



A Supervised Laplacian Eigenmap Algorithm for Visualization of Multi-label Data: SLE-ML

Mariko Tai^(✉) and Mineichi Kudo

Graduate School of Information Science and Technology, Hokkaido University,
Sapporo, Japan
marikotai@ist.hokudai.ac.jp

Abstract. A novel supervised Laplacian eigenmap algorithm is proposed especially aiming at visualization of multi-label data. Supervised Laplacian eigenmap algorithms proposed so far suffer from hardness in the setting of parameters or the lack of the ability of incorporating the label space information into the feature space information. Most of all, they cannot deal with multi-label data. To cope with these difficulties, we consider the neighborhood relationship between two samples both in the feature space and in the label space. As a result, multiple labels are consistently dealt with as the case of single labels. However, the proposed algorithm may produce apparent/fake separability of classes. To mitigate such a bad effect, we recommend to use two values of the parameter at once. The experiments demonstrated the advantages of the proposed method over the compared four algorithms in the visualization quality and understandability, and in the easiness of parameter setting.

Keywords: Supervised Laplacian eigenmap · Multi-Label data · Feature and label spaces

1 Introduction

In recent years, various kinds of information, such as location information, search history, and videos, have been converted to numerical/categorical/binary data. Those data are often expressed by vectors of a high dimension. Therefore, it is difficult for us to observe directly the data in order to grasp how data are distributed and what relationship exists among data. To make use of our high-order brain functions and intuition to analyze such data, dimension reduction into a two- or three-dimensional space is effective. Dimension reduction, not limited to two- or three-dimensional, is also useful to avoid the “curse of dimensionality”, a common obstacle in regression and classification. Many visualization methods proposed so far are categorized into two of unsupervised methods and supervised methods. They are furthermore divided into two of linear and nonlinear methods. The unsupervised methods do not use class labels as seen in principal

component analysis (PCA) and multidimensional scaling (MDS). PCA is a linear mapping and maximizes the variance of mapped data. MDS is a nonlinear mapping and preserves the distance between data, before and after mapping, as much as possible. On the other hand, the supervised methods use class labels as supervision information. Fisher linear discriminant analysis (FLDA) is a representative example. FLDA is a linear mapping and minimizes the ratio of within-class variance to between-class variance in the mapped data. The visual neural classifier [5] is an example of supervised nonlinear method.

Unsupervised methods are useful for revealing hidden structure, typically manifolds formed from data. On the contrary, supervised methods are effective for revealing the separability of classes. Linear-methods keep the linear structure of data but cannot express the manifold structures with varying curvature. On the other hand, nonlinear-methods can effectively catch the manifold structure but may produce fake structure which can mislead the analysts. Laplacian Eigenmaps (LEs), our main concerns, are, originally unsupervised, nonlinear mappings and preserve the neighbor relationship of data by graph Laplacians over adjacency graphs. In this paper, we propose a novel *supervised* LE, which combines feature and label information into a single neighborhood relation between data.

2 Related Works

In this section, we provide an overview of supervised LEs. So far, CCDD [2], Constraint Score [8], S-LapEig [4] and S-LE [6] have been proposed. In the following, the detail of each algorithm will be introduced. Note that some parameter symbols are changed from the original papers for keeping consistency through this paper. In fact, k is used in common for the number of nearest neighbors, σ^2 for a variance of an exponential, τ^2 for a variance of a second exponential, β for a parameter on label-agreement, and λ for a parameter on the balance between feature space and label space information. In addition, necessary parameters of each algorithm are also shown with the name.

2.1 Laplacian Eigenmaps: LE(k) (Original LE)

Given n data points $\{\mathbf{x}_i\}_{i=1}^n$ in a high-dimensional space \mathbb{R}^M , the original LE [1] maps them into points $\{\mathbf{z}_i\}_{i=1}^n$ in a low-dimensional space \mathbb{R}^m on the basis of a neighbor relation represented by $\{w_{ij}(\geq 0)\}_{i,j=1}^n$ over $\{\mathbf{x}_i\}_{i=1}^n$ in such a way to minimize

$$J_{\text{LE}} = \sum_i \sum_j \|\mathbf{z}_i - \mathbf{z}_j\|^2 w_{ij}. \quad (1)$$

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

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

where $\mathbf{x}_i \in \text{kNN}(\mathbf{x}_j)$ shows that \mathbf{x}_i is a member of k nearest neighbors of \mathbf{x}_j .

Let Z be a matrix of $n \times m$ and let \mathbf{z}_i^T (T denotes the transpose) be the i th row. Then J_{LE} becomes $J_{\text{LE}} = 2\text{tr } Z^T LZ$ (tr denotes the trace), where $L = D - W$ with $D = \text{diag}(\sum_j w_{1j}, \dots, \sum_j w_{nj})$. We can find $\{\mathbf{z}_i\}_{i=1}^n$ by minimizing $\text{tr } Z^T LZ$, subject to $Z^T DZ = I$. The solution is given by solving the generalized eigenvalue problem, $LZ = DZA$, and, avoiding the trivial eigenvector of $\mathbf{1}$ with $\lambda = 0$, the second to $(m + 1)$ th smallest (in the corresponding eigenvalue) eigenvectors are used for Z . Note that L is positive semi-definite.

2.2 Classification Constrained Dimensional Reduction: CCDDR(k, σ^2, λ)

CCDDR [2] introduces a hypothetical node for each class, called a *class center*, and requires the points of the same class to gather around the class center in the mapped space. Let $\mu_k \in \mathbb{R}^m$ be the class center of class k in the mapped space and $C = (c_{ki})$ be the class membership matrix, i.e., $c_{ki} = 1$ if $\mathbf{x}_i \in \mathbb{R}^M$ has label k and $c_{ki} = 0$ otherwise. CCDDR minimizes the cost function

$$J_{\text{CCDDR}} = \lambda \sum_{i,j} \|\mathbf{z}_i - \mathbf{z}_j\|^2 w_{ij} + (1 - \lambda) \sum_{k,i} \|\mu_k - \mathbf{z}_i\|^2 c_{ki}, \quad (2)$$

where

$$w_{ij} = \begin{cases} \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / \sigma^2) & (\mathbf{x}_i \in \text{kNN}(\mathbf{x}_j) \vee \mathbf{x}_j \in \text{kNN}(\mathbf{x}_i)) \\ 0 & (\text{otherwise}) \end{cases}, c_{ki} = \begin{cases} 1 & (y_i = k) \\ 0 & (y_i \neq k) \end{cases}.$$

Here y_i is the class label of \mathbf{x}_i and λ ($0 \leq \lambda \leq 1$) is a balance parameter between feature space information and label space information. In [2], σ^2 is determined as ten times the average of the squared nearest neighbor distances and $\lambda = 1/2$.

2.3 Constraint Score: CS(β)

The Constraint Score [8] is not proposed directly for dimension reduction nor visualization, but for feature selection. However, we can use the criterion for LE. In fact, it is a naïve way to deal with sample pairs of different classes: if the classes are the same, then multiply +1 to (1), otherwise -1.

Although two cost functions, division type and subtraction type, are shown in [8], we consider only the subtraction type that minimizes the cost function

$$J_{\text{CS}} = \sum_{i,j} \|\mathbf{z}_i - \mathbf{z}_j\|^2 w_{ij}^M - \beta \sum_{i,j} \|\mathbf{z}_i - \mathbf{z}_j\|^2 w_{ij}^C,$$

where $w_{ij}^M = \mathbf{1}(y_i = y_j)$ and $w_{ij}^C = \mathbf{1}(y_i \neq y_j)$ ($\mathbf{1}(\cdot)$ is the indication function that takes 1 if the argument is true, 0 otherwise).

2.4 S-LapEig(k, σ^2, τ^2)

S-LapEig [4] modifies the distance between data points $\{\mathbf{x}_i\}_{i=1}^n$ in the original space such that data of the same class label become closer and data of the different class labels become more distant. The criterion to minimize is the same as the original LE: $J_{\text{S-LapEig}} = J_{\text{LE}}$. However, the weight is determined at two stages as

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

where

$$d^2(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} 1 - \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/\tau^2) & (y_i = y_j) \\ \exp(\|\mathbf{x}_i - \mathbf{x}_j\|^2/\tau^2) & (y_i \neq y_j) \end{cases}.$$

Here τ^2 is taken as the square of the average Euclidean distance between all pairs of data points in [4].

2.5 S-LE(σ^2, β)

S-LE [6] computes the adjacency matrix W as follows. Let $AS(\mathbf{x}_i) = 1/n \cdot \sum_{j=1}^n s(\mathbf{x}_i, \mathbf{x}_j)$, where $s(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/\sigma^2)$. If $(s(\mathbf{x}_i, \mathbf{x}_j) > AS(\mathbf{x}_i)) \wedge (y_i = y_j)$, then \mathbf{x}_j is judged as the neighbor of \mathbf{x}_i and denoted by $\mathbf{x}_j \in N_w(\mathbf{x}_i)$. On the contrary, if $(s(\mathbf{x}_i, \mathbf{x}_j) > AS(\mathbf{x}_i)) \wedge (y_i \neq y_j)$, then $\mathbf{x}_j \in N_b(\mathbf{x}_i)$. Under these definitions, S-LE maximizes (not minimizes)

$$J_{\text{SLE}} = \beta \sum_{i,j} \|\mathbf{z}_i - \mathbf{z}_j\|^2 w_{ij}^B - (1 - \beta) \sum_{i,j} \|\mathbf{z}_i - \mathbf{z}_j\|^2 w_{ij}^W,$$

where

$$w_{ij}^W = \begin{cases} s(\mathbf{x}_i, \mathbf{x}_j) & (\mathbf{x}_i \in N_w(\mathbf{x}_j) \vee \mathbf{x}_j \in N_w(\mathbf{x}_i)) \\ 0 & (\text{otherwise}), \end{cases}, w_{ij}^B = \begin{cases} 1 & (\mathbf{x}_i \in N_b(\mathbf{x}_j) \vee \mathbf{x}_j \in N_b(\mathbf{x}_i)) \\ 0 & (\text{otherwise}) \end{cases}.$$

3 Supervised Laplacian Eigenmaps

Almost all supervised LE algorithms that we refer to in Sect. 2 basically separate a pair $(\mathbf{x}_i, \mathbf{x}_j)$ into a same-class pair or a different-class pair and evaluate them separately. Therefore, we need to pay a special attention to the difference of the number of two kinds of pairs. In addition, some algorithms cannot control the degree to which we mix the label information and the feature information. Most of all, they cannot deal with multi-label datasets where a single data is associated with multiple class labels. Only CCDR can deal with multi-label data, if we want to do that, but it has its own problem as will be discussed later. For the other three algorithms, it is also not easy to extend because they deal with sample pairs

differently depending on if they share the same class or not. To cope with these limitations, we propose a novel supervised LE, called the *Supervised Laplacian Eigenmaps for Multi-Label datasets* (shortly, SLE-ML), for visualization mainly. We combine neighbor information in the feature space and that in the class-label space into one with a balance parameter λ ($0 \leq \lambda \leq 1$).

3.1 Supervised Laplacian Eigenmaps for Multi-label Datasets: SLE-ML(k, λ)

SLE-ML minimizes the same cost function as the original LE using a different weight

$$J_{\text{SLE-ML}} = \lambda \sum_{i,j} \|\mathbf{z}_i - \mathbf{z}_j\|^2 w_{ij}^F + (1 - \lambda) \sum_{i,j} \|\mathbf{z}_i - \mathbf{z}_j\|^2 w_{ij}^L, \quad (3)$$

where

$$w_{ij}^F = (\mathbf{1}(\mathbf{x}_i \in kNN(\mathbf{x}_j)) + \mathbf{1}(\mathbf{x}_j \in kNN(\mathbf{x}_i))) / 2, \text{ and } w_{ij}^L = \frac{|\mathbf{y}_i \wedge \mathbf{y}_j|}{|\mathbf{y}_i \vee \mathbf{y}_j|}.$$

Here, superscript ‘F’ stands for “feature space” and ‘L’ stands for “label space”. In addition, w_{ij}^L is the Jaccard similarity coefficient, the ratio of common labels to 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 i and j share the same label, and $w_{ij}^L = 0$ otherwise. The original (unsupervised) LE is a special case of SLE-ML with $\lambda = 1.0$. Unlike many of previous supervised LEs that take a trade-off in the feature space between same-class pairs and different-class pairs, SLE-ML take a trade-off of similarity between the feature space and the label space.

3.2 Parameters

All algorithms have their own parameters: CCDD(k, σ^2, λ), CS(β), S-LapEig(k, σ^2, τ^2), SLE(σ^2, β), and SLE-ML(k, λ). It is often critical to choose an appropriate value for each parameter. We first discuss how to determine the values and how sensitive they are to the results. The variance parameters σ^2 and τ^2 are often determined from data. A typical way is to use the average squared Euclidean distance between all pairs of data points. As for the value of k , we need to use the same value in common to all algorithms. In the following experiments, the value of k is set to 1.5 times the average sample size per class in order to relate each sample to other samples of different classes. As for the other parameter, β (as for the label agreement) and λ (as for trade-off between feature and label information), we need to be more careful about the setting. Let us consider β in CS(β) and S-LE(σ^2, β). When the number of classes is large, the cases when two samples have the same label are far less than the counter part. So, we have to set the value of β in accordance with the given dataset. When we consider supervised LEs, the most important thing is how we incorporate

the label information into feature information. In contrast to LE that uses the feature information only, if we use the label information only, then all the points of the same label concentrate on a single point in the mapped space, as seen in SLE-ML($k, \lambda = 0.0$). Therefore, we need to be careful about the value of λ more than the other parameters. CCDD(k, σ^2, λ) has the same parameter λ , but it has another problem. The criterion (2) has two terms: the size of the first term is $O(n^2)$ and the size of the second term is $O(Kn)$ where K is the number of classes. Therefore, if $K \ll n$ or its converse (as seen in extreme multi-label problems), the effect of the same value of λ changes. So, it needs to be set carefully. In SLE-ML(k, λ), the two terms in (3) have the same size of $O(n^2)$. Therefore, we do not need to be careful about the number of classes and can consider the value of λ independently of datasets. That is, SLE-ML is problem-independent. The algorithms except for CCDD(k, σ^2, λ) and SLE-ML(k, λ) do not have even a trade-off parameter between feature and label information. This means we cannot control it.

Table 1. Datasets. (A) stands for artificial datasets and (N) for natural datasets. In the artificial datasets, garbage features are added; 7 for Torus and 8 for Clusdat.

Dataset	#samples	#classes	#samples in each class	#features (Intrinsic)
Torus (A)	1000	2	500 500	10 (3)
Clusdat (A)	1600	2	800 800	10 (2)
Digits (N) [†]	1797	10	178 182 ... 180	64 (64)
Scene (N) [‡]	1211	6(14)	194 165 ... 1	294(294)

[†]:UCI Machine Learning Repository [3]

[‡]:Mulan: A Java Library for Multi-Label Learning [7]

4 Experiments

We evaluated the performance of the proposed method on several high-dimensional datasets (Table 1). The dataset `digits` consist of 1797 images of hand-written digits (0–9). In our experiments, the parameter k for nearest neighbors was set to 1.5 times the average number of samples of each class as described before.

Figure 1 is the visualization result of `digits` by SLE-ML. To confirm the effect of the parameter λ , we varied the value from 0 to 1 by step 0.2. We see that $\lambda = 1.0$ (the feature space only) derives the same mapping as LE, and $\lambda = 0.0$ (the label space only) derives the class-isolated mapping. For a middle value of λ , we can see the result by a trade-off between feature and label spaces. It should be noted that a smaller value of λ tends to enhance the separability among classes more than the reality. So, we recommend to use two values of λ as $\lambda = 0.5$ and 0.9 at once for analyzing data. We compared four algorithms, CCDD, CS, S-LapEig and S-LE, with SLE-ML. The parameters were chosen so as to produce almost the best results except for SLE-ML. In Fig. 2, the results of

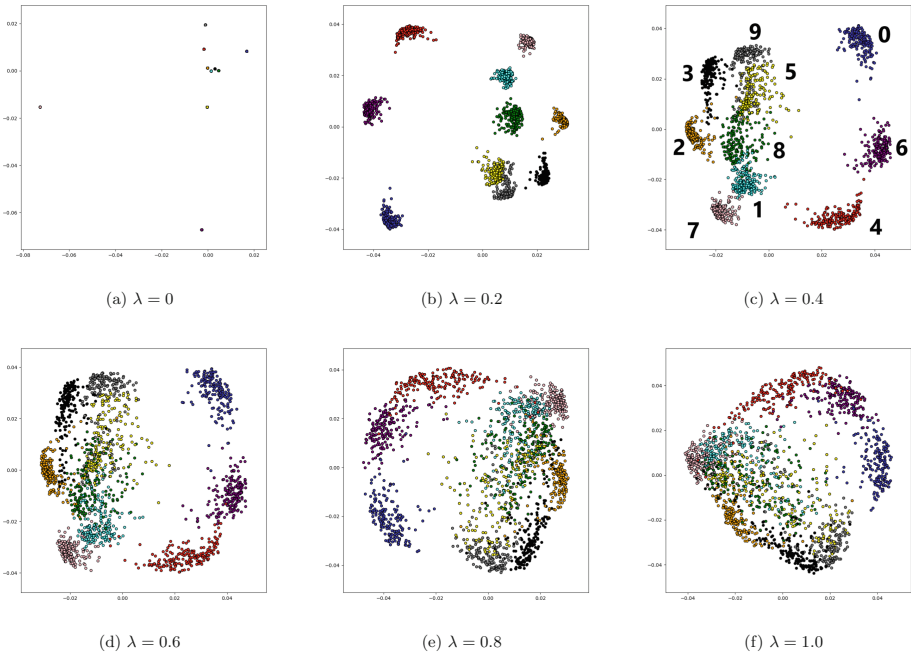


Fig. 1. Effect of the balance parameter λ in the proposed SLE-ML in **digits** dataset (Each color corresponds to a class as seen in the case of $\lambda = 0.4$). (Color figure online)

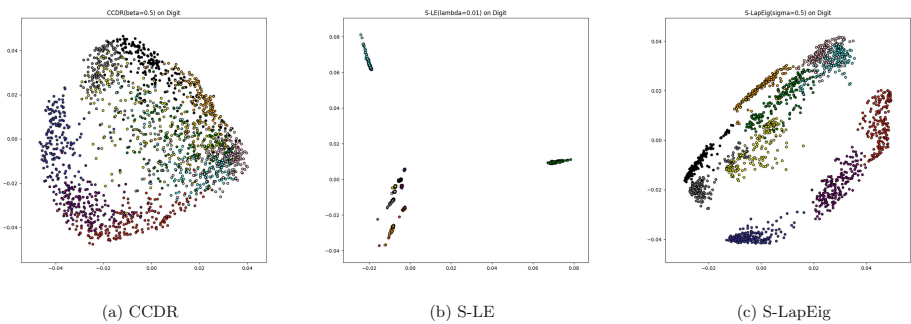


Fig. 2. Visualization of **digits** by three algorithms

CCDR, S-LE and S-LapEig are shown. Since CS did not produce any good result, the result is not shown. We see that a high separability of classes is visualized by CCDR, S-LapEig and SLE-ML($\lambda = 0.4, 0.6$). The other algorithms fail to reveal the separability that actually exists. In the following, therefore, we compared these three only.

To make clear the difference of those algorithms, we visualized two artificial datasets **Torus** and **Clusdat**. Note that these data are contaminated by garbage

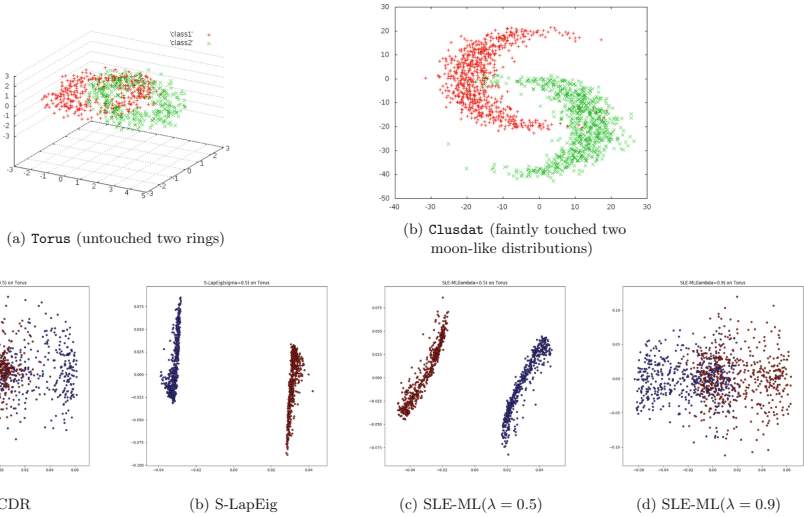


Fig. 3. Visualization of Torus

features. The results are shown in Figs. 3 and 4. We see that CCDR and SLE-ML ($\lambda = 0.9$) expose the manifold structure to some extent, while SLE-ML ($\lambda = 0.5$) and S-LapEig succeed to show the separability.

Next we dealt with multi-label datasets. Figure 5 is the visualization result of scene by SLE-ML ($\lambda = 0.5$). We observe that multi-label data are mapped the same as single-label data. In Fig. 5, we see that data with two labels {Fall foliage, Field} locate in the middle of data with {Fall foliage} and data with {Field}. Such an observation reveals the relationship between a composite class and its component classes in the original feature space.

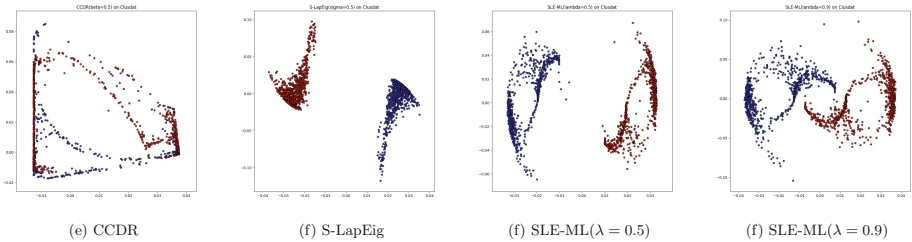


Fig. 4. Visualization of Clusdat

5 Discussion

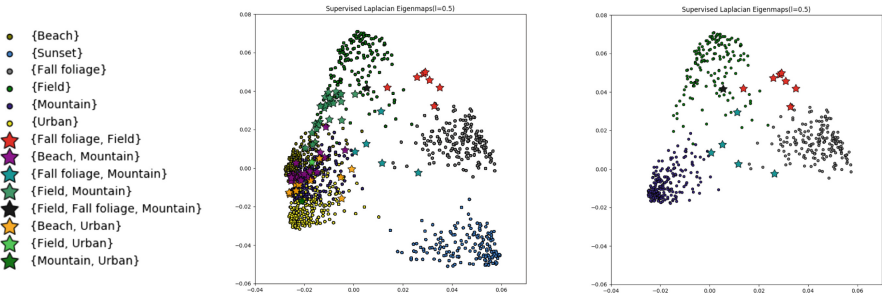
The proposed SLE-ML is advantageous to the compared four algorithms in the sense that the results give more information than the others. This is mainly



(a) {Fall foliage, Field} ★



(b) {Beach, Urban} ★

(a) SLE-ML ($\lambda = 0.5$)

(b) Relationship between some multi-labels and its children single labels

Fig. 5. Visualization of scene with multiple labels.

because the control parameter λ is intuitive and the multiple results with different values of it help us to analyze data. However, there still remain many more challenges that the original LEs had and maybe many LEs still have. First of all, we need to resolve the “out-of-sample” problem. Since the mapping in SLE-ML is not explicit, we cannot apply this mapping to a newly arrived data. We are now thinking to simulate the mapping linearly or nonlinearly. If it is succeeded, we may choose the parameter value under which separability is held high. Next, we need to cope with “imbalance problem.” SLE-ML needs to be modified to emphasize minority classes. Last, we have to devise some way to visualize a hundred of thousands of data and data with a large number of multiple labels.

6 Conclusion

In this paper, we have proposed a novel supervised Laplacian eigenmap algorithm that can handle multi-label data in addition to single-label data. The experiment demonstrated the advantages of the algorithm over the compared the state-of-the-art algorithms in the visualization quality and understandability, and in the easiness of parameter setting.

In the proposed algorithm, we combine the feature information and the label information into one, and control the balance by a parameter. To mitigate the

risk of being cheated by an apparent separability with a small value of the parameter, we recommend to use two different values of the parameter at once. We also analyzed how appropriately we can give the values in the parameters of previous four algorithms and, as a result, pointed out some careful points.

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

References

1. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.* **15**(6), 1373–1396 (2003)
2. Costa, J.A., Hero, A.O.: Classification constrained dimensionality reduction. In: *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP 2005)*, vol. 5, pp. v/1077–v/1080, March 2005
3. Dua, D., Graff, C.: UCI ML repository (2017). <http://archive.ics.uci.edu/ml>
4. Jiang, Q., Jia, M.: Supervised Laplacian eigenmaps for machinery fault classification. In: *2009 WRI World Congress on Computer Science and Information Engineering*, vol. 7, pp. 116–120, March 2009. <https://doi.org/10.1109/CSIE.2009.765>
5. Ornes, C., Sklansky, J.: A visual neural classifier. *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)* **28**(4), 620–625 (1998)
6. Raducanu, B., Dornaika, F.: A supervised non-linear dimensionality reduction approach for manifold learning. *Pattern Recogn.* **45**(6), 2432–2444 (2012)
7. Tsoumakas, G., et al.: Mulan: a Java library for multi-label learning. *J. Mach. Learn. Res.* **12**, 2411–2414 (2011)
8. Zhang, D., Chen, S., Zhou, Z.H.: Constraint score: a new filter method for feature selection with pairwise constraints. *Pattern Recogn.* **41**(5), 1440–1451 (2008)