2009 International Symposium on Nonlinear Theory and its Applications
NOLTA'09, Sapporo, Japan, October 18-21, 2009

Robust Laplacian Eigenmaps for Semi-Supervised Pattern Classification

Weiwei Du† and Kiichi Urahama‡

†Dept. of Information Science, Kyoto Institute of Technology, Kyoto 606-9585 Japan
‡Dept. of Visual Communication Design, Kyushu University, Fukuoka 815-8540 Japan
Email: duweiwei@kit.ac.jp, urahama@design.kyushu-u.ac.jp

Abstract—The Laplacian eigenmap (LEM) is a popularly used graph spectral algorithm for mapping data nonlinearly into a low-dimensional space. In this paper, we robustify LEM and use it for semi-supervised pattern classification. The distance metric for feature vectors is modulated with a semi-supervised feature selection method and data are mapped into a low-dimensional classification space with the robustified LEM. Test data are classified by the nearest neighbor rule with the modulated distance metric between data.

1. Introduction

The Laplacian eigenmap (LEM)[1] is a representative graph-spectral technique for mapping data nonlinearly into a low-dimensional space. While LEM has been used popularly for clustering, data visualization and pattern classification, its weakness lies in the high sensitivity to outlier data similarly to the principal component analysis (PCA). Its robustification is desired for enhancing separability between classes for nonlinearly entangled distribution of data.

The PCA and LEM are unsupervised algorithms for distance metric learning[2]. The LEM has been extended to semi-supervised learning where class labels are given only limited number of data[3]. In the semi-supervised LEM[3], the distance metric is distorted locally on the basis of label information of partial data. While this local distortion is propagated along graph links to unlabeled data, it is difficult for this method to capture globally consistent distance metric due to its sensitivity to outlier data.

Alternative to such local metric modulation, if we incorporate a semi-supervised metric learning technique giving a global distance metric into a robustified LEM, we can obtain a new semi-supervised LEM algorithm with higher classification rates.

In this paper, we robustify the LEM to which we incorporate the semi-supervised feature scoring technique by Zhao et al.[4] and propose a new semi-supervised pattern classification method. Robustification enables the LEM to map data with enhanced separation between classes. This enhancement in class separation improves its classification rate. We verify this improvement with some experiments for a synthetic toy dataset and real datasets popularly used for benchmark test of classifiers.

2. Related Works

Robustness has attracted attention in signal processing and statistical data manipulation.

In image processing, linear filters cannot remove impulsive noises nor preserve edges on the boundary between objects. This is due to their sensitivity to outlier signals. Their robustification leads to nonlinear filters. The median filter preserves edges, while more strong nonlinear filters such as the bilateral filter or the mode filter sharpen the slope of edges. The edge in signals corresponds to class boundary between data in pattern classification.

In statistical data analysis, linear methods such as the least mean square estimation is also sensitive to outliers. Their robustification leads to nonlinear algorithms such as the least median of squares estimation.

The graph spectral method is a class of statistical data analysis techniques. The most popular graph spectral algorithm is the LEM[1] which is used for spectral clustering of data. The spectral clustering is the "cluster *partition*" type algorithm which is sensitive to outlier data. Another type is the "cluster *extraction*" algorithm which is robust to outliers[5].

In this paper, we incorporate the simple trick in the cluster extraction algorithm[5] into the LEM and used it for semi-supervised pattern classification.

3. Laplacian Eigenmaps

Let there be given m training data of feature vectors f_i from which similarity between data i and j is expressed by

$$s_{ij} = e^{-\alpha\|f_i - f_j\|^2} \quad (i, j = 1, ..., m) \tag{1}$$

3.1. Laplacian Eigenmaps

In the LEM[1], data are mapped to the coordinate $\mathbf{x} = [x_1, ..., x_m]^T$ given by

$$\begin{array}{ll} \min & \mathbf{x}^T(D - S)\mathbf{x} \\ \text{subj.to} & \mathbf{x}^T D\mathbf{x} = 1 \end{array} \tag{2}$$

where $S = [s_{ij}]$ is the similarity matrix and $D = \mathrm{diag}(d_1, ..., d_m), d_i = \sum_j s_{ij}$. With this mapping, mutually similar data with large s_{ij} are projected close together with near x_i and x_j. The solution of eq.(2) is the generalized eigenvector of $S\mathbf{x} = \lambda D\mathbf{x}$ of which principal eigenvector

with the eigenvalue 1 is constant $[1, ..., 1]/\sqrt{\sum_i d_i}$, hence we discard it and use the eigenvectors from the second to the $(p+1)$th when we project the data into a p-dimensional space. This algorithm is sensitive to outlier data due to strong homogenization effect of $\mathbf{x}^T D\mathbf{x} = 1$ which deals with all data equally even for outliers.

3.2. Robust Laplacian Eigenmaps

This LEM is an unsupervised algorithm where the distance between data is Euclidean as is in eq.(1), i.e. $(f_i - f_j)^T I(f_i - f_j)$ where I is the identity matrix. In the semi-supervised learning, label information is given for some training data, which induces the modulation of distance metric into a generalized quadratic form $(f_i - f_j)^T A(f_i - f_j)$ with a metric matrix A with which the similarity s_{ij} is modified to $\tilde{s}_{ij} = e^{-\alpha(f_i-f_j)^T A(f_i-f_j)}$ which enhances class separability on the basis of label information.

Then a straightforward extension of eq.(2) is modification of both S and D to $\tilde{S} = [\tilde{s}_{ij}]$ and $\tilde{D} = \mathrm{diag}(\tilde{d}_1, ..., \tilde{d}_m), \tilde{d}_i = \sum_j \tilde{s}_{ij}$. This simple scheme, however, does not work well because the normalization $\mathbf{x}^T \tilde{D}\mathbf{x} = 1$ uniformizes $\mathbf{x}$ as mentioned above. This equalization effect in eq.(2) cancels the enhanced class separability gained with $\tilde{S}$ and makes the mapping sensitive to outliers and blurs the boundary between classes.

Hence we preserve D in its unmodified form and propose, in this paper, to modify eq.(2) into a robust form

$$\begin{array}{ll} \max & \mathbf{x}^T(D - \tilde{S})\mathbf{x} \\ \text{subj.to} & \mathbf{x}^T D\mathbf{x} = 1 \end{array} \tag{3}$$

of which solution is the generalized eigenvector of $\tilde{S}\mathbf{x} = \mu D\mathbf{x}$. Different from eq.(2) of which first eigenvector is constant and is discarded, the first eigenvector of eq.(3) is not constant and contains useful information about data structure. So, we use its eigenvectors from the first to the pth one for embedding data into p-dimensional space.

Similar trick has been introduced to graph spectral algorithms for clustering data[5]. Eq.(3) is its incorporation into the semi-supervised LEM. Similar to the robustness of the clustering algorithm[5], eq.(3) is robust to outliers hence enhances the boundary between classes. We call this mapping the *Robust Laplacian Eigenmaps* (RLEM).

4. Semi-Supervised Feature Selection

In this section, we review the semi-supervised feature selection method by Zhao et al.[4]. Assume the training data to be partially labeled. We construct the within class similarity $s_{w,ij}$ and between class similarity $s_{b,ij}$ as

$$s_{w,ij} = \begin{cases} \gamma_1 & i, j \in \text{same class} \\ 1 & i \text{ or } j \text{ is unlabeled and} \\ & i \in \mathrm{kNN}(j) \text{ or } j \in \mathrm{kNN}(i) \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

$$s_{b,ij} = \begin{cases} \gamma_2 & i, j \in \text{different classes} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where $\mathrm{kNN}(i)$ is the set of k nearest neighbors of i. We set $k = 5$ in the following experiments.

We next construct the within class Laplacian matrix L_w and the between class Laplacian matrix L_b as

$$\begin{array}{l} L_w = D_w - S_w \\ L_b = D_b - S_b \end{array} \tag{6}$$

where D_w=diag$(d_{w,1}, ..., d_{w,m})$, $d_{w,i} = \sum_j s_{w,ij}$ and D_b=diag$(d_{b,1}, ..., d_{b,m})$, $d_{b,i} = \sum_j s_{b,ij}$.

We finally compute the importance score of each feature. Let the feature be an n-dimensional vector $f_i = [f_{1i}, ..., f_{ni}]^T$. We define the vector of the r-th feature as $g_r = [f_{r1}, ..., f_{rm}]^T$ from which the score is computed by

$$L_r = \frac{g_r^T L_b g_r}{g_r^T L_w g_r} \quad (r = 1, ..., n) \tag{7}$$

which is large for an important feature similarly to Fisher's discriminant criterion. Zhao et al.[4] select the features with L_r greater than a threshold and called this technique the Locality Sensitive Discriminant Feature (LSDF).

5. Semi-Supervised Pattern Classification

We incorporate this LSDF into the RLEM in section 3.2. Since the LSDF gives the score of each feature, we restrict the metric matrix A diagonal and set it as $A = L^2$ where $L = \mathrm{diag}(L_1, ..., L_n)$ with the LSDF score L_r, that is, we modify the similarity in eq.(1) to

$$\tilde{s}_{ij} = e^{-\alpha(f_i-f_j)^T L^2 (f_i-f_j)} \tag{8}$$

where we set L_r below a threshold to 0.

As was mentioned in section 3.2, we construct the similarity matrix $\tilde{S}$ in eq.(3) from these modified $\tilde{s}_{ij}$, while maintaining the matrix D in the original form calculated from the unmodulated similarity in eq.(1).

We map the training data with this RLEM into $(c-1)$-dimensional classification spaces where c is the number of classes.

Our proposed technique is summarized as

Step 1: We compute the original similarity $s_{ij} = e^{-\alpha_2 \|f_i - f_j\|^2}$ and construct $D = \mathrm{diag}(d_1, ..., d_m), d_i = \sum_j s_{ij}$.

Step 2: We compute the feature score L_r with the LSDF in section 4 and construct $L = \mathrm{diag}(L_1, ..., L_n)$.

Step 3: We compute the modified similarity $\tilde{s}_{ij} =$

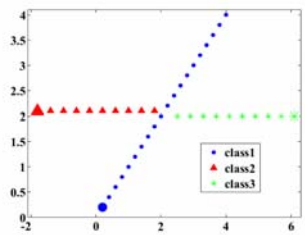
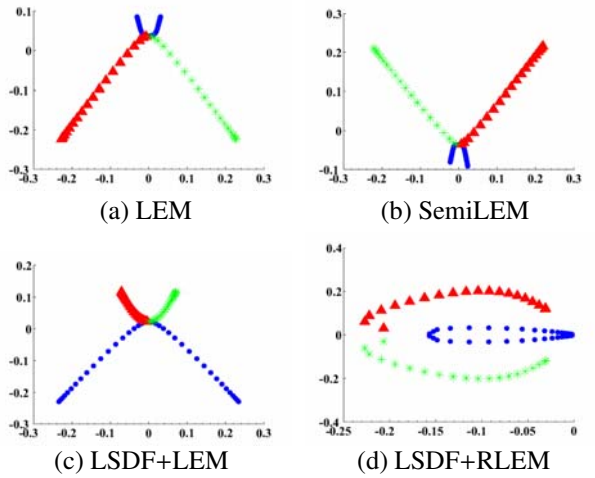


Figure 1: Synthetic data.



(a) LEM (b) SemiLEM

(c) LSDF+LEM (d) LSDF+RLEM

Figure 2: Mapped data.

Table 1: Error rates for test data.

	test error(%)
LEM	18.75
semiLEM	17.50
LSDF	33.75
LSDF+LEM	33.75
LSDF+RLEM	0

$e^{-\alpha_1(f_i-f_j)^T L^2(f_i-f_j)}$ and construct $\tilde{S} = [\tilde{s}_{ij}]$.

Step 4: We execute the RLEM in section 3.2 and compute the eigenvectors from the first to $(c-1)$th one.

Step 5: We map every training datum into $(c-1)$-dimensional space and we label all the unlabeled data by the nearest neighbor rule with the weighted distance $(f_i - f_j)^T L^2 (f_i - f_j)$ to labeled data.

This finishes the learning phase where all training data are labeled. In a test phase, we classify test data by the nearest neighbor rule with the weighted distance between test data and all the training data.

6. Experiments

We compare the performance of the proposed method LSDF+RLEM with LEM[1], semi-supervised LEM (SemiLEM)[3], LSDF[4] and LSDF+LEM. In each method, we adjust their parameters to the value with their best performance.

6.1. Synthetic Data

We firstly experiment with the data in Fig.1 which includes 3 classes. Data are arranged on three straight lines. Two horizontal lines are composed of 40 data points and the central inclined line includes 80 points. At each line, data are separated into the training and test data interleaving one by one. Only sampled training data are plotted in Fig.1 where the large marks at the left, right and bottom ends are labeled data which exist only one in each class.

Mapped training data are shown in Fig.2 where ▲ and ∗ are two horizontal line classes in Fig.1 and ● is the center slant line class. The proposed method succeeds to separate three classes while they are glued together in other methods. We verify with this figure that the RLEM in section 3.2 is efficient for enhancing the separation of classes and hence improve the classification rate. The error rates for test data are shown in table 1 where our proposed method (LSDF+RLEM) can classify test data perfectly. Note that no improvement is gained by the combination of LSDF and LEM. Thus the proposed robustification for the LEM in section 3.2 is essential for the classifier.

Table 2: Data configuration.

dataset	dim.	class	data	labeled	test
iris	4	3	150	9	60
liver	6	2	345	6	173
iono.	34	2	351	6	176
vote	16	2	435	6	218
crx	15	2	690	6	345

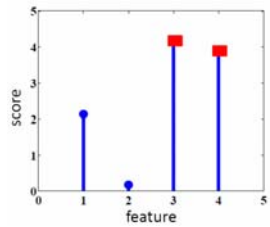


Figure 3: LSDF score for iris dataset.

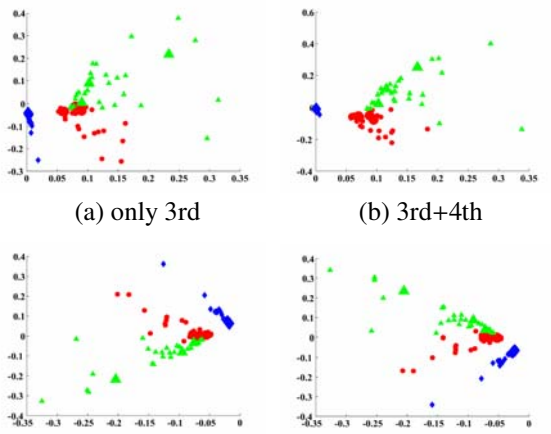
6.2. Real Data

We next experiment with five datasets: iris, liver, ionosphere, vote and crx in the UCI benchmark data[6] popularly used for testing the performance of classifiers. Their data configuration is shown in table 2.

6.2.1. Feature Score

We firstly examine the feature score in the LSDF for the iris dataset which includes four features: sepal length, sepal width, petal length and petal width. Their LSDF scores are 2.13, 0.18, 4.16 and 3.90 as is shown in Fig.3 where ■ marks on the third (petal length) and the fourth feature (petal width) denotes that the classification rate is highest when we select these two features, i.e. set L_1 and L_2 to zero.

Mapped training data are shown in Fig.4 where four cases of combination of features are examined in the order of value of L_r: (1) only 3rd feature, (2) 3rd and 4th features, (3) 3rd+4th+1st features and (4) all 3rd+4th+1st+2nd features. In Fig.4, labeled data are shown with large marks.



(a) only 3rd (b) 3rd+4th

(c) 3rd+4th+1st (d) 3rd+4th+1st+2nd

Figure 4: Mapped iris data.

Table 3: Error rates for each combination of features.

combination	test error(%)
only 3rd	5.00
3rd+4th	1.67
3rd+4th+1st	8.33
3rd+4th+1st+2nd	11.67

Class separation is largest in Fig.4(b), i.e. selection of the 3rd and the 4th features is the best.

The error rates of test data are shown in table 3 for these feature combinations. Coincident with the result of Fig.4, the classification rate is highest when the 3rd and the 4th features are selected. This superiority of the 3rd and the 4th features is the well known fact for the iris dataset.

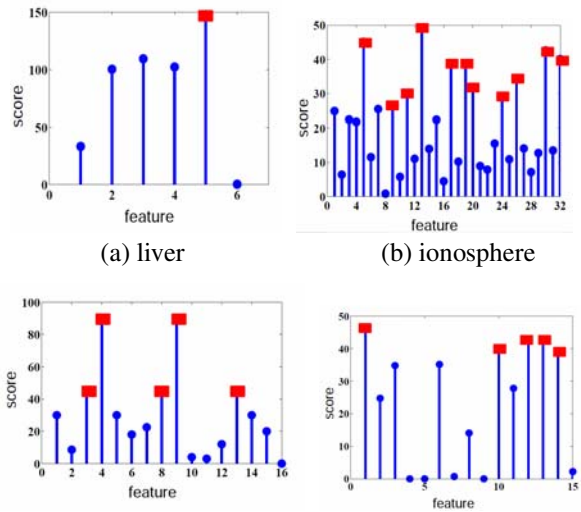
The LSDF scores in other four datasets are shown in Fig.5 where ■ marks denote the best selection of features similarly to Fig.3.

6.2.2. *Classification Rates*

The error rates of five algorithms for these iris, liver, ionosphere, vote and crx datasets are shown in table 4. We use the selected features marked with ■ in Fig.3 and Fig.5. The classification rate of the proposed method (LSDF+RLEM) is highest among these methods.

7. Conclusion

We have presented a robust graph spectral method where the semi-supervised feature scoring technique by Zhao et al. is incorporated into the Laplacian eigenmaps. Theoretical elaboration of the proposed method is a subject of future researches.



(a) liver (b) ionosphere

(c) vote (d) crx

Figure 5: LSDF scores.

Table 4: Error rates for UCI benchmark datasets.

test error	iris	liver	iono.	vote	crx
LEM	3.33	49.71	36.36	15.14	32.75
SemiLEM	3.33	49.13	32.95	15.14	32.46
LSDF	3.33	39.31	18.75	14.22	33.91
LSDF+LEM	3.33	37.57	32.95	14.22	33.62
LSDF+RLEM	1.67	35.84	16.48	11.93	27.83

References

[1] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Comp.*, vol.15, no.6, pp.1373–1396, 2003.

[2] L. Yang, "Distance metric learning: a comprehensive survey," *Tech. Rep. Michigan State Univ.*, 2006.

[3] W. Du and K. Urahama, "Semi-supervised pattern classification utilizing fuzzy clustering and nonlinear mapping of data," *J. Adv. Comput. Intell. Inform.*, vol.11, no.9, pp.1159–1164, 2007.

[4] J. Zhao, K Lu and X. He, "Locality sensitive semi-supervised feature selection," *Neurocomputing*, vol.71, no.10-12, pp.1842–1849, 2008.

[5] W. Du and K. Urahama, "Unsupervised and semi-supervised graph-spectral algorithms for robust extraction of arbitrarily shaped fuzzy clusters," *J. Adv. Comput. Intell. Inform.*, vol.11, no.6, pp.554–560, 2007.

[6] D. J. Newman, S. Hettich, C. L. Blake and C. J. Merz, "UCI repository of machine learning databases," http://www1.ics.uci.edu/mlearn/MLRepository.html, 1998.