Pattern Recognition 123 (2022) 108399

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Kernelized Supervised Laplacian Eigenmap for Visualization and Classification of Multi-Label Data☆

Mariko Tai[a,1], Mineichi Kudo[a,*], Akira Tanaka[a], Hideyuki Imai[a,b], Keigo Kimura[a]

[a] Graduate School of Information Science and Technology, Hokkaido University, Sapporo 060-0814, Japan
[b] Global Station for Big Data and Cybersecurity, Hokkaido University, Japan

ARTICLE INFO

Article history:
Received 28 January 2021
Revised 22 October 2021
Accepted 23 October 2021
Available online 26 October 2021

Keywords:
Supervised Laplacian eigenmaps
Out-of-sample problem
Multi-label problems
Kernel trick
Separability-guided feature extraction

ABSTRACT

We had previously proposed a supervised Laplacian eigenmap for visualization (SLE-ML) that can handle multi-label data. In addition, SLE-ML can control the trade-off between the class separability and local structure by a single trade-off parameter. However, SLE-ML cannot transform new data, that is, it has the "out-of-sample" problem. In this paper, we show that this problem is solvable, that is, it is possible to simulate the same transformation perfectly using a set of linear sums of reproducing kernels (KSLE-ML) with a nonsingular Gram matrix. We experimentally showed that the difference between training and testing is not large; thus, a high separability of classes in a low-dimensional space is realizable with KSLE-ML by assigning an appropriate value to the trade-off parameter. This offers the possibility of separability-guided feature extraction for classification. In addition, to optimize the performance of KSLE-ML, we conducted both kernel selection and parameter selection. As a result, it is shown that parameter selection is more important than kernel selection.

We experimentally demonstrated the advantage of using KSLE-ML for visualization and for feature extraction compared with a few typical algorithms.

1. Introduction

In recent years, various types of information, such as location information, search history, and videos, have been converted into numerical, categorical, or binary data. These data are often represented as vectors of a finite but high dimensionality. Unfortunately, it is difficult to directly observe high-dimensional data with the aim of understanding the data distribution and the mutual relationships that exist among the data. To address this issue, dimension reduction into a two- or three-dimensional space would be effective. Dimension reduction, which is not limited to a two- or three-dimensional space, is also useful to remedy the *curse of dimensionality*, a common obstacle in regression and classification problems.

On the other hand, multi-label classification (MLC) has been gathering a great deal of attention recently (for example, see [1]). MLC has a wide range of applications including text categorization, image annotation, and medical diagnosis. Unlike in single-label classification, in MLC, it is important to deal with label correlation and label-specific feature appropriately [2,3]. In addition, label imbalance problems become more remarkable when the number of labels increases (extreme multi-label classification) [4]. Some works have been already tackling this problem [5,6].

Among many visualization methods proposed thus far, Laplacian eigenmaps (LEs) [7] are one of the most effective. The popularity comes from the following: 1) the criterion requiring that a close pair should remain close even after a low-dimensional mapping is intuitive, 2) it is capable of representing a manifold structure of data hidden in a high-dimensional space, and 3) the locality-preserving character makes it insensitive to outliers and noise. LEs were originally proposed as unsupervised visualization methods that do not use class label information. Subsequently, supervised LE algorithms (SLEs) were proposed [8–10], but they are problematic in that the optimal parameter setting strongly depends on the dataset. In addition, they cannot deal with multi-label data. This motivated us to develop another supervised Laplacian eigenmap algorithm for multi-label data (SLE-ML) [11]. This algorithm can deal with multi-label data as well as single-label data. The balance parameter controlling the ratio of the amount of label information to that of feature information is intuitive and problem-independent. However, the mapping of SLE-ML as an

☆ source code is available at https://github.com/KudoMineichi/KSLE-ML

* Corresponding author.
E-mail address: mine@ist.hokudai.ac.jp (M. Kudo).

[1] She is now with Accenture Japan Ltd.

https://doi.org/10.1016/j.patcog.2021.108399

embedding is implicit rather than explicit; thus, similar to other methods [8–10], this mapping cannot be applied to newly arrived data. This is widely known as the "out-of-sample problem".

In this paper, we propose a method that uses a kernel trick to realize the mapping explicitly. This method is named the Kernelized Supervised Laplacian Eigenmap algorithm for Multi-Label data (KSLE-ML). We also discuss the optimal kernel selection and parameter setting. Finally, we demonstrate the realization of separability-guided feature extraction using KSLE-ML.

The contributions of this study are as follows:

- We show that Laplacian eigenmaps can be explicitly realized by kernels and that such mappings are exchangeable between kernels with nonsingular Gram matrices.
- We experimentally show that parameter selection is more important than kernel selection in KSLE-ML; this ensures that the RBF (Radial Basis Function) kernel can be used solely for the general purpose.
- We propose a method for separability-guided feature extraction that is based on the high-separability of classes in 2D visualization.
- We empirically confirm the effectiveness of the separability-guided feature extraction by showing that the separability is retained even for mapping newly arrived samples without class labels in KSLE-ML, and by showing that the separability increases as the number of samples or the mapping dimension or both increase.

2. Related Work

In this section, we provide an overview of typical visualization methods and Laplacian eigenmaps. A comprehensive survey of LEs is available in the literature [14].

2.1. Visualization methods

Many visualization methods that have been proposed thus far can be divided into unsupervised and supervised methods. These methods can be further categorized as either linear or nonlinear methods. Herein, we list one method from each of these categories. PCA [15] is a representative unsupervised linear method that maximizes the variance of mapped data. LDA [16] is a supervised linear method that minimizes the ratio of within-class variance to between-class variance in mapped data. t-SNE [17] is an unsupervised nonlinear method that makes the distance distribution of data pairs in the original space close to that in the mapped spaces in KL divergence. UMAP [18] is an unsupervised/supervised nonlinear method that aims to preserve the topological structure before and after mapping. Compared with t-SNE, UMAP produces more effective clustering and is faster. A detailed comparison of additional methods, including LEs, is available elsewhere [19]. In this study, we focused on LEs because of their simplicity and the excellent results that they have yielded thus far. The original LE [7] is an unsupervised nonlinear method.

2.2. (Unsupervised) Laplacian Eigenmaps (LEs) [7]

LE [7] first expresses the adjacency relationship between data as a weighted graph and then performs a nonlinear mapping of data to a low-dimensional space using the Laplacian eigenvectors of the graph. LE can be considered to be a manifold learning method. Given n data points $\{\boldsymbol{x}_i\}_{i=1}^n$ in a high-dimensional space $\mathbb{R}^M$, the original LE maps them into points $\{\boldsymbol{z}_i\}_{i=1}^n$ in a low-dimensional space $\mathbb{R}^m$ ($m \ll M$) on the basis of the neighboring relationship represented by $\{w_{ij}(\geq 0)\}_{i,j=1}^n$ over $\{\boldsymbol{x}_i\}_{i=1}^n$, to minimize

$$J_{LE} = \sum_i \sum_j ||\boldsymbol{z}_i - \boldsymbol{z}_j||^2 w_{ij}. \tag{1}$$

This formulation corresponds to the graph Laplacian with the adjacency relation $W = (w_{ij})$. Typically, W is given by

$$w_{ij} = \begin{cases} \exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/\sigma^2) & (\boldsymbol{x}_i/\boldsymbol{x}_j \in \mathrm{kNN}(\boldsymbol{x}_j/\boldsymbol{x}_i)) \\ 0 & (\text{otherwise}) \end{cases},$$

where $\boldsymbol{x}_i/\boldsymbol{x}_j \in \mathrm{kNN}(\boldsymbol{x}_j/\boldsymbol{x}_i)$ indicates that $\boldsymbol{x}_i$ is a member of the k nearest neighbors of $\boldsymbol{x}_j$ or $\boldsymbol{x}_j$ is a member of the k nearest neighbors of $\boldsymbol{x}_i$. Let $D = \mathrm{diag}(\sum_j w_{1j}, \sum_j w_{2j}, \ldots, \sum_j w_{nj})$ and $L = D - W$. Furthermore, let Z be an $n \times m$ matrix with $\boldsymbol{z}_i^T$ ('T' denotes the transpose) in the ith row. Then, J_{LE} becomes

$$J_{LE} = 2\,\mathrm{tr}\, Z^T L Z.$$

Note that L is positive semi-definite from (1). We solve this minimization problem, subject to $Z^T D Z = I$. The solution is obtained by solving the generalized eigenvalue problem:

$$L\mathbb{z}_j = \lambda_j D\mathbb{z}_j, \;\; j = 1, \ldots, m, \;\; \text{for } Z = (\mathbb{z}_1, \ldots, \mathbb{z}_m), \tag{2}$$

where $0 = \lambda_0 \leq \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_m$. Avoiding the trivial eigenvector of $\mathbf{1}$ with $\lambda_0 = 0$, the eigenvectors corresponding to $\lambda_1, \lambda_2, \ldots, \lambda_m$ are used for Z column-wise.

The eigenvalue problem (2) can be rewritten as

$$\tilde{L}\mathbb{u}_j = \lambda_j \mathbb{u}_j, \;\; \mathbb{z}_j = D^{-\frac{1}{2}}\mathbb{u}_j, \;\; j = 1, \ldots, m, \tag{3}$$

where $\tilde{L} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$.

2.3. Supervised Laplacian Eigenmaps (SLEs)

Unsupervised LEs have been extended to supervised LEs (SLEs) using several methods [8–10]. CCDR [8] introduces a hypothetical class center and sets the weights between it and the samples of the class to one in the criterion (1). S-LE [9] and S-LapEig [10] reduce the distances of same-class data pairs from the actual distances in the original space. Note that S-LE and S-LapEig are strongly affected by the ratio of same-class data pairs to all data pairs; they become less effective when the number of classes is large.

SLE-ML [11] combines the feature information and label information into one and uses a single parameter to control the balance between these two types of information. In addition, SLE-ML can process multi-label data, where data points $\{\boldsymbol{x}_i\}_{i=1}^n$ are given with the multi-label information $\{\boldsymbol{y}_i \in \{0,1\}^L\}_{i=1}^n$. Here, L is the number of (class) labels. SLE-ML minimizes

$$\begin{aligned} J_{SLE-ML} &= \sum_{i,j} ||\boldsymbol{z}_i - \boldsymbol{z}_j||^2 w_{ij} \\ &= \sum_{i,j} ||\boldsymbol{z}_i - \boldsymbol{z}_j||^2 \left(\lambda w_{ij}^F + (1-\lambda) w_{ij}^L\right), \end{aligned} \tag{4}$$

where $0 \leq \lambda \leq 1$, $w_{ij} = \lambda w_{ij}^F + (1-\lambda) w_{ij}^L$,

$w_{ij}^F = \left(\mathbb{1}(\boldsymbol{x}_i \in kNN(\boldsymbol{x}_j)) + \mathbb{1}(\boldsymbol{x}_j \in k + -NN(\boldsymbol{x}_i))\right)/2$, and

$$w_{ij}^L = \frac{|\boldsymbol{y}_i \wedge \boldsymbol{y}_j|}{|\boldsymbol{y}_i \vee \boldsymbol{y}_j|}. \tag{5}$$

Here, the superscripts 'F' and 'L' denote the *feature space* and *label space*, respectively. In addition, w_{ij}^L is the Jaccard similarity coefficient, that is, the ratio of the number of common labels to that of the union of their labels, and takes a value between 0 and 1. For a single-label problem,$w_{ij}^L = 1$ if data points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ share a label, and $w_{ij}^L = 0$ otherwise. The original LE is a special case of SLE-ML when $\lambda = 1$. The value of λ controls the extent to which the feature information is prioritized against the label information, such that λ is independent of the particular problem. An example of this is shown in Fig. 1.

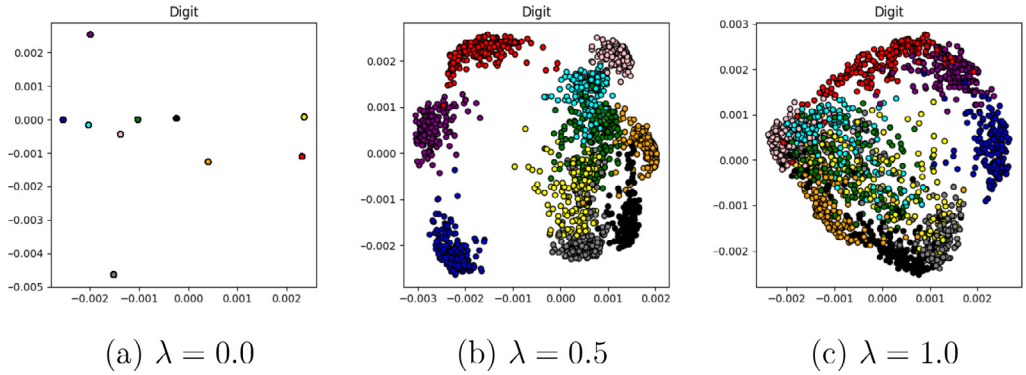


Fig. 1. Effect of the balance parameter λ in SLE-ML [11] in `digits` (see Table 1 and Fig. 2).

Fig. 2. Examples of images included in the datasets.

Table 1
Details of the five datasets used in our experiments. The numbers in parentheses are the numbers of distinctly labeled subsets.

Dataset	#samples	#classes	#samples per class	#features	domain
Digits	1797	10	178 182 ⋯ 180	64	images
CIFAR-10	60000	10	6000	3072	images
FashionMNIST	70000	10	7000	784	images
Scene	1211	6 (14)	194 165 ⋯ 1	294	images
Emotions	593	6 (27)	72 12 ⋯ 12	72	music

3. Out-of-Sample Problem and Kernel Trick Solution

In this section, we will not distinguish between supervised-LE and unsupervised-LE, because the label information w_{ij}^L is incorporated into w_{ij} in (4) with a balance parameter λ in SLE-ML.

Many previously proposed LEs perform mapping implicitly, that is, an embedding; hence, they cannot map newly arrived data (the "out-of-sample" problem).

A technique to solve the out-of-sample problem has been proposed for unsupervised LE [12]. It realizes an explicit mapping using a linear combination of kernels associated with every sample. The kernel matrix is given as $\tilde{L} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$ in (3) and the (i, j)th element of $\tilde{L}$ is denoted by $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. When the kernel matrix is updated with a new sample $\boldsymbol{x}$ as the $(n+1)$th sample, to find the corresponding jth mapping value u_j $(j = 1, 2, \ldots, m)$, according to (3), we have to solve the eigenvalue problem

$$\begin{pmatrix} \tilde{L} & K(\boldsymbol{x}) \\ K^T(\boldsymbol{x}) & k(\boldsymbol{x}, \boldsymbol{x}) \end{pmatrix} \begin{pmatrix} \mho_j \\ u_j \end{pmatrix} = \lambda_j \begin{pmatrix} \mho_j \\ u_j \end{pmatrix}, \tag{6}$$

where $K(\boldsymbol{x}) = (k(\boldsymbol{x}, \boldsymbol{x}_1), k(\boldsymbol{x}, \boldsymbol{x}_2), \ldots, k(\boldsymbol{x}, \boldsymbol{x}_n))^T$. Here, we assume that $\mho_j$ and λ_j are the eigenvector and eigenvalue of $\tilde{L}$, respectively. The idea is based on the convergence of an n-dimensional vector to a function such that

$$\frac{1}{n} K(\boldsymbol{x})^T \mho \to K_p u := \int k(\boldsymbol{x}, \boldsymbol{y}) u(\boldsymbol{y}) p(\boldsymbol{y}) d\boldsymbol{y},$$

for some probability density $p(\boldsymbol{x})$, where $\mho = (u_1, \ldots, u_n)^T = (u(\boldsymbol{x}_1), \ldots, u(\boldsymbol{x}_n))^T$, $\forall i\ \boldsymbol{x}_i \sim p(\boldsymbol{x})$ (the law of large numbers). When $u(\boldsymbol{x})$ is an eigenfunction such that $K_p u = \mu u$, the convergence property derives an approximation of the linear system (6) as

$$\begin{pmatrix} \tilde{L} & \mathbf{0} \\ K^T(\boldsymbol{x}) & 0 \end{pmatrix} \begin{pmatrix} \mho_j \\ u_j \end{pmatrix} = \lambda_j \begin{pmatrix} \mho_j \\ u_j \end{pmatrix}.$$

See [12] for details. This leads to the derivation of 1) $\tilde{L}\mho_j = \lambda_j \mho_j$ and 2) $K(\boldsymbol{x})^T \mho_j = \lambda_j u_j$, so we have $u_j = (1/\lambda_j) K(\boldsymbol{x})^T \mho_j$. The realized explicit function is $u_j = (1/\lambda_j) K(\boldsymbol{x})^T \mho_j$ $(j = 1, 2, \ldots, m)$. The idea is attractive, but the treatment of $K(\boldsymbol{x})$ is not consistent with that of $K(\boldsymbol{x}_i)$, the ith column of $\tilde{L}$, $i = 1, 2, \ldots, n$, if we retain property 1). This is because all the training samples $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$ are candidates of the k nearest neighbors of $\boldsymbol{x}$, whereas any $\boldsymbol{x}_i$ does not consider $\boldsymbol{x}$ as a candidate. Conversely, if we update $\tilde{L}$ with $\boldsymbol{x}$, then the eigenvalue λ_j and eigenvector $\mho_j$ will change. It also incurs the cost of sorting for finding the k nearest neighbors, in addition to the distance calculation with respect to $\boldsymbol{x}$.

Our idea that will be shown soon is similar to the method of [12] but different essentially in two points. First, the method of [12] is based on updating of the kernel matrix $\tilde{L}$ for mapping a new data, assuming a sufficient number of training samples. On the contrary, we simulate the implicit but already realized mapping in an explicit way instead of updating $\tilde{L}$. Therefore, our method is valid even for a limited number of training samples. Second, their kernel depends on the data, but ours is independent of the data. Therefore we can select kernels optimally based on some criterion. Moreover, the computational complexity of kernels is lower than that in the case of the abovementioned method.

With data-independent kernels, we propose a kernelized supervised LE for multi-label data, KSLE-ML, which simulates the embedding explicitly to enable new data to be mapped in the same manner. The concept is as follows. We map the samples from $\mathbb{R}^M$ to a higher-dimensional space $\mathbb{R}^{\mathcal{M}}$, typically an infinite-dimensional space, using a nonlinear mapping ϕ, and then map the data to a lower-dimensional space, typically $m = 2$ or 3, using a linear mapping:

$$\boldsymbol{x}(\in \mathbb{R}^M) \xrightarrow{\text{nonlinear}} \boldsymbol{\phi}(\boldsymbol{x})(\in \mathbb{R}^{\mathcal{M}}) \xrightarrow{\text{linear}} \boldsymbol{z}(\in \mathbb{R}^m).$$

Specifically, using a reproducing kernel $k(\cdot,\cdot)$, we map $\boldsymbol{x}$ to $\phi(\boldsymbol{x}) = k(\cdot,\boldsymbol{x})$ in the reproducing kernel Hilbert space (RKHS) $\mathcal{H}_k$, where the inner product is given by $\phi(\boldsymbol{x})^T\phi(\boldsymbol{y}) = k(\boldsymbol{x},\boldsymbol{y})$ (kernel trick) ([13], Chap. 14). We limit ourselves into $\mathrm{Span}(k(\cdot,\boldsymbol{x}_1), k(\cdot,\boldsymbol{x}_2), \ldots, k(\cdot,\boldsymbol{x}_n)) \subseteq \mathcal{H}_k$ and consider a function f on the subspace as a linear combination $f(\boldsymbol{x}) = \sum_{i=1}^n c_i k(\boldsymbol{x},\boldsymbol{x}_i)$. Then, m such functions enable us to map $\boldsymbol{x}$ to $\boldsymbol{z} = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x}))^T$. This can be written in a matrix form as :

$$\boldsymbol{z} = CK(\boldsymbol{x}) = \begin{pmatrix} \mathbb{c}_1^T \\ \mathbb{c}_2^T \\ \vdots \\ \mathbb{c}_m^T \end{pmatrix} \begin{pmatrix} k(\boldsymbol{x},\boldsymbol{x}_1) \\ k(\boldsymbol{x},\boldsymbol{x}_2) \\ \vdots \\ k(\boldsymbol{x},\boldsymbol{x}_n) \end{pmatrix} \in \mathbb{R}^m, \tag{7}$$

where $\mathbb{c}_j \in \mathbb{R}^n \quad (j = 1, 2, \ldots, m)$ is a coefficient vector. From (7), the objective function (1) or (4) can be represented as

$$\begin{aligned} J &= \sum_{i,j} ||CK(\boldsymbol{x}_i) - CK(\boldsymbol{x}_j)||^2 w_{ij} \\ &= 2\,\mathrm{tr}\, C\mathbb{K}^T L \mathbb{K} C^T, \end{aligned} \tag{8}$$

where $\mathbb{K}$ is the Gram matrix of size $n \times n$:

$$\mathbb{K} = (K(\boldsymbol{x}_1), K(\boldsymbol{x}_2), \ldots, K(\boldsymbol{x}_n)) = (k(\boldsymbol{x}_i,\boldsymbol{x}_j)).$$

Here, recall that we express Z in a row- and column-wise manner as

$$Z = \begin{pmatrix} \boldsymbol{z}_1^T \\ \boldsymbol{z}_2^T \\ \vdots \\ \boldsymbol{z}_n^T \end{pmatrix} = (\mathbb{z}_1, \mathbb{z}_2, \ldots, \mathbb{z}_m) = \mathbb{K}C^T. \tag{9}$$

The solution C that minimizes (8), subject to $C\mathbb{K}^T D \mathbb{K} C^T = I$, is obtained by solving the generalized eigenvalue problem, corresponding to (2),

$$(\mathbb{K}L\mathbb{K})\mathbb{c}_j = \lambda_j (\mathbb{K}D\mathbb{K})\mathbb{c}_j. \tag{10}$$

Note that $\mathbb{K}^T = \mathbb{K}$. For visualization, we display $\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n\}$ in $m = 2$ or 3. It is worth nothing that a linear relationship exists between $\mathbb{z}_j$ and $\mathbb{c}_j$ both belonging to $\mathbb{R}^n$. Indeed, from (7) and (9), we have

$$\mathbb{z}_j = \mathbb{K}\mathbb{c}_j \quad (j = 1, \ldots, m). \tag{11}$$

Thus, if $\mathbb{K}$ is invertible, the mapping realized by the kernel is the same as that realized by the LE. In fact, we can calculate $\mathbb{c}_j = \mathbb{K}^{-1}\mathbb{z}_j$ directly from $\mathbb{z}_j$, which is obtained by solving (2). Conversely, the use of (11) allows us to calculate $\mathbb{z}_j$ from $\mathbb{c}_j$, which is obtained by solving (10) for any kernel. This also means that $\mathbb{c}_j$ and $\mathbb{c}'_j$ are exchangeable between two different kernels $k(\cdot,\cdot)$ and $k'(\cdot,\cdot)$, or different parameters in the same kernel, such as $\mathbb{c}_j = \mathbb{K}^{-1}\mathbb{K}'\mathbb{c}'_j$.

3.1. Kernels and Parameters

After the out-of-sample problem is solved, the next challenge is to choose the kernel and the parameters optimally. To achieve this goal, we compare three kernels: an RBF (Gaussian) kernel, a polynomial kernel, and a sinc kernel, which are defined as

1) **RBF (Gaussian) kernel**

$$k(\boldsymbol{x},\boldsymbol{y}) = \exp\left(-\frac{||\boldsymbol{x}-\boldsymbol{y}||^2}{\sigma^2}\right),$$

2) **Polynomial kernel**

$$k(\boldsymbol{x},\boldsymbol{y}) = \left(\frac{\boldsymbol{x}^T\boldsymbol{y}}{M} + 1\right)^g, \quad \text{and}$$

3) **Sinc kernel**

$$k(\boldsymbol{x},\boldsymbol{y}) = \frac{\sin\sigma(||\boldsymbol{x}-\boldsymbol{y}||)}{\sigma\pi||\boldsymbol{x}-\boldsymbol{y}||}.$$

Here, M is the dimensionality of the original feature space. In the following experiment, we compare these kernels with certain parameter values.

3.2. Nonsingularity of Gram matrices

Eq. (10) has already been derived by following a different route in locally preserving projection (LPP) [20]. Linear LPP was first proposed and then extended to nonlinear kernelized LPP (KLPP) using a kernel trick [21,22]; thus, the LPP did not experience the out-of-sample problem from the beginning. Instead, LPP does not always optimally comply with the LE criterion (1) or (4). In contrast, starting from the implicit nonlinear mapping of LEs, we obtain (10) to solve the out-of-sample problem. This route differs from that of LPP and led to the finding that the nonsingularity of the Gram matrix is a sufficient condition for the mapping to be the same as that of the LE. In formal terms,

Proposition 1. *If the Gram matrix over n samples is nonsingular, then the function realized by linear combinations of the kernels (7) maps those samples as the same as LE or SLE ((1) or (4)).*

The question that then arises is under which condition the Gram matrix is nonsingular (invertible). In general, if a set $\{k(\cdot,\boldsymbol{x}_1), k(\cdot,\boldsymbol{x}_1), \ldots, k(\cdot,\boldsymbol{x}_n)\}$ is linearly independent, then the Gram matrix $\mathbb{K} = (k(\boldsymbol{x}_i,\boldsymbol{x}_j))$ is nonsingular. Indeed, because of this property, RBF (Gaussian) kernel is proven to be always nonsingular, regardless of the value of the parameter σ^2, as long as the samples are all distinct [23]. As for the sinc kernel, a more intuitive analysis is possible. We notice that the Gram matrix $\mathbb{K}$ approaches to the identity matrix I_n as the parameter σ approaches infinity. Because the determinant of $\mathbb{K}$ is continuous in σ and the determinant of the identity is one, a value of σ exists such that the determinant of $\mathbb{K}$ is nonzero for this or larger values of σ. Therefore, there is a value of σ to make $\mathbb{K}$ nonsingular. This analysis is also valid for an RBF kernel. For polynomial kernels, we know empirically that the Gram matrix $\mathbb{K}$ becomes nonsingular when we assign a large value of dimension g in many cases, but we are not aware of a theoretical result. Conversely, it can be shown that $\mathbb{K}$ is singular if $n > \binom{M+g}{g}$ [23].

Each of the three kernels described above forms a family of nested Hilbert spaces. The RBF kernel has a *variance* parameter σ^2 and has the following property: $\mathcal{H}_{\sigma_1} \subseteq \mathcal{H}_{\sigma_2}$ if $\sigma_2 \leq \sigma_1$ [24], where $\mathcal{H}_\sigma$ is an infinite-dimensional RKHS with σ. The polynomial kernel with a *dimension* parameter g produces an RKHS of a finite dimension that is not larger than $\binom{M+g}{g}$. The sinc kernel has a *cut-off frequency* parameter σ and produces an RKHS of an infinite dimension.

In general, the narrower RKHS is, the more *stable* the mapping is. Here, we consider a mapping to be *stable* when it maps the training and testing samples in a similar manner. We confirm this in the subsequent sections.

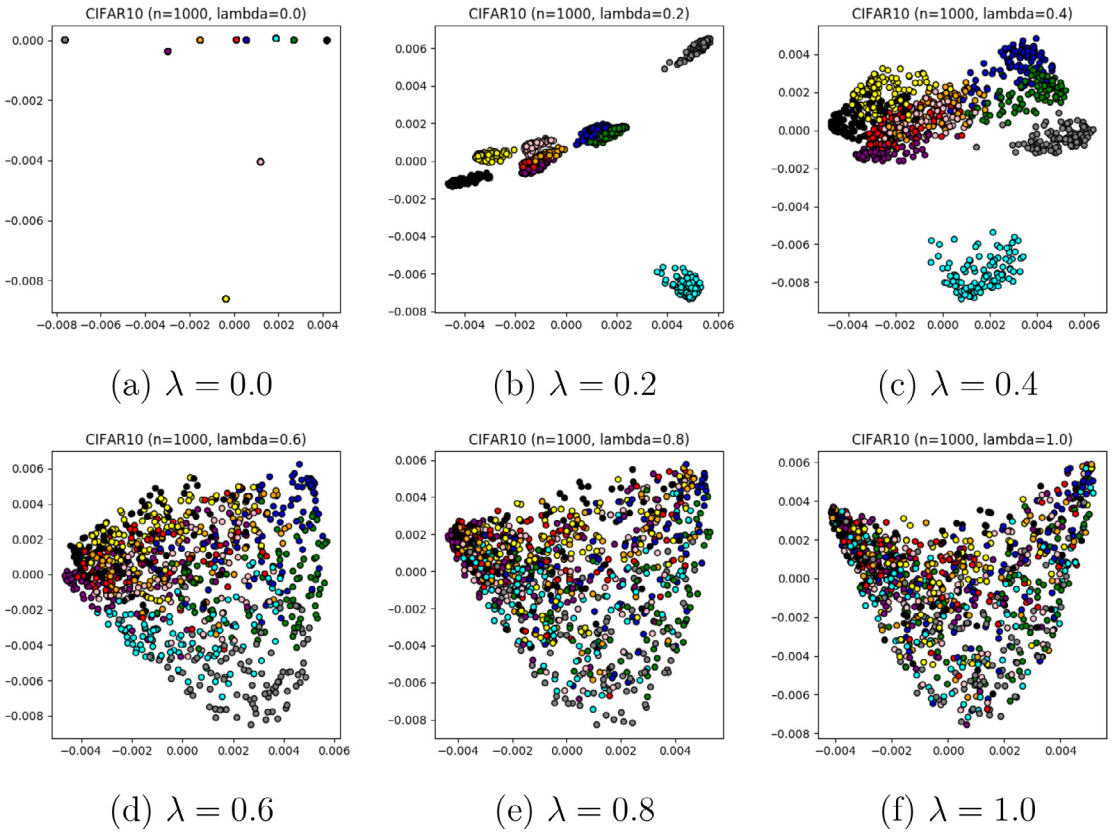


Fig. 3. Effect of the balance parameter λ on the mapping performance of SLE-ML/KSLE-ML on `CIFAR-10`.

4. Experiments

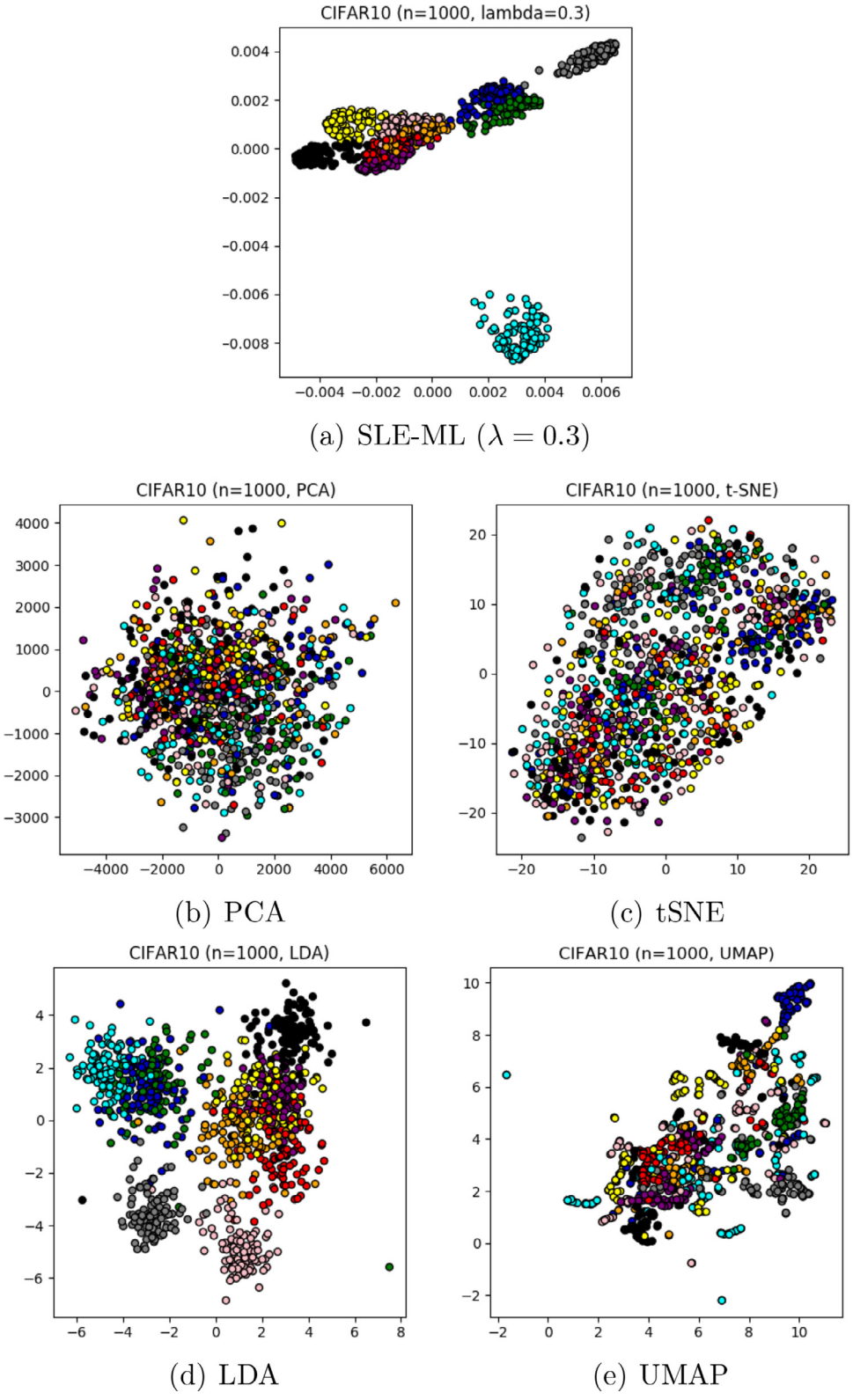
4.1. Datasets

In the following experiments, we used five high-dimensional datasets, `Digits`, `FashionMNIST`, `CIFAR-10`, `Scene`, and `Emotions`. `Digits`, `CIFAR-10`, and `FashionMNIST` are single-label datasets, whereas `Scene` and `Emotions` are multi-label datasets. The dataset `Digits` consists of 1797 images of handwritten digits (0–9). The dataset `CIFAR-10` consists of 60,000 32×32 color images (50,000 for training and 10,000 for testing) in 10 classes, with 6000 images per class. The classes in the dataset are {*airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck*}. The dataset `FashionMNIST` is a collection of labeled fashion images. Each sample is a 28×28 grayscale image associated with a label from 10 classes. This dataset contains 70,000 samples (60,000 for training and 10,000 for testing). The multi-label dataset `Scene` is a collection of 1,211 images of landscapes, with the average of 1.07 labels. The other multi-label dataset, `Emotions`, contains 593 songs categorized into six classes of emotions, with the average of 1.87 labels. Table 1 provides the details of these datasets. Examples of data in these datasets, other than `Emotions` are shown in Fig. 2 and Fig. 5.

4.2. Effectiveness in Visualization

First, we examined the changes in the visualization result for different values of λ in SLE-ML/KSLE-ML. In what follows, since the mapping is the same for the training samples between SLE-ML and KSLE-ML, we will use the term "SLE-ML" in the case when the testing samples are not being considered; otherwise, we will use the term "KSLE-ML". Fig. 3 is the visualization result of `CIFAR-10`. To facilitate observation, we sampled 1000 data randomly. The separation between classes and the local structure in each class are balanced differently according to the value of parameter λ. Two extreme cases were considered: all samples of a class are mapped to a single point for $\lambda = 0.0$, whereas LE is realized for $\lambda = 1.0$. The most effective balance was obtained for $\lambda = 0.2$ or 0.4 (Fig. 3 (b) and (c)). This shows the effectiveness of incorporating label information with the sample similarity in the feature space. Next, we compared SLE-ML with four representative algorithms: PCA [15], tSNE [17], LDA [16], and UMAP [18]. PCA and tSNE are unsupervised linear and nonlinear algorithms, respectively, LDA is a supervised linear algorithm, and UMAP is an unsupervised/supervised nonlinear algorithm. In this experiment, we used UMAP as the supervised algorithm. In these methods, the default values were used for their parameters, if any, other than $m = 2$. The parameter k for nearest neighbors in SLE-ML was set to 1.5 times the average number of samples in a class. We used this setting in SLE-ML throughout this study. The results for `CIFAR-10` are shown in Fig. 4. This dataset has a large dimensionality of $M = 3072$; therefore, the robustness of each algorithm to dimensionality was clarified. Except for SLE-ML and LDA, all the other algorithms fail to detect the separability that actually exists (as confirmed by the high accuracies of over 90% attained by CNNs [25]). With LDA, certain classes overlap to a larger extent than with SLE-ML. This is natural because LDA is an unsupervised method. In addition, it is difficult for LDA to extract a manifold structure, unlike for SLE-ML. tSNE and UMAP seem to fail to extract the global structure, such as the geometrical relationship between classes. Apart from these difficulties, although the results are not shown here, these two algorithms output different results in different trials owing to the nature of random algorithms. This is not preferable for a visualization method.

These results demonstrate the importance of reducing the degree of the effect of the curse of dimensionality by other informa-



(a) SLE-ML ($\lambda = 0.3$) (b) PCA (c) tSNE (d) LDA (e) UMAP

Fig. 4. Comparison of mapping algorithms on `CIFAR-10`.

tion or constraint, such as the label information used in SLE-ML and the linearity imposed in LDA.

In addition, we processed two multi-label datasets using SLE-ML, which is the only method applicable to multi-label datasets among the SLEs proposed so far. Fig. 5 and Fig. 6 show the mapping results of `Scene` ($\lambda = 0.4$) and `Emotions` ($\lambda = 0.2$), respectively. In both sets of results, samples with multiple labels are located between the samples with the corresponding single labels, for example, {Fall foliage, Fields} for {Fall foliage} and {Fields} in `Scene`, {relaxing, quiet} for {relaxing} and {quiet}, and {amazed, sad} for {amazed} and {sad} in `Emotions`. The mapping result for `Scene` exhibits significant overlap among {Urban}, {Mountain},

Fig. 5. Mapping result of data in `Scene` using SLE-ML/KSLE-ML ($\lambda = 0.4$).

and {Beach}; thus, these scenes would be expected to be difficult to distinguish. In `Emotions`, the group of negative emotions (angry and sad) is classified separately from that of positive emotions (happy and relaxing). The data in `Emotions` are labeled according to the Tellegen-Watson-Clark model [26]. This mapping faithfully represents the model in the pairs of opposite emotions: {happy} vs. {sad}, {relaxing} vs. {angry}, and {quiet} vs. {amazed}.

4.3. Kernel selection and parameter setting

Next, we conducted experiments for kernel and parameter selection. The criterion is the stability of the mapping realized by KSLE-ML. For $\lambda = 0.5$, we compared nine combinations of kernels and parameters (three kernels by three values of the parameter). The results were evaluated through human inspection and by the mean squared error (MSE) using 10-fold cross-validation. The MSE was obtained by calculating the location errors of the testing samples between the mapping learned only from the training samples and that learned from both the training and testing samples. The parameter k for nearest neighbors was set to 1.5 times the average number of samples in a class. In this experiment, we first determined $\mathbb{z}_j$ and then converted $\mathbb{z}_j$ to $\mathbb{c}_j$ by $\mathbb{c}_j = \mathbb{K}^{-1}\mathbb{z}_j$. We used the generalized Moore–Penrose inverse $\mathbb{K}^{+}$ instead of $\mathbb{K}^{-1}$ when $\mathbb{K}$ is singular.

The results are shown in Fig. 7. First, we note that the 10 randomly chosen testing samples without labels (one per class) are mapped fairly close to the training samples of the same class independently of the kernels if the parameter is chosen appropriately (the color of the testing samples is added after mapping). Second, we note that the best results are similar independently

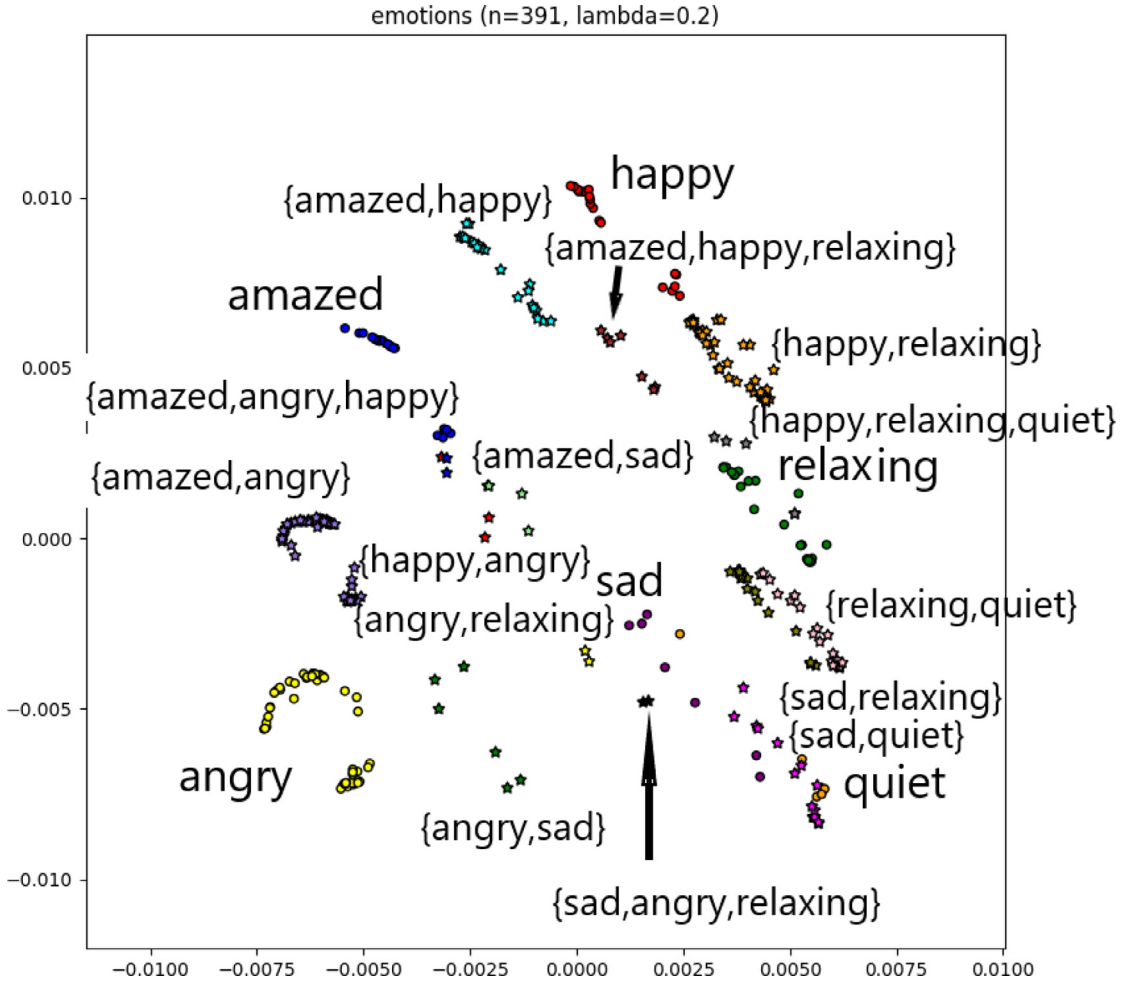


Fig. 6. Mapping result of data in `Emotions` using SLE-ML/KSLE-ML ($\lambda = 0.2$).

of kernels as long as their parameter values are optimally chosen; this indicates that the parameter selection is more important than the kernel selection. Finally, the parameter value corresponding to the minimal RKHS (the leftmost figure with mark (R) in each row) in a nested family appears to be a suitable choice, provided that the Gram matrix is nonsingular ((a) for RBF, (e) for polynomial, and (h) for sinc in Fig. 7). This criterion is almost consistent with the MSE minimization criterion except for the sinc kernel, for which the MSE is approximately eight times larger than that of the other two kernels. Note that in the singular cases of the Gram matrix (Fig. 7(d),(g)), the mapping of the training samples is slightly different from that in the nonsingular case in Fig. 7; that is, SLE-ML is not perfectly simulated by KSLE-ML, as predicted in Prop. 1.

In the RBF kernel with σ^2, the Gram matrix is always nonsingular; however, because a very large value of σ^2 can cause a numerical problem, we propose, as a rule of thumb,

$$\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{k}\sum_{\boldsymbol{x}_j \in kNN(\boldsymbol{x}_i)} ||\boldsymbol{x}_j - \boldsymbol{x}_i||^2. \tag{12}$$

This is the average of the local variances. In the following, we use the RBF kernel with this estimated variance.

4.4. Separability control

Next, we consider methods to select the value of the balance parameter λ in KSLE-ML after the kernel and parameter have been determined. The smaller the value of λ is, the larger the class separability is obtained. However, a smaller value of λ results in the loss of a greater amount of information on the feature space. In KSLE-ML, because the mapping is realized using only the information of the feature space, more specifically, by $\mathbb{K}$ over $\{\boldsymbol{x}_i\}_{i=1}^n$ and a coefficient matrix C, the mismatch between the training samples and testing samples becomes larger for a smaller value of λ. This is clearly visible in Fig. 8. Therefore, to determine the value of λ, we recommend choosing the largest value for which a necessary separability is maintained through human inspection ($\lambda = 0.2$ or $\lambda = 0.4$ in Fig. 8). Alternatively, we may rely on the classification performance or the MSE of the mapping in cross-validation. Hereafter, we typically use $\lambda = 0.3$.

4.5. Separability-Guided Feature Extraction

Finally, we consider the availability of KSLE-ML for feature extraction in classification. The idea is as follows. Once samples are mapped in a low-dimensional (not limited to two) space such that their class separability is sufficiently preserved, the classification performance of a classifier carried out in the space is expected to be high. To confirm this, we conducted classification experiments in a low-dimensional space using KSLE-ML to map samples, with $\lambda = 0.3$. In the mapped space of KSLE-ML, we used the nearest mean (centroid) classifier. The motivation of this subsection is to give alternative way for finding discriminative features instead of CNN or the other type of neural networks.

We first examined the variation in the classification rate by varying the number of the training samples n and the mapping di-

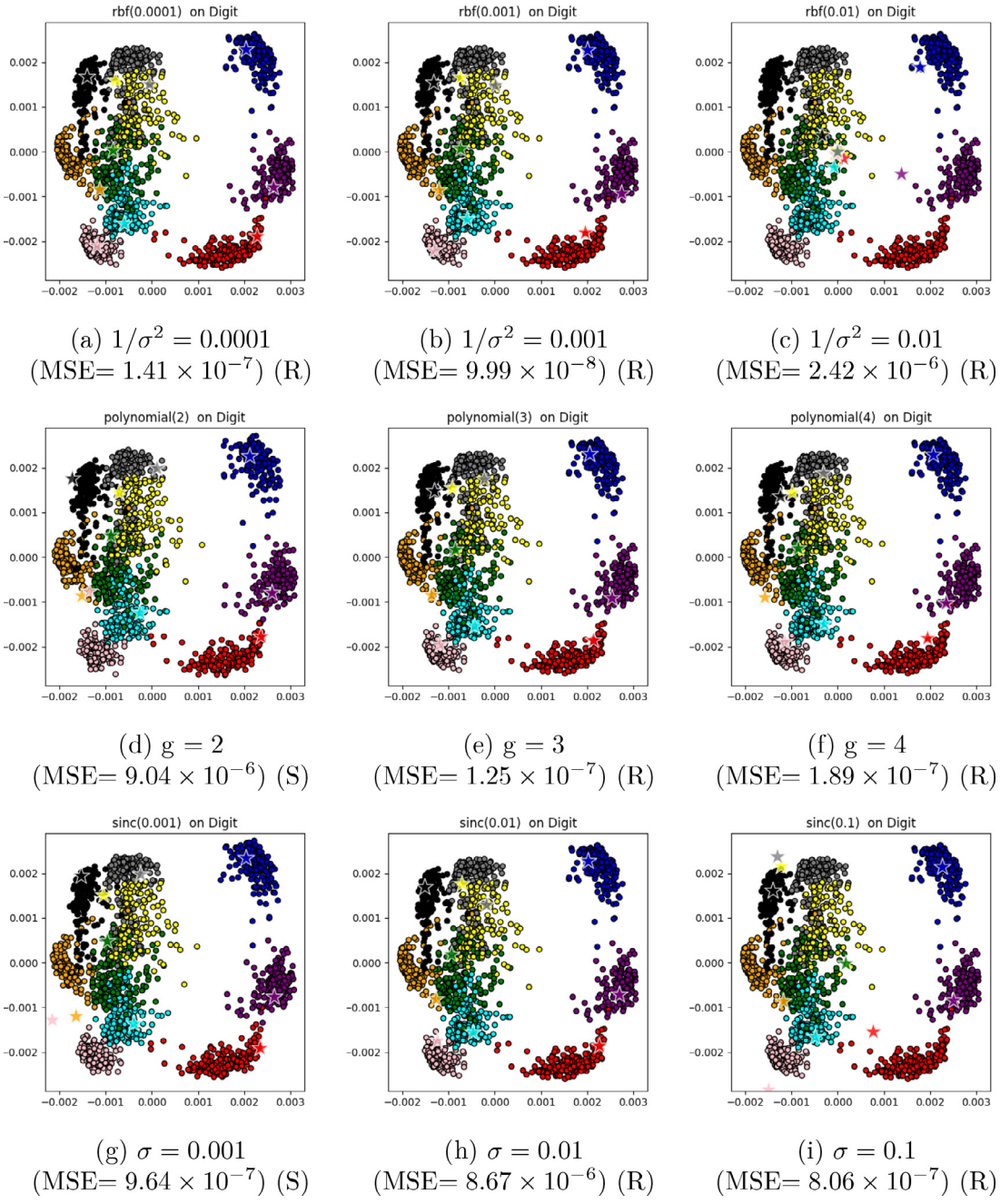


Fig. 7. Mapping results of ten testing samples (one per class) for nine (kernel, parameter) pairs in `digit`, with each of the different samples in each mapping represented by a star. The circles represent the other training samples that were used to design the KSLE-ML mapping. In each row (RBF, polynomial, and sinc, from top to bottom), the models are enlarged from left to right in a nested family. MSE represents the mean squared error for the 10-fold cross-validation, and (S)/(R) indicates whether the Gram matrix is singular/nonsingular.

mension m in KSLE-ML. The training samples were randomly sampled to obtain the required number of samples n. The results for `CIFAR-10` and `FashionMNIST` are shown in Fig. 9. The more training samples and the larger the dimension we used, the higher the classification rate was obtained. The results show that an increase in the mapping dimension is more effective, as seen in Fig. 9. The effect of increasing the dimension is shown in detail in Fig. 10. In Fig. 10, we can see that different classes are separated from the other classes in different feature pairs.

Because the recognition rate was confirmed to increase as the dimension increased, we set the number of dimensions to $m = L - 1$, where L is the number of classes. This dimension is necessary to distinguish L classes in a fair manner, for example, by placing each class center in a vertex of a $(L-1)$ – simplex, for example, a line segment for two classes or a triangle for three classes.

We compared the performance of KSLE-ML with $m = L - 1$ with that of three classifiers: Linear-SVM [27], 5-NN [28], and Random Forest [29]. These classifiers were carried out in the original high-dimensional space. The results for `CIFAR-10` and `FashionMNIST` (using 10,000 testing samples for each) are shown in Fig. 11. The performance of KSLE-ML is superior to that of these classifiers despite the fact that KSLE-ML uses a simpler classifier. The classification rates are not comparable to those achieved by recent deep neural networks, for example, 90.2% by CNN for `CIFAR-10` and 92.0% by LSTM for `FashionMNIST` [25]; this is

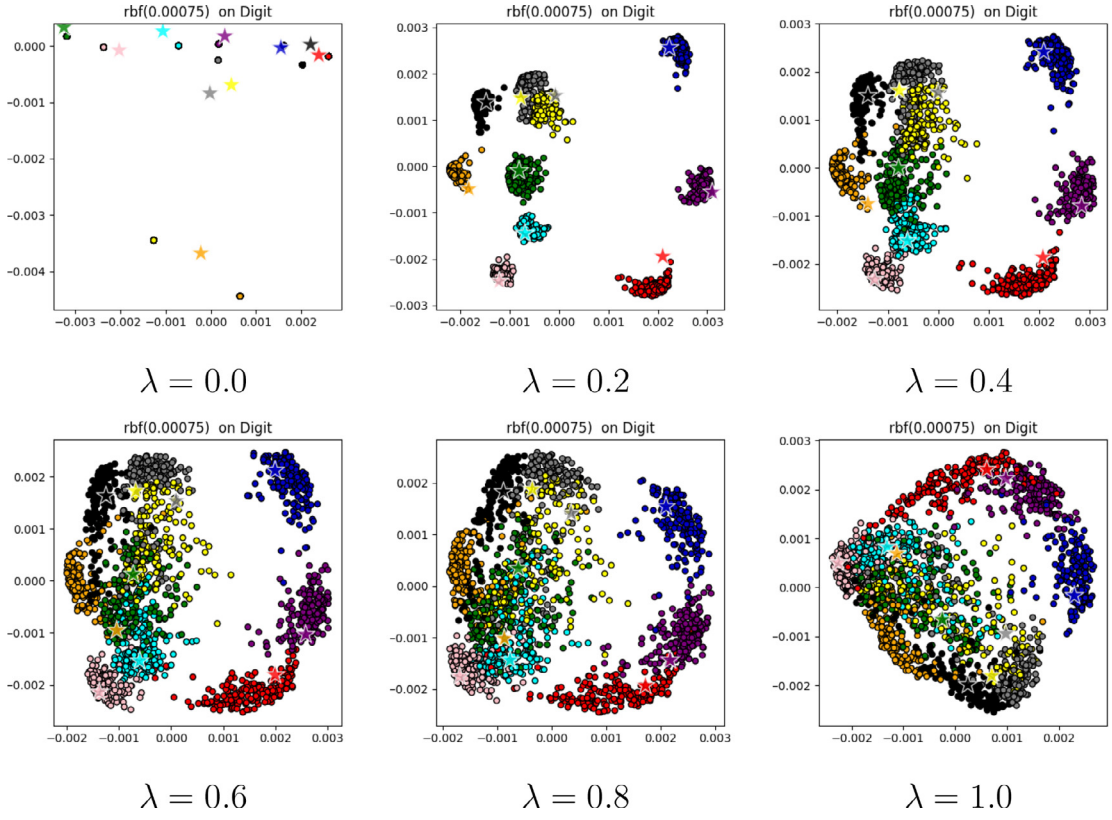


Fig. 8. Mapping results of 10 testing samples (represented by stars) for various values of λ in KSLE-ML with the RBF kernel of the estimated variance ($1/\sigma^2 = 0.00075$) in `digit`.

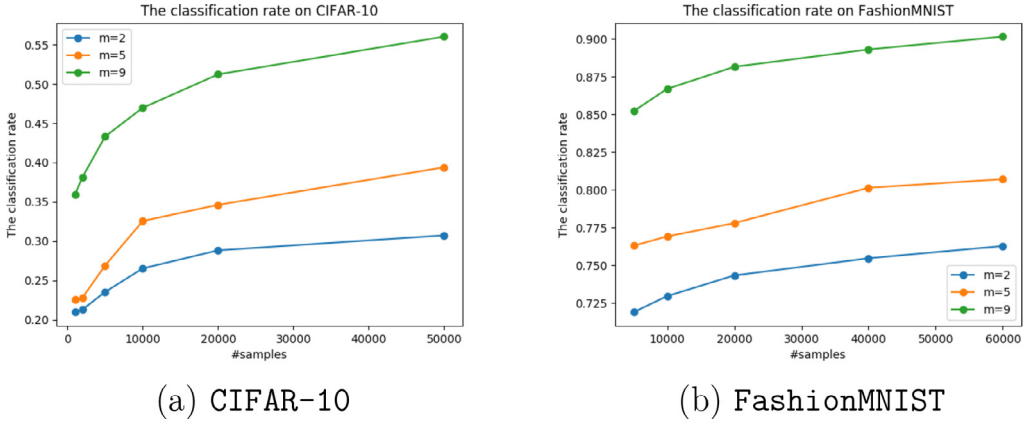


Fig. 9. The classification rates on two datasets for various pairs of n and m, where n is the number of training samples and m is the mapping dimension. The nearest mean classifier was applied to the data mapped by KSLE-ML with $\lambda = 0.3$.

probably because the convolution operation used in these networks works effectively.

Finally, we conducted a similar experiment using the multi-label dataset `Scene`. For evaluation, we used the following definition of $P@k$ to determine the classification accuracy:

$$P@k = \frac{1}{k} \sum_{l \in \mathrm{rank}_k(\hat{\boldsymbol{y}})} \boldsymbol{y}_l \ .$$

Here, $\boldsymbol{y} \in \{0,1\}^L$ is the true label vector, and $\hat{\boldsymbol{y}} \in \mathbb{R}^L$ is the predicted score vector. In this experiment, we compared our method with five typical multi-label classifiers: BR [30], CPLST [31], ECC [32], MLkNN [33], and MLSI [34]. The results are shown in Fig. 12, which shows that KSLE-ML outperforms the other classifiers, especially at P@1.

5. Discussion

We summarize the procedure for visualization ($m = 2$) with KSLE-ML and for separability-guided feature extraction:

1. (Choice of λ) First, using SLE-ML/KSLE-ML, choose the value of λ such that the class separability and the local structure are well-balanced in the two-dimensional mapping ($m = 2$). The default value is 0.3.
2. (Choice of kernel and parameter) Second, choose any kernel. The default is the RBF kernel. Then, choose the value of the kernel parameter such that the corresponding Hilbert space is minimal, subject to the nonsingularity of the Gram matrix. As a rule of thumb, use the average of the local variances for the

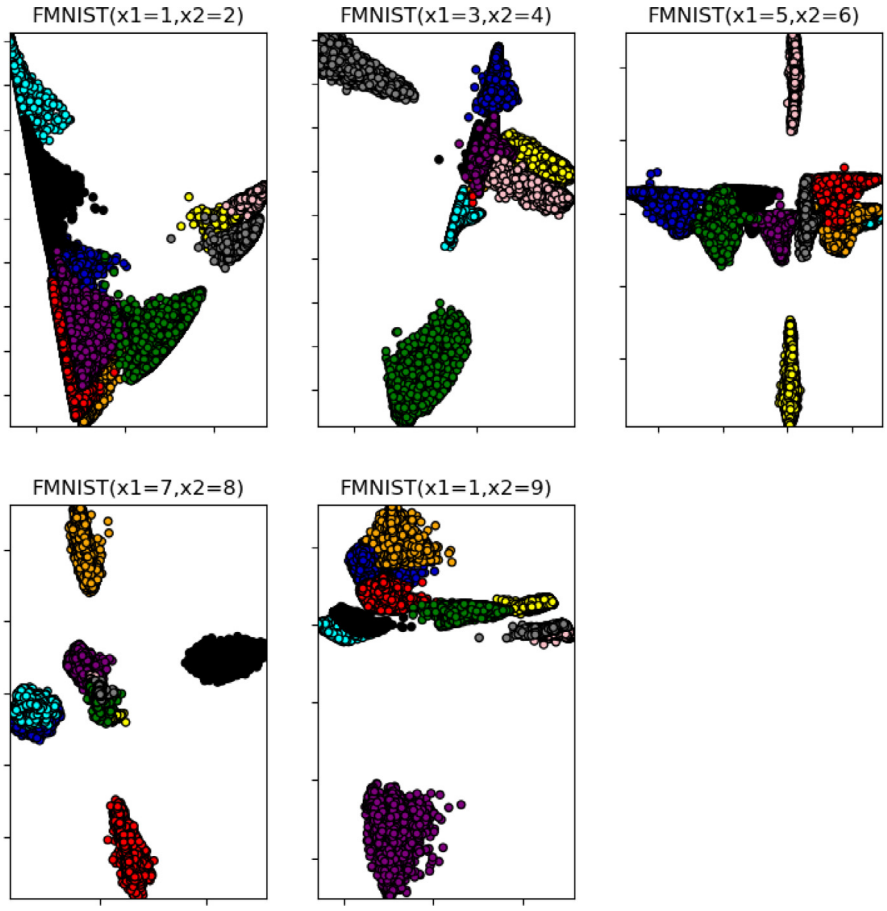


Fig. 10. Data distributions in different pairs of features in `FashionMNIST` ($n = 60,000$). Different classes are separated from the other classes in different feature pairs.

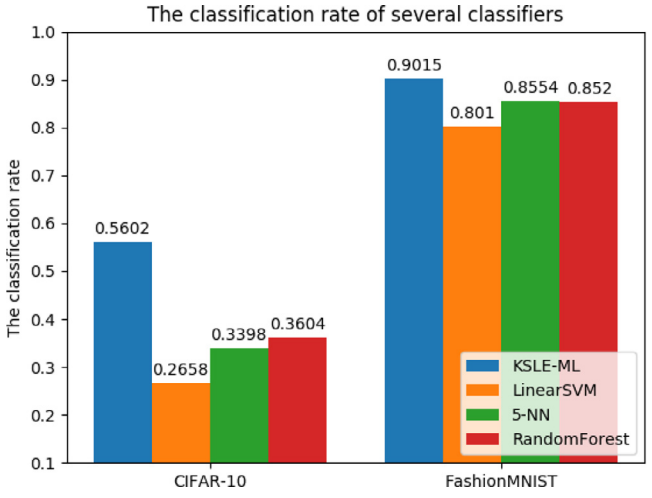


Fig. 11. Comparison of classification results for `CIFAR-10` and `FashionMNIST`. KSLE-ML denotes the nearest mean classifier executed on the nine-dimensional data ($m = 9$) mapped by KSLE-ML with $\lambda = 0.3$.

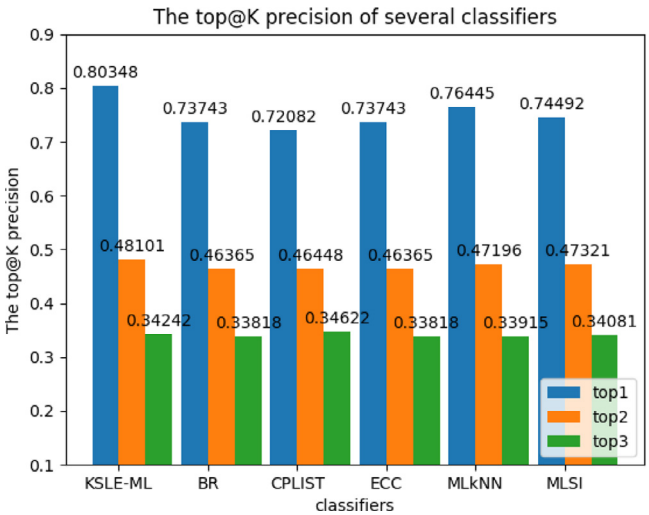


Fig. 12. Comparison of classifiers in 10-fold cross-validation on the multi-label `Scene` dataset. KSLE-ML denotes the nearest mean classifier executed on the five-dimensional data (the number of classes minus one) mapped by KSLE-ML with $\lambda = 0.3$. CPLST represents a label-space dimension reduction method, ECC represents an ensemble method, and MLkNN is a method based on kNN. Top@k is the relative ratio of the correct predictions among the k predictions.

RBF kernel because the Gram matrix is always nonsingular. Determine the coefficient matrix C in KSLE-ML.

3. (Feature extraction) Raise the value of m to the number of classes minus one and execute KSLE-ML again using the same values of λ and σ^2 as in Step 2. Re-calculate C for KSLE-ML.

Note that the use of the default values obviates the need to specify parameter values, which is a distinct advantage of KSLE-ML. Usually, it suffices to examine a few values of $\lambda = 0.2, 0.3, 0.4$, with other default values. However, this method is problematic in certain respects, most notably the time and space complexity. It requires $O(n^3)$ time, mainly for the inversion of a Gram matrix $\mathbb{K}$

of size n by n. Although it is possible to solve the linear equation $\mathbb{z}_j = \mathbb{K}\mathbb{c}_j$ directly without taking the inverse of $\mathbb{K}$, a clear advantage has not yet become apparent. It also requires $O(nM + n^2)$ memory, where M is the data dimensionality. In a PC with 128GB memory (Ubuntu 18.04, Intel(R) Xeon(R) Gold 6136 CPU, 3.00GHz), we were able to deal with at most $n = 60,000$ samples of $M = 789$ features extracted from `FashionMNIST`. The next problem is the difficulty of visualizing hundreds of thousands of samples, although this is a common problem with all visualization methods. In this study, we chose an appropriate number of samples using a random sampling; however, certain detailed information is lost when using this method. For both complexity reduction and visualization, we will research the possibility of resampling at the expense of non-singularity and that of adoption of an optimization framework with a penalty term to reduce the number of samples, in the future. Finally, special attention must be paid to the visualized class separability, because we could increase the separability to an unlimited extent by assigning a small value to λ. Hence, we recommend the use of the largest value of λ among the values that would result in a sufficient degree of class separability.

Techniques that are related to separability-guided feature extraction are known as *feature (set) embedding*. In this regard, two major approaches are LPP [20] and LLE [35]. As explained previously, LPP shares the objective function with LE. A similar objective function is adopted for LLE. First, we extract $\{w_{ij}\}$ by minimizing

$$J_{LLE} = \sum_i ||\boldsymbol{x}_i - \sum_j w_{ij}\boldsymbol{x}_j||^2.$$

Then, embedding is required to maintain this relation $\{w_{ij}\}$ by minimizing J_{LLE} with respect to $\{\boldsymbol{z}_i\}$. Their supervised variants are obtained by embedding the label information in the objective functions. However, these approaches do not enable the class separability to be controlled. In contrast, the proposed feature extraction method enables us to control the class separability in 2D visualization, which provides the basis for KSLE-ML feature extraction in a higher-dimensional space. Moreover, many feature embedding methods experience the out-of-sample problem and are unable to accommodate multi-label data. Further, there are neural networks available for finding discriminative features [36,37]. Using KSLE-ML is more advantageous in practical problems compared to using those neural networks, as these networks generally require more samples than those available. Hence, KSLE-ML is also advantageous for *extreme multi-label problems* [4] where minority classes contain very few samples.

In reality, in KSLE-ML feature extraction, the degree of class separability is evaluated on the subjective perception in 2D visualization, which is not objective. However, human perception is often superior to a single or a few values of the objective criteria, such as the estimated recognition rate and/or the MSE. For example, 2D visualization enables us to infer a class distribution from a limited number of the samples and to estimate the relative closeness among classes. In the proposed separability-guided feature extraction, we increased the dimension of the mapping, relying on the class separability observed through 2D visualization. Therefore, the effectiveness depends on the assurance that the difference in the mappings of the training and testing data is small and that at least the same degree of class separability is preserved even when the dimension increases. These things were confirmed experimentally.

There is another possibility for the proposed framework, that is, the simulation of an already existing mapping by linear combinations of the kernels associated with the data. This means that any mapping can be simulated by this way, once a set of input and output pairs has been provided (end-to-end learning). For example, we can simulate partially a deep neural network by choosing any two of its layers. We can also combine several of these simulations.

6. Conclusion

In this study, we solved the out-of-sample problem associated with the previously proposed supervised Laplacian eigenmap for multi-label classification (SLE-ML). The problem was solved by simulating the embedding using a set of linear sums of kernels (KSLE-ML). In addition, we identified a sufficient condition under which the simulated mapping needs to be the same as that of the original embedding, that is, the nonsingularity of the Gram matrix. We confirmed experimentally that the extent to which the mappings of the training samples differ from those of the testing samples is acceptable. Because kernels are easily exchanged in KSLE-ML, we conducted both kernel selection and parameter selection. This revealed the importance of parameter selection over kernel selection. Raising the mapping dimension was confirmed to be effective for increasing the class separability. This offers the possibility of separability-guided feature extraction to increase the classification performance of any classifier. Experimentally, we demonstrated that KSLE-ML shows an excellent performance in visualization of single- and multi-label data. Furthermore, we confirmed that the feature extraction realized by KSLE-ML enables a simple classifier to achieve a higher accuracy compared with typical classifiers.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was partially supported by JSPS KAKENHI (Grant Number 19H04128).

References

[1] A.N. Tarekegn, M. Giacobini, K. Michalak, A review of methods for imbalanced multi-label classification, Pattern Recognition 118 (2021) 107965.
[2] J. Li, et al., Learning common and label-specific features for multi-label classification with correlation information, Pattern Recognition 121 (2022) 108259.
[3] L. Maltoudoglou, et al., Well-calibrated confidence measures for multi-label text classification with a large number of labels, Pattern Recognition 122 (2022) 108271.
[4] K. Bhatia, et al., The extreme classification repository: Multi-label datasets and code, 2016. http://manikvarma.org/downloads/XC/XMLRepository.html.
[5] S. Aoki, M. Kudo, Balancing of samples in class hierarchy,. Proceedings of VII International Workshop on Pattern Recognition and Artificial Intelligence (IWAIPR'2021), to appear in LNCS 13055.
[6] T. Horio, M. Kudo, Feature selection with class hierarchy for imbalance problems, Proceedings of VII International Workshop on Pattern Recognition and Artificial Intelligence (IWAIPR'2021), to appear in LNCS 13055.
[7] M. Belkin, P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, Neural Comput. 15 (6) (2003) 1373–1396.
[8] J.A. Costa, A.O. Hero, Classification constrained dimensionality reduction, in: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '05), volume 5, 2005, pp. 1077–1080.
[9] B. Raducanu, F. Dornaika, A supervised non-linear dimensionality reduction approach for manifold learning, Pattern Recognition 45 (6) (2012) 2432–2444.
[10] Q. Jiang, M. Jia, Supervised laplacian eigenmaps for machinery fault classification, in: Proceedins of 2009 WRI World Congress on Computer Science and Information Engineering, volume 7, 2009, pp. 116–120. https://doi.org/10.1109/CSIE.2009.765
[11] M. Tai, M. Kudo, A supervised laplacian eigenmap algorithm for visualization of multi-label data: SLE-ML, in: Proceedings of Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, 2019, pp. 525–534.
[12] Y. Bengio, Yoshua, J.-F. Paiement, P. Vincent, O. Delalleau, N. Roux, M. Ouimet, Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering, in: Proc. of the 16th International Conference on Neural Information Processing Systems (NIPS'03), 2003, pp. 177–184.
[13] K.P. Murphy, Machine Learning: A Probablistic Perspective, MIT Press, 2012.
[14] B. Li, Y.-R. Li, X.L. Zhang, A survey on laplacian eigenmaps based manifold learning methods, Neurocomputing 335 (2019) 336–351. https://doi.org/10.1016/j.neucom.2018.06.077
[15] K. Pearson, On lines and planes of closest fit to systems of points in space, Philosophical Magazine 2 (1901) 559–572.

[16] R.A. Fisher, The use of multiple measurements in taxonomic problems, Annals of Eugenics 7 (7) (1936) 179–188.
[17] L. van der Maaten, G. Hinton, Visualizing data using t-SNE, Journal of Machine Learning Research 9 (2008) 2579–2605. http://www.jmlr.org/papers/v9/vandermaaten08a.html
[18] L. McInnes, J. Healy, J. Melville, Umap: Uniform manifold approximation and projection for dimension reduction, 2020. http://arxiv.org/abs/1802.03426.
[19] M. Espadoto, R.M. Martins, A. Kerren, N.S.T. Hirata, A.C. Telea, Towards a quantitative survey of dimension reduction techniques, IEEE Trans. on Visual. and Comp. Graph. (2019). 1–1 https://doi.org/10.1109/TVCG.2019.2944182
[20] X. He, P. Niyogi, Locality preserving projections, in: Proceedings of the 16th International Conference on Neural Information Processing Systems (NIPS'03), MIT Press, Cambridge, MA, USA, 2003, pp. 153–160.
[21] G. Feng, D. Hu, D. Zhang, Z. Zhou, An alternative formulation of kernel lpp with application to image recognition, Neurocomputing 69 (13) (2006) 1733–1738. https://doi.org/10.1016/j.neucom.2006.01.006
[22] J.-B. Li, J.-S. Pan, S.C. Chu, Kernel class-wise locality preserving projection, Information Sciences 178 (7) (2008) 1825–1835. https://doi.org/10.1016/j.ins.2007.12.001
[23] H. Knaf, Kernel fisher discriminant functions - a concise and rigorous introduction, Fraunhofer (ITWM), 2007 Tech. rep. 117.
[24] A. Tanaka, H. Imai, M. Kudo, M. Miyakoshi, Theoretical analyses on a class of nested rkhs's, in: Proceedings of 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2011), 2011, pp. 2072–2075.
[25] N. Karthika, B. Janet, H. Shukla, A novel deep neural network model for image classification, International Journal of Engineering and Advanced Technology 8 (6) (2019) 3241–3249.
[26] K. Trohidis, G. Tsoumakas, G. Kalliris, I.P. Vlahavas, Multi-label classification of music into emotions, in: J.P. Bello, E. Chew, D. Turnbull (Eds.), Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR 2008), 2008, pp. 325–330. Drexel University, Philadelphia, PA, USA, September 14-18
[27] V. Vapnik, A. Chervonenkis, A note on one class of perceptrons, Automation and Remote Control 25 (1964) 821–837.
[28] B.V. Dasarathy, Nearest neighbor (nn) norms : Nn pattern classification techniques, IEEE Computer Society Press, 1991.
[29] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32. https://doi.org/10.1023/A:1010933404324
[30] M. Boutell, X.S.J. Luo, C. Brown, Learning multi-label scene classification, Pattern Recognition 37 (9) (2004) 1757–1771.
[31] Y.n. Chen, H.t. Lin, Feature-aware label space dimension reduction for multi-label classification, in: F. Pereira, C.J.C. Burges, L. Bottou, K.Q. Weinberger (Eds.), Advances in Neural Information Processing Systems, volume 25, Curran Associates, Inc., 2012, pp. 1529–1537.
[32] J. Read, B. Pfahringer, G. Holmes, E. Frank, Classifier chains for multi-label classification, Mach. Learn. 85 (3) (2011) 333–359. https://doi.org/10.1007/s10994-011-5256-5
[33] M.-L. Zhang, Z.H. Zhou, Ml-knn: A lazy learning approach to multi-label learning, Pattern Recognition 40 (7) (2007) 2038–2048.
[34] K. Yu, S. Yu, V. Tresp, Multi-label informed latent semantic indexing, in: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '05), Association for Computing Machinery, New York, NY, USA, 2005, pp. 258–265. https://doi.org/10.1145/1076034.1076080
[35] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500) (2000) 2323–2326. https://science.sciencemag.org/content/290/5500/2323
[36] A. Stuhlsatz, J. Lippel, T. Zielke, Discriminative feature extraction with deep neural networks, in: Proc. of the 2010 International Joint Conference on Neural Networks (IJCNN), 2010, pp. 1–8.
[37] Z. Jiang, et al., Learning discriminative features via label consistent neural network, in: Proc. of 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), 2017, pp. 207–216.

Mariko Tai received her Master Degree of Information Science from the Hokkaido University in 2021. She is now with Accenture Japan Ltd.

Mineichi Kudo received his Dr. Eng. degree in Information Engineering from the Hokkaido University in 1988. He is a professor in Hokkaido University. His research interests include design of pattern recognition systems, image processing, data mining and computational learning theory. He is fellows of IAPR and the IEICE Japan.

Akira Tanaka received the D.E. degree from Hokkaido University, Sapporo, Japan, in 2000. He is with the Faculty of Information Science and Technology, Hokkaido University. His research interests include image processing, acoustic signal processing, and machine learning theory.

Hideyuki Imai received the D.E. from Hokkaido University in 1999. He joined the Faculty of Information Science and Technology. His research interests include data analysis and statistical inference.

Keigo Kimura recieved Ph.D. degree in Information Engineering from Hokkaido University in 2017. He is an assistant professor in Hokkaido University. His research interests are in Machine Learning, Pattern Recognition and Data Mining.