Semi-Supervised Local Fisher Discriminant Analysis for Dimensionality Reduction

Masashi Sugiyama[1], Tsuyoshi Idé[2], Shinichi Nakajima[3], and Jun Sese[4]

[1] Tokyo Institute of Technology, Tokyo, Japan
sugi@cs.titech.ac.jp, http://sugiyama-www.cs.titech.ac.jp/~sugi/
[2] IBM Research, Kanagawa, Japan
goodidea@jp.ibm.com
[3] Nikon Corporation, Saitama, Japan
nakajima.s@nikon.co.jp
[4] Ochanomizu University, Tokyo, 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 due to 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 has an analytic form of the globally optimal solution and it can be computed based on eigendecompositions. Therefore, the proposed method is computationally reliable and efficient. We show the effectiveness of the proposed method through extensive simulations with benchmark data sets.

1 Introduction

The goal of dimensionality reduction is to obtain a low-dimensional representation of high-dimensional data samples while preserving most of 'intrinsic information' contained in the original data. Once dimensionality reduction is carried out appropriately, the compact representation of the data can be used for various succeeding tasks such as visualization and classification.

In supervised learning scenarios where data samples are accompanied with class labels, *Fisher discriminant analysis* (FDA) [1] is a popular dimensionality reduction method. FDA seeks an embedding transformation such that between-class scatter is maximized and within-class scatter is minimized. FDA works very well if samples in each class are Gaussian with the common covariance structure. However, it tends to give undesired results if samples in a class form several separate clusters or there exist outliers [1]. To overcome this drawback, *local FDA (LFDA)* has been proposed [2], which localizes the between-class and within-class scatter matrices. LFDA works well even when within-class multimodality or outliers exist. Furthermore, LFDA overcomes critical limitation of original FDA in dimensionality reduction—the dimension of the FDA embedding space

should be less than the number of classes [1], while LFDA does not suffer from this restriction in general.

However, the performance of LFDA (and all other supervised dimensionality reduction methods) tend to be degraded when only a small number of labeled samples are available. Thus, the supervised methods overfit embedding spaces to the labeled samples. In such cases, it is effective to make use of *unlabeled* samples which are often available abundantly, i.e., *semi-supervised learning*. The book [3] showed through extensive simulations that *principal component analysis* (PCA), which is an unsupervised dimensionality reduction method for preserving the global data structure, works moderately well in semi-supervised learning scenarios.

Although PCA is reported to work well, it may not be the best choice in semi-supervised learning due to its unsupervised nature. In this paper, we propose a new semi-supervised dimensionality reduction method which smoothly bridges LFDA and PCA so that we can control our reliance on the global structure of unlabeled samples and information brought by (a small number of) labeled samples. We experimentally show 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 eigendecompositions. Therefore, SELF is still computationally efficient and reliable.

2 Preliminaries

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

2.1 Formulation

Let $\boldsymbol{x}_i \in \mathbb{R}^d$ $(i = 1, 2, \ldots, n)$ be d-dimensional samples and let $\boldsymbol{X} \equiv (\boldsymbol{x}_1|\boldsymbol{x}_2|\cdots|\boldsymbol{x}_n)$. 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. We focus on linear dimensionality reduction, i.e., using a $d \times r$ transformation matrix $\boldsymbol{T}$, an embedded representation $\boldsymbol{z}$ of a sample $\boldsymbol{x}$ is obtained as

$$\boldsymbol{z} = \boldsymbol{T}^\top \boldsymbol{x}, \tag{1}$$

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

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

$$\boldsymbol{T}_{OPT} \equiv \underset{\boldsymbol{T} \in \mathbb{R}^{d \times r}}{\operatorname{argmax}} \left[\operatorname{tr} \left(\boldsymbol{T}^\top \overline{\boldsymbol{C}} \boldsymbol{T} (\boldsymbol{T}^\top \underline{\boldsymbol{C}} \boldsymbol{T})^{-1} \right) \right]. \tag{2}$$

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:

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

We assume that the generalized eigenvalues are sorted as $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$ and the generalized eigenvectors are normalized as $\boldsymbol{\varphi}_k^\top \underline{\boldsymbol{C}} \boldsymbol{\varphi}_k = 1$ for $k = 1, 2, \ldots, d$. Note that this normalization is often automatically carried out by an eigensolver. Then a solution $\boldsymbol{T}_{OPT}$ is analytically given as $(\boldsymbol{\varphi}_1 | \boldsymbol{\varphi}_2 | \cdots | \boldsymbol{\varphi}_r)$ (e.g., [1]):.

When addressing dimensionality reduction problems, we often face with a matrix of the following pairwise form [2]:

$$\boldsymbol{S} \equiv \frac{1}{2} \sum_{i,j=1}^{n} W_{i,j} (\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top, \tag{4}$$

where $\boldsymbol{W}$ is some n-dimensional matrix. Let $\boldsymbol{D}$ be the n-dimensional diagonal matrix with $D_{i,i} \equiv \sum_{j=1}^{n} W_{i,j}$, and let $\boldsymbol{L} \equiv \boldsymbol{D} - \boldsymbol{W}$. Then $\boldsymbol{S}$ is expressed as $\boldsymbol{S} = \boldsymbol{X} \boldsymbol{L} \boldsymbol{X}^\top$, which is positive semi-definite.

2.2 Principal Component Analysis (PCA)

A fundamental unsupervised dimensionality reduction method is *principal component analysis* (PCA).

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

$$\boldsymbol{S}^{(t)} \equiv \sum_{i=1}^{n} (\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^\top, \tag{5}$$

where $\boldsymbol{\mu} \equiv \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{x}_i$. The PCA transformation matrix $\boldsymbol{T}_{PCA}$ is defined as

$$\boldsymbol{T}_{PCA} \equiv \underset{\boldsymbol{T} \in \mathbb{R}^{d \times r}}{\operatorname{argmax}} \left[\operatorname{tr} \left(\boldsymbol{T}^\top \boldsymbol{S}^{(t)} \boldsymbol{T} (\boldsymbol{T}^\top \boldsymbol{T})^{-1} \right) \right]. \tag{6}$$

That is, PCA seeks a transformation matrix $\boldsymbol{T}$ such that scatter in the embedding space is maximized. A solution $\boldsymbol{T}_{PCA}$ is given with $\overline{\boldsymbol{C}} = \boldsymbol{S}^{(t)}$ and $\underline{\boldsymbol{C}} = \boldsymbol{I}_d$, where $\boldsymbol{I}_d$ is the identity matrix on $\mathbb{R}^d$.

2.3 Locality-Preserving Projection (LPP)

Another useful unsupervised dimensionality reduction technique is *locality-preserving projection* (LPP) [4].

Let $\boldsymbol{A}$ be the *affinity matrix*, i.e., the n-dimensional square matrix with $A_{i,j}$ being the affinity between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. We assume that $A_{i,j} \in [0, 1]$; $A_{i,j}$ 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 $\boldsymbol{A}$, e.g., based on nearest neighbors or the heat kernel Through the paper, we use the *local scaling heuristic* [5] as the definition of the affinity matrix $\boldsymbol{A}$, i.e.,

$$A_{i,j} = \exp \left(- \frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{\sigma_i \sigma_j} \right). \tag{7}$$

σ_i is 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 $k = 7$ has shown to be useful through extensive simulations [5, 2].

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

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

where $\boldsymbol{D}^{(n)}$ is the n-dimensional diagonal matrix with $D_{i,i}^{(n)} \equiv \frac{1}{n}\sum_{j=1}^{n} A_{i,j}$ and $W_{i,j}^{(l)} \equiv \frac{1}{n}A_{i,j}$. The LPP transformation matrix $\boldsymbol{T}_{LPP}$ is defined as

$$\boldsymbol{T}_{LPP} \equiv \underset{\boldsymbol{T}\in\mathbb{R}^{d\times r}}{\operatorname{argmax}} \left[\operatorname{tr}\left(\boldsymbol{T}^\top \boldsymbol{S}^{(n)}\boldsymbol{T}(\boldsymbol{T}^\top \boldsymbol{S}^{(l)}\boldsymbol{T})^{-1}\right)\right]. \tag{9}$$

That is, 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 $\mathbb{R}^r$. Thus, LPP tends to preserve the local structure of the data. A solution $\boldsymbol{T}_{LPP}$ is given with $\overline{\boldsymbol{C}} = \boldsymbol{S}^{(n)}$ and $\underline{\boldsymbol{C}} = \boldsymbol{S}^{(l)}$.

2.4 Fisher Discriminant Analysis (FDA)

A popular supervised dimensionality reduction technique is *Fisher discriminant analysis* (FDA) [1]. 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 $m \in \{1, 2, \ldots, c\}$.

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

$$\boldsymbol{S}^{(b)} \equiv \sum_{m=1}^{c} n'_m(\boldsymbol{\mu}_m - \boldsymbol{\mu})(\boldsymbol{\mu}_m - \boldsymbol{\mu})^\top, \quad \boldsymbol{S}^{(w)} \equiv \sum_{m=1}^{c}\sum_{i:y_i=m}(\boldsymbol{x}_i - \boldsymbol{\mu}_m)(\boldsymbol{x}_i - \boldsymbol{\mu}_m)^\top, \tag{10}$$

where $\boldsymbol{\mu}_m \equiv \frac{1}{n'_m}\sum_{i:y_i=m}\boldsymbol{x}_i$. The FDA transformation matrix $\boldsymbol{T}_{FDA}$ is defined as

$$\boldsymbol{T}_{FDA} \equiv \underset{\boldsymbol{T}\in\mathbb{R}^{d\times r}}{\operatorname{argmax}} \left[\operatorname{tr}\left(\boldsymbol{T}^\top \boldsymbol{S}^{(b)}\boldsymbol{T}(\boldsymbol{T}^\top \boldsymbol{S}^{(w)}\boldsymbol{T})^{-1}\right)\right]. \tag{11}$$

That is, FDA seeks a transformation matrix $\boldsymbol{T}$ such that between-class scatter is maximized and within-class scatter is minimized in the embedding space $\mathbb{R}^r$. A solution $\boldsymbol{T}_{FDA}$ is given with $\overline{\boldsymbol{C}} = \boldsymbol{S}^{(b)}$ and $\underline{\boldsymbol{C}} = \boldsymbol{S}^{(w)}$.

The between-class scatter matrix $\boldsymbol{S}^{(b)}$ has at most rank $c-1$ [1]. This implies that FDA allows us to obtain at most $c-1$ meaningful features; the remaining features found by FDA are arbitrary in the null space of $\boldsymbol{S}^{(b)}$. This is an essential limitation of FDA in dimensionality reduction.

2.5 Local Fisher Discriminant Analysis (LFDA)

Local Fisher discriminant analysis (LFDA) is a supervised dimensionality reduction method [2] which overcomes vulnerability of original FDA against within-class multimodality or outliers [1].

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

$$\boldsymbol{S}^{(lb)} \equiv \sum_{i,j=1}^{n'} \frac{W_{i,j}^{(lb)}}{2}(\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top, \quad \boldsymbol{S}^{(lw)} \equiv \sum_{i,j=1}^{n'} \frac{W_{i,j}^{(lw)}}{2}(\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top, \tag{12}$$

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

$$W_{i,j}^{(lb)} \equiv \begin{cases} A_{i,j}(1/n' - 1/n'_{y_i}) & \text{if } y_i = y_j, \\ 1/n' & \text{if } y_i \neq y_j, \end{cases} \quad W_{i,j}^{(lw)} \equiv \begin{cases} A_{i,j}/n'_{y_i} & \text{if } y_i = y_j, \\ 0 & \text{if } y_i \neq y_j. \end{cases} \tag{13}$$

The LFDA transformation matrix $\boldsymbol{T}_{LFDA}$ is defined as

$$\boldsymbol{T}_{LFDA} \equiv \underset{\boldsymbol{T} \in \mathbb{R}^{d \times r}}{\operatorname{argmax}} \left[\operatorname{tr}\left(\boldsymbol{T}^\top \boldsymbol{S}^{(lb)} \boldsymbol{T} (\boldsymbol{T}^\top \boldsymbol{S}^{(lw)} \boldsymbol{T})^{-1} \right) \right]. \tag{14}$$

$A_{i,j}(1/n' - 1/n'_{y_i})$ is negative while $A_{i,j}/n'_{y_i}$ and $1/n'$ are non-negative. Thus, LFDA seeks a transformation matrix $\boldsymbol{T}$ such that nearby data pairs in the same class are made close and the data pairs in different classes are made apart; far apart data pairs in the same class are not imposed to be close. Samples in different classes are separated from each other irrespective of their affinity values. A solution $\boldsymbol{T}_{LFDA}$ is given with $\overline{\boldsymbol{C}} = \boldsymbol{S}^{(lb)}$ and $\underline{\boldsymbol{C}} = \boldsymbol{S}^{(lw)}$.

When $A_{i,j} = 1$ for all i, j (i.e., no locality), $\boldsymbol{S}^{(lw)}$ and $\boldsymbol{S}^{(lb)}$ are reduced to $\boldsymbol{S}^{(w)}$ and $\boldsymbol{S}^{(b)}$ [2]. Thus, LFDA could be regarded as a localized variant of FDA. The between-class scatter matrix $\boldsymbol{S}^{(b)}$ has at most rank $c - 1$, while its local counterpart $\boldsymbol{S}^{(lb)}$ usually has full rank (given $n' \geq d$). Therefore, LFDA can be applied to dimensionality reduction into *any* dimensional spaces.

3 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 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.

3.1 Basic Idea

When only a small number of labeled samples are available, supervised dimensionality reduction methods tend to find embedding spaces which are overfitted to the labeled samples. In such situations, using unlabeled samples is often

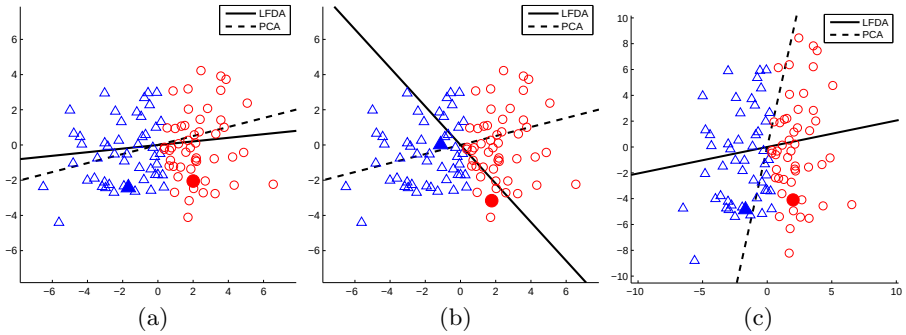

Fig. 1. Illustrative examples of LFDA and PCA for toy data sets. Circle/triangle symbols denote samples in positive/negative classes and filled/unfilled symbols denote labeled/unlabeled samples; solid and dashed lines denote 1-dimensional embedding spaces found by LFDA and PCA, respectively (onto which data samples will be projected).

effective—indeed, the book [3] showed through extensive simulations that PCA works well on the whole; our experimental results in Section 4 also show that PCA is sometimes better than LFDA. This means that preserving the global structure of all samples in an unsupervised manner can be better than strongly relying on class information provided by a small number of labeled samples.

Figure 1 depicts 2-dimensional 2-class examples; circle/triangle symbols denote samples in positive/negative classes and filled/unfilled symbols denote labeled/unlabeled samples; solid and dashed lines denote 1-dimensional embedding spaces found by LFDA and PCA, respectively (onto which data samples will be projected). For the data set in Figure 1(a), both LFDA and PCA can find good embedding spaces which well separate unlabeled samples in different classes from each other. However, for the data set in Figure 1(b), LFDA finds an embedding space that is overfitted to the labeled samples. On the other hand, in the case of Figure 1(c), PCA does not work well due to its unsupervised nature.

The above result implies that LFDA and PCA can compensate for the weakness of each other, i.e., LFDA can utilize label information, while PCA can avoid overfitting. Our simulation results with benchmark data sets in Section 4 also show that LFDA and PCA work in a complementary manner. Motivated by these facts, we propose *bridging* LFDA and PCA so that we can smoothly control our reliance on the global structure of unlabeled samples and class information brought by labeled samples. We refer to the proposed method as *semi-supervised LFDA* (SELF).

The embedding transformations of LFDA and PCA can be analytically computed based on the eigendecompositions. So we combine the eigenvalue problems of LFDA and PCA and solve them together. This allows us to maintain the computational efficiency and reliability of LFDA and PCA.

3.2 Definition

More specifically, we propose solving the following generalized eigenvalue problem:

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

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

$$\boldsymbol{S}^{(rlb)} \equiv (1-\beta)\boldsymbol{S}^{(lb)} + \beta \boldsymbol{S}^{(t)}, \quad \boldsymbol{S}^{(rlw)} \equiv (1-\beta)\boldsymbol{S}^{(lw)} + \beta \boldsymbol{I}_d. \tag{16}$$

β ($\in [0,1]$) 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 inherits characteristics of both LFDA and PCA (this will be discussed in detail in Section 3.3). The solution of SELF can be computed in the same way as LFDA or PCA.

3.3 Properties

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

$$\boldsymbol{S}^{(rlb)} \equiv \frac{1}{2}\sum_{i,j=1}^{n} W_{i,j}^{(rlb)} (\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top, \tag{17}$$

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

$$W_{i,j}^{(rlb)} \equiv \begin{cases} (1-\beta)A_{i,j}(1/n' - 1/n'_{y_i}) + \beta/n & \text{if } y_i = y_j, \\ (1-\beta)/n' + \beta/n & \text{if } y_i \neq y_j, \\ \beta/n & \text{otherwise.} \end{cases} \tag{18}$$

The first case in Eq.(18) is negative if $\beta < \frac{A_{i,j}n(n'-n'_{y_i})}{A_{i,j}n(n'-n'_{y_i}) + n'n'_{y_i}}$ (< 1). This implies that SELF tries to make sample pairs in the same class close if β is small, while it separates them from each other if β is large. Thus the local data structure in the same class tends to be preserved when β is small, but it is no longer preserved when β is large. The second case in Eq.(18) is always positive for any $\beta \in [0,1]$, implying that SELF always tries to make sample pairs in different classes apart for any β. This would be natural in semi-supervised learning scenarios. The third case in Eq.(18) is always non-negative, implying that unlabeled samples are separated from each other for preserving the global data structure.

Next, we give an interpretation of $\boldsymbol{S}^{(rlw)}$. When $\beta = 0$, $\boldsymbol{S}^{(rlw)}$ ($= \boldsymbol{S}^{(lw)}$) could be ill-conditioned—this is crucial particularly 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 $\boldsymbol{S}^{(rlw)}$ works as a *regularizer* and SELF can avoid overfitting to the labeled samples. Therefore, SELF is regarded as a regularized variant of LFDA and would be more stable and reliable than original LFDA particularly when the number of labeled samples is small. Note that unlike Eq.(17), $\boldsymbol{S}^{(rlw)}$ does not have a pairwise expression since $\boldsymbol{I}_d$ can not be expressed in a pairwise form.

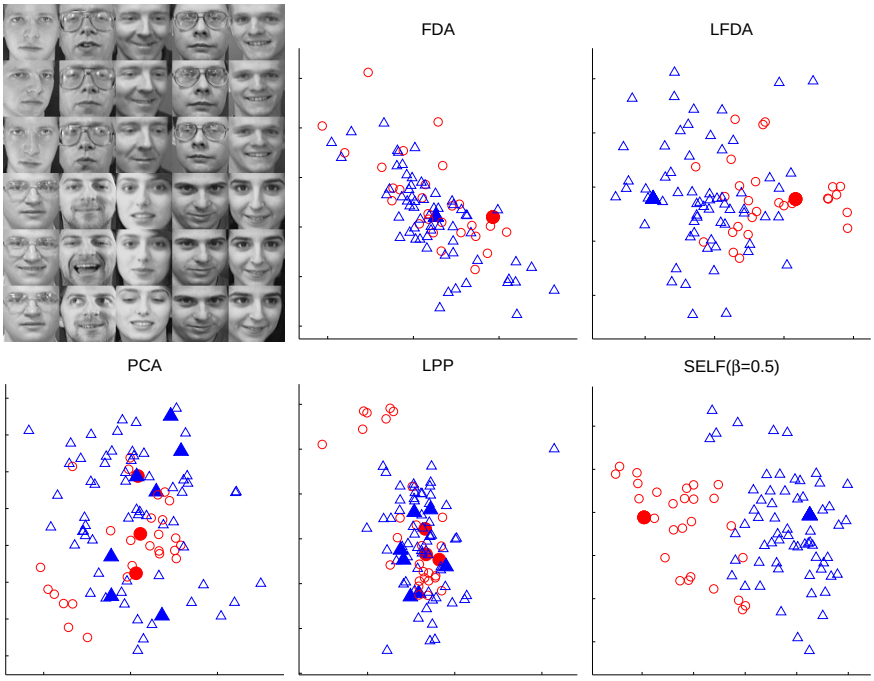

Fig. 2. Embedded face samples (glasses vs. non-glasses). Circle/triangle symbols are faces with/without glasses and filled/unfilled symbols are labeled/unlabeled samples.

3.4 Numerical Examples

For illustrating how SELF behaves, let us use the *Olivetti face* data set[5]. The data set consists of 400 gray-scale face images (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 simulation, we use the image samples of only 10 subjects (i.e., totally 100 images) for making the visualization results clear. We note that the result does not change essentially (but visually denser) when all 400 images are used.

Among 10 people used for the experiments, 3 subjects are with glasses and other 7 are without glasses (see the top-left pictures of Figure 2). Our task is to embed the face images into a two-dimensional space so that the subjects *with* and *without* glasses are separated from each other. We treat 1 image per person as labeled (i.e., totally 3 faces with glasses and 7 faces without glasses) and the rest are treated as unlabeled. Since each class contains several different subjects, this data set is thought to possess within-class multimodality.

The embedded results are shown in Figure 2, where circle/triangle symbols are faces with/without glasses and filled/unfilled symbols are labeled/unlabeled samples. The figure shows that FDA and LFDA perfectly separate the labeled

[5] '`http://www.cs.toronto.edu/~roweis/data.html`'

samples in different classes from each other. However, unlabeled samples tend to be mixed due to an overfitting phenomenon. PCA and LPP tend to mix the labeled samples in different classes due to the unsupervised nature. Consequently, unlabeled samples in different classes are also mixed. On the other hand, SELF with $\beta = 0.5$ clearly separates the labeled samples in different classes from each other, and at the same time, it also nicely separates the unlabeled samples in different classes from each other. We note that, in this visualization simulation, 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 Simulations

In this section, we experimentally evaluate the performance of relevant dimensionality reduction methods using standard classification benchmark data sets.

The book [3] conducted systematic experiments for comparing semi-supervised learning methods. The results showed that each method performs very well for a particular type of data sets. However, at the same time, it tends to be poor for other kinds of data sets. Thus, the performance of semi-supervised learning methods is highly dependent on the type of data sets and there seems to be no single best method. On the other hand, 1-*nearest neighbor classifier* is shown to be stable for various data sets, although it may not be the best possible method in semi-supervised classification. For avoiding the 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 monotone increasing as the dimensionality is reduced[6]. 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 dimensionality reduction methods in a meaningful way. Prefixing the reduced dimensionality r to some number is a possible option for avoiding the above problem, but the evaluation results can significantly depend on the choice of the dimensionality. Based on this argument, we decided to use the *average* misclassification rate over 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 employ the benchmark data sets taken from the book [3], which consist of 9 semi-supervised data sets. We refer to them as the *SSL* data sets. We did not test the *SSL8* and *SSL9* data sets since they are too huge. Note that the *SSL6* data set contains 6 classes, while the other data sets have 2 classes. Table 1 describes the mean and standard deviation of the misclassification rate over repetitions. Since we had a numerical problem when computing LFDA, we slightly regularized it and consider SELF with $\beta = 0.001$ as LFDA. The fulfillment of the *cluster assumption* [3] is described as 'CA', which is the correct

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

classification rate by the 1-nearest-neighbor classifier when both training and test labels are used for classifying all the training and test samples. Note that CA is computed *before* dimensionality reduction is applied, so it represents the fulfillment of the cluster assumption of the original data samples. The larger the value of CA is, the more reliable the cluster assumption would be (although the values are coarse).

When the number of labeled samples is 100 (see the upper half of the table), LFDA and PCA tend to work well in a complementary way—LFDA works well if CA is small while PCA works well if CA is large. SELF with $\beta = 0.5$ tends to make up the deficit of each method; moreover it can outperform both LFDA and PCA for some cases. We also test 'SELF(CV)', where β in SELF is chosen from $\{0, 0.25, 0.5, 0.75, 1\}$ by 10-fold cross validation. The results shown in the table show that SELF(CV) further improves the performance over SELF with $\beta = 0.5$. LPP does not work so well on the whole. The combination of LFDA and LPP in a similar way (indicated by SELF'(CV) in the table) also does not perform as good as SELF(CV). We also tested the combination of LFDA, PCA, and LPP, but this did not further improve the performance over SELF so we omit the detail.

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

We also conducted similar experiments using the *IDA* data sets [6], where we randomly extracted labeled and unlabeled samples from the pool of all samples; we tested $n' = 100, 30$. The results are summarized in Table 2, showing that SELF(CV) still compares favorably with alternative methods.

Overall, SELFreg is shown to be a useful dimensionality reduction.

5 Conclusions and Future Prospects

Our approach to dimensionality reduction in this paper is called the *filter* approach, i.e., the dimensionality reduction procedure is independent of subsequent classification algorithms. Our experimental results showed that the proposed method, *SELF*, works well when it is combined with the 1-nearest-neighbor classifier. An important future direction is to develop a *wrapper* method of semi-supervised dimensionality reduction, which explicitly takes properties of subsequent classification algorithms into account. We expect that a wrapper approach is promising in semi-supervised learning since the performance of elaborate semi-supervised learning methods is highly dependent on the reliability of the assumption behind unlabeled samples such as the cluster or manifold structure [3].

In this paper, we focused on linear dimensionality reduction. However, we can show that a non-linear variant of SELF is obtained by employing the standard *kernel trick*. This kernelized variant also allows us to reduce the dimensionality of *non-vectorial structured data* such as strings, trees, and graphs [7]. However, kernelized SELF shares the common difficulty in kernel methods, i.e., how to

Table 1. Misclassification rate for the SSL data sets. The numbers in the bracket are the standard deviation over repetitions. For each data set, the best method and comparable ones based on the *t-test* at the significance level 5% are described in bold face. 'CA' denotes the fulfillment of the cluster assumption. SELF(CV) denotes SELF with β chosen by cross validation. SELF' denotes the combination of LFDA and LPP in a similar manner.

Data	CA	LFDA	SELF ($\beta = 0.5$)	PCA	SELF (CV)	LPP	SELF' (CV)
SSL1	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	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	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	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	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	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	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	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	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	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	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	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	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	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 rate for the IDA data sets.

Data	CA	LFDA	SELF ($\beta = 0.5$)	PCA	SELF (CV)	LPP	SELF' (CV)
banana	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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

choose the kernel functions. This needs to be investigated in the context of semi-supervised dimensionality reduction. In the future work, we will also explore semi-supervised dimensionality reduction of structured data using kernel SELF.

A remaining important issue to be discussed—which is common to all semi-supervised learning techniques—is how to optimize tuning parameters. We may simply employ cross-validation for this purpose, but it has two potential problems. The first problem is that the number of labeled samples is typically small in semi-supervised learning scenarios and thus cross-validation is not reliable [3]. Fortunately, our experiments showed that SELF is not so sensitive to the trade-off parameter β in small sample cases, but there is still room for further 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* in statistics and ordinary cross-validation is known to be significantly biased; *importance-weighted* cross-validation is unbiased under covariate shift [8]. In the future work, we will investigate how the covariate shift adaptation techniques could be employed in the context of semi-supervised dimensionality reduction.

Acknowledgments: The authors would like to thank members of T-PRIMAL (Tokyo PRobabilistic Inference and MAchine Learning) for their fruitful comments. MS acknowledges financial support from MEXT (Grant-in-Aid for Young Scientists 17700142 and Grant-in-Aid for Scientific Research (B) 18300057) and Tateishi Science and Technology Foundation.

References

1. Fukunaga, K.: Introduction to Statistical Pattern Recognition. Second edn. Academic Press, Inc., Boston (1990)
2. Sugiyama, M.: Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis. Journal of Machine Learning Research **8** (May 2007) 1027–1061
3. Chapelle, O., Schölkopf, B., Zien, A., eds.: Semi-Supervised Learning. MIT Press, Cambridge (2006)
4. He, X., Niyogi, P.: Locality preserving projections. In Thrun, S., Saul, L., Schölkopf, B., eds.: Advances in Neural Information Processing Systems 16. MIT Press, Cambridge, MA (2004)
5. Zelnik-Manor, L., Perona, P.: Self-tuning spectral clustering. In Saul, L.K., Weiss, Y., Bottou, L., eds.: Advances in Neural Information Processing Systems 17. MIT Press, Cambridge, MA (2005) 1601–1608
6. Rätsch, G., Onoda, T., Müller, K.R.: Soft margins for adaboost. Machine Learning **42**(3) (2001) 287–320
7. Kashima, H., Koyanagi, T.: Kernels for semi-structured data. In: Proceedings of the Nineteenth International Conference on Machine Learning, San Francisco, CA, Morgan Kaufmann (2002) 291–298
8. Sugiyama, M., Krauledat, M., Müller, K.R.: Covariate shift adaptation by importance weighted cross validation. Journal of Machine Learning Research **8** (May 2007) 985–1005