UNSUPERVISED LOCALLY EMBEDDED CLUSTERING FOR AUTOMATIC HIGH-DIMENSIONAL DATA LABELING

Yun Fu and Thomas S. Huang*

Beckman Institute for Advanced Science and Technology
University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA
{yunfu2, huang}@ifp.uiuc.edu

ABSTRACT

In most machine learning and pattern recognition problems, the large number of high-dimensional sensory data, such as images and videos, are often labeled manually for training classifiers and modeling features, which is time-consuming and tedious. To automatically execute this process by machine, we present the unsupervised high-dimensional data clustering and automatic labeling algorithms, called Locally Embedded Clustering (LEC): (i) Constructing the neighborhood weighted graph with an appropriate distance metric; (ii) Tuning the regularization parameter to smooth the approximated manifold; (iii) Calculating the unified projection in a closed-form solution for the embedding and dimensionality reduction; (iv) Choosing the top or bottom coordinates of the embedded low-dimensional space for data representation; (v) Normalizing the low-dimensional representation to have unit length; (vi) Clustering and labeling the data via K-means. Experimental results demonstrate that LEC provides better data representation, more efficient dimensionality reduction and better clustering performance than many existing methods.

Index Terms— LEA, LEC, manifold, high-dimensional data clustering, dimensionality reduction.

1. INTRODUCTION

The clustering and dimensionality reduction in supervised or unsupervised manner have been the focus of considerable issues in computer vision and pattern recognition. Usually, the large number of sensory inputs, such as images and videos, are often viewed as high-dimensional vectors with large percent of dimensionality redundancy. The basic information processing task is to label the high-dimensional raw data manually, which is very tedious and time-consuming. In order to understand and learning the multivariate data with automatic machine processing, we need to reduce the dimensionality and find more compact representations for unsupervised high-dimensional data clustering. Conceptually, if the variance of the multivariate data is faithfully represented as a set of parameters, the data can be considered as a set of geometrically related points lying on a smooth low-dimensional manifold. An interesting issue, which will be investigated in the paper, is how to cluster the high-dimensional multivariate data automatically with a suitable dimensionality reduction technique.

The fundamental issue in dimensionality reduction is how to model the geometry structure of the manifold and produce a faithful embedding for data projection. The existing nonlinear methods such as LLE[10], Laplacian Eigenmaps (LE)[1], Isomap[12] and SDE[13], focus on preserving the geodesic distances which reflect the real geometry of the low-dimensional manifold. They have been successfully applied to some standard data sets and generate satisfying results in dimensionality reduction and manifold visualization. However, most nonlinear methods only provide the mapping from input to manifold, instead of a reversible mapping from manifold space to original space. Some papers have demonstrated that the nonlinear methods can be associated with particular linearization formulations, e.g. KPCA⇔PCA [8], ISOMAP ⇔MDS, LE⇔LPP [4] and LLE⇔LEA [3] (NPE [5]). A general framework, Graph Embedding(GE)[14], reveals the essential objective characteristic shared by these methods.

Another technique related to this topic is the Spectral Clustering (SC) [2, 9, 15], which uses the top or bottom eigenvectors of a normalized affinity matrix for K-means clustering[18]. These methods start with well-motivated objective functions, optimization eventually leading to eigenvectors, with many clear and interesting algebraic properties. Better than traditional clustering methods, SC algorithms do not need to learn an explicit model of data distribution, in which EM is used to learn the mixture density. However, most existing SC algorithms only focus on the detailed theoretical analysis for low-dimensional synthetic data clustering, instead of the high-dimensional real data, which cause more complicated and difficult problems because of the "*curse of dimensionality*". To handle the dimensionality reduction in SC, we suggest using GE as the preprocessing part of the clustering algorithms.

We are interested in exploiting the *Locally Embedded Clustering* (LEC) technique for the real-world scenario: automatic

*This work was funded in part by the Disruptive Technology Office VACE III Contract NBCHC060160 issued by DOI-NBC, Ft. Huachuca, AZ; and in part by the National Science Foundation Grant CCF 04-26627.

high-dimensional data labeling. Our idea essentially integrates the SC technique with the LEA dimensionality reduction approach in an unsupervised learning manner.

2. LOCALLY EMBEDDED ANALYSIS

Locally Embedded Analysis (LEA) [3] is the linearization form of LLE. The basic idea is to represent each vertex of a neighborhood graph as a certain measurement value, which reveals the essential manifold structure of the original data, by preserving the similarities of vertex pairs obtained from the graph similarity matrix. Define a one-to-one mapping between $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$ and $\mathcal{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_n\}$. Suppose $\mathcal{G} = \{\mathbf{X}, \mathbf{W}\}$ is a weighted graph with similarity matrix $\mathbf{W} \in \mathbb{R}^{n\times n}$ and Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where $D_{ii} = \sum W_{ij}, \forall i \neq j$. Then we find the graph embedding of graph $\mathcal{G}$ and the low-dimensional representation through $\mathbf{y}^* = \arg\min \mathbf{y}^T\mathbf{L}\mathbf{y}$ (1-D case), with the constraint $\mathbf{y}^T\mathbf{B}\mathbf{y} = q$, where q is a constant. The space projection of linearization is defined by a transformation $P : \mathbb{R}^D \to \mathbb{R}^d$. The $D \times d$ projection matrix is denoted by $\mathbf{P} = [\mathbf{p}_1\ \mathbf{p}_2\ \cdots\ \mathbf{p}_d]$ which satisfies $\mathbf{y}_i = \mathbf{P}^T\mathbf{x}_i$. The D-to-d projection can be written as a single matrix equation $\mathbf{Y} = \mathbf{P}^T\mathbf{X}$ where $\mathbf{x}_i$ and $\mathbf{y}_i$ are respectively viewed as columns of $D \times n$ matrix $\mathbf{X}$ and $d \times n$ matrix $\mathbf{Y}$. When $d = 1$, the original D-dimensional data set is projected onto a line. We obtain the D-to-1 projection $\mathbf{y} = \mathbf{p}^T\mathbf{X}$. The vector $\mathbf{p} = [p_1\ p_2\ \cdots\ p_n]^T$ is the line to project, and $\mathbf{y} = [y_1\ y_2\ \cdots\ y_n]^T$ is the vector of the coordinates, where $y_i = \mathbf{p}^T\mathbf{x}_i$. Subject to the constraint $\mathbf{p}^T\mathbf{X}\mathbf{B}\mathbf{X}^T\mathbf{p} = q$ or $\mathbf{p}^T\mathbf{p} = q$, we finally get the objective function $\mathbf{p}^* = \arg\min \mathbf{p}^T\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{p}$ in matrix form.

3. LOCALLY EMBEDDED CLUSTERING

We present the LEC algorithms associated with two different embedding cases, subject to different constraints in formulations. In the supervised embedding, assuming that the desired low-dimensional degree d to project is known, the problem of embedding is to calculate the mapping from the D-dimensional space to the d-dimensional subspace. In the unsupervised embedding, assuming that the desired low- dimensional degree d to project is unknown, we formulate the embedding problem as a D-to-1 projection, that is, the mapping from D-dimensional space to each axis of the low-dimensional space is optimal. In geometrical sense, the high-dimensional data are linearly projected onto an optimal line preserving the underlying neighborhood structure.

Suppose a high-dimensional data set with n elements is denoted by $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$, where $\mathbf{x}_i \in \mathbb{R}^D$ and $i = 1, 2, \cdots, n$, which is prone to cluster into c subsets. Denote $\mathcal{X}_N^{(i)} = \{\mathbf{x}_{N(1)}^{(i)}, \mathbf{x}_{N(2)}^{(i)}, \cdots, \mathbf{x}_{N(k)}^{(i)}\}$ as the set of $\mathbf{x}_i$'s k nearest neighbors, where $\mathbf{x}_{N(j)}^{(i)} \in \mathbb{R}^D$ and $j = 1, 2, \cdots, k$. We summarize the two LEC algorithms as follows:

LEC Algorithm 1:

1. *Form the local Gram matrix* $\mathbf{G}_i \in \mathbb{R}^{k\times k}$ *for* $\mathbf{x}_i$.

- $\mathbf{G}_i[j,l] = (\mathbf{x}_i - \mathbf{x}_{N(j)}^{(i)})^T(\mathbf{x}_i - \mathbf{x}_{N(l)}^{(i)})$.
- $\mathbf{G}_i = (\mathbf{x}_i\mathbf{1}^T - \mathbf{X}^{(i)})^T(\mathbf{x}_i\mathbf{1}^T - \mathbf{X}^{(i)})$, where the k-by-1 column vector $\mathbf{1}$ consists of ones and the columns of the D-by-k matrix $\mathbf{X}^{(i)}$ contains $\mathbf{x}_i$'s k nearest neighbors.

2. *Tuning the regularization parameter* r.

- $\mathbf{G}_i^{-1} = (\mathbf{G}_i + (r \cdot \sum_{j=1}^k \lambda_G^{(j)}) \cdot \mathbf{I})^{-1}$, where $\lambda_G^{(j)}$ is the eigenvalue of matrix $\mathbf{G}_i$.

3. *Construct the similarity matrix* $\mathbf{W} \in \mathbb{R}^{n\times n}$.

- $\mathbf{W}[i, N(j)] = \mathbf{w}_i(j) = w_j^{(i)}$, and the other elements of $\mathbf{W}$ are 0.
- $\mathbf{w}_i = \frac{\mathbf{G}_i^{-1}\mathbf{1}}{\mathbf{1}^T\mathbf{G}_i^{-1}\mathbf{1}}$ and $\mathbf{w}_i = [w_1^{(i)}\ w_2^{(i)}\ \cdots\ w_k^{(i)}]^T$.

4. *Solve for D-to-d embedding* $\mathbf{Y} = \mathbf{P}^T\mathbf{X}$, and $\mathbf{Y} \in \mathbb{R}^{d\times n}$.

- Solve the eigenvalue problem $\mathbf{X}(\mathbf{I} - \mathbf{W})^T(\mathbf{I} - \mathbf{W})\mathbf{X}^T\mathbf{P} = \Lambda\mathbf{X}\mathbf{X}^T\mathbf{P}$, where Λ is the diagonal Lagrange multiplier matrix.
- The columns of $\mathbf{P}$ are the smallest d eigenvectors of matrix $(\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{X}(\mathbf{I} - \mathbf{W})^T(\mathbf{I} - \mathbf{W})\mathbf{X}^T$ after discarding the bottom eigenvector (the mean of $\mathbf{P}^T\mathbf{X}$).

5. *Normalize each column of* $\mathbf{Y}$ *to form* $\mathbf{Z} \in \mathbb{R}^{c\times n}$.

- $\mathbf{Z}[i,j] = \frac{\mathbf{Y}[i,j]}{\sqrt{\sum_i \mathbf{Y}[i,j]^2}}$.

6. *Treat each column of* $\mathbf{Z}$ *as a point in* $\mathbb{R}^c$ *and cluster via K-means.*

7. *Assign the original point* $\mathbf{x}_i$ *to cluster* c *if and only if* $\mathbf{z}_i$ *is assigned to cluster* c.

LEC Algorithm 2:

1. **2**. **3**. are exactly the same as **LEC Algorithm 1**.

4. *Solve for D-to-1 embedding* $\mathbf{y} = \mathbf{p}^T\mathbf{X}$ *for* $\mathbf{y} \in \mathbb{R}^n$.

- Form the diagonal matrix $\mathbf{D} \in \mathbb{R}^{n\times n}$, where $\mathbf{D}[i,i] = \sum_{j=1}^n \mathbf{W}[i,j]$.
- Obtain the projection vector $\mathbf{p}$ by solving the eigenvalue problem $\mathbf{X}(\mathbf{D} - \mathbf{W})^T(\mathbf{D} - \mathbf{W})\mathbf{X}^T\mathbf{p} = \lambda\mathbf{X}(\mathbf{D}^T\mathbf{D} + \mathbf{W}^T\mathbf{W})\mathbf{X}^T\mathbf{p}$.
- The vector $\mathbf{p}$ is the second smallest eigenvector of $(\mathbf{X}(\mathbf{D}^T\mathbf{D}+\mathbf{W}^T\mathbf{W})\mathbf{X}^T)^{-1}\mathbf{X}(\mathbf{D}-\mathbf{W})^T(\mathbf{D}-\mathbf{W})\mathbf{X}^T$.

5. *Treat each component of* $\mathbf{y}$ *as a point in* $\mathbb{R}$ *and cluster via K-means.*

6. *Assign the original point* $\mathbf{x}_i$ *to cluster* c *if and only if* y_i *is assigned to cluster* c.

Recall that each diagonal value $\mathbf{D}[i,i]$ of matrix $\mathbf{D}$ corresponds to the weights summation for the k nearest neighbors of a particular data point. Moreover, since the non-symmetric rows and columns of sparse matrix $\mathbf{W}$ have very few overlaps, it follows that $\mathbf{W}^T\mathbf{W}$ is almost a diagonal matrix. Therefore, the matrix $\mathbf{D}^T\mathbf{D} + \mathbf{W}^T\mathbf{W}$ indicates the local distribution around y_i because a large value in the diagonal means a short distance between neighbors. Hence, we choose the constraint $\mathbf{p}^T\mathbf{X}(\mathbf{D}^T\mathbf{D} + \mathbf{W}^T\mathbf{W})\mathbf{X}^T\mathbf{p} = 1$ in LEC-2.

4. COMMENTS ON LEC ALGORITHMS

First, theoretically, LEC-1 algorithm is a particular case of LEC-2 algorithm. Matrices $\mathbf{X}(\mathbf{D}-\mathbf{W})^T(\mathbf{D}-\mathbf{W})\mathbf{X}^T$ and $\mathbf{X}(\mathbf{D}^T\mathbf{D}+\mathbf{W}^T\mathbf{W})\mathbf{X}^T$ are both positive semidefinite and symmetric. Subject to the constraint $\sum_{j=1}^{k} w_j^{(i)} = 1$ and $\mathbf{D} = \mathbf{I}$, the equation in step 4 of LEC-2 can be rewritten as $\mathbf{X}(\mathbf{I}-\mathbf{W})^T(\mathbf{I}-\mathbf{W})\mathbf{X}^T\mathbf{p} = \lambda\mathbf{X}(\mathbf{I}+\mathbf{W}^T\mathbf{W})\mathbf{X}^T\mathbf{p}$, which is similar to step 4 of LEC-1 when $\mathbf{W}^T\mathbf{W} \approx \mathbf{I}$ and $d = 1$.

Secondly, since LEC algorithms focus on revealing the approximate local structure, the display of the manifold is often very rough. One efficient way to improve the clustering accuracy is to smooth the manifold, which can actually get rid of the boundary errors between different clusters. In order to smooth the approximated manifold, we introduce a regularization parameter r [11] in step 2 of both LEC algorithms, that is $\mathbf{G}_i^{-1} = (\mathbf{G}_i + (r \cdot \sum_{j=1}^{k} \lambda_G^{(j)}) \cdot \mathbf{I})^{-1}$. The r is usually chosen in the range of $[0, 1]$ in the case of LEC clustering.

Thirdly, in some cases, especially when the original data are normalized in the preprocessing, we should be very careful in choosing the bottom eigenvectors for low-dimensional representation. In the presented LEC-1 and LEC-2 algorithms, we throw away the smallest eigenvector. Since the parameter r play an important role in the clustering, sometimes the smallest eigenvector is still useful, especially the LEC-2 algorithm. It is reasonable to check the bottom eigenvector in some particular high-dimensional data clustering cases.

Finally, the LEC algorithms are very general, which can be extended to the linearization type of any particular graph embedding methods. We can also generalize the LEC algorithms using different type of distance metric for neighborhood measurement. For example, the distance between neighbors in high-dimensional space can be measured with *cosine angles*, that is $m = \exp(-\|\mathbf{x}_i\|^T \cdot |\mathbf{x}_j\|/\sigma^2)$.

To evaluate the performance of LEC, we compare the proposed LEC-2 algorithms with 4 following listed state-of-the-art methods:

1. *K-means*: Apply K-means clustering on the original high-dimensional data;
2. *PCA+K-means*: Apply K-means clustering on the low-dimensional PCA representation of the original data after re-normalization;
3. *K-Whitening+K-means*[16]: First project the original data into the kernel space using a Kernel Whitening transformation. Then apply K-means clustering on the low-dimensional Kernel space representation of the original data after centering the kernel matrices;
4. *Ng-Jordan-Weiss (NJW)*[15]: First, form the affinity matrix. Secondly, construct the normalized affinity matrix. Thirdly, find the desired number of largest eigenvectors of the normalized affinity matrix. Fourthly, normalize each trimmed eigenvector to have unit length. Fifthly, treat each normalized vector as a low-dimensional representation of original data point. Finally, cluster and label the original data via K-means.

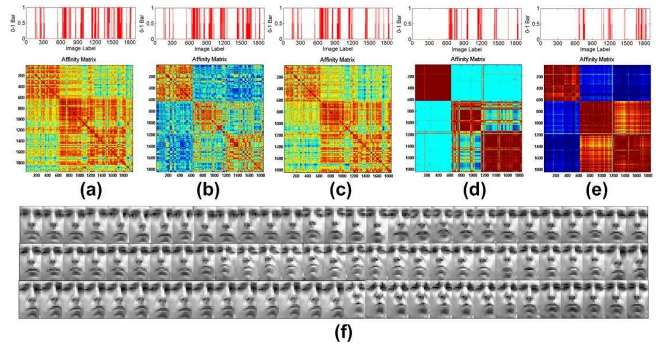

Fig. 1. Clustering results of Frey's face images via 5 different methods. The top row shows the error distribution. The bottom row shows the clustering affinity matrix. (a) K-means. (b) PCA (top 10) + K-means. (c) K-Whitening + K-means. (d) Ng-Jordan-Weiss (NJW). (e) LEC-2. (f) The 87 images of LEC-2 clustering errors.

The LEC algorithms inherit the basic properties of GE and SC techniques. The essential new feature of LEC is the joint property for dimensionality reduction and unsupervised clustering. Unlike the NJW algorithm, LEC has the properties of dimensionality reduction and manifold analysis, which is the important advantage of LEC for high-dimensional data clustering. Moreover, LEC has clear objective functions with closed-form solutions. Unlike GE, LEC aims at data clustering other than a pure dimensionality reduction.

5. EXPERIMENTS

We demonstrate the properties of LEC using Frey faces [10] and AAI database [19]. The Frey faces contain 1965 gray-scale face images taken from a video. The images, in a resolution of 28×20, show variations in face expression and view rotations. The images are cropped to the resolution of 24×14 with a rectangular mask. We manually assign each 1965 image with one of the four labels: happy, neutral, unhappy1 or unhappy2. The unhappy1 subset contains most normal sad images, while the unhappy2 contains some unexplained unhappy images. The original data set is finally partitioned into 4 subsets [Happy, Neutral, Unhappy1, Unhappy2], which contain 618, 587, 634 and 126 images respectively. We use an automatic 3D PBVD tracker to localize and crop the subject's face in the AAI video sequence, which generates 25 frames of facial texture map per video second. We obtained 5230 frames (1372 happy, 851 unhappy, 3007 neutral, labeled by a psychologist) for the male subject. The facial texture maps are normalized and re-scaled to the images in the resolution of 30×26 with 256 grey levels per pixel. We sample 200 key frames from each of the 3 subsets respectively.

The top 3 sub sets of totally 1893 Frey's face images are chosen to test the LEC performance and compare the results with other 4 reference methods. Figure 1 shows the clustering

Table 1. Clustering results on Frey's face images.

Method	**Parameter** (r, σ^2)	**Dim.** $\sharp$ (T., B.)	**Error Rate** (%) ($\sharp$ / 1839)
K-means	None	336	15.72 (289/1839)
PCA10+K-means	None	10 (T)	13.32 (245/1839)
PCA1+K-means	None	1 (T)	15.39 (283/1839)
K-W+K-means	$\sigma^2 = 0.5$	10 (T)	9.84 (181/1839)
NJW	$\sigma^2 = 0.15$	3 (T)	7.34 (135/1839)
LEC-2	$r = 0.6$	1 (B)	**4.73 (87/1839)**

Table 2. Clustering results on AAI male face images.

Method	**Parameter** (r, σ^2)	**Dim.** $\sharp$ (T., B.)	**Error Rate** (%) ($\sharp$ / 600)
K-means	None	780	40.83 (245/600)
PCA100+K-means	None	100 (T)	40.83 (245/600)
PCA1+K-means	None	1 (T)	35.33 (212/600)
K-W+K-means	$\sigma^2 = 0.5$	3 (T)	29.00 (174/600)
NJW	$\sigma^2 = 0.1$	3 (T)	30.67 (184/600)
LEC-2	$r = 0.005$	1 (B)	**20.67 (124/600)**

results of 5 different methods: (a) K-means, (b) PCA (top 10) + K-means, (c) K-Whitening + K-means, (d) NJW, and (e) LEC-2. The top row shows the error distribution. Each vertical red bar represents a clustering error for the particular image. More red bars displays more errors in the clustering. The bottom row shows the clustering affinity matrix, which represents the distance between each data point pairs in color ranging from blue to red, and passing through the cyan, yellow, and orange. The red color represents small distance value, while the blue color represents large distance value. More distinct blocks in affinity matrix, such as (e), displays better clustering results. The distribution of red bars and the affinity matrix, especially in (d)(e), reveal that the neutral faces and unhappy faces are hard to separate. Table 1 shows more quantitative results on this experiment. Set the cluster numbers as 3 and replicates as 50 for K-means clustering. The bottom 3 values are obviously better than the top 3 values. The LEC-2 outperforms all the other 5 methods with lowest error rate (4.73%) and degree (only 1) of dimensionality reduction. The 87 images of LEC-2 clustering errors are shown in (f). We observe that many of these error images contain large variations, such as pose rotation, blinking, grimace and transition expressions. It is even hard for human to recognize and categorize their emotional types, which can be considered as outliers.

Table 2 shows the clustering results of AAI male face images via 6 different methods: K-means, PCA (top 1) + K-means, PCA (top 100) + K-means, K-Whitening + K-means, NJW, and LEC-2. Set the cluster numbers as 3 and replicates as 50 for K-means clustering. The bottom 3 values are still better than the top 3 values. The LEC-2 still outperforms all the other 5 methods with lowest error rate (20.67%) and degree (only 1) of dimensionality reduction.

6. CONCLUSIONS

The LEC algorithms are demonstrated to be more applicable for high-dimensional data clustering than many existing methods. Their key advantage is the generalization ability, since the GE type can be changed to any suitable embedding cases [14, 17], and the clustering part is also flexible to be substituted by any advanced algorithms [9]. The future work will be focused on self-parameter-tuning [18], estimating the number of clusters automatically, and iterative kernelized LEC [6].

7. REFERENCES

[1] M. Belkin and P. Niyogi, "Laplacian Eigenmaps for Dimensionality Reduction and Data Representation", *Neural Computation*, vol. 15, no. 6, pp. 1373-1396, 2003.

[2] C. Ding, "A Tutorial on Spectral Clustering,"*ICML'04*, 2004, http://crd.lbl.gov/ cding/Spectral/notes.html

[3] Y. Fu and T.S. Huang, *Locally Linear Embedded Eigenspace Analysis*, www.ifp.uiuc.edu/∼yunfu2/papers/LEA-Yun05.pdf.

[4] X.F. He and P. Niyogi, "Locality Preserving Projections", *Proc. of NIPS'03*, 2003.

[5] X.F. He, D. Cai, S.C. Yan and H.J. Zhang, " Neighborhood Preserving Embedding", *IEEE Conf. on ICCV'05*, 2005.

[6] J. Ham, D.D. Lee, S. Mika and B. Schölkopf,"A kernel View of the Dimensionality Reduction of Manifolds", *ICML'04*, 2004.

[7] M. Polito and P. Perona, "Grouping and Dimensionality Reduction by Locally Linear Embedding", *NIPS'01*, 2001.

[8] A.M. Martinez and A.C. Kak, "PCA versus LDA", *IEEE Trans. on PAMI*, vol. 23, no. 2, pp. 228-233, 2001.

[9] D. Verma and M. Meila, "Comparison of Spectral Clustering Methods", http://www.ms.washington.edu/ spectral/, 2003.

[10] S.T. Roweis and L.K. Saul, "Nonlinear Dimensionality Reduction by Locally Linear Embedding", *Science*, vol. 290, pp. 2323-2326, 2000.

[11] D. de Ridder and R.P.W. Duin, *Locally Linear Embedding for Classification*, TR. PH-2002-01, Delft Uni. of Tech., 2002.

[12] J.B. Tenenbaum, V.de Silva, and J.C. Langford, "A Global Geometric Framework for Nonlinear Dimensionality Reduction", *Science*, vol. 290, pp. 2319-2323, 2000.

[13] K.Q. Weinberger and L.K. Saul, "Unsupervised Learning of Image Manifolds by Semidefinite Programming", *IEEE Conf. on CVPR'04*, vol. 2, pp. 988-995, 2004.

[14] S.C. Yan, D. Xu, B.Y. Zhang and H.J. Zhang," Graph Embedding: A General Framework for Dimensionality Reduction", *IEEE Conf. on CVPR'05*, pp. 830-837, 2005.

[15] A.Y. Ng, M.I. Jordan and Y. Weiss, "On Spectral Clustering: Analysis and An Algorithm", *NIPS'01*, 2001.

[16] D.M.J. Tax and P. Juszczak, "Kernel Whitening for One-Class Classification", *IJPRAI*, vol.17, no.3, pp. 333-347, 2003.

[17] Y. Weiss, "Segmentation using Eigenvectors: A Unifying View", *IEEE Conf. on ICCV'99*, pp. 975-982, 1999.

[18] L. Zelnik-Manor and P. Perona, "Self-Tuning Spectral Clustering", *NIPS'04*, 2004.

[19] Z.H. Zeng, Y. Fu, G.I. Roisman, Z. Wen, Y.X. Hu, and T. S. Huang, "Spontaneous Emotional Facial Expression Detection", *Journal of Multimedia (JMM)*, vol. 1, no. 5, pp. 1-8, 2006.