This allows us to retain the computational efficiency of LFDA and PCA.

As described in Section 3.4, LFDA includes FDA as a special case. Therefore, the idea of combining LFDA and PCA detailed below is also applicable to FDA.

4.2 Definition

More specifically, we propose to solve the following generalized eigenvalue problem:

$$\boldsymbol{S}^{(\mathrm{rlb})}\boldsymbol{\varphi} = \lambda \boldsymbol{S}^{(\mathrm{rlw})}\boldsymbol{\varphi}, \tag{15}$$

where $\boldsymbol{S}^{(\mathrm{rlb})}$ and $\boldsymbol{S}^{(\mathrm{rlw})}$ are the *regularized* local between-class scatter matrix and the *regularized* local within-class scatter matrix defined by

$$\boldsymbol{S}^{(\mathrm{rlb})} := (1-\beta)\boldsymbol{S}^{(\mathrm{lb})} + \beta\boldsymbol{S}^{(\mathrm{t})}, \tag{16}$$

$$\boldsymbol{S}^{(\mathrm{rlw})} := (1-\beta)\boldsymbol{S}^{(\mathrm{lw})} + \beta\boldsymbol{I}_d. \tag{17}$$

$\beta \in [0,1]$ is a trade-off parameter—SELF is reduced to LFDA when $\beta = 0$ and SELF is reduced to PCA when $\beta = 1$. In general, SELF with $0 < \beta < 1$ inherits the characteristics of both LFDA and PCA (discussed in detail in Section 4.3). One may use different trade-off parameters in $\boldsymbol{S}^{(\mathrm{rlb})}$ and $\boldsymbol{S}^{(\mathrm{rlw})}$ to increase the flexibility. However, this in turn makes the trade-off parameter choice laborious. For this reason, we focus on using the single shared trade-off parameter β for $\boldsymbol{S}^{(\mathrm{rlb})}$ and $\boldsymbol{S}^{(\mathrm{rlw})}$ below.

The optimization problem of SELF is expressed as

$$\boldsymbol{T}^{(\mathrm{SELF})} := \underset{\boldsymbol{T}\in\mathbb{R}^{d\times r}}{\mathrm{argmax}}\left[\mathrm{tr}\left(\boldsymbol{T}^\top \boldsymbol{S}^{(\mathrm{rlb})}\boldsymbol{T}(\boldsymbol{T}^\top \boldsymbol{S}^{(\mathrm{rlw})}\boldsymbol{T})^{-1}\right)\right].$$

In other words, SELF seeks a transformation matrix $\boldsymbol{T}$ such that the regularized local between-class scatter in the embedding space (i.e., $\boldsymbol{T}^\top \boldsymbol{S}^{(\mathrm{rlb})}\boldsymbol{T}$) is 'maximized' and the regularized local within-class scatter in the embedding space (i.e., $\boldsymbol{T}^\top \boldsymbol{S}^{(\mathrm{rlw})}\boldsymbol{T}$) is 'minimized'. Since this optimization problem is the same form as LFDA and PCA, a solution $\boldsymbol{T}^{(\mathrm{SELF})}$ can be computed as

$$\boldsymbol{T}^{(\mathrm{SELF})} = (\sqrt{\lambda_1}\boldsymbol{\varphi}_1|\sqrt{\lambda_2}\boldsymbol{\varphi}_2|\cdots|\sqrt{\lambda_r}\boldsymbol{\varphi}_r), \tag{18}$$

where $\{\boldsymbol{\varphi}_k\}_{k=1}^d$ are the generalized eigenvectors of Eq.(15) associated with the generalized eigenvalues $\{\lambda_k\}_{k=1}^d$. We assume that $\{\lambda_k\}_{k=1}^d$ are sorted in descending order as in Eq.(4) and $\{\boldsymbol{\varphi}_k\}_{k=1}^d$ are normalized as in Eq.(5). In Section 4.3, we will prove that all the generalized eigenvalues are non-negative, which guarantees that the solution (18) is always valid.

In the original LFDA, the nearest neighbor search (involved in the computation of local scaling σ_i in the affinity matrix $\boldsymbol{A}$) is carried out in a classwise manner (Sugiyama, 2007). On the other hand, in SELF, we determine the local scaling using all of the samples $\{\boldsymbol{x}_i\}_{i=1}^n$ since the number of labeled samples is typically small in semi-supervised learning. SELF requires affinity values $A_{i,j}$ only for the pairs of labeled samples in the same class. This means that we need to compute local scaling values only for the labeled samples and affinity values only for the labeled sample pairs in the same class. This contributes greatly to reducing the computational costs. The total scatter matrix $\boldsymbol{S}^{(\mathrm{t})}$ in the original PCA is computed for unlabeled samples, but we use all of the samples $\{\boldsymbol{x}_i\}_{i=1}^n$ (i.e., both the labeled and unlabeled samples) in SELF. The pseudo-code for SELF appears in Figure 2.

4.3 Properties

First, we give an interpretation of $\boldsymbol{S}^{(\mathrm{rlb})}$. The matrix $\boldsymbol{S}^{(\mathrm{rlb})}$ can be expressed in a pairwise form as

$$\boldsymbol{S}^{(\mathrm{rlb})} := \frac{1}{2}\sum_{i,j=1}^n W_{i,j}^{(\mathrm{rlb})}(\boldsymbol{x}_i-\boldsymbol{x}_j)(\boldsymbol{x}_i-\boldsymbol{x}_j)^\top, \tag{19}$$

where $\boldsymbol{W}^{(\mathrm{rlb})}$ is the $n\times n$ matrix with

$$W_{i,j}^{(\mathrm{rlb})} := \begin{cases} (1-\beta)A_{i,j}(1/n' - 1/n'_{y_i}) + \beta/n & \text{if } y_i = y_j, \\ (1-\beta)/n' + \beta/n & \text{if } y_i \neq y_j, \\ \beta/n & \text{otherwise.} \end{cases} \tag{20}$$

The first case in Eq.(20) is negative if

$$\beta < B_{i,j},$$

```
Input:   Labeled samples $\{(\boldsymbol{x}_i, y_i) \mid \boldsymbol{x}_i \in \mathbb{R}^d, y_i \in \{1, 2, \ldots, c\}\}_{i=1}^{n'}$
         Unlabeled samples $\{\boldsymbol{x}_i \mid \boldsymbol{x}_i \in \mathbb{R}^d\}_{i=n'+1}^{n}$
         Dimensionality of embedding space $r$ $(1 \leq r \leq d)$
         Trade-off parameter $\beta$ $(0 \leq \beta \leq 1)$
Output:  $d \times r$ transformation matrix $\boldsymbol{T}^{(\mathrm{SELF})}$

for $i = 1, 2, \ldots, n'$
    $\boldsymbol{x}_i^{(7)} \longleftarrow$ 7th nearest neighbor of $\boldsymbol{x}_i$ among $\{\boldsymbol{x}_j\}_{j=1}^{n}$;
    $\sigma_i \longleftarrow \|\boldsymbol{x}_i - \boldsymbol{x}_i^{(7)}\|$;
end
for $i, j = 1, 2, \ldots, n'$
    if $y_i = y_j$
        $A_{i,j} \longleftarrow \exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/(\sigma_i\sigma_j))$;
        $W_{i,j}^{(\mathrm{lb}')} \longleftarrow A_{i,j}(1/n' - 1/n'_{y_i})$;
        $W_{i,j}^{(\mathrm{lw}')} \longleftarrow A_{i,j}/n'_{y_i}$;
    else
        $W_{i,j}^{(\mathrm{lb}')} \longleftarrow 1/n'$;
        $W_{i,j}^{(\mathrm{lw}')} \longleftarrow 0$;
    end
end
$\boldsymbol{X}' \longleftarrow (\boldsymbol{x}_1|\boldsymbol{x}_2|\cdots|\boldsymbol{x}_{n'})$;
$\boldsymbol{S}^{(\mathrm{lb})} \longleftarrow \boldsymbol{X}' \left\{\mathrm{diag}(\boldsymbol{W}^{(\mathrm{lb}')}\boldsymbol{1}_{n'}) - \boldsymbol{W}^{(\mathrm{lb}')}\right\} \boldsymbol{X}'^{\top}$;
$\boldsymbol{S}^{(\mathrm{lw})} \longleftarrow \boldsymbol{X}' \left\{\mathrm{diag}(\boldsymbol{W}^{(\mathrm{lw}')}\boldsymbol{1}_{n'}) - \boldsymbol{W}^{(\mathrm{lw}')}\right\} \boldsymbol{X}'^{\top}$;
$\boldsymbol{X} \longleftarrow (\boldsymbol{x}_1|\boldsymbol{x}_2|\cdots|\boldsymbol{x}_n)$;
$\boldsymbol{\mu} \longleftarrow \boldsymbol{X}\boldsymbol{1}_n/n$;
$\boldsymbol{S}^{(\mathrm{t})} \longleftarrow \boldsymbol{X}\boldsymbol{X}^{\top} - n\boldsymbol{\mu}\boldsymbol{\mu}^{\top}$;
$\boldsymbol{S}^{(\mathrm{rlb})} \longleftarrow (1-\beta)\boldsymbol{S}^{(\mathrm{lb})} + \beta\boldsymbol{S}^{(\mathrm{t})}$;
$\boldsymbol{S}^{(\mathrm{rlw})} \longleftarrow (1-\beta)\boldsymbol{S}^{(\mathrm{lw})} + \beta\boldsymbol{I}_d$;
$\{\lambda_k, \boldsymbol{\varphi}_k\}_{k=1}^{r} \longleftarrow$ Generalized eigenvalues and eigenvectors of $\boldsymbol{S}^{(\mathrm{rlb})}\boldsymbol{\varphi} = \lambda\boldsymbol{S}^{(\mathrm{rlw})}\boldsymbol{\varphi}$,
                        where $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$ and $\boldsymbol{\varphi}_k^{\top}\boldsymbol{S}^{(\mathrm{rlw})}\boldsymbol{\varphi}_k = 1$;
$\boldsymbol{T}^{(\mathrm{SELF})} = (\sqrt{\lambda_1}\boldsymbol{\varphi}_1|\sqrt{\lambda_2}\boldsymbol{\varphi}_2|\cdots|\sqrt{\lambda_r}\boldsymbol{\varphi}_r)$;
```

Figure 2: Pseudo-code for SELF. $\boldsymbol{1}_n$ denotes the n-dimensional vector with ones and $\mathrm{diag}(\boldsymbol{w})$ denotes the diagonal matrix with the diagonal elements specified by a vector $\boldsymbol{w}$. A MATLAB implementation of SELF is available from '`http://sugiyama-www.cs.titech.ac.jp/~sugi/software/SELF`'.

where

$$B_{i,j} := \frac{A_{i,j} n(n' - n'_{y_i})}{A_{i,j} n(n' - n'_{y_i}) + n' n'_{y_i}}.$$

Note that $0 \leq B_{i,j} < 1$. This implies that SELF tries to make sample pairs in the same class close together if β is smaller than $B_{i,j}$, while it separates them farther from each other if β is larger than $B_{i,j}$. Thus the local data structures in the same class tend to be preserved when β is small, but are no longer preserved when β is large. $B_{i,j}$ is reduced when $A_{i,j}$ is increased, so $B_{i,j}$ is smallest in the case of FDA where $A_{i,j} = 1$ for all i, j.

The second case in Eq.(20) is always positive for any $\beta \in [0, 1]$, implying that SELF always tries to make sample pairs in different classes farther apart for any β. This would be natural in (semi-)supervised learning scenarios. The third case in Eq.(20) is always non-negative, implying that unlabeled samples are separated from each other to preserve the global data structure.

$\boldsymbol{S}^{(\mathrm{rlb})}$ includes the total scatter matrix $\boldsymbol{S}^{(\mathrm{t})}$ (see Eq.(16)), which is equivalent to the sum of $\boldsymbol{S}^{(\mathrm{b})}$ and $\boldsymbol{S}^{(\mathrm{w})}$ (see Eq.(12)). If samples in different classes were highly localized and clearly separated from each other, $\boldsymbol{S}^{(\mathrm{b})}$ would be dominant in $\boldsymbol{S}^{(\mathrm{t})}$ and thus $\boldsymbol{S}^{(\mathrm{t})}$ and $\boldsymbol{S}^{(\mathrm{b})}$ would be similar to each other. However, since $\boldsymbol{S}^{(\mathrm{b})}$ needs to be computed from a small number of labeled samples in semi-supervised learning, it is often unreliable. In contrast, $\boldsymbol{S}^{(\mathrm{t})}$ can be computed in a more reliable manner using a large number of unlabeled samples[1]. For this reason, including $\boldsymbol{S}^{(\mathrm{t})}$ in $\boldsymbol{S}^{(\mathrm{rlb})}$ will improve the reliability of the solution.

Next, we give an interpretation of $\boldsymbol{S}^{(\mathrm{rlw})}$. When $\beta = 0$, $\boldsymbol{S}^{(\mathrm{rlw})}$ $(= \boldsymbol{S}^{(\mathrm{lw})})$ could be ill-conditioned. This is particularly crucial when the dimension d of the original data space is larger than the number n' of labeled samples. In such situations, $\beta \boldsymbol{I}_d$ included in $\boldsymbol{S}^{(\mathrm{rlw})}$ (see Eq.(17)) works as a *regularizer* and SELF can avoid overfitting the labeled samples (cf. Friedman, 1989; Mika et al., 2003). Therefore, SELF is regarded as a regularized variant of LFDA and would be more stable and more reliable than the original LFDA, particularly when the number of labeled samples is small. Note that unlike Eq.(19), $\boldsymbol{S}^{(\mathrm{rlw})}$ does not have a pairwise expression since $\boldsymbol{I}_d$ cannot be expressed in a pairwise form.

Finally, we investigate the positive (semi-)definiteness of $\boldsymbol{S}^{(\mathrm{rlb})}$ and $\boldsymbol{S}^{(\mathrm{rlw})}$. Let

$$\boldsymbol{W}^{(\Delta \mathrm{lb})} = \boldsymbol{W}^{(\mathrm{lb})} - \boldsymbol{W}^{(\mathrm{b})},$$

which means that $\boldsymbol{W}^{(\Delta \mathrm{lb})}$ is the $n \times n$ matrix with

$$W^{(\Delta \mathrm{lb})}_{i,j} := \begin{cases} (A_{i,j} - 1)(1/n' - 1/n'_{y_i}) & \text{if } y_i = y_j, \\ 0 & \text{otherwise.} \end{cases}$$

Note that $(A_{i,j} - 1)(1/n' - 1/n'_{y_i})$ is non-negative. Then taking into account the relation (9), we can express $\boldsymbol{S}^{(\mathrm{lb})}$ as

$$\boldsymbol{S}^{(\mathrm{lb})} = \boldsymbol{S}^{(\mathrm{b})} + \boldsymbol{X} \boldsymbol{L}^{(\Delta \mathrm{lb})} \boldsymbol{X}^\top,$$

[1] This may partially explain why PCA is useful under the *cluster assumption*—samples in the same cluster are likely to have a common label (Chapelle et al., 2006).

where $\boldsymbol{L}^{(\Delta\mathrm{lb})}$ is defined with $\boldsymbol{W}^{(\Delta\mathrm{lb})}$ (see Section 2.3). Since $\boldsymbol{S}^{(\mathrm{b})}$ and $\boldsymbol{L}^{(\Delta\mathrm{lb})}$ are both symmetric positive semi-definite, $\boldsymbol{S}^{(\mathrm{lb})}$ is also symmetric positive semi-definite. In addition, since $\boldsymbol{S}^{(\mathrm{t})}$ is symmetric positive semi-definite and β and $1-\beta$ are non-negative, $\boldsymbol{S}^{(\mathrm{rlb})}$ is also symmetric positive semi-definite (see Eq.(16)). On the other hand, since $\boldsymbol{S}^{(\mathrm{lw})}$ is symmetric positive semi-definite and $\boldsymbol{I}_d$ is symmetric positive definite, $\boldsymbol{S}^{(\mathrm{rlw})}$ is symmetric positive definite if $\beta > 0$. The facts that $\boldsymbol{S}^{(\mathrm{rlb})}$ is symmetric positive semi-definite and $\boldsymbol{S}^{(\mathrm{rlw})}$ is symmetric positive definite guarantee that the generalized eigenvalues of Eq.(15) are non-negative (Bai et al., 2000). Thus, the solution (18) is always valid.

4.4 Numerical Examples

To illustrate how SELF behaves, we used the *Olivetti face* dataset[2]. The dataset consists of 400 gray-scale images of faces (40 people, 10 images per person). Each image consists of 4096 ($= 64 \times 64$) pixels and each pixel takes an integer value between 0 and 255 as the intensity level. In this experiment, we used the image samples of only 10 subjects (i.e., 100 images in total) to make the visualization results clear. We experimentally confirmed that the results do not change significantly (though points are more overlapped) when all 400 images are used.

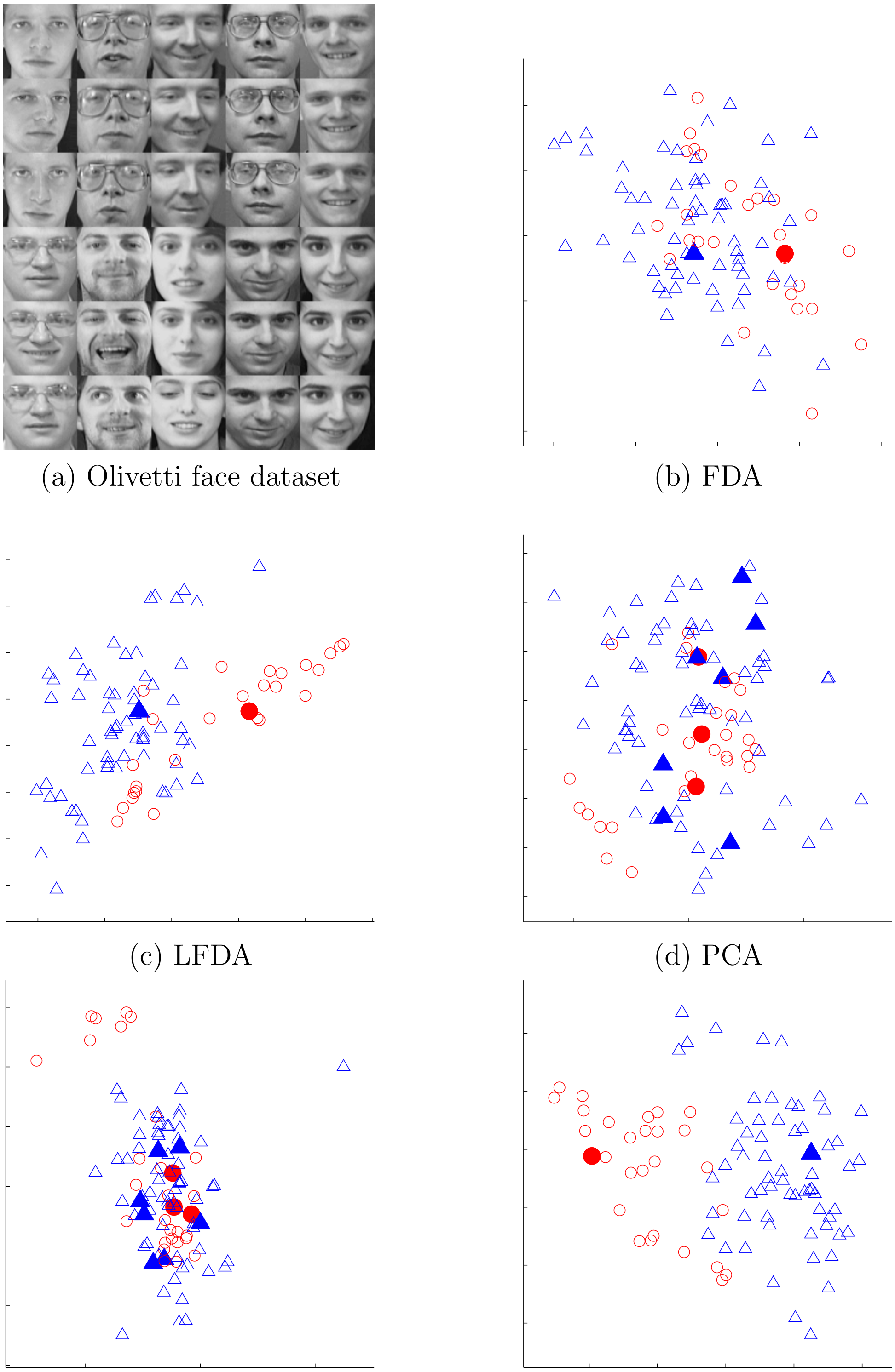
Among the 10 people used for the experiments, 3 subjects are wearing glasses and the other 7 subjects are without glasses (see Figure 3(a)). Our task was to embed the face images into a two-dimensional space so that the subjects *with* and *without* glasses were separated from each other. We labeled 1 image per person (so 3 faces are labeled as with glasses and 7 faces as without glasses in total) and the rest are treated as unlabeled. Since each class contains several different subjects, this dataset is thought to possess within-class multimodality.

The embedded results are depicted in Figure 3, where the circles and triangles denote the faces with or without glasses and the filled or unfilled symbols denote the labeled or unlabeled samples. The figure shows that FDA and LFDA perfectly separate the labeled samples in the two classes from each other. However, the unlabeled samples tend to be mixed because of an overfitting phenomenon. PCA and iLPP tend to mix the labeled samples in different classes because of their unsupervised natures. As a result, the unlabeled samples in different classes are also mixed. In contrast, SELF with $\beta = 0.5$ clearly separates the labeled samples in the two classes from each other, and at the same time, it also effectively separates the unlabeled samples in the two classes. We note that, in this visualization experiment, the result of SELF is not sensitive to the choice of the trade-off parameter β. The results are almost unchanged for $0.01 \leq \beta \leq 0.99$.

4.5 Discussion

Here we discuss several issues related to SELF.

[2] The dataset is available from '`http://www.cs.toronto.edu/~roweis/data.html`'.



(a) Olivetti face dataset

(b) FDA

(c) LFDA

(d) PCA

(e) iLPP

(f) SELF ($\beta = 0.5$)

Figure 3: Embedded face samples (glasses vs. non-glasses). The circles and triangles are the faces with or without glasses and the filled or unfilled symbols are the labeled or unlabeled samples. In the plots of FDA, LFDA, and SELF, all the labeled points in the same class are concentrated in one point.

4.5.1 Combination of LFDA and iLPP

Semi-supervised learning is regarded as a situation between supervised learning and unsupervised learning. Similarly, SELF may also be interpreted as a dimensionality reduction method between supervised and unsupervised methods. This implies that our choice does not have to be restricted to LFDA and PCA—other powerful supervised and unsupervised methods could also be combined in a similar manner. Sugiyama (2007) showed that LFDA is a useful supervised dimensionality reduction method through experiments, so the use of LFDA in the semi-supervised dimensionality reduction method would be reasonable.

On the other hand, the performance of an unsupervised dimensionality reduction method is heavily dependent on label distributions. Clearly there are situations where PCA performs poorly (as in Figure 1(c)). An alternative choice of the unsupervised counterpart would be iLPP (see Section 3.2), which results in

$$\begin{aligned}\boldsymbol{B} &= (1-\beta)\boldsymbol{S}^{(\mathrm{lb})} + \beta\boldsymbol{S}^{(\mathrm{n})},\\ \boldsymbol{C} &= (1-\beta)\boldsymbol{S}^{(\mathrm{lw})} + \beta\boldsymbol{S}^{(\mathrm{l})}.\end{aligned}$$

Although this variant is still computationally as efficient as the original SELF, the combination of LFDA and iLPP was shown to be less useful in our experiments (see Section 5.1). This was because the global data structure is not taken into account. That is, iLPP tries to make samples in the same cluster close together, but it does not impose different clusters to be separated from each other. Therefore, several clusters may merge without any penalties and iLPP may lose the global cluster structure.

We also tested a combination of three methods—LFDA, PCA, and iLPP—with two trade-off parameters, but this did not improve the performance over the original SELF.

4.5.2 Distance Metric Learning

The performance of distance-based learning methods such as nearest neighbor classifiers depends heavily on the definition of the distances between samples. The idea of *distance metric learning* is to optimize a metric $\boldsymbol{M}$ used for computing the distances between samples (Xing et al., 2003; Goldberger et al., 2005; Globerson & Roweis, 2006; Weinberger et al., 2006):

$$\mathrm{dist}(\boldsymbol{x}_i, \boldsymbol{x}_j; \boldsymbol{M}) = (\boldsymbol{x}_i - \boldsymbol{x}_j)^\top \boldsymbol{M}(\boldsymbol{x}_i - \boldsymbol{x}_j).$$

By definition, the metric matrix $\boldsymbol{M}$ is symmetric and positive semi-definite. For this reason, metric learning is typically formulated as a *semi-definite programming* problem, which is a convex optimization problem for which the unique global solution can be obtained (Boyd & Vandenberghe, 2004; Weinberger et al., 2006).

If the rank of the $d \times d$ matrix $\boldsymbol{M}$ is constrained to r, then the distance metric learning methods are automatically causing implicit dimensionality reduction. More specifically, the symmetricity and positive semi-definiteness of the metric matrix $\boldsymbol{M}$ implies that $\boldsymbol{M}$ can be decomposed as

$$\boldsymbol{M} = \boldsymbol{T}\boldsymbol{T}^\top,$$

where $\boldsymbol{T}$ is a $d \times r$ matrix. Then $\boldsymbol{T}^\top \boldsymbol{x}_i$ could be regarded as an explicit expression of a sample $\boldsymbol{x}_i$ after dimensionality reduction. However, simultaneously reducing the dimensionality of samples and learning the distance metric is usually hard since the rank constraint is non-convex (Boyd & Vandenberghe, 2004). Thus it may not be possible to obtain the global optimal solution.

In contrast, our approach to dimensionality reduction is formulated by Eq.(1), which is not convex but which still allows us to access the global solution in terms of the range of the embedding space. This means that we can obtain the unique solution for the metric matrix by combining SELF (or any other dimensionality reduction method formulated by Eq.(1)) with a convex metric learning method (such as Weinberger et al., 2006). That is, a two-stage procedure of first reducing the dimensionality (i.e., determining the range of the embedding space) with SELF and then learning the metric in the embedding space without the rank constraint. We expect that this procedure is practically useful.

4.5.3 Kernelization

So far, we focused on linear dimensionality reduction. Using the standard *kernel trick* (Schölkopf et al., 1998), we can easily obtain a non-linear variant of SELF.

Let

$$\boldsymbol{L}^{(\mathrm{rlw})} = (1-\beta)\boldsymbol{L}^{(\mathrm{lw})} + \beta(\boldsymbol{X}^\top \boldsymbol{X})^\dagger,$$

where † denotes the Moore-Penrose generalized inverse (Albert, 1972). Recalling that $\boldsymbol{S} = \boldsymbol{X}\boldsymbol{L}\boldsymbol{X}^\top$ (see Eq.(9)), we can express the eigenvalue problem solved in SELF as

$$\boldsymbol{X}\boldsymbol{L}^{(\mathrm{rlb})}\boldsymbol{X}^\top \boldsymbol{\varphi} = \lambda \boldsymbol{X}\boldsymbol{L}^{(\mathrm{rlw})}\boldsymbol{X}^\top \boldsymbol{\varphi}. \tag{21}$$

In the derivation of this expression, we used the fact that $\boldsymbol{I}_d$ in Eq.(17) can be replaced with a projection matrix $\boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X})^\dagger \boldsymbol{X}^\top$ without essentially changing the solution when $\boldsymbol{X}^\top \boldsymbol{X}$ is not invertible.

Since $\boldsymbol{X}^\top \boldsymbol{\varphi}$ in Eq.(21) belongs to the range of $\boldsymbol{X}^\top$, it can be expressed by using some vector $\boldsymbol{\alpha} \in \mathbb{R}^n$ as follows[3]:

$$\boldsymbol{X}^\top \boldsymbol{\varphi} = \boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{\alpha} = \boldsymbol{K}\boldsymbol{\alpha},$$

where $\boldsymbol{K}$ is the $n \times n$ matrix with

$$K_{i,j} := \boldsymbol{x}_i^\top \boldsymbol{x}_j.$$

Then multiplying Eq.(21) by $\boldsymbol{X}^\top$ from the left-hand side yields

$$\boldsymbol{K}\boldsymbol{L}^{(\mathrm{rlb})}\boldsymbol{K}\boldsymbol{\alpha} = \lambda \boldsymbol{K}\boldsymbol{L}^{(\mathrm{rlw})}\boldsymbol{K}\boldsymbol{\alpha}.$$

Note that one of the properties of the Moore-Penrose generalized inverse implies that $\boldsymbol{K}\boldsymbol{L}^{(\mathrm{rlw})}\boldsymbol{K}$ can be simply computed as

$$\boldsymbol{K}\boldsymbol{L}^{(\mathrm{rlw})}\boldsymbol{K} = (1-\beta)\boldsymbol{K}\boldsymbol{L}^{(\mathrm{lw})}\boldsymbol{K} + \beta\boldsymbol{K}.$$

[3] Here, we are not equating $\boldsymbol{\varphi}$ with $\boldsymbol{X}\boldsymbol{\alpha}$, but we equate $\boldsymbol{X}^\top \boldsymbol{\varphi}$ with $\boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{\alpha}$.

When $\boldsymbol{K}\boldsymbol{L}^{(\mathrm{rlw})}\boldsymbol{K}$ is not of full rank, we may need to regularize it (Schölkopf et al., 1998), i.e., for a small positive scalar ϵ, we replace (21) with

$$\boldsymbol{K}\boldsymbol{L}^{(\mathrm{rlb})}\boldsymbol{K}\boldsymbol{\alpha} = \lambda(\boldsymbol{K}\boldsymbol{L}^{(\mathrm{rlw})}\boldsymbol{K} + \epsilon\boldsymbol{I}_n)\boldsymbol{\alpha}. \tag{22}$$

Let $\{\boldsymbol{\alpha}_k\}_{k=1}^{d}$ be the generalized eigenvectors associated with the generalized eigenvalues $\{\lambda_k\}_{k=1}^{d}$ of Eq.(22), where they are sorted and normalized as

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$$

and

$$\boldsymbol{\alpha}_k^\top(\boldsymbol{K}\boldsymbol{L}^{(\mathrm{rlw})}\boldsymbol{K} + \epsilon\boldsymbol{I}_n)\boldsymbol{\alpha}_k = 1 \quad \text{for } k = 1, 2, \ldots, d.$$

Then the embedded representation $\boldsymbol{z}$ of an original sample $\boldsymbol{x}$ can be computed in terms of $\{\boldsymbol{\alpha}_k\}_{k=1}^{r}$ as

$$\boldsymbol{z} = (\sqrt{\lambda_1}\boldsymbol{\alpha}_1 | \sqrt{\lambda_2}\boldsymbol{\alpha}_2 | \cdots | \sqrt{\lambda_r}\boldsymbol{\alpha}_r)^\top(\boldsymbol{x}_1^\top\boldsymbol{x}, \boldsymbol{x}_2^\top\boldsymbol{x}, \ldots, \boldsymbol{x}_n^\top\boldsymbol{x})^\top.$$

This implies that the data samples appear only via their *inner products*. We note that the affinity values as well as the local scaling can also be computed in terms of the inner products between data samples. Therefore, if the inner product $\boldsymbol{x}_i^\top\boldsymbol{x}_j$ is replaced by a *reproducing kernel* $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ (Aronszajn, 1950), we can obtain a non-linear variant of SELF—linear dimensionality reduction is carried out in an implicit kernel feature space (Schölkopf et al., 1998).

Beyond non-linearization, kernel SELF is also useful in the following two scenarios. The first is that the kernelized variant also allows us to reduce the dimensionality of *non-vectorial structured data* such as strings, trees, and graphs by employing kernel functions defined for such structured data (Lodhi et al., 2002; Duffy & Collins, 2002; Kashima & Koyanagi, 2002; Kondor & Lafferty, 2002; Kashima et al., 2003; Gärtner et al., 2003; Gärtner, 2003).

Another possible usage of the kernel formulation would be for computational efficiency. The size of matrices to be eigen-decomposed in the kernel formulation depends only on the number of samples, not on the input dimensionality. Thus when the number of samples is smaller than the input dimensionality, using the kernel formulation with the linear kernel could be more efficient in terms of both computation time and memory space consumption than the original formulation (see also Section 5.2).

5 Experiments

In this section, we experimentally evaluate the performance of SELF and other dimensionality reduction methods using standard classification benchmark datasets.

5.1 Benchmark Datasets

In Chapter 21 of Chapelle et al. (2006), systematic experiments were conducted for comparing various semi-supervised learning methods. The results showed that each method performs very well for a particular type of dataset, but at the same time, it tends to be poor for other kinds of datasets. Thus, the performance of semi-supervised learning methods is highly dependent on the types of the datasets and there seems to be no single best method. In contrast, although it may not be the best possible method in semi-supervised classification, the 1-*nearest neighbor classifier* has been shown to perform reasonably well across various datasets. In order to avoid any bias caused by the choice of the learning methods, we decided to use the 1-nearest neighbor classifier in our experiments.

The misclassification rate is sometimes monotonically decreasing as the dimensionality is reduced[4] (see Figure 4). In such cases, if the best dimensionality is chosen (e.g., by cross-validation), the largest dimension is mostly chosen (i.e., no dimensionality reduction). Then we may not be able to compare the performance of the dimensionality reduction methods in a meaningful way. Fixing the reduced dimensionality r to some number in advance would be a possible option for avoiding this comparison problem, but the evaluation results can strongly depend on the choice of the dimensionality. For this reason, we decided to use the *average* misclassification rate over the reduced dimensions (or equivalently the area under the classification error curve) as our error metric, which we believe to be reasonable in the current experiments.

First we use the benchmark datasets used in Chapelle et al. (2006), which consist of 9 semi-supervised learning datasets[5]. We refer to them as the *SSL* datasets. We did not test the *SSL8* and *SSL9* datasets since the *SSL8* dataset contains too many samples (n is over one million) and the *SSL9* dataset has too many dimensions (d is over ten thousand). The *SSL6* dataset contains 6 classes, while the other datasets have 2 classes. Table 1 describes the means and standard deviations of the misclassification rates over 12 repetitions. Since we encountered a numerical problem when computing LFDA, we slightly regularized it and treat SELF with $\beta = 0.001$ as LFDA.

The *cluster assumption* that the samples in the same cluster are likely to have the common label is often regarded as an important assumption for the success of semi-supervised classification (Chapelle et al., 2006). We roughly evaluated the correctness of the cluster assumption (denoted as 'CA' in Table 1) by the correct classification rate of all the training and test samples using the 1-nearest-neighbor classifier (the cases in which the label of the target point is correctly predicted by the label of the nearest sample). Note that CA is computed *before* the dimensionality reduction, so it represents the correctness of the cluster assumption for the original data samples. The larger the value of CA is, the more reliable the cluster assumption becomes.

When the number of labeled samples is 100 (see the upper half of Table 1), LFDA and PCA tend to work well in a complementary way—LFDA works well if CA is small while

[4] Even so, dimensionality reduction is still useful since a compact representation of the data can yield faster computation in the test phase.

[5] The datasets are available from '`http://www.kyb.tuebingen.mpg.de/ssl-book/`'.

PCA works well if CA is large[6]. 'SELF(0.5)' (SELF with $\beta = 0.5$) tends to compensate for the weaknesses of each method. It even outperforms both LFDA and PCA in some cases. We also tested 'SELF(CV)', where β in SELF is chosen from $\{0.001, 0.25, 0.5, 0.75, 1\}$ by using 10-fold cross-validation. The results in Table 1 show that SELF(CV) further improves the performance over SELF(0.5). These results also show that iLPP does not work so well. The combination of LFDA and iLPP (indicated by SELF'(CV) in the table) also does not perform as well as SELF(CV). We also tested the combination of LFDA, PCA, and iLPP with two trade-off parameters, but this did not further improve the performance over SELF, so we omit these details.

Figure 4 depicts the mean misclassification rates as a function of the reduced dimensionality for LFDA, PCA, and SELF(CV). This also shows that LFDA and PCA tend to work well in a complementary way and SELF(CV) tends to compensate for the weaknesses of each method. We note that the curves are almost flat for large dimensions since minor eigenvectors are deemphasized according to the square root of eigenvalues (see Eq.(7)).

When the number of labeled samples is only 10 (see the lower half of Table 1), the performance differences among the methods shrink but SELF(CV) is still slightly better than the other methods.

We also conducted similar experiments using the *IDA* datasets[7] (Rätsch et al., 2001), which consist of supervised classification tasks. We randomly extracted labeled and unlabeled samples from the pool of all samples, testing $n' = 100$ and 30. The results are summarized in Table 2, showing that SELF(CV) still compares favorably with the alternative methods. From these results, we demonstrated that SELF(CV) performs reasonably well across various datasets.

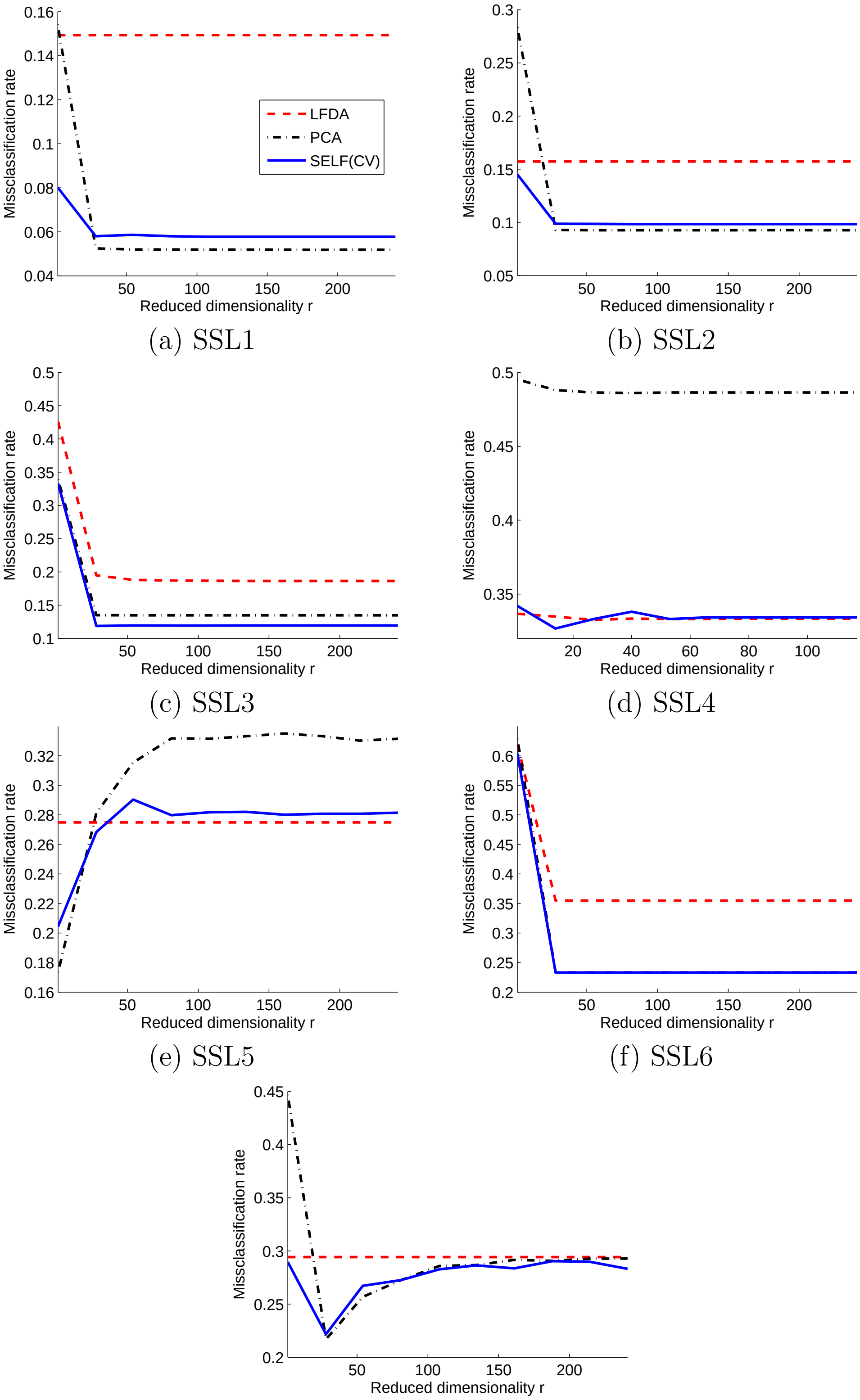
5.2 Document Classification

Here, we apply the proposed dimensionality technique, SELF, to real-world document classification tasks and evaluate its performance. We used the datasets in the *Technion Repository of Text Categorization*[8] (TechTC; Davidov et al., 2004). The TechTC repository contains 295 binary document classification tasks. Each task contains a few hundred documents with category labels and a document is expressed by a *bag-of-words vector* of term frequencies, which has an entry in the vector corresponding to each word in the dictionary and its number of occurrences in the document. Following convention (Joachims, 2002), we multiply the term frequency by the logarithm of the inverse ratio of the documents containing the corresponding word. The feature vectors constructed in this way is called the *term frequency-inverse document frequency* (TFIDF) vector and TFIDF is widely used as a standard feature extraction scheme in the document analysis community.

[6] The success of PCA depends, of course, on the scaling of the data (see Figure 1 again). However, for the SSL datasets, it was shown through extensive experiments that PCA works well on the whole (see Chapter 21 of Chapelle et al., 2006). This implies that the scaling of the data is well-conditioned for PCA in the SSL datasets.

[7] The datasets are available from '`http://ida.first.fhg.de/projects/bench/benchmarks.htm`'.

[8] The datasets are available from '`http://techtc.cs.technion.ac.il/techtc300/techtc300.html`'.



(a) SSL1

(b) SSL2

(c) SSL3

(d) SSL4

(e) SSL5

(f) SSL6

(g) SSL7

Figure 4: Mean misclassification rates for the SSL datasets as a function of the reduced dimensionality r when $n = 100$.

Table 1: Misclassification rates for the SSL datasets averaged over reduced dimensions. The numbers in the parentheses are the standard deviations over repetitions. For each dataset, the best method with the smallest mean and the methods judged to have no significant difference from the best method based on the *t-test* at the significance level 5% are in bold. ‘Dim’, ‘Lab’, ‘Unlab’, ‘Rep’, and ‘CA’ denote the dimensionality of original samples, the number of labeled samples, the number of unlabeled samples, the number of repetitions, and the correctness of the cluster assumption (see the text for details), respectively. SELF(CV) denotes SELF with β chosen by cross-validation. SELF’ denotes the combination of LFDA and iLPP in a similar manner.

Data	Dim	Lab	Unlab	Rep	CA	LFDA	SELF(0.5)	PCA	SELF(CV)	iLPP	SELF’(CV)
SSL1	241	100	1400	12	0.98	14.9 (1.8)	**6.0 (1.3)**	**6.2 (1.1)**	**6.0 (1.4)**	27.4 (1.4)	28.4 (2.6)
SSL2	241	100	1400	12	0.97	15.7 (0.9)	**9.6 (1.1)**	11.2 (0.8)	**10.3 (2.4)**	24.1 (2.2)	21.9 (1.9)
SSL3	241	100	1400	12	1.00	21.1 (3.9)	**14.3 (1.8)**	15.5 (1.0)	**14.1 (1.4)**	18.0 (2.4)	18.5 (2.4)
SSL4	117	100	300	12	0.58	**33.4 (3.5)**	36.6 (2.4)	48.7 (2.4)	**33.4 (3.7)**	46.7 (1.7)	**36.0 (4.7)**
SSL5	241	100	1400	12	0.64	**27.5 (2.3)**	**27.2 (2.3)**	31.0 (1.9)	**27.3 (2.9)**	37.0 (1.3)	35.3 (1.9)
SSL6	241	100	1400	12	0.98	38.1 (1.5)	35.4 (2.4)	**27.3 (2.7)**	**27.0 (2.7)**	35.2 (1.7)	36.9 (3.2)
SSL7	241	100	1400	12	0.68	29.4 (2.4)	**29.1 (2.4)**	29.3 (1.6)	**27.7 (1.4)**	32.0 (0.9)	32.8 (1.5)
# Bests						2	5	2	7	0	1
SSL1	241	10	1490	12	0.98	**22.9 (5.1)**	26.3 (6.1)	**19.2 (4.2)**	**22.3 (5.4)**	45.9 (2.3)	48.5 (2.4)
SSL2	241	10	1490	12	0.97	**22.3 (3.0)**	**21.3 (2.9)**	25.8 (4.2)	**21.5 (2.5)**	31.2 (7.5)	**21.4 (0.8)**
SSL3	241	10	1490	12	1.00	**42.7 (2.9)**	**42.9 (3.0)**	**42.7 (4.2)**	43.6 (3.2)	**40.4 (4.1)**	**41.0 (5.2)**
SSL4	117	10	390	12	0.58	**47.3 (2.9)**	**47.7 (2.7)**	49.9 (2.2)	**48.3 (3.3)**	**49.5 (2.5)**	**48.5 (1.9)**
SSL5	241	10	1490	12	0.64	45.4 (4.4)	45.4 (4.4)	**36.3 (5.5)**	**40.2 (6.9)**	41.2 (3.3)	44.5 (3.6)
SSL6	241	10	1490	12	0.98	**67.7 (4.6)**	**67.0 (4.0)**	**67.7 (4.1)**	**67.6 (4.6)**	71.4 (4.0)	73.7 (2.9)
SSL7	241	10	1490	12	0.68	43.6 (5.2)	43.6 (5.2)	**38.9 (5.7)**	**40.1 (7.1)**	**40.3 (4.2)**	**42.7 (5.3)**
# Bests						5	4	5	6	3	4

Table 2: Misclassification rates for the IDA datasets averaged over reduced dimensions.

Data	Dim	Lab	Unlab	Rep	CA	LFDA	SELF(0.5)	PCA	SELF(CV)	iLPP	SELF'(CV)
banana	2	100	2000	100	0.87	**27.0 (2.6)**	**26.6 (2.1)**	**26.4 (1.9)**	**26.5 (2.1)**	**26.4 (1.9)**	**26.5 (2.0)**
b-cancer	9	100	77	100	0.68	**34.5 (4.4)**	**34.4 (4.4)**	**34.4 (4.1)**	**34.3 (4.3)**	**34.8 (4.0)**	**34.7 (4.1)**
diabetes	8	100	300	100	0.70	**32.7 (2.8)**	**33.0 (2.7)**	34.4 (2.7)	**33.0 (2.7)**	34.4 (2.6)	**33.2 (2.7)**
f-solar	9	100	400	100	0.63	**39.5 (5.1)**	**40.1 (5.1)**	**40.1 (5.2)**	**39.7 (5.2)**	**39.7 (5.4)**	**39.5 (5.4)**
german	20	100	300	100	0.69	**31.2 (2.9)**	**31.2 (3.0)**	33.7 (2.8)	**31.5 (2.9)**	33.7 (2.6)	32.1 (3.0)
heart	13	100	100	100	0.77	**22.8 (2.9)**	**22.6 (2.8)**	24.1 (2.7)	**23.1 (2.8)**	**23.4 (2.9)**	**23.1 (2.8)**
image	18	100	1010	20	0.81	**17.2 (1.3)**	18.8 (1.3)	19.9 (1.5)	17.8 (1.7)	18.8 (2.1)	**16.6 (1.3)**
ringnorm	20	100	2000	100	0.71	28.1 (1.9)	28.9 (1.9)	29.1 (1.6)	28.1 (1.8)	**27.1 (1.6)**	**27.6 (1.8)**
splice	60	100	2000	20	0.71	29.9 (3.5)	**27.8 (3.5)**	30.8 (2.3)	**27.7 (3.0)**	42.1 (1.9)	**30.1 (4.6)**
thyroid	5	100	75	100	0.96	**4.8 (2.0)**	**5.3 (2.1)**	5.5 (2.1)	**5.0 (1.9)**	5.9 (2.1)	**5.1 (2.0)**
titanic	3	100	2000	100	0.68	**33.2 (11.9)**	**33.2 (11.9)**	**33.2 (11.9)**	**33.2 (11.9)**	40.0 (12.3)	37.4 (12.5)
twonorm	20	100	2000	100	0.94	4.8 (1.3)	4.5 (1.2)	**4.1 (1.1)**	**4.3 (1.1)**	**4.0 (1.0)**	4.5 (1.2)
waveform	21	100	2000	100	0.85	15.5 (1.4)	14.5 (1.5)	**14.1 (1.4)**	**14.2 (1.7)**	**13.8 (1.4)**	14.4 (1.9)
# Bests						9	9	6	11	7	9
banana	2	30	2000	100	0.87	31.1 (4.0)	**30.6 (3.5)**	**30.0 (4.1)**	**29.6 (3.4)**	**30.0 (4.1)**	**30.3 (3.6)**
b-cancer	9	30	77	100	0.67	**36.1 (6.4)**	**35.4 (6.2)**	**36.1 (6.3)**	**35.6 (6.4)**	**36.1 (5.8)**	**36.0 (6.2)**
diabetes	8	30	300	100	0.70	**35.0 (4.8)**	**34.7 (4.3)**	36.0 (4.1)	**34.9 (4.4)**	35.9 (3.7)	**35.1 (4.2)**
f-solar	9	30	400	100	0.63	**41.5 (5.5)**	42.6 (5.4)	42.7 (5.1)	42.0 (5.4)	**40.6 (5.3)**	**40.4 (5.4)**
german	20	30	300	100	0.69	36.6 (4.7)	**32.8 (3.8)**	35.6 (4.1)	**33.9 (4.3)**	36.0 (4.0)	34.5 (4.1)
heart	13	30	100	100	0.76	25.6 (5.4)	**23.7 (4.9)**	**24.4 (4.1)**	**24.6 (4.7)**	**24.2 (4.0)**	**24.9 (4.2)**
image	18	30	1010	20	0.81	**24.5 (3.8)**	**26.2 (3.2)**	27.6 (3.8)	**26.0 (3.8)**	27.9 (4.2)	**24.5 (3.5)**
ringnorm	20	30	2000	100	0.70	35.5 (4.2)	34.0 (3.7)	33.8 (2.8)	33.1 (3.2)	**31.1 (3.3)**	32.5 (3.8)
splice	60	30	2000	20	0.71	**34.0 (3.1)**	**33.1 (3.1)**	**34.6 (2.5)**	**33.2 (2.7)**	45.2 (2.5)	39.9 (4.6)
thyroid	5	30	75	100	0.94	9.9 (4.5)	**8.3 (4.1)**	**8.4 (3.6)**	**8.7 (4.2)**	**8.2 (3.3)**	**8.9 (4.2)**
titanic	3	30	2000	100	0.68	**33.9 (12.1)**	**34.0 (12.2)**	**34.0 (12.1)**	**33.9 (12.1)**	40.8 (12.3)	37.5 (12.9)
twonorm	20	30	2000	100	0.94	15.3 (6.5)	6.3 (2.0)	**4.3 (1.3)**	6.7 (3.9)	**4.2 (1.3)**	6.9 (3.8)
waveform	21	30	2000	100	0.85	27.5 (4.3)	16.6 (3.1)	**15.6 (2.3)**	16.9 (3.2)	**15.3 (2.2)**	17.8 (3.6)
# Bests						6	9	8	9	8	7

The TFIDF vector $\boldsymbol{x}$ usually has a large number of dimensions. In our experiments, its dimensionality ranged from thousands to tens of thousands (depending on the tasks since we removed the entries of zero occurrences for all of the documents). In general it is not possible to directly solve eigenvalue problems in such high dimensional spaces. Here, we used the kernel formulation (see Section 4.5.3; we used the linear kernel so that SELF is still a linear dimensionality reduction), relying on the number of samples being much smaller than the input dimensionality in our experiments.

We compare the performance of 'Plain' (without dimensionality reduction), LFDA, PCA, 'SELF(0.5)' (SELF with $\beta = 0.5$), and 'SELF(CV)' (SELF with β chosen by using 5-fold cross-validation). In each method, the dimensionality of the reduced space r is chosen by using 5-fold CV from[9] $\{1, 2, \ldots, 10\}$. For each dataset, we consider 4 configurations with different degrees of supervision. Given n document samples, we randomly choose 20%, 40%, 60%, and 80% of them as the training data and the rest are treated as unlabeled data. The 1-nearest neighbor method was again used to evaluate the classification accuracy of the unlabeled samples. For each dataset and each training sample configuration, the experiments were repeated 100 times with randomly selected training samples.

The means and standard deviations of the misclassification rates are summarized in Table 3. The table shows that all of the dimensionality reduction methods perform better than Plain, so dimensionality reduction evidently contributes to improving the accuracy of document classification. Among these methods, SELF consistently works better than LFDA and PCA.

The mean value of β in SELF(CV) for the four configurations, 20%, 40%, 60%, and 80%, are 0.57, 0.52, 0.48, and 0.46, respectively. This shows that, as the degree of supervision increases, the value of β decreases and therefore SELF approaches LFDA. This agrees well with our intuition. However, since all of the values are rather close to 0.5 in this experiment, SELF(0.5) tends to perform slightly better (and is computationally more efficient) than SELF(CV). It is also intuitive that LFDA tends to outperform PCA as the degree of supervision increases.

Overall, SELF—a combination of LFDA and PCA—was shown to be a useful dimensionality reduction method in practical document classification tasks.

6 Conclusions, Discussion, and Future Work

Our approach to dimensionality reduction in the current work is called the *filter* approach, meaning that the dimensionality reduction procedure is independent of subsequent classification algorithms (Guyon & Elisseeff, 2003). Our experimental results showed that the proposed method, SELF, works well when it is combined with the 1-nearest-neighbor classifier. On the other hand, it is also important to explore *wrapper* methods (Ko-

[9] We set the upper limit of r to 10 mainly for computational reasons. However, as shown later, the value of r chosen by CV is typically less than 10, so this restriction does not cause a serious performance change.

Table 3: Means and standard deviations of the misclassification rates of the document classification tasks over 295 datasets for 100 runs each (i.e., 29, 500 total trials). The mean value of the reduced dimensionality r chosen by CV was also included in the table.

$n' = 0.2n$	Plain	LFDA	SELF(0.5)	PCA	SELF(CV)
Mean error	20.8	18.4	16.3	18.5	16.4
Std. error	3.4	2.3	2.1	1.7	1.7
# Bests	41	76	252	46	242
Mean chosen r	—	2.2	3.3	4.6	3.5
$n' = 0.4n$					
Mean error	20.8	15.0	13.8	17.0	13.9
Std. error	3.8	1.6	1.5	1.6	1.5
# Bests	10	115	239	34	220
Mean chosen r	—	3.0	4.0	5.3	3.9
$n' = 0.6n$					
Mean error	21.1	13.9	12.7	16.2	12.7
Std. error	3.4	1.5	1.6	1.7	1.6
# Bests	8	126	235	50	242
Mean chosen r	—	3.5	4.2	5.7	3.9
$n' = 0.8n$					
Mean error	21.5	13.6	12.0	15.6	12.1
Std. error	2.9	2.0	2.2	2.4	2.3
# Bests	9	134	245	72	240
Mean chosen r	—	3.8	4.1	6.0	3.8

havi & John, 1997) for semi-supervised dimensionality reduction, which explicitly take the properties of subsequent classification algorithms into account. A wrapper approach would be particularly useful in semi-supervised learning scenarios since the performance of elaborate semi-supervised learning methods is highly dependent on the reliability of the assumptions on the unlabeled samples, such as cluster or manifold structure (Chapelle et al., 2006).

We showed in Section 4.5.3 that a non-linear variant of SELF can be created by employing the standard kernel trick. However, a kernelized SELF shares the common difficulty of kernel methods, the question of how to choose the kernel functions. This must be investigated in the context of semi-supervised dimensionality reduction. In future work, we will explore semi-supervised dimensionality reduction of structured data using kernel SELF.

In SELF, we linearly combined the eigenvalue problems of LFDA and PCA since this approach allows us to maintain the computational advantages of LFDA and PCA. This approach was demonstrated to be useful through our experiments in Section 5. Although we examined some properties of the combined method in Section 4.3, it is important to provide a better understanding of the mechanism of the proposed method. Also, our proposed approach for combining LFDA and PCA is not the only possibility. A future direction would be to explore other ways to combine supervised and unsupervised methods for further performance improvement.

An advantage of SELF is that its solution can be obtained analytically by solving a generalized eigenvalue problem. When the number of samples is very large, solving the eigenvalue problem by using the algorithm in Figure 2 would be still computationally tractable as long as the input dimensionality is not too high. On the other hand, when the input dimensionality is very high, the kernel formulation with the linear kernel (see Section 4.5.3) is still computationally tractable as long as the number of samples is moderate (as demonstrated by the document classification experiments in Section 5.2). However, when the number and dimensionality of the samples are both very large, a naive implementation may not be computationally tractable. Thus an important future work along this line is to further investigate the computational aspects of SELF and develop efficient algorithms that can deal with high-dimensional and large-scale datasets, perhaps by utilizing the sparsity of the data matrix or the kernel matrix.

A remaining important issue to be discussed, which is common to all semi-supervised learning techniques, is how to optimize the tuning parameters. We may simply use cross-validation for this purpose, but that approach has two potential problems. The first problem is that the number of labeled samples is typically small in semi-supervised learning scenarios, so cross-validation is not reliable (Chapelle et al., 2006). Fortunately, our experiments showed that SELF is not very sensitive to the choice of the trade-off parameter β in small sample cases, but there is still room for improvement. The second problem is that labeled samples and unlabeled samples can have different (input) distributions. Such a situation is referred to as *covariate shift* (Shimodaira, 2000; Quiñonero-Candela et al., 2009) and ordinary cross-validation is known to be significantly biased in such situations (Zadrozny, 2004), while *importance-weighted* cross-validation is unbiased under covariate shift (Sugiyama et al., 2007). In future work, we will investigate how such covariate shift adaptation techniques can be used in the context of semi-supervised dimensionality reduction.

The properties of a family of linear discriminant analysis algorithms were studied in Ye (2005; 2008) and Loog (2007; 2008), but the methods discussed in these papers do not take the locality of the data into account as LFDA does. Therefore our current work is essentially different from these existing methods. Another alternative to our approach involves regularized linear discriminant analysis methods for semi-supervised dimensionality reduction based on LPP (Cai et al., 2007) or manifold regularization (Belkin et al., 2006; Song et al., 2008). These methods suffer from the weakness of the original FDA, i.e., the maximum dimension of the reduced subspace is dominated by the number of classes. In contrast, our method offers advantages for classification tasks with rather small numbers of classes. A relevant dimensionality reduction method has also been proposed in the context of semi-supervised clustering (Zhang et al., 2007). However, the locality of the data is still not addressed. Recently, a non-linear dimensionality reduction method based on a neural network has been proposed (Hinton & Salakhutdinov, 2006). However, neural-network-based methods are prone to suffer from local optimality because of the non-convexity of optimization. Also, this optimization is usually carried out via a gradient method and is computationally inefficient. Therefore another important research direction is to extend such neural-network-based methods to semi-supervised setups and compare their accuracy and computational efficiency with discriminant-analysis-based methods.

Acknowledgments

The authors would like to thank the members of the T-PRIMAL (Tokyo PRobabilistic Inference and MAchine Learning) group for their fruitful comments and insights. Our gratitude also goes to the anonymous reviewers for their valuable comments. MS acknowledges financial support from MEXT (20680007 and 1907364) and the Tateishi Science and Technology Foundation.

References

Albert, A. (1972). *Regression and the Moore-Penrose pseudoinverse.* New York and London: Academic Press.

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

Bai, Z., Demmel, J., Dongarra, J., Ruhe, A., & van der Vorst, H. (Eds.). (2000). *Templates for the solution of eigenvalue problems: A practical guide.* Philadelphia, PA, USA: Society for Industrial and Applied Mathematics.

Belkin, M., & Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, *15*, 1373–1396.

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

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization.* Cambridge: Cambridge University Press.

Cai, D., He, X., & Han, J. (2007). Semi-supervised discriminant analysis. *Proceedings of the IEEE International Conference on Computer Vision* (pp. 1–7). Rio de Janeiro, Brazil.

Chapelle, O., Schölkopf, B., & Zien, A. (Eds.). (2006). *Semi-supervised learning.* Cambridge: MIT Press.

Chung, F. R. K. (1997). *Spectral graph theory.* Providence, R.I.: American Mathematical Society.

Davidov, D., Gabrilovich, E., & Markovitch, S. (2004). Parameterized generation of labeled datasets for text categorization based on a hierarchical directory. *The 27th Annual International ACM SIGIR Conference* (pp. 250–257). Sheffield, UK.

Duffy, N., & Collins, M. (2002). Convolution kernels for natural language. *Advances in Neural Information Processing Systems 14* (pp. 625–632). Cambridge, MA: MIT Press.

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

Friedman, J. H. (1989). Regularized discriminant analysis. *Journal of the American Statistical Association*, *84*, 165–175.

Fukunaga, K. (1990). *Introduction to statistical pattern recognition.* Boston: Academic Press, Inc. Second edition.

Gärtner, T. (2003). A survey of kernels for structured data. *SIGKDD Explorations*, *5*, S268–S275.

Gärtner, T., Flach, P., & Wrobel, S. (2003). On graph kernels: Hardness results and efficient alternatives. *Proceedings of the Sixteenth Annual Conference on Computational Learning Theory* (pp. 129–143).

Globerson, A., & Roweis, S. (2006). Metric learning by collapsing classes. *Advances in Neural Information Processing Systems 18* (pp. 451–458). Cambridge, MA: MIT Press.

Goldberger, J., Roweis, S., Hinton, G., & Salakhutdinov, R. (2005). Neighbourhood components analysis. *Advances in Neural Information Processing Systems 17* (pp. 513–520). Cambridge, MA: MIT Press.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, *3*, 1157–1182.

He, X., & Niyogi, P. (2004). Locality preserving projections. *Advances in Neural Information Processing Systems 16* (pp. 153–160). Cambridge, MA: MIT Press.

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, *313*, 504–507.

Joachims, T. (2002). *Learning to classify text using support vector machines: Methods, theory and algorithms.* Boston: Kluwer Academic Publishers.

Jolliffe, I. T. (1986). *Principal component analysis.* New York: Springer-Verlag.

Kashima, H., & Koyanagi, T. (2002). Kernels for semi-structured data. *Proceedings of the Nineteenth International Conference on Machine Learning* (pp. 291–298). San Francisco, CA: Morgan Kaufmann.

Kashima, H., Tsuda, K., & Inokuchi, A. (2003). Marginalized kernels between labeled graphs. *Proceedings of the Twentieth International Conference on Machine Learning* (pp. 321–328). San Francisco, CA: Morgan Kaufmann.

Kohavi, R., & John, G. (1997). Wrappers for feature selection. *Artificial Intelligence*, *97*, 273–324.

Kondor, R. I., & Lafferty, J. (2002). Diffusion kernels on graphs and other discrete input spaces. *Proceedings of the Nineteenth International Conference on Machine Learning* (pp. 315–322).

Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., & Watkins, C. (2002). Text classification using string kernels. *Journal of Machine Learning Research*, *2*, 419–444.

Loog, M. (2007). A complete characterization of a family of solutions to a generalized fisher criterion. *Journal of Machine Learning Research*, *8*, 2121–2123.

Loog, M. (2008). On the equivalence of linear dimensionality-reducing transformations. *Journal of Machine Learning Research*, *9*, 2489–2490.

Mika, S., Rätsch, G., Weston, J., Schölkopf, B., Smola, A., & Müller, K.-R. (2003). Constructing descriptive and discriminative nonlinear features: Rayleigh coefficients in kernel feature spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *25*, 623–628.

Quiñonero-Candela, J., Sugiyama, M., Schwaighofer, A., & Lawrence, N. (Eds.). (2009). *Dataset shift in machine learning*. Cambridge, MA: MIT Press.

Rätsch, G., Onoda, T., & Müller, K.-R. (2001). Soft margins for adaboost. *Machine Learning*, *42*, 287–320.

Roweis, S., & Saul, L. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, *290*, 2323–2326.

Schölkopf, B., Smola, A., & Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, *10*, 1299–1319.

Shimodaira, H. (2000). Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, *90*, 227–244.

Song, Y., Nie, F., Zhang, C., & Xiang, S. (2008). A unified framework for semi-supervised dimensionality reduction. *Pattern Recognition*, *41*, 2789–2799.

Sugiyama, M. (2007). Dimensionality reduction of multimodal labeled data by local Fisher discriminant analysis. *Journal of Machine Learning Research*, *8*, 1027–1061.

Sugiyama, M., Ide, T., Nakajima, S., & Sese, J. (2008). Semi-supervised local Fisher discriminant analysis for dimensionality reduction. *Advances in Knowledge Discovery and Data Mining* (pp. 333–344). Berlin: Springer.

Sugiyama, M., Krauledat, M., & Müller, K.-R. (2007). Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, *8*, 985–1005.

Tenenbaum, J. B., de Silva, V., & Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, *290*, 2319–2323.

Weinberger, K., Blitzer, J., & Saul, L. (2006). Distance metric learning for large margin nearest neighbor classification. *Advances in Neural Information Processing Systems 18* (pp. 1473–1480). Cambridge, MA: MIT Press.

Xing, E. P., Ng, A. Y., Jordan, M. I., & Russell, S. (2003). Distance metric learning with application to clustering with side-information. *Advances in Neural Information Processing Systems 15* (pp. 505–512). Cambridge, MA: MIT Press.

Ye, J. (2005). Characterization of a family of algorithms for generalized discriminant analysis on undersampled problems. *Journal of Machine Learning Research*, *6*, 483–502.

Ye, J. (2008). Comments on the complete characterization of a family of solutions to a generalized fisher criterion. *Journal of Machine Learning Research*, *9*, 517–519.

Zadrozny, B. (2004). Learning and evaluating classifiers under sample selection bias. *Proceedings of the Twenty-First International Conference on Machine Learning* (pp. 903–910). New York, NY: ACM Press.

Zelnik-Manor, L., & Perona, P. (2005). Self-tuning spectral clustering. *Advances in Neural Information Processing Systems 17* (pp. 1601–1608). Cambridge, MA: MIT Press.

Zhang, D., Zhou, Z.-H., & Chen, S. (2007). Semi-supervised dimensionality reduction. *Proceedings of the 7th SIAM International Conference on Data Mining* (pp. 629–634). Minneapolis, MN, USA.