Multiclass spectral feature scaling method for dimensionality reduction

Momo Matsuda*, Keiichi Morikuni, Akira Imakura, Xiucai Ye, Tetsuya Sakurai

University of Tsukuba, Tsukuba, Japan

Abstract

Irregular features disrupt the desired classification. In this paper, we consider aggressively modifying scales of features in the original space according to the label information to form well-separated clusters in low-dimensional space. The proposed method exploits spectral clustering to derive scaling factors that are used to modify the features. Specifically, we reformulate the Laplacian eigenproblem of the spectral clustering as an eigenproblem of a linear matrix pencil whose eigenvector has the scaling factors. Numerical experiments show that the proposed method outperforms well-established supervised dimensionality reduction methods for toy problems with more samples than features and real-world problems with more features than samples.

Keywords: Dimensionality reduction; spectral clustering; feature scaling; machine learning

1 Introduction

Dimensionality reduction is a technique for reducing the number of variables of data samples and has been successfully applied in many fields to make machine learning algorithms faster and more accurate, including the pathological diagnoses of gene expression data [26], the analysis of chemical sensor data [16], the community detection in social networks [27], the analyses of neural spike sorting [1], and others [22]. Due to their dependence on label information, dimensionality reduction methods can be divided into supervised and unsupervised methods. Typical unsupervised dimensionality reduction methods are the principal component analysis (PCA) [12, 15], the classical multidimensional scaling (MDS) [4], the locality preserving projections (LPP) [11], and the t-distributed stochastic neighbor embedding (t-SNE) [28].

To make use of prior knowledge on the labels, we focus on supervised dimensionality reduction methods. Supervised dimensionality reduction methods map data samples into an optimal low-dimensional space for satisfactory classification while incorporating the label information. One of the most popular supervised dimensionality reduction methods is the linear discriminant analysis (LDA) [3], which maximizes the between-class scatter and reduces the within-class scatter in a low-dimensional space. However, LDA is based on the assumption that the input data obeys a Gaussian

---

*Corresponding author

Complete address of corresponding author: Momo Matsuda, Department of Computer Science, University of Tsukuba, 1-1-1 Tennodai, Tsukuba, Ibaraki, Japan; Tel: +81-29-853-6574; Fax: +81-29-853-6574; Email: matsuda.momo.ww@alumni.tsukuba.ac.jp

distribution and may fail to capture the underlying local data structure. To overcome its drawback, variants of LDA have been proposed. Bressan and Vitrià [5] proposed a variant that preserves the local data structure by redefining the between-class scatter and within-class scatter matrices based on the *k*-nearest neighbors of each data sample. Weinberger and Saul [30] considered penalizing large distances among data samples with the same label and the *k*-nearest neighbors and small distances among data samples with different labels according to a cost function. Sugiyama proposed the local Fisher discriminant analysis (LFDA) [24], which maximizes the between-class separability and preserves the within-class local structure, incorporating LPP for supervised dimensionality reduction. Cai et al. [7] proposed another variant, which captures the local geometrical data structure to map *k*-nearest neighbors in the same classes and those in different classes to a low-dimensional space as close and as distant as possible, respectively. Instead of preserving the local data structure based on the distances among data samples in the original data space, Li et al. [17] introduced a weight between a pair of samples in the same class and exploited the local manifold structure of data samples in a low-dimensional space.

In addition to capturing the local structure, as in LDA and its variants, we consider aggressively modifying the scales of the features in the original space before mapping the data samples to a low-dimensional space. This results in reducing the effect of features that interfere in the desirable classification as well as enhancing the separations of different clusters. For these purposes, we derive quantities called feature scaling factors, which are used to scale features based on spectral clustering (SC) for dimensionality reduction [18]. SC projects given data samples to a low-dimensional space formed by eigenvectors of a similarity or Laplacian matrix. In [33, 32], it was reported that SC is effective in detecting cluster structures, and in [31], it was reported that SC preserves the local data structure for feature selection. These reports motivated us to also use SC to learn local cluster structures.

Our purpose is to scale features of given data samples and form (well-separated) clusters in a lower dimensional space. Previous spectral feature scaling methods aimed at binary classification [18]: in this paper, we extend the scope of classification problems to handle multiclassing. Multiclass classification problems with more than two classes have typically been solved by combining independently produced binary classification problems [13]. We take an approach different from [13] to enable a multiclass classification. The contributions of this study can be summarized as follows:

Extension to multiclass classification
> We extend the scope of problems that the spectral feature scaling method can handle to multiclass classification. Previous formulations allow only binary classification and require repeated applications of the method for multiclass classification.

Automatic parameter tuning
> We propose an automatic tuning technique to determine the values of parameters that specify entries of the (Fielder) eigenvectors according to the labels of training data samples. The extension of the spectral feature scaling method to multiclass classification naively increases the number of hyperparameters and the computational cost. Our technique can save this cost in the multiclass case.

The remainder of this paper is organized as follows. In Section 2, we present preliminaries. In Section 3, we present the proposed spectral feature scaling method. In Section 4, we describe the difference between the proposed method and previous methods. In Section 5, we present experi-

mental results comparing the proposed method with previous methods. In Section 6, we conclude the paper. Throughout the paper, we use italic bold characters to represent column vectors.

2 Preliminaries

Conventional spectral clustering nonlinearly reduces the dimensionality of data samples but may not form the desired data sample clusters due to irregularity of the data features. To improve the classification of reduced data samples, we may modify the scales of features. Simple feature scaling techniques, such as centering, standardization, and normalization, serve as preprocessing for the support vector machine and k-means algorithm. This motivated us to propose supervised feature scaling that enhances the separation of reduced data samples based on prior knowledge from training data.

The spectral clustering partitions a weighted undirected graph in which the edge weight $w_{i,j}$ between nodes i and j is defined according to the similarity between data samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, where $i, j = 1, 2, \ldots, n$ and n is the number of data samples. We use the locally scaled Gaussian kernel function [34] to define the similarity between data samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ as follows:

$$w_{i,j} = \exp\left(-\frac{(\boldsymbol{x}_i - \boldsymbol{x}_j)^\mathsf{T} S (\boldsymbol{x}_i - \boldsymbol{x}_j)}{\sigma_i' \sigma_j'}\right), \tag{1}$$

where $S = \mathrm{diag}\,(s_1, s_2, \ldots, s_m) \in \mathbb{R}^{m \times m}$, and σ_i' denotes the local scaling of data samples around $\boldsymbol{x}_i$. Thus, we convert a given set of data samples to a graph according to similarities or distances among data samples. A similarity graph is given by a model of the local relationship between data samples. We set $\sigma_i' = \left(\boldsymbol{x}_i - \boldsymbol{x}_i^{(k)}\right)^\mathsf{T} S \left(\boldsymbol{x}_i - \boldsymbol{x}_i^{(k)}\right)$, where $\boldsymbol{x}_i^{(k)}$ is the kth nearest neighbor of $\boldsymbol{x}_i$. Zelnik-Manor and Perona [34] recommended the value $k = 7$ for $S = I$ in spectral clustering in practice, and we adopted this choice.

Spectral clustering partitions a graph into subgraphs that are contradictory to each other and forms clusters from the corresponding data samples. Let $W_s = (w_{i,j}) \in \mathbb{R}^{n \times n}$ be the similarity matrix,

$$D_s = \mathrm{diag}\,(d_1, d_2, \ldots, d_n) \in \mathbb{R}^{n \times n} \quad \text{with} \quad d_i = \sum_{j=1}^{n} w_{i,j}, \tag{2}$$

$\boldsymbol{e}$ be a vector of all ones, and $t_i \in \{\pm 1\}$ be an indicator such that $t_i = 1$ when data sample i belongs to a cluster and $t_i = -1$ when data sample i belongs to another cluster. Then, the graph partitioning is given by the constrained minimization problem of the normalized cut (Ncut) function [21]:

$$\min_{\boldsymbol{v}} \frac{\boldsymbol{v}^\mathsf{T} (D_s - W_s) \boldsymbol{v}}{\boldsymbol{v}^\mathsf{T} D_s \boldsymbol{v}},$$

$$\text{subject to} \quad \boldsymbol{e}^\mathsf{T} D_s^{(p)} \boldsymbol{v} = 0, \quad v_i \in \{1, -b\}, \ b = \frac{\sum_{i:t_i>0} d_i}{\sum_{i:t_i<0} d_i}. \tag{3}$$

The entry $v_i \in \{1, -b\}$ takes two discrete values to devide the set of data samples into two clusters. By the continuous relaxation $v_i \in \{1, -b\}$ to $v_i \in \mathbb{R}$, (3) is reduced to finding the Fiedler vector $\boldsymbol{v} \in \mathbb{R}^n \setminus \{\boldsymbol{0}\}$ [9] corresponding to the minimum eigenvalue of the constrained generalized eigenvalue problem

$$L_s \boldsymbol{v} = \lambda_s D_s \boldsymbol{v}, \quad \lambda_s \in \mathbb{R}, \quad \text{subject to} \quad \boldsymbol{e}^\mathsf{T} D_s \boldsymbol{v} = 0, \tag{4}$$

if the minimum eigenvalue has multiplicity one, where $L_s = D_s - W_s$ is the Laplacian matrix. The entries of the Fiedler vector are referred to as the coordinates of the data samples in the reduced space. In practice, we use more eigenvectors corresponding to the minimum eigenvalues of (4) to form a larger space for accurate classification. When clustering well-separated data samples, we will find hyperplanes that separate the data samples with reduced dimensions formed by a few eigenvectors in (4). Here, if $S = I$ in (1), i.e., S is the identity matrix, (4) is reduced to a conventional spectral clustering method with relaxed Ncut [21]. If $S \neq I$, (4) is a generalization of the method [21] and we can calculate the diagonal entries s_i $(i = 1, 2, \ldots, m)$ of S, which are called the scaling factors, for each data feature for binary classification problems [18].

3 Proposed method

In this section, we formulate the spectral feature scaling method (SFS) for dimensionality reduction. The method was originally proposed for binary clustering and classification, but it is here extended to multiclass classification. The proposed method does not use typical means of extension to multiclass classification, such as multiclass support vector machines [13]. The new formulation of SFS computes r scaling factors for each feature and merges them into a single scaling factor. Here, the parameter r is the number of combinations that divide the set of data samples into two classes: we will discuss the selection of r in Section 3.3.

If the labels of the training data samples are known, we use discrete variable $v_i \in \{1, -b\}$ in (3) to prescribe the entries of the r eigenvectors $\boldsymbol{v}^{(p)} \in \mathbb{R}^n \setminus \{\boldsymbol{0}\}$ $(p = 1, 2, \ldots, r)$ of the Laplacian eigenvalue problem

$$L_s^{(p)} \boldsymbol{v}^{(p)} = \lambda_s^{(p)} D_s^{(p)} \boldsymbol{v}^{(p)}, \quad \boldsymbol{e}^\mathsf{T} D_s \boldsymbol{v}^{(p)} = 0, \quad \lambda_s^{(p)} \in \mathbb{R}, \tag{5}$$

where $\boldsymbol{v}^{(p)} \in \{1, -b^{(p)}\}$, $L_s^{(p)} = D_s^{(p)} - W_s^{(p)}$, $D_s^{(p)} = \mathrm{diag}\left(d_1^{(p)}, d_2^{(p)}, \ldots, d_n^{(p)}\right)$, and $W_s^{(p)} = \left(w_{i,j}^{(p)}\right) \in \mathbb{R}^{n \times n}$ depend on the feature scaling factors $s_i^{(p)} \in \mathbb{R}, i = 1, 2, \ldots, m$, and $b^{(p)}$ is a parameter: we will discuss how to estimate the values of $b^{(p)}$ in Section 3.3. The entries $w_{i,j}^{(p)}$ and $d_i^{(p)}$ are analogously defined as (2). We form the scaling matrix S from $s_i^{(p)}$ and will discuss the detail in Section 3.2. Then, we apply $S^{1/2}$ to data $[X^\mathsf{T}, T^\mathsf{T}]^\mathsf{T}$, which consists of the training data $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n]^\mathsf{T} \in \mathbb{R}^{n \times m}$ and the test data $Y \in \mathbb{R}^{N \times m}$ as follows:

$$Z = \begin{bmatrix} X \\ Y \end{bmatrix} S^{1/2}. \tag{6}$$

With the feature scaling S, we enhance the separation of the reduced data samples as desired in the low-dimensional space formed by spectral clustering. This will improve the classification results. Finally, we classify the scaled data $Z \in \mathbb{R}^{(N+n) \times m}$ using ℓ eigenvectors corresponding to the positive minimum eigenvalue of the Laplacian eigenvalue problem

$$L' \boldsymbol{u} = \lambda' D' \boldsymbol{u}, \quad \boldsymbol{e}^\mathsf{T} D' \boldsymbol{u} = 0, \quad \boldsymbol{u} \in \mathbb{R}^N \setminus \{\boldsymbol{0}\}, \quad \lambda' \in \mathbb{R},$$

where $L' = D' - W' \in \mathbb{R}^{N \times N}, W' = \left(w'_{i,j}\right) \in \mathbb{R}^{N \times N}$,

$$w'_{i,j} = \exp\left(-\frac{\|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2^2}{\sigma_i \sigma_j}\right), \quad i, j = 1, 2 \ldots, N, \tag{7}$$

$\boldsymbol{z}_i \in \mathbb{R}^m$ is the ith column of Z^T, $d'_i = \sum_{j=1}^n w'_{i,j}, D' = \mathrm{diag}\left(d'_1, d'_2, \ldots, d'_N\right)$, and $\sigma_i = \|\boldsymbol{z}_i - \boldsymbol{z}_i^{(k)}\|_2$. The values of parameters ℓ and r can be determined by cross-validation.

4

3.1 Formulation

Now, we reformulate (5) as another eigenvalue problem to extract the scaling factors $\sqrt{s_i^{(p)}}$ $(i = 1, 2, \ldots, m)$ as an eigenvector. Denote the scaling matrix by $S^{(p)1/2} = \mathrm{diag}(\sqrt{s_1^{(p)}}, \sqrt{s_2^{(p)}}, \ldots, \sqrt{s_m^{(p)}}) \in \mathbb{R}^{m \times m}$ and the (i, j) entry of the similarity matrix $W_s^{(p)}$ of scaled data $XS^{(p)1/2}$ by

$$
w_{i,j}^{(p)} = \begin{cases} 1 - \dfrac{(\boldsymbol{x}_i - \boldsymbol{x}_j)^\mathsf{T} S^{(p)} (\boldsymbol{x}_i - \boldsymbol{x}_j)}{\sigma_i \sigma_j} \\ = 1 - \dfrac{\boldsymbol{s}^{(p)\mathsf{T}} \boldsymbol{x}_{i,j}}{\sigma_i \sigma_j} \simeq \exp\left(-\dfrac{(\boldsymbol{x}_i - \boldsymbol{x}_j)^\mathsf{T} S^{(p)} (\boldsymbol{x}_i - \boldsymbol{x}_j)}{\sigma_i' \sigma_j'}\right), & i \neq j, \\ 0, & i = j, \end{cases} \tag{8}
$$

where $\sigma_i = \|\boldsymbol{x}_i - \boldsymbol{x}_i^{(k)}\|_2$, $\boldsymbol{s}^{(p)} = \left[s_1^{(p)}, s_2^{(p)}, \ldots, s_m^{(p)}\right]^\mathsf{T} \in \mathbb{R}^m$, and the kth entry of $\boldsymbol{x}_{i,j} \in \mathbb{R}^m$ is $(x_{i,k} - x_{j,k})^2$ [20]. Here, we used the first-order approximation of the exponential function $\exp(-x) \simeq 1 - x$ for $0 < x \ll 1$. Then, the ith row of $W_s^{(p)}$ is

$$
\boldsymbol{w}_i^{(p)\mathsf{T}} = [1, \ldots, 1, 0, 1, \ldots, 1] - \boldsymbol{s}^{(p)\mathsf{T}} \left[\frac{\boldsymbol{x}_{i,1}}{\sigma_i \sigma_1}, \frac{\boldsymbol{x}_{i,2}}{\sigma_i \sigma_2}, \ldots, \frac{\boldsymbol{x}_{i,n}}{\sigma_i \sigma_n}\right] = \tilde{\boldsymbol{e}}_i^\mathsf{T} - \boldsymbol{s}^{(p)\mathsf{T}} X_i,
$$

where $\tilde{\boldsymbol{e}}_i$ is an n-dimensional vector with zero in the ith entry and ones in all other entries, and $X_i = \left[\frac{\boldsymbol{x}_{i,1}}{\sigma_i \sigma_1}, \frac{\boldsymbol{x}_{i,2}}{\sigma_i \sigma_2}, \ldots, \frac{\boldsymbol{x}_{i,n}}{\sigma_i \sigma_n}\right] \in \mathbb{R}^{m \times n}$. Hence, we have

$$
W_s^{(p)} \boldsymbol{v}^{(p)} = \left[\tilde{\boldsymbol{e}}_1^\mathsf{T}, \tilde{\boldsymbol{e}}_2^\mathsf{T}, \ldots, \tilde{\boldsymbol{e}}_n^\mathsf{T}\right] \boldsymbol{v}^{(p)} - \left[X_1 \boldsymbol{v}^{(p)}, X_2 \boldsymbol{v}^{(p)}, \ldots, X_n \boldsymbol{v}^{(p)}\right]^\mathsf{T} \boldsymbol{s}^{(p)}.
$$

Let $\hat{\boldsymbol{x}}_i = \sum_{j=1}^n \frac{\boldsymbol{x}_{i,j}}{\sigma_i \sigma_j}$. Then, the ith diagonal entry of $D_s^{(p)}$ is

$$
d_i^{(p)} = \sum_{j=1}^n w_{i,j}^{(s)} = (n - 1) - \boldsymbol{s}^{(p)\mathsf{T}} \hat{\boldsymbol{x}}_i.
$$

Hence, denoting the eigenvector by $\boldsymbol{v}^{(p)} = \left[v_1^{(p)}, v_2^{(p)}, \ldots, v_n^{(p)}\right]^\mathsf{T}$, we have

$$
D_s^{(p)} \boldsymbol{v}^{(p)} = (n - 1)\, \boldsymbol{v}^{(p)} - \left[v_1^{(p)} \hat{\boldsymbol{x}}_1, v_2^{(p)} \hat{\boldsymbol{x}}_2, \ldots, v_n^{(p)} \hat{\boldsymbol{x}}_n\right]^\mathsf{T} \boldsymbol{s}^{(p)}.
$$

Thus, (5) can be written as

$$
\begin{aligned}
& L_s^{(p)} \boldsymbol{v}^{(p)} = \lambda_s^{(p)} D_s^{(p)} \boldsymbol{v}^{(p)} \\
\Leftrightarrow\quad & W_s^{(p)} \boldsymbol{v}^{(p)} = \left(1 - \lambda_s^{(p)}\right) D_s^{(p)} \boldsymbol{v}^{(p)} \\
\Leftrightarrow\quad & \begin{bmatrix} A^{(p)} & \boldsymbol{\alpha}^{(p)} \end{bmatrix} \begin{bmatrix} \boldsymbol{s}^{(p)} \\ -1 \end{bmatrix} = \mu^{(p)} \begin{bmatrix} B^{(p)} & \boldsymbol{\beta}^{(p)} \end{bmatrix} \begin{bmatrix} \boldsymbol{s}^{(p)} \\ -1 \end{bmatrix},
\end{aligned} \tag{9}
$$

where $\mu^{(p)} = 1 - \lambda_s^{(p)}$,

$$
A^{(p)} = \begin{bmatrix} \left(X_1 \boldsymbol{v}^{(p)}\right)^\mathsf{T} \\ \left(X_2 \boldsymbol{v}^{(p)}\right)^\mathsf{T} \\ \vdots \\ \left(X_n \boldsymbol{v}^{(p)}\right)^\mathsf{T} \end{bmatrix} \in \mathbb{R}^{n \times m}, \quad B^{(p)} = \begin{bmatrix} v_1^{(p)} \hat{\boldsymbol{x}}_1^\mathsf{T} \\ v_2^{(p)} \hat{\boldsymbol{x}}_2^\mathsf{T} \\ \vdots \\ v_n^{(p)} \hat{\boldsymbol{x}}_n^\mathsf{T} \end{bmatrix} \in \mathbb{R}^{n \times m}, \tag{10}
$$

$$
\boldsymbol{\alpha}^{(p)} = [\tilde{\boldsymbol{e}}_1, \tilde{\boldsymbol{e}}_2, \ldots, \tilde{\boldsymbol{e}}_n]^\mathsf{T} \boldsymbol{v}^{(p)} \in \mathbb{R}^n, \quad \boldsymbol{\beta}^{(p)} = (n - 1) \boldsymbol{v}^{(p)} \in \mathbb{R}^n. \tag{11}
$$

5

Furthermore, the feature scaling factors have the constraint

$$\boldsymbol{e}^{\mathsf{T}} D_s^{(p)} \boldsymbol{v}^{(p)} = \sum_{i=1}^{n} \left((n-1)\, v_i^{(p)} - v_i^{(p)} \boldsymbol{s}^{(p)} \hat{\boldsymbol{x}}_i \right) = 0$$

$$\Leftrightarrow \left(\sum_{i=1}^{n} v_i^{(p)} \hat{\boldsymbol{x}}_i^{\mathsf{T}} \right) \boldsymbol{s}^{(p)} - (n-1) \sum_{i=1}^{n} v_i^{(p)} = 0. \tag{12}$$

Combining (9)–(12), we can obtain the generalized eigenvalue problem of a linear matrix pencil

$$\mathcal{A}^{(p)} \begin{bmatrix} \boldsymbol{s}^{(p)} \\ -1 \end{bmatrix} = \mu^{(p)} \mathcal{B}^{(p)} \begin{bmatrix} \boldsymbol{s}^{(p)} \\ -1 \end{bmatrix}, \quad \boldsymbol{s}^{(p)} \in \mathbb{R}^m, \quad \mu^{(p)} \in \mathbb{R}, \tag{13}$$

where

$$\mathcal{A}^{(p)} = \begin{bmatrix} A^{(p)} & \boldsymbol{\alpha}^{(p)} \\ \boldsymbol{\gamma}^{(p)\mathsf{T}} & \rho^{(p)} \end{bmatrix} \in \mathbb{R}^{(n+1)\times(m+1)}, \tag{14}$$

$$\mathcal{B}^{(p)} = \begin{bmatrix} B^{(p)} & \boldsymbol{\beta}^{(p)} \\ \boldsymbol{0}^{\mathsf{T}} & 0 \end{bmatrix} \in \mathbb{R}^{(n+1)\times(m+1)}, \tag{15}$$

$$\boldsymbol{\gamma}^{(p)} = \sum_{i=1}^{n} v_i^{(p)} \hat{\boldsymbol{x}}_i \in \mathbb{R}^m, \quad \rho^{(p)} = (n-1) \sum_{i=1}^{n} v_i^{(p)} \in \mathbb{R}, \tag{16}$$

and we solve (13) for the scaling factors $\boldsymbol{s}^{(p)}$ for each $p = 1, 2, \ldots, r$. If the r eigenproblems (13) have a common eigenpair, they can be combined as

$$\begin{bmatrix} \mathcal{A}^{(1)} \\ \mathcal{A}^{(2)} \\ \vdots \\ \mathcal{A}^{(r)} \end{bmatrix} \begin{bmatrix} \boldsymbol{s} \\ -1 \end{bmatrix} = \mu_s \begin{bmatrix} \mathcal{B}^{(1)} \\ \mathcal{B}^{(2)} \\ \vdots \\ \mathcal{B}^{(r)} \end{bmatrix} \begin{bmatrix} \boldsymbol{s} \\ -1 \end{bmatrix}, \quad \boldsymbol{s} \in \mathbb{R}^m, \quad \mu_s \in \mathbb{R}, \tag{17}$$

where $\mu_s = 1 - \lambda_s$. The existence of an eigenvalue $\mu^{(p)}$ of (13) is assured by the equality $\begin{bmatrix} \boldsymbol{e}^{\mathsf{T}} & 0 \end{bmatrix} (\mathcal{A} - \mathcal{B})^{\mathsf{T}} = \boldsymbol{0}^{\mathsf{T}}$. Because the Fiedler vector is associated with the smallest eigenvalue, we adopt the eigenvector of (13) and (17) associated with the eigenvalues $\mu^{(p)} = 1 - \lambda_s^{(p)}$ and μ_s closest to but less than one as a candidate feature scaling factors, respectively. Here, the r eigenproblem (17) may not have the same eigenpair, so we find an eigenpair $\left(\lambda, [\boldsymbol{s}^{\mathsf{T}}, -1]^{\mathsf{T}} \right)$ of (14) and (15) as close as possible in the r eigenproblems. To solve (13) or (17), we can use the contour integral method [19] for $n < m$; otherwise, we can also use the minimal perturbation approach [14].

3.2 Integrating candidates of feature scaling factors

Here, we describe how to determine the scaling factors from the candidates in the multiclass case. The scaling factors $\sqrt{s_i^{(p)}}$ given by solving (13) are used to form a single feature scaling matrix $S^{1/2} = \mathrm{diag}\,(s_1, s_2, \ldots, s_m)^{1/2}$. We have the following five approaches to computing the diagonal elements $s_i^{1/2}$ $(i = 1, 2, \ldots, m)$ of $S^{1/2}$:

- Principal component analysis (PCA) [12, 15]

  We can obtain $s_i^{1/2}$ by using the principal component, i.e., the eigenvector $\boldsymbol{\phi} \in \mathbb{R}^r$ associated with the maximum eigenvalue $\tau \in \mathbb{R}$ of the eigenproblem

  $$M\boldsymbol{\phi} = \tau\boldsymbol{\phi}, \quad M = \sum_{i=1}^{m} \left(\hat{\boldsymbol{s}}_i - \hat{\boldsymbol{\mu}}\right)\left(\hat{\boldsymbol{s}}_i - \hat{\boldsymbol{\mu}}\right)^{\mathsf{T}},$$

  where $\hat{\boldsymbol{s}}_i^{\mathsf{T}} \in \mathbb{R}^r$ is ith row of the matrix

  $$\begin{bmatrix} \sqrt{s_1^{(1)}} & \sqrt{s_1^{(2)}} & \cdots & \sqrt{s_1^{(r)}} \\ \sqrt{s_2^{(1)}} & \sqrt{s_2^{(2)}} & \cdots & \sqrt{s_2^{(r)}} \\ \vdots & \vdots & \ddots & \vdots \\ \sqrt{s_m^{(1)}} & \sqrt{s_m^{(2)}} & \cdots & \sqrt{s_m^{(r)}} \end{bmatrix} \in \mathbb{R}^{m \times r},$$

  and $\hat{\boldsymbol{\mu}} = \frac{1}{m}\sum_{i=1}^{m} \hat{\boldsymbol{s}}_i$. Thus, $s_i^{1/2} = \hat{\boldsymbol{s}}_i^{\mathsf{T}}\boldsymbol{\phi}$ is obtained.

- Arithmetic mean $s_i^{1/2} = \frac{1}{r}\sum_{p=1}^{r} \sqrt{s_i^{(p)}}$.

- Geometric mean $s_i^{1/2} = \left(\prod_{p=1}^{r} \sqrt{s_i^{(p)}}\right)^{1/r}$.

- Root mean square $s_i^{1/2} = \sqrt{\frac{1}{r}\sum_{p=1}^{r} s_i^{(p)}}$.

- Harmonic mean $s_i^{1/2} = r \left/ \sum_{p=1}^{r} \left(1/\sqrt{s_i^{(p)}}\right)\right.$.

We will compare the classification accuracy of each approach for integrating scaling factors in the numerical experiments.

3.3 Number of prescribed eigenvectors and prescription of the entries

Consider determining a reasonable number r of prescribed eigenvectors $\boldsymbol{v}^{(p)}$ to prevent solving many eigenvalue problems (13) or a large eigenvalue problem (17). Let K be the number of classes. In particular, if $K = 2$, we prescribe only one eigenvector $\boldsymbol{v}^{(p)}, p = 1, r = 1$ in (5), as in [18]. This is because one eigenvector is sufficient for binary classification if the clusters are well-separated [8]. If $K > 2$, we split the set of training data samples into r combinations of binary classes with two options: take $r = K$ or take at least $r = \lceil \log_2 K \rceil$ and determine the entries of the eigenvector $\boldsymbol{v}^{(p)}$ as

$$v_i^{(p)} = \begin{cases} 1 & \text{if sample } i \text{ is in class } c, \quad c = 1, 2, \ldots, K, \\ -b^{(p)} & \text{otherwise} \end{cases} \tag{18}$$

for $p = 1, 2, \ldots, r$ according to the label information of the training data, where $b^{(p)}$ is a parameter we discuss how to determine later in this section. Remark that the r-dimensional indicator vector $[v_i^{(1)}, v_i^{(2)}, \ldots, v_i^{(r)}]$ for sample i is uniquely chosen to a label. This is a solution of discrete optimization for a binary clustering (3) reduced from multiclass classification. That is, one cluster consists of

data samples belonging to several classes, and the other clusters consist of data samples belonging to other classes. The classification accuracy varies depending on the combination of such binary classifications. By setting $r = K$, the computational cost for solving (13) or (17) is large, while the classification accuracy tends to be less sensitive to the choice of combinations. When setting r as small, the classification accuracy tends to be sensitive and the computational cost for solving (13) or (17) is small. Hence, there is a trade-off between the cost and sensitivity of classification accuracy for these approaches.

In the previous SFS with $r = 1$ in (18), we empirically determined the value of b [18]. In this study, we found that the values of $b^{(p)}$ in (18) determined by

$$b^{(p)} = \frac{\sum_{i:t_i>0} d_i^{(p)}}{\sum_{i:t_i<0} d_i^{(p)}}, \quad p = 1, 2, \ldots, r \tag{19}$$

according to the label information of training data X give favorable results, where $t_i = 1$ indicates that data sample i belongs to a cluster, and $t_i = -1$ indicates that data sample i belongs to another cluster. The choice of $t_i = 1$ or $t_i = -1$ is flexible in (19) and can be fixed by cross-validation in practice (see Section 5). Note that the ideal Fiedler vector does not necessarily take such an entry and is not necessarily unique.

We summarize the procedures of the proposed method in Algorithm 1.

Algorithm 1 Supervised dimensionality reduction method with spectral features scaling

---

Input: Training data $X \in \mathbb{R}^{n \times m}$ with n samples and m features, test data $Y \in \mathbb{R}^{N \times m}$, number of reduced dimensions ℓ.
Output: Low-dimensional data samples $U = [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_\ell]^\mathsf{T} \in \mathbb{R}^{\ell \times (N+n)}$.
 1: Decide the number of eigenvectors to prescribe r.
 2: Compute the parameter $b^{(p)}$ using X and set the eigenvectors $\boldsymbol{v}^{(p)}$, $(p = 1, 2, \ldots, r)$.
 3: Compute $A^{(p)}, B^{(p)}, \boldsymbol{\alpha}^{(p)}, \boldsymbol{\beta}^{(p)}, \boldsymbol{\gamma}^{(p)}$, and $\rho^{(p)}$ using (10), (11), and (16).
 4: Solve (13) for an eigenvector $\boldsymbol{s}^{(p)}$.
 5: Compute the scaling matrix $S^{1/2}$ from the scaling factors $\boldsymbol{s}^{(p)}$ (Section 3.3).
 6: Compute the scaled data matrix Z (6).
 7: Compute the similarity matrix $W' \in \mathbb{R}^{(N+n) \times (N+n)}$ (7) and D'.
 8: Compute the Laplacian matrix $L' = D' - W'$.
 9: Solve $L'\boldsymbol{u} = \lambda' D'\boldsymbol{u}$ for the eigenvectors $\boldsymbol{u}_1, \boldsymbol{u}_2 \ldots \boldsymbol{u}_\ell$ corresponding to the ℓ smallest nonzero eigenvalues.

---

4   Related methods

In this section, we discuss the difference between the proposed method and previous methods. Many supervised dimensionality reduction methods, such as LDA and its variants, utilize knowledge from labels and map given data samples to an optimal low-dimensional space. The objective of these methods is to find a transformation matrix T to map given data samples X to low-dimensional data samples U

$$U = TX^\mathsf{T},$$

whereas the objective of our method is to generate a scaling matrix $S^{1/2}$ and modify the scales of the features of given data samples X as $XS^{1/2}$ and then map them to low-dimensional data samples U' using a transformation matrix T':

$$U' = T'\phi\left(S^{1/2}X^{\mathsf{T}}\right),$$

where $\phi : \mathbb{R}^{m \times n} \to \mathbb{R}^{m \times n}$ is a transformation with a nonlinear kernel function. The methods in [17, 29] modify the weights among data samples, whereas the proposed method modifies the scales of features in the original space and uses the same feature scaling factors for all data samples.

5   Numerical experiments

We compare the proposed method (SFS) with previous supervised dimensionality reduction methods in terms of accuracy by numerical experiments on artificial data and practical datasets and show that the proposed method outperforms previous methods in some cases, and is more robust than the previous methods. The previous methods were kernel versions of LDA (KDA) [2, 6] and LFDA (KLFDA)[25] and LADA [17]. The performance measures used for comparisons were overall accuracy (OA), average accuracy (AA), and normalized mutual information (NMI) [%] [23]. Here, OA and AA are defined as

$$\mathrm{OA} = \frac{\sum_{i=1}^{K} \hat{n}_i}{\sum_{i=1}^{K} n_i}, \quad \mathrm{AA} = \frac{1}{K}\sum_{i=1}^{K} \frac{\hat{n}_i}{n_i},$$

where n_i is the number of samples in class i, $\hat{n}_i$ is the number of samples classified by a method to class i, and K is the number of classes. We performed five-fold cross-validation and show the mean and standard deviation of each performance measure. The reduced dimension ℓ was chosen by four-fold cross-validation regarding OA between one to the number of features when the number of features is smaller than the number of training data samples and between one to the number of training data samples, otherwise. In SFS, we used the seven nearest neighbors to sparsify the matrix W'. We chose optional values of the parameters λ and γ of LADA [29] among $1, 10^{\pm 1}, 10^{\pm 2}$, and $10^{\pm 3}$ according to cross-validation. In SFS, we chose the value of $t_i = 1$ or $t_i = -1$ to which class i is set when using the number of classes as the number of reduced dimensions (Section 3.3), chose the number of reduced dimension ℓ, and chose the value of t_i again. We used the KLFDA code from the Sugiyama–Sato–Honda Lab site[1] and programmed the LADA and KDA codes. All programs were coded and run in MATLAB 2018a.

5.1   Artificial datasets

In this subsection, we report results on a toy problem consisting of 600 data samples with ten features. The data samples had three classes, and each class had 200 samples. Figure 1 shows three of the ten features, where the symbols $\circ$, $\triangle$, and $\square$ represent data samples in classes 1, 2, and 3, respectively, and the data samples in each class are normally distributed along a ring. The rings of class 1 and 2, and those of classes 2 and 3 intersect, respectively. The variances of the first to third features are 0.35, 2.01, and 0.19, respectively. The remaining seven features are normally distributed with center zero and variance σ^2. We changed the variance σ^2 from 1 to 25 in increments of 1. These

---

[1]http://www.ms.k.u-tokyo.ac.jp/

seven features interfere with data sample classification according to the first three features. The classification of data samples with the reduced dimension was done by logistic regression. Results for other classifiers are given in the Appendix. In SFS, the number of prescribed eigenvectors $\boldsymbol{v}^{(p)}$ was set to the number of classes.

Figure 2 shows the performance measures for variances $\sigma^2 = 1, 2, \ldots, 25$ where SFS (one-versus-all) represents the "one-vs-all" approach [10, p. 658] for the previous SFS for a binary classification problem. Regarding the performance measures, some of the proposed methods were more accurate than the previous methods, and KDA was the worst regardless of the variance. SFS (PCA) was the best when the variance was small, while SFS(arithmetic) and SFS (RMS) were the best when the variance was large. SFS(RMS), SFS(arithmetic), and LADA tended to be robust against interfering features. SFS(one-vs-all) was almost as accurate as KDA, and the line was hidden by the KDA line. LADA was more accurate than KLFDA, though LADA is not kernelized. Among the compared SFS methods, the RMS approach was most accurate for many values of σ^2. The proposed SFS methods were more accurate than the conventional "one-vs-all" approach for the multiclass classification problem.

Figure 3 and 4 show the first and second features of the original data in parts (a) and the two-dimensional data samples reduced by each method in parts (b)–(d) for $\sigma^2 = 1$ and 25, respectively. LADA and KLFDA failed to enhance the separations of clusters, while the proposed method separated clusters well.

Figure 5 shows the absolute values of the obtained scaling factors of SFS(RMS) for $\sigma^2 = 1$ and $\sigma^2 = 25$. Because the fourth to tenth features given by random numbers are not involved in the classification, the values of the scaling factors of these seven features were small. Thus, the values of the scaling factors indicate which features are effective in the desired classification.

5.2   Real-world datasets

In this section, we report results for five datasets with more features than samples in the real world from a public repository[2]. Table 1 gives the numbers of samples, features, and classes of the datasets. Datasets (a)–(d) are of gene expressions, and dataset (e) is of face images. In SFS, the number of prescribed eigenvectors was set to the number of classes, and we used RMS to generate the feature scaling matrix. The classification was done by KNN for $k = 1$ since it was the best in the experiment in Section 5.1.

Table 2 gives the mean and standard deviation of the performance measures for each method. For the colon and lymphoma, the NMI of KDA was Not-a-Number (NaN) because KDA gave the same label to all test data samples. In all performance measures, SFS (RMS) was the best or second best for all datasets. KLFDA was the best for warpPIE10P.

6   Conclusions

In this paper, to deal with irregularity or uncertainty in features, we extended previously proposed feature scaling methods to multiclass classification and proposed a supervised dimensionality reduction method that exploits knowledge of labels of training data samples. The proposed method aggressively modifies the scales of features, and feature scaling can reduce the effects of features that prevent us from obtaining the desired clusters. To obtain the factors for scaling the features based
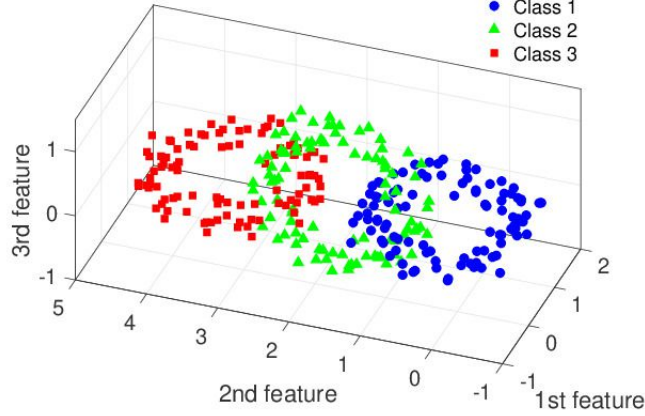
---

[2]http://featureselection.asu.edu/datasets.php

Figure 1: Visualization of the artificial data in three dimensions
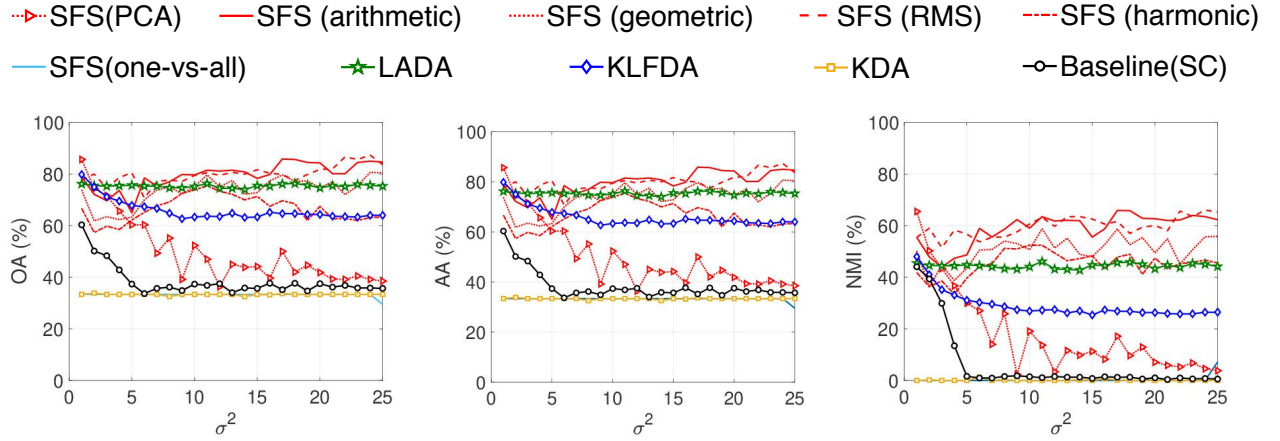
Figure 2: Mean of each performance measure vs. variance σ^2

(a) Original data   (b) LADA   (c) KLFDA   (d) SFS (RMS)
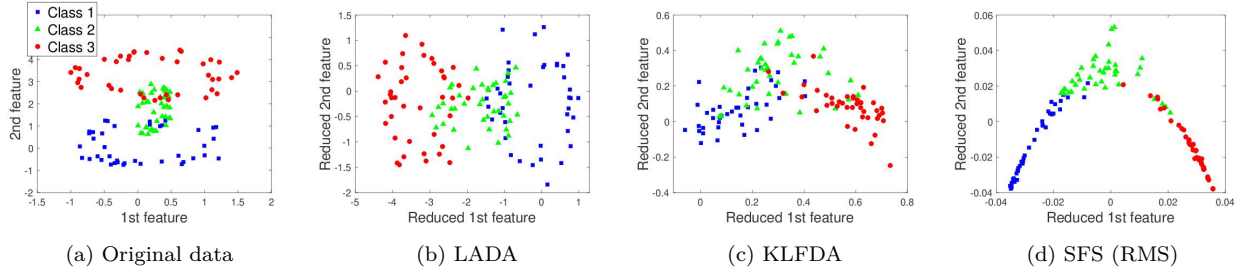
Figure 3: Test data samples for $\sigma^2 = 1$ in the original and reduced two-dimensional space

on spectral clustering, we derived matrix eigenproblems whose eigenvectors have the feature scaling factors and described the procedures for the proposed method. The multiclass extension asks for more hyperparameters and computational cost. To reduce the number of hyperparameters, we pro-
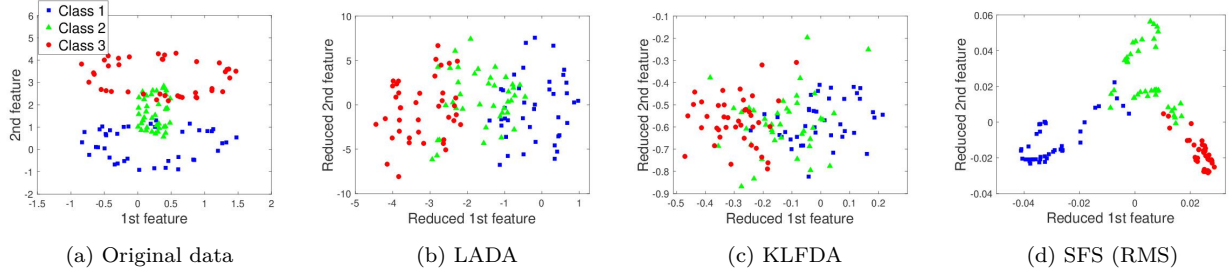
(a) Original data   (b) LADA   (c) KLFDA   (d) SFS (RMS)

Figure 4: Test data samples for $\sigma^2 = 25$ in the original and reduced two-dimensional space

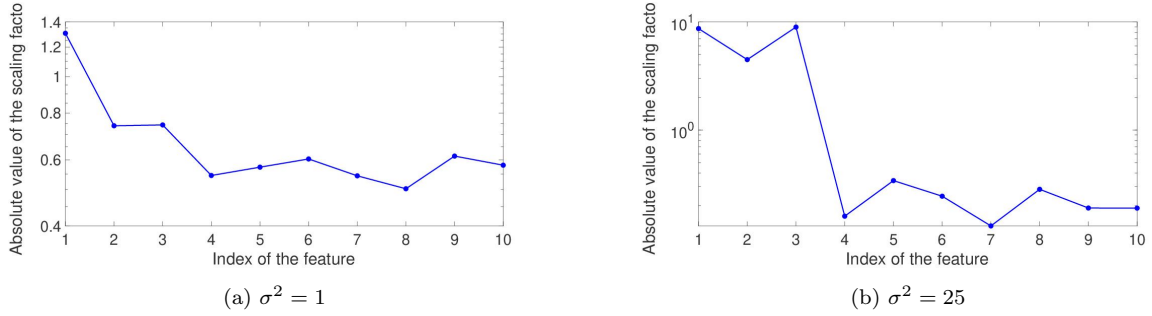(a) $\sigma^2 = 1$ (b) $\sigma^2 = 25$

Figure 5: Values of the scaling factors for (a) $\sigma^2 = 1$ and (b) $\sigma^2 = 25$ when using the root mean square

Table 1: Specifications of datasets

	# of samples	# of features	# of classes
(a) Colon	62	2000	2
(b) CLL_SUB_111	111	11340	3
(c) Lung	203	3312	5
(d) Lymphoma	96	4026	9
(e) WarpPIE10P	210	2420	10

posed an automatic tuning technique based on the formula for discrete optimization to determine the entries of prescribed eigenvectors. The demanding combinations of entries of prescribed vectors are related to the number of eigenproblems to solve and can be reduced with a trade-off between accuracy and computational cost. Numerical experiments showed that feature scaling is effective when combined with the proposed classification method. For toy problems with more samples than features, feature scaling improved accuracy and was more robust to interfering features. In terms of flexibility in merging candidates of scaling factors for each feature, our experimental results indicate that the RMS approach is preferable in practice. For real-world problems with more features than samples, the proposed method was more robust than previous methods and outperformed previous methods in some cases.

Table 2: Performance rates (mean% ± std) for real-world datasets

(a) Colon

	OA	AA	NMI
SFS (RMS)	$81.7_{6.2}$	$\mathbf{81.3}_{8.8}$	$35.1_{19.2}$
LADA	$63.3_{6.7}$	$57.5_{8.3}$	$12.5_{8.0}$
KLFDA	$\mathbf{83.3}_{7.5}$	$78.8_{8.5}$	$\mathbf{36.7}_{22.2}$
KDA	$66.7_{0.0}$	$50.0_{0.0}$	NaN

(b) CLL_SUB_111

	OA	AA	NMI
SFS (RMS)	$\mathbf{68.2}_{5.0}$	$\mathbf{72.7}_{8.8}$	$\mathbf{37.3}_{2.2}$
LADA	$53.6_{7.3}$	$38.0_{6.5}$	$11.2_{8.3}$
KLFDA	$67.3_{7.8}$	$70.7_{9.3}$	$34.2_{7.9}$
KDA	$50.0_{5.0}$	$35.3_{4.0}$	$15.9_{13.3}$

(c) Lung

	OA	AA	NMI
SFS (RMS)	$\mathbf{94.5}_{1.9}$	$\mathbf{85.4}_{11.1}$	$\mathbf{85.7}_{6.3}$
LADA	$93.5_{3.4}$	$79.4_{13.1}$	$79.8_{10.2}$
KLFDA	$93.0_{2.9}$	$82.0_{10.7}$	$78.7_{8.3}$
KDA	$64.5_{8.7}$	$20.4_{4.4}$	$8.5_{10.0}$

(d) Lymphoma

	OA	AA	NMI
SFS (RMS)	$\mathbf{94.8}_{9.2}$	$\mathbf{94.8}_{9.2}$	$\mathbf{94.6}_{5.2}$
LADA	$60.0_{7.8}$	$42.8_{15.4}$	$62.7_{7.4}$
KLFDA	$90.5_{2.1}$	$80.0_{4.4}$	$90.7_{2.2}$
KDA	$47.4_{0.0}$	$11.1_{0.0}$	NaN

(e) WarpPIE10P

	OA	AA	NMI
SFS (RMS)	$88.6_{2.8}$	$88.9_{2.6}$	$90.3_{2.4}$
LADA	$82.9_{2.8}$	$82.6_{3.2}$	$83.7_{3.0}$
KLFDA	$\mathbf{99.5}_{1.0}$	$\mathbf{99.5}_{1.0}$	$\mathbf{99.5}_{1.0}$
KDA	$23.3_{7.6}$	$8.7_{2.5}$	$24.1_{14.1}$

Acknowledgements

Appendix A.   Comparison of classifiers

In Section 5.1, we used logistic regression to classify data samples with reduced dimensions. In fact, the proposed method is not specialized to a particular classifier, which means we can choose an arbitrary classifier after the dimensionality reduction. In this appendix, we observe the dependence of the accuracy on classifiers and compare four classifiers: logistic regression (LR), k-nearest neighbor algorithm (KNN), support vector machine (SVM), and naive Bayes classifier (NB). Among the proposed methods, we used SFS (RMS) because it performed the best in many cases.

Figure 6 shows the mean OA for each method and each classifier. SFS(RMS) was more accurate than other methods for every classifier. SFS (RMS) with KNN was more accurate than the other classifiers. The relative positions of the compared methods are the same for all performance measures.

Figure 7 shows the OA of each method for $k = 1, 2, \ldots, 50$ in the KNN. For $k{=}1$, SFS (RMS) was the best and its OA was 93.7%. The accuracies of the compared methods did not largely depend on the value of k. The relative positions of the compared methods are also the same for all k in the KNN.
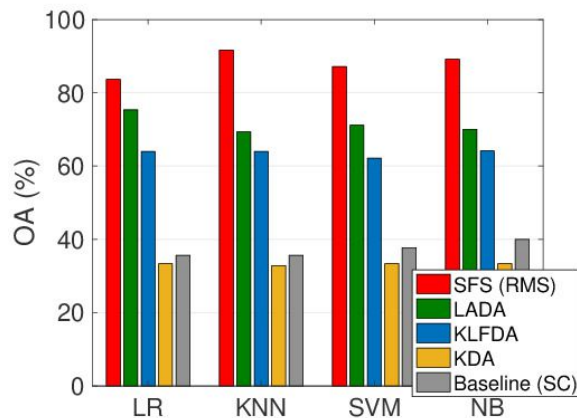

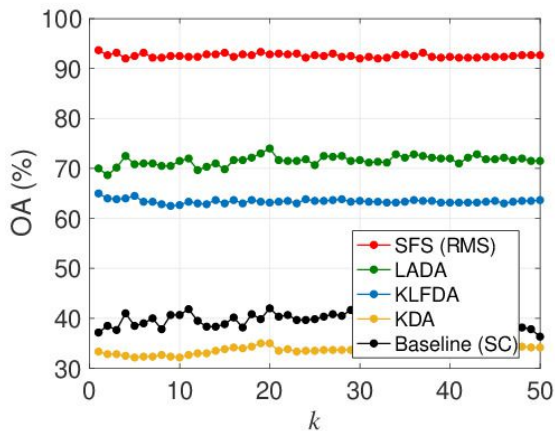Figure 6: Mean performance measure (OA) vs. classifier

Figure 7: Mean of performance measure (OA) vs. k in the k-NN classifier

References

[1] A. Bar-Hillel, A. Spiro, and E. Stark. Spike sorting: Bayesian clustering of non-stationary data. *Journal of Neuroscience Methods*, 157(2):303–316, 2006.

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

[3] P. N. Belhumeur., J. P. Hespanha., and D. J. Kriegman. Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.

[4] I. Borg and P. J. F. Groenen. *Modern Multidimensional Scaling: Theory and Applications.* Springer, New York, 1997.

[5] M. Bressan and J. Vitrià. Nonparametric discriminant analysis and nearest neighbor classification. *Pattern Recognition Letters*, 24(15):2743–2749, 2003.

[6] D. Cai, X. He, and J. Han. Efficient kernel discriminant analysis via spectral regression. *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, pages 427–432, 2007.

[7] D. Cai, X. He, K. Zhou, J. Han, and H. Bao. Locality sensitive discriminant analysis. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2011)*, pages 708–713, 2007.

[8] F. R. K. Chung. *Spectral Graph Theory.* American Mathematical Society, Providence, RI, 1996.

[9] M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(2):298–305, 1973.

[10] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning.* Springer, New York, 2nd edition, 2009.

[11] X. He and P. Niyogi. Locality preserving projections. In *Proceedings of the 16th International Conference on Neural Information Processing Systems*, pages 153–160, 2003.

[12] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417–441, 1933.

[13] C. W. Hsu and C.-J. Lin. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, 2002.

[14] S. Ito and K. Murota. An algorithm for the generalized eigenvalue problem for nonsquare matrix pencils by minimal perturbation approach. *SIAM Journal on Matrix Analysis and Applications*, 37(1):409–419, 2016.

[15] I. T. Jolliffe. *Principal component analysis and factor analysis*, pages 115–128. *Principal Component Analysis*, 1986.

[16] P. C. Jurs, G. A. Bakken, and H. E. McClelland. Computational methods for the analysis of chemical sensor array data from volatile analytes. *Chemical Reviews*, 100(7):2649–2678, 2000.

[17] X. Li, M. Chen, F. Nie, and Q. Wang. Locality adaptive discriminant analysis. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI 2017)*, pages 2201–2207, 2017.

[18] M. Matsuda, K. Morikuni, and T. Sakurai. Spectral feature scaling method for supervised dimensionality reduction. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI 2018)*, pages 2560–2566, 2018.

[19] K. Morikuni. Contour integral-type method for eigenproblems of rectangular matrix pencils. In *Book of Abstracts, Annual Meeting 2016 of the Japan Society for Industrial and Applied Mathematics*, pages 352–353, Kitakyushu, Japan, 2016.

[20] A. Y. Ng, M. I. Jordan, and W. Yair. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, pages 849–856, 2002.

[21] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[22] T. S. Sogh. Wired and wireless intrusion detection system: Classifications, good characteristics and state-of-the-art. *Computer Standards & Interfaces*, 28(6):670–694, 2006.

[23] A. Strehl and J. Ghosh. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617, 2002.

[24] M. Sugiyama. Local Fisher discriminant analysis for supervised dimensionality reduction. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 905–912, 2006.

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

[26] A. L. Tarca, R. Romero, and S. Draghici. Analysis of microarray experiments of gene expression profiling. *American Journal of Obstetrics and Gynecology*, 195(2):373–388, 2006.

[27] N. M. Tichy, M. L. Tushman, and C. Fombrun. Social network analysis for organizations. *Academy of Management Review*, 4(4):507–519, 1979.

[28] L. van der Maaten and G. Hinton. Visualizing high-dimensional data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[29] Q. Wang, Z. Meng, and X. Li. Locality adaptive discriminant analysis for spectral–spatial classification of hyperspectral images. *IEEE Geoscience and Remote Sensing Letters*, 14(11):2077–2081, 2017.

[30] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10:207–244, 2009.

[31] X. Ye, K. Ji, and T. Sakurai. Spectral clustering and discriminant analysis for unsupervised feature selection. In *Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pages 563–568, 2016.

[32] X. Ye, H. Li, T. Sakurai, and Z. Liu. Large scale spectral clustering using sparse representation based on hubness. In *Proceeding of the IEEE International Conference on Cloud and Big Data Computing*, pages 1731–1737, 2018.

[33] X. Ye and T. Sakurai. Spectral clustering with adaptive similarity measure in Kernel space. *Intelligent Data Analysis*, 22(4):751–765, 2018.

[34] L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. In *Proceedings of Advances in Neural Information Processing Systems 17*, pages 1601–1608, 2005.