Research Article

Spectral Nonlinearly Embedded Clustering Algorithm

Mingming Liu,¹ Bing Liu,² Chen Zhang,² and Wei Sun¹

¹*School of Information and Electrical Engineering, China University of Mining and Technology, Xuzhou, Jiangsu 221116, China*

²*School of Computer Science and Technology, China University of Mining and Technology, Xuzhou, Jiangsu 221116, China*

Correspondence should be addressed to Bing Liu; liubing@cumt.edu.cn and Chen Zhang; zc@cumt.edu.cn

Received 22 November 2015; Revised 16 May 2016; Accepted 1 June 2016

Academic Editor: Babak Shotorban

Copyright © 2016 Mingming Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

As is well known, traditional spectral clustering (SC) methods are developed based on the *manifold assumption*, namely, that two nearby data points in the high-density region of a low-dimensional data manifold have the same cluster label. But, for some high-dimensional and sparse data, such an assumption might be invalid. Consequently, the clustering performance of SC will be degraded sharply in this case. To solve this problem, in this paper, we propose a general spectral embedded framework, which embeds the true cluster assignment matrix for high-dimensional data into a nonlinear space by a predefined embedding function. Based on this framework, several algorithms are presented by using different embedding functions, which aim at learning the final cluster assignment matrix and a transformation into a low dimensionality space simultaneously. More importantly, the proposed method can naturally handle the out-of-sample extension problem. The experimental results on benchmark datasets demonstrate that the proposed method significantly outperforms existing clustering methods.

1. Introduction

As one of the fundamental topics in data mining and machine learning, clustering has been successfully applied in various fields. Generally speaking, the target of clustering is to group the examples into a number of classes, or clusters. Over the past decades, a large family of clustering algorithms has been studied extensively, which is mainly divided into two categories: generative clustering approaches and discriminative clustering models. Generative clustering approaches, for example, mixture models [1, 2], generally integrate Bayesian approaches into its models. However, generative models add restrict assumptions on the class-conditional densities, which might lead to unconvincing clustering results when these assumptions do not hold. Discriminative methods, such as spectral clustering (SC) [3] and *K*-means clustering [4], learn discriminative models based on loss functions from unlabeled data through the low-density separation assumption.

Recently, discriminative clustering methods, such as the variants of kernel-based clustering and spectral clustering, have attracted more and more renewed attentions. It is easy to perform them to capture nonlinear cluster structures. Motivated by the outstanding performance of support vector

machine (SVM) in supervised learning, maximum margin clustering (MMC) [5–7] methods have been developed to obtain a decision boundary that can separate data points into different clusters to the utmost extent. Although these clustering methods have the ability of exploiting nonlinear data structures, they are still sensitive to high-dimensional data points. For example, *K*-means clustering iteratively computes the distance between each data point and the center of each cluster. Hence, its clustering performance severely depends on the distance measurement. However, high-dimensional data, such as some image data, would have a bad influence on the similarity computation by virtue of Euclidian distance, and the performance of *K*-means clustering would be degraded dramatically. SC can perform clustering by utilizing the spectrum of the similarity matrix to discover the nonlinear and low-dimensional manifold structure of data points. In other words, it heavily relies on the manifold assumption [8, 9], namely, that two nearby data points of a low-dimensional manifold have the same class label. However, for high-dimensional and sparse data, the manifold assumption may not hold due to the bias caused by the curse of dimensionality. Nie et al. [10] have validated that graph-based spectral clustering methods cannot always

exploit the low-dimensional manifold structure, which would result in the performance degradation of SC. Another challenge for traditional SC methods is that they do not solve the out-of-sample extension problem; that is, the discrete cluster assignment vectors for some new unseen samples cannot be automatically obtained. The algorithm proposed in [11] takes advantage of the Nyström method to approximate the eigenfunction for the unseen data points. The method described in [12] makes good use of some heuristics to evaluate the implicit eigenfunction for the new data points. But, the performance of these methods heavily relies on the estimated affinity matrix defined between training and new data points.

To improve the clustering performance of SC for high-dimensional data further, in this paper, we firstly propose a general spectral embedded clustering framework, which incorporates dimensionality reduction methods into the model of SC. Secondly, by using different low-dimensional embedding functions, we derive the corresponding optimization models and develop the spectral nonlinearly embedded algorithms based on extreme learning machine (ELM) and kernel functions, respectively. Our main contributions include the following:

- (1) A general spectral embedded clustering framework is presented by imposing a linearity regularization on the objective function of SC. The proposed framework introduces dimensionality reduction of the training data by controlling the error between the cluster assignment matrix and the low-dimensional embedding of the data.
- (2) Based on the proposed general framework, several models can be derived by using different embedding functions, which include the linear embedding functions and the nonlinear functions in Reproducing Kernel Hilbert Space (RKHS) as well as in ELM feature space. The spectral embedded clustering model (SEC) proposed in [12] can be considered as the special case of the general framework.
- (3) We prove that the spectral nonlinearly embedded clustering model based on ELM (ESEC) is an approximation of the kernel-based spectral nonlinearly embedded clustering (KSEC) method under some conditions. The fast spectral nonlinearly embedded clustering algorithm is proposed based on ESEC by utilizing the efficient learning ability of ELM.
- (4) The out-of-sample extension problem can be naturally solved for the clustering methods under our proposed SEC framework.
- (5) Experimental results on benchmark datasets demonstrate that the proposed ESEC outperforms the existing SC methods, K -means clustering, and SEC and KSEC for in-sample clustering. For out-of-sample clustering, ESEC also has better generalization capability over the Nyström method and superior performance than K -means clustering, SEC, and KSEC.

The rest of this paper is organized as follows. Related works are introduced in Section 2. In Section 3, we present

the general spectral embedded clustering framework and derive several different models by using different embedding functions. The relationship between ESEC and KSEC is demonstrated and the ESEC clustering algorithm is described in detail. In addition, clustering for out-of-sample data is also discussed. To validate our model, experimental results are reported in Section 4. Finally, we give the related conclusions and a discussion of future works in Section 5. In order to avoid confusion, we give a list of the main notations used in this paper in Notations section.

2. Related Works

2.1. Spectral Clustering. Given a dataset $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$, the main task of clustering is to partition $\mathbf{X}$ into c clusters. SC aims at finding a cluster assignment matrix of the training data by a weighted graph $\mathbf{G}$ whose vertices are over $\mathbf{X}$. Several SC algorithms have been proposed in [3, 13, 14]. In this paper, we mainly discuss the SC algorithm with k -way normalized cuts [3].

Specifically, denote an undirected weighted graph by $\mathbf{G} = \{\mathbf{X}, \mathbf{W}\}$, where $\mathbf{X}$ is a vertex set and $\mathbf{W} \in \mathbb{R}^{n \times n}$ represents an affinity matrix. Each entry W_{ij} of the symmetric matrix $\mathbf{W}$ is used to record the edge weights that characterize the similarity relationship between a pair of vertices of $\mathbf{G}$. W_{ij} is commonly defined by

$$W_{ij} = \begin{cases} \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma^2}\right) & \text{if } \mathbf{x}_i, \mathbf{x}_j \text{ are neighbors,} \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

The Laplacian graph $\mathbf{L}$ is defined by $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where $\mathbf{D}$ is a diagonal matrix with the diagonal elements as $D_{ii} = \sum_j W_{ij}$, $\forall i$. Based on the normalized cut criterion, where the size of a subset of a graph is measured by the weights of its edges and the normalized Laplacian matrix is used, the optimization problem can be transformed into the following trace maximization problem [3]:

$$\max_{\mathbf{F}^T \mathbf{F} = \mathbf{I}_c} \text{tr}(\mathbf{F}^T \mathbf{D}^{-1/2} \mathbf{W} \mathbf{D}^{-1/2} \mathbf{F}), \quad (2)$$

where $\mathbf{I}_c$ denotes the identity matrix of size c by c and $\mathbf{F} \in \mathbb{R}^{n \times c}$ represents the cluster assignment matrix with continuous values by relaxation. Then optimal solution $\mathbf{F}$ of (2) can be obtained by eigenvalue decomposition of the matrix $\mathbf{D}^{-1/2} \mathbf{W} \mathbf{D}^{-1/2}$.

2.2. Extreme Learning Machine. The output function of ELM for generalized single-hidden-layer feedforward neural networks (SLFNs) in the case of one output node is

$$f_L(\mathbf{x}) = \sum_{i=1}^L \beta_i h_i(\mathbf{x}) = \mathbf{h}(\mathbf{x}) \boldsymbol{\beta}, \quad (3)$$

where $\boldsymbol{\beta} = [\beta_1, \dots, \beta_L]^T$ is the vector of the output weights between the hidden layer of L nodes and the output node

and $\mathbf{h}(\mathbf{x}) = [h_1(\mathbf{x}), \dots, h_L(\mathbf{x})]$ is the output (row) vector of the hidden layer with respect to the input $\mathbf{x}$. In fact, $\mathbf{h}(\mathbf{x})$ maps the data from the d -dimensional input space to the L -dimensional hidden-layer feature space (ELM feature space) $\mathbf{H}$. ELM is to minimize the training error as well as the norm of the output weights [15]

$$\min_{\beta} \frac{C}{2} \|\mathbf{H}\beta - \mathbf{T}\|^2 + \frac{1}{2} \|\beta\|^2, \quad (4)$$

where C is a tradeoff parameter between the complexity and fitness of the decision function and $\mathbf{H}$ is the hidden-layer output matrix denoted by

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}(\mathbf{x}_1) \\ \mathbf{h}(\mathbf{x}_2) \\ \vdots \\ \mathbf{h}(\mathbf{x}_n) \end{bmatrix} = \begin{bmatrix} h_1(\mathbf{x}_1) & \dots & h_L(\mathbf{x}_1) \\ h_1(\mathbf{x}_2) & \dots & h_L(\mathbf{x}_2) \\ \vdots & \vdots & \vdots \\ h_1(\mathbf{x}_n) & \dots & h_L(\mathbf{x}_n) \end{bmatrix}. \quad (5)$$

Similar to support vector machine (SVM), to minimize the norm of the output weights $\|\beta\|$ is actually to maximize the distance of the separating margins of the two different classes in the ELM feature space: $2/\|\beta\|$, which actually controls the complexity of the function in the ELM feature space.

3. General Spectral Embedded Clustering Framework

As mentioned above, SC methods greatly depend on the construction of the affinity matrix $\mathbf{W}$. For some high-dimensional data, it might not exhibit an evident low-dimensional manifold structure. In this case, the clustering performance of SC may be inferior to the K -means clustering.

In the following subsections, we will firstly propose a general spectral embedded clustering framework, which incorporates a linearity regularization into the traditional normalized SC model. By using different embedding functions, this framework can generate a family of spectral embedded clustering algorithms, such as SEC, KSEC, and ESEC. Secondly, we demonstrate the relationship between ESEC and KSEC. The ESEC algorithm is then proposed for high-dimensional data clustering. Finally, the out-of-sample extension problem is discussed for our proposed ESEC method.

3.1. Formulation. Generally, clustering models of traditional SC methods can be transformed into the following minimization problem:

$$\min_{\mathbf{F}^T \mathbf{F} = \mathbf{I}_c} \text{tr}(\mathbf{F}^T \tilde{\mathbf{L}} \mathbf{F}), \quad (6)$$

where $\tilde{\mathbf{L}} = \mathbf{D}^{-(1/2)} \mathbf{L} \mathbf{D}^{-(1/2)} = \mathbf{I}_n - \mathbf{D}^{-(1/2)} \mathbf{W} \mathbf{D}^{-(1/2)}$ is the normalized Laplacian matrix.

To make use of the underlying dense grouping structure of data in a low-dimensional subspace, the proposed general framework introduces a regularization term into the optimization problem (6), which controls the error between the

cluster assignment matrix and the low-dimensional embedding of the data. Specifically, we minimize the following objective function:

$$\min_{\mathbf{F}^T \mathbf{F} = \mathbf{I}_c} \text{tr}(\mathbf{F}^T \tilde{\mathbf{L}} \mathbf{F}) + \mu (\|\phi(\mathbf{X}) - \mathbf{F}\|^2 + \gamma_g \|\phi\|^2), \quad (7)$$

where μ and γ_g are two regularization parameters and $\phi(\mathbf{X}) = (\psi_1(\mathbf{X}), \dots, \psi_c(\mathbf{X})) \in \mathbb{R}^{n \times c}$ is the low-dimensional embedding of training data. The second term represents the error between the relaxed cluster assignment matrix $\mathbf{F}$ and the low-dimensional embedding of the data. The third term is the norm penalty of $\phi(\mathbf{X})$ and represents the complexity of functions in a high-dimensional feature space.

In dimensionality reduction, linear embedding functions and nonlinear embedding functions are commonly used to address out-of-sample problems. This is due to the fact that they contain few parameters, which are not expensive in computational time and memory. In this paper, we mainly discuss kernel-based and ELM-based nonlinear embedding functions.

If we choose a linear embedding function $\mathbf{y} = \mathbf{W}^T \mathbf{x} + \mathbf{b}$, $\phi(\mathbf{X}) = \mathbf{X}^T \mathbf{W} + \mathbf{1}_n \mathbf{b}^T$, where $\mathbf{W} \in \mathbb{R}^{d \times c}$ and $\mathbf{b} \in \mathbb{R}^{c \times 1}$, the optimization problem (7) becomes

$$\begin{aligned} \min_{\substack{\mathbf{F}, \mathbf{W}, \mathbf{b} \\ \mathbf{F}^T \mathbf{F} = \mathbf{I}_c}} \text{tr}(\mathbf{F}^T \tilde{\mathbf{L}} \mathbf{F}) \\ + \mu (\|\mathbf{X}^T \mathbf{W} + \mathbf{1}_n \mathbf{b}^T - \mathbf{F}\|^2 + \gamma_g \text{tr}(\mathbf{W}^T \mathbf{W})) \end{aligned} \quad (8)$$

which is equivalent to the SEC method proposed in [12].

If we use a nonlinear embedding function in RKHS, that is, $\mathbf{y} = f(\mathbf{x}) = \sum_{i=1}^n \alpha_i k(\mathbf{x}, \mathbf{x}_i)$, then $\phi(\mathbf{X}) = \mathbf{K} \alpha$, where $\mathbf{K}$ is a symmetric kernel matrix and $\alpha = [\alpha_1, \dots, \alpha_n]^T \in \mathbb{R}^{n \times c}$; problem (7) can be rewritten as

$$\min_{\substack{\mathbf{F}, \alpha \\ \mathbf{F}^T \mathbf{F} = \mathbf{I}_c}} \text{tr}(\mathbf{F}^T \tilde{\mathbf{L}} \mathbf{F}) + \mu (\|\mathbf{K} \alpha - \mathbf{F}\|^2 + \gamma_g \text{tr}(\alpha^T \mathbf{K} \alpha)) \quad (9)$$

which is referred to as KSEC.

Alternatively, if we consider an embedding function in ELM feature space, that is, $f(\mathbf{x}) = \sum_{i=1}^L \beta_i h_i(\mathbf{x}) = \mathbf{h}(\mathbf{x}) \beta$, then $\phi(\mathbf{X}) = \mathbf{H} \beta$, where $\mathbf{H}$ represents the hidden-layer output matrix of ELM. Problem (7) can be reformulated as

$$\min_{\substack{\mathbf{F}, \beta \\ \mathbf{F}^T \mathbf{F} = \mathbf{I}_c}} \text{tr}(\mathbf{F}^T \tilde{\mathbf{L}} \mathbf{F}) + \mu (\|\mathbf{H} \beta - \mathbf{F}\|^2 + \gamma_g \text{tr}(\beta^T \beta)), \quad (10)$$

which is referred to as ESEC.

3.2. Method. Firstly, to solve the optimization problems (9), we transform them into another simple form and have the following theorem.

Theorem 1. *The optimization problems (9) can be transformed into the following minimization problem:*

$$\min_{\mathbf{F}^T \mathbf{F} = \mathbf{I}_c} \text{tr}(\mathbf{F}^T (\tilde{\mathbf{L}} + \mu \mathbf{L}_F) \mathbf{F}), \quad (11)$$

where $\mathbf{L}_F = \mathbf{I}_n - \mathbf{K}(\mathbf{K} + \gamma_g \mathbf{I}_n)^{-1}$ and $\mathbf{I}_n$ denotes the identity matrix of size n by n .

Proof. Problem (9) is firstly transformed into the following form:

$$\begin{aligned} \min_{\substack{\mathbf{F}, \boldsymbol{\alpha} \\ \mathbf{F}^T \mathbf{F} = \mathbf{I}_c}} \text{tr}(\mathbf{F}^T \tilde{\mathbf{L}} \mathbf{F}) \\ + \mu \left(\text{tr}((\mathbf{F} - \mathbf{K}\boldsymbol{\alpha})^T (\mathbf{F} - \mathbf{K}\boldsymbol{\alpha})) + \gamma_g \text{tr}(\boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha}) \right), \end{aligned} \quad (12)$$

where $\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_n]^T \in \mathbb{R}^{n \times c}$.

By setting the derivatives of the objective function (16) with respect to $\boldsymbol{\alpha}$ to zero, we have

$$\boldsymbol{\alpha} = (\gamma_g \mathbf{I}_n + \mathbf{K})^{-1} \mathbf{F}. \quad (13)$$

By substituting $\boldsymbol{\alpha}$ in (12) by (13), the optimization problem (12) becomes

$$\min_{\mathbf{F}^T \mathbf{F} = \mathbf{I}_c} \text{tr}(\mathbf{F}^T \tilde{\mathbf{L}} \mathbf{F}) + \mu \left(\text{tr} \mathbf{F}^T (\mathbf{I}_n - \mathbf{K}(\mathbf{K} + \gamma_g \mathbf{I}_n)^{-1}) \mathbf{F} \right) \quad (14)$$

which can be denoted as follows:

$$\min_{\mathbf{F}^T \mathbf{F} = \mathbf{I}_c} \text{tr}(\mathbf{F}^T (\tilde{\mathbf{L}} + \mu \mathbf{L}_F) \mathbf{F}), \quad (15)$$

where $\mathbf{L}_F = \mathbf{I}_n - \mathbf{K}(\mathbf{K} + \gamma_g \mathbf{I}_n)^{-1}$. This completes the proof of Theorem 1. $\square$

Based on Theorem 1, the relaxed cluster assignment matrix $\mathbf{F}^*$ of KSEC can be achieved by computing the eigenvectors of $\tilde{\mathbf{L}} + \mu \mathbf{L}_F$ corresponding to the c smallest eigenvalues. The columns of $\mathbf{F}^*$ are corresponding to the top c eigenvectors. Finally, the discrete-valued cluster assignment matrix can be obtained by clustering each row of $\mathbf{F}^*$.

To inherit the advantage of fast learning speed of ELM, we mainly discuss ESEC based on ELM with multioutputs, since ELM with single output can be regarded as a special case of it. We have the following theorem on ESEC, which is the foundation of the proposed ESEC algorithm.

Theorem 2. *The optimization problem (10) can be transformed into the following minimization problem:*

$$\min_{\mathbf{F}^T \mathbf{F} = \mathbf{I}_c} \text{tr}(\mathbf{F}^T (\tilde{\mathbf{L}} + \mu \mathbf{L}_H) \mathbf{F}), \quad (16)$$

where $\mathbf{L}_H = \mathbf{I}_n - \mathbf{H} \mathbf{H}^T (\gamma_g \mathbf{I}_n + \mathbf{H} \mathbf{H}^T)^{-1}$ or $\mathbf{L}_H = \mathbf{I}_n - \mathbf{H}(\gamma_g \mathbf{I}_L + \mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T$. $\mathbf{I}_L$ denotes the identity matrix of size c by c and L is the number of hidden layer nodes in ELM.

Proof. By setting the derivatives of the objective function (10) with respect to $\boldsymbol{\beta}$ to zero, we have

$$\boldsymbol{\beta} = \mathbf{H}^T (\gamma_g \mathbf{I}_n + \mathbf{H} \mathbf{H}^T)^{-1} \mathbf{F}. \quad (17)$$

By substituting $\boldsymbol{\beta}$ in (10) by (17), the optimization problem (10) becomes

$$\begin{aligned} \min_{\mathbf{F}^T \mathbf{F} = \mathbf{I}_c} \text{tr}(\mathbf{F}^T \tilde{\mathbf{L}} \mathbf{F}) + \mu \left(\text{tr} \left((\mathbf{F} - \mathbf{H} \mathbf{H}^T (\gamma_g \mathbf{I}_n + \mathbf{H} \mathbf{H}^T)^{-1} \mathbf{F})^T \right. \right. \\ \left. \left. \cdot (\mathbf{F} - \mathbf{H} \mathbf{H}^T (\gamma_g \mathbf{I}_n + \mathbf{H} \mathbf{H}^T)^{-1} \mathbf{F}) \right) \right) \\ + \gamma_g \text{tr} \left(\mathbf{F}^T (\gamma_g \mathbf{I}_n + \mathbf{H} \mathbf{H}^T)^{-1} \mathbf{H} \mathbf{H}^T (\gamma_g \mathbf{I}_n + \mathbf{H} \mathbf{H}^T)^{-1} \mathbf{F} \right). \end{aligned} \quad (18)$$

Problem (18) can be further transformed into the following objective function:

$$\begin{aligned} \min_{\mathbf{F}^T \mathbf{F} = \mathbf{I}_c} \text{tr}(\mathbf{F}^T \tilde{\mathbf{L}} \mathbf{F}) \\ + \mu \text{tr} \left(\mathbf{F}^T (\mathbf{I}_n - \mathbf{H} \mathbf{H}^T (\gamma_g \mathbf{I}_n + \mathbf{H} \mathbf{H}^T)^{-1}) \mathbf{F} \right) \end{aligned} \quad (19)$$

which can be denoted as follows:

$$\min_{\mathbf{F}^T \mathbf{F} = \mathbf{I}_c} \text{tr}(\mathbf{F}^T (\tilde{\mathbf{L}} + \mu \mathbf{L}_H) \mathbf{F}), \quad (20)$$

where $\mathbf{L}_H = \mathbf{I}_n - \mathbf{H} \mathbf{H}^T (\gamma_g \mathbf{I}_n + \mathbf{H} \mathbf{H}^T)^{-1}$. $\mathbf{L}_H$ can be transformed into another form as follows:

$$\begin{aligned} \mathbf{L}_H &= \mathbf{I}_n - \mathbf{H} (\gamma_g \mathbf{I}_L + \mathbf{H}^T \mathbf{H})^{-1} \cdot (\gamma_g \mathbf{I}_L + \mathbf{H}^T \mathbf{H}) \\ &\cdot \mathbf{H}^T (\gamma_g \mathbf{I}_n + \mathbf{H} \mathbf{H}^T)^{-1} \\ &= \mathbf{I}_n - \mathbf{H} (\gamma_g \mathbf{I}_L + \mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T. \end{aligned} \quad (21)$$

This completes the proof of Theorem 2. $\square$

ESEC makes good use of an embedding function in ELM feature space instead of RKHS. Thus, the form of ESEC is similar to that of KSEC. It can be proved that there is a link between ESEC and KSEC. We have the following theorem.

Theorem 3. *If the mapping $\mathbf{h}(\mathbf{x})$ in ELM is $(1/\sqrt{L})(k_{b_1}(\mathbf{x}, \mathbf{a}_1), k_{b_2}(\mathbf{x}, \mathbf{a}_2), \dots, k_{b_L}(\mathbf{x}, \mathbf{a}_L))$, where k denotes any kernel function and L is the number of hidden nodes in ELM and $\{(\mathbf{a}_i, b_i)\}_{i=1}^L$ (b_i is the parameter of kernel function $k(\mathbf{x}_1, \mathbf{x}_2)$) are random sampling points from any continuous probability distribution, then ESEC is an approximation of KSEC by discretizing the embedding function $f(\mathbf{x}) = \sum_{i=1}^n \alpha_i k(\mathbf{x}_i, \mathbf{x})$.*

Proof. Since $k(\mathbf{x}, \mathbf{x}_i) = k(\mathbf{x}_i, \cdot) \cdot k(\mathbf{x}, \cdot)$ in RKHS, $f(\mathbf{x})$ can be denoted as

$$f(\mathbf{x}) = [k(\mathbf{x}_1, \cdot) \cdot k(\mathbf{x}, \cdot), \dots, k(\mathbf{x}_n, \cdot) \cdot k(\mathbf{x}, \cdot)] \boldsymbol{\alpha}, \quad (22)$$

where $\boldsymbol{\alpha} = (\gamma_g \mathbf{I}_n + \mathbf{K})^{-1} \mathbf{F}$. Let $h(\mathbf{x}) = (1/\sqrt{L})(k_{b_1}(\mathbf{x}, \mathbf{a}_1), k_{b_2}(\mathbf{x}, \mathbf{a}_2), \dots, k_{b_L}(\mathbf{x}, \mathbf{a}_L))$ and $\mathbf{H} = [\mathbf{h}(\mathbf{x}_1), \mathbf{h}(\mathbf{x}_2), \dots, \mathbf{h}(\mathbf{x}_n)]$; then

$$f(\mathbf{x}) \approx h(\mathbf{x}) \mathbf{H}^T (\gamma_g \mathbf{I}_n + \mathbf{H} \mathbf{H}^T)^{-1} \mathbf{F} = h(\mathbf{x}) \boldsymbol{\beta}. \quad (23)$$

Thus, we approximately derive the embedding function of ESEC from KSEC. This completes the proof of Theorem 3. $\square$

The proposed ESEC algorithm is described as follows.

Algorithm 1.

Input. The input is the training dataset $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n] \in \mathbb{R}^{d \times n}$ and the number of clusters c .

Output. The output is the class assignment matrix of cluster $\mathbf{Y}$.

Step 1. Construct the graph Laplacian $\mathbf{L}$ from $\mathbf{X}$.

Step 2. Randomly generate input weights $\{(\mathbf{a}_i, b_i)\}_{i=1}^L$ and initiate an ELM network of L hidden neurons; calculate the output matrix of the hidden layer.

Step 3. If $L \leq n$; let $\mathbf{L}_H = \mathbf{I}_n - \mathbf{H}(\gamma_g \mathbf{I}_L + \mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T$.

Else $\mathbf{L}_H = \mathbf{I}_n - \mathbf{H} \mathbf{H}^T (\gamma_g \mathbf{I}_n + \mathbf{H} \mathbf{H}^T)^{-1}$.

Step 4. Compute the matrix $\tilde{\mathbf{L}} + \mu \mathbf{L}_H$.

Step 5. Find the eigenvectors of $\tilde{\mathbf{L}} + \mu \mathbf{L}_H$ corresponding to the c smallest eigenvalues, which form the optimal $\mathbf{F}^*$.

Step 6. Treat each row of $\mathbf{F}^*$ as a new training sample, and use the K -means algorithm to cluster the n training samples into c clusters. Let $\mathbf{F}_D^*$ be the final discrete class assignment matrix of cluster for training data.

Return the class assignment matrix of cluster $\mathbf{F}_D^*$.

3.3. Computational Complexity. From Algorithm 1, we can see that the most costly computation is computing the matrix $\mathbf{L}_H$ and carrying out the eigen-decomposition of $\tilde{\mathbf{L}} + \mu \mathbf{L}_H$. If $L \ll n$, computing $\mathbf{L}_H$ needs to obtain the inversion of $\gamma_g \mathbf{I}_L + \mathbf{H}^T \mathbf{H}$, whose computational complexity is $O(L^3 + Ln^2)$. In addition, the computational complexity of eigenvalue decomposition of $\tilde{\mathbf{L}} + \mu \mathbf{L}_H$ is $O(n^3)$. Thus, the total computational complexity of ESEC is $O(n^3 + L^3 + Ln^2)$, where $L \ll n$. Correspondingly, for KSEC, computational complexity of calculating $\mathbf{L}_F$ is $O(n^3)$ and its total computational complexity is $O(2n^3)$. Consequently, ESEC has lower computational complexity than KSEC.

3.4. Clustering for Out-of-Sample Data. By performing Algorithm 1, we can obtain the cluster assignment matrix $\mathbf{F}_D^*$ for the training data. Thus, β can be easily computed by using formula (17). Then, for any new data point $\mathbf{x} \in \mathbb{R}^d$, we can obtain the prediction result

$$\mathbf{y} = h(\mathbf{x}) \beta = h(\mathbf{x}) \mathbf{H}^T (\gamma_g \mathbf{I}_n + \mathbf{H} \mathbf{H}^T)^{-1} \mathbf{F}_D^*. \quad (24)$$

In this paper, we use the spectral rotation method to calculate the discrete cluster assignment vector for $\mathbf{x}$. Firstly, an

TABLE 1: Dataset description.

Dataset	Size	Dimensions	Classes
Iris	150	4	3
Glass	214	10	6
Wine	178	13	3
WPBC	198	33	2
SpectfHeart	267	44	2
USPS	900	256	3
Yale	165	1024	15
COIL20	1440	16384	20
Isolet	390	1234	13
ORL	100	4096	10

orthogonal matrix $\mathbf{R}$ is computed by the following spectral rotation method:

$$\begin{aligned} \min_{\mathbf{Y} \in \mathbb{B}^{n \times c}, \mathbf{R} \in \mathbb{R}^{c \times c}} \quad & \|\mathbf{Y} - \mathbf{Y}^* \mathbf{R}\|^2 \\ \text{s.t.} \quad & \mathbf{Y} \mathbf{1}_c = \mathbf{1}_n, \\ & \mathbf{R}^T \mathbf{R} = \mathbf{I}_c, \end{aligned} \quad (25)$$

where $\mathbf{1}_c$ and $\mathbf{1}_n$ denote the $c \times 1$ and $n \times 1$ vectors of all 1s, respectively. $\mathbf{R} \in \mathbb{R}^{c \times c}$ is an orthogonal matrix and $\mathbf{Y}^*$ is defined by

$$\mathbf{Y}^* = \text{Diag}(\mathbf{F}^* \mathbf{F}^{*T})^{-1/2} \mathbf{F}^*, \quad (26)$$

where $\text{Diag}(\mathbf{A})$ represents a diagonal matrix with the same diagonal elements as the square matrix $\mathbf{A}$. Secondly, the discrete cluster assignment vector for $\mathbf{x}$ is calculated as follows:

$$\tilde{\mathbf{y}} = \frac{\mathbf{R}^T \mathbf{y}}{\sqrt{\|\mathbf{y}\|^2}}. \quad (27)$$

Finally, the class of the data point $\mathbf{x}$ is

$$\text{label} = \arg \max_i \tilde{\mathbf{y}}(i), \quad (28)$$

where $\tilde{\mathbf{y}}(i)$ is the i th element in the vector $\tilde{\mathbf{y}}$.

4. Experiments

To evaluate the in-sample clustering and out-sample clustering performance of different clustering methods, we test all algorithms on UCI datasets (Iris, Glass, Wine, WPBC, SpectHeart, and Isolet (<http://archive.ics.uci.edu/ml/datasets.html>)), face recognition datasets (Yale (<http://vision.ucsd.edu/~leekc/ExtYaleDatabase/Yale%20Face%20Database.htm>)), ORL (<http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>)), digits recognition datasets (USPS (<http://www-i6.informatik.rwth-aachen.de/~keyser/usps.html>)), and object recognition datasets (COIL-20 (<http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php>)). Some datasets are resized, and the basic information of datasets is listed in Table 1. All the experiments have been performed in

TABLE 2: Performance comparison of clustering accuracy using KM, SC, SEC, KSEC, and ESEC for the in-sample clustering on ten datasets.

Dataset	KM	SC	SEC	KSEC	ESEC
Iris	82.1 $\pm$ 1.9	83.3 $\pm$ 1.9	68.9 $\pm$ 1.5	83.3 $\pm$ 1.8	90.4 $\pm$ 2.3
Glass	54.9 $\pm$ 0.9	58.9 $\pm$ 1.3	59.3 $\pm$ 1.6	59.8 $\pm$ 2.1	72.2 $\pm$ 1.6
Wine	57.1 $\pm$ 1.8	57.9 $\pm$ 1.9	44.9 $\pm$ 1.8	60.1 $\pm$ 2.4	61.8 $\pm$ 1.8
WPBC	63.6 $\pm$ 1.9	74.2 $\pm$ 1.4	77.2 $\pm$ 1.4	77.2 $\pm$ 1.3	74.1 $\pm$ 1.2
SpectfHeart	61.6 $\pm$ 3.1	78.7 $\pm$ 3.7	55.4 $\pm$ 1.3	79.8 $\pm$ 3.9	76.7 $\pm$ 2.5
USPS	63.7 $\pm$ 1.2	61.3 $\pm$ 2.4	65.1 $\pm$ 1.7	61.9 $\pm$ 2.5	65.9 $\pm$ 2.6
Yale	37.8 $\pm$ 1.5	35.9 $\pm$ 1.7	30.9 $\pm$ 3.4	37.3 $\pm$ 1.6	40.8 $\pm$ 1.9
COIL20	67.0 $\pm$ 3.4	76.2 $\pm$ 3.3	67.6 $\pm$ 1.4	78.8 $\pm$ 2.8	77.0 $\pm$ 2.4
Isolet	66.6 $\pm$ 0.9	65.4 $\pm$ 2.6	68.7 $\pm$ 1.8	76.4 $\pm$ 2.2	70.6 $\pm$ 2.6
ORL	59.3 $\pm$ 1.5	58.2 $\pm$ 2.9	50.9 $\pm$ 3.8	61.4 $\pm$ 1.9	60.7 $\pm$ 2.7

MATLAB R2013a running in a 3.10 GHZ Intel Core™i5-2400 with 4 GB RAM.

In in-sample clustering, we assign a cluster label to each unlabeled in-sample data point. The proposed ESEC algorithm is compared with K -means (KM) clustering, SC [3], SEC [12], and KSEC. For KM, an EM-like algorithm is used to assign cluster labels as in [16]. In out-sample clustering, the cluster label of each unseen data point is assigned to the closest cluster center learned from the in-sample data points for KM. We use the proposed out-of-sample approach to cope with unseen data for ESEC. Similar out-of-sample method is also used for KSEC and SEC by using different embedding functions. Since the Nyström method [10] can be used to deal with unseen data, we also compare ESEC with the Nyström method for the out-of-sample SC.

4.1. Experimental Setup. Each dataset is randomly divided into seen and unseen samples, and we use the seen data to obtain the optimal parameters of different clustering methods by cross-validation. Then, we use the unseen data to test the performance of all algorithms using the obtained optimal parameters. In the experiments, 80% of the data are randomly selected as seen data and the remaining data are used as unseen data.

The self-tuning SC method [17] is used to determine the parameter σ in (1) for SC, SEC, KSEC, and ESEC. The parameter K of K -nearest-neighbor graph is set to 5 empirically. For fair comparisons, we set the parameters γ_g in SEC, KSEC, and ESEC as 1 and select the parameter μ in these methods from $\{10^{-9}, 10^{-6}, 10^{-3}, \dots, 10^0, \dots, 10^3, 10^6, 10^9, 10^{12}, 10^{15}\}$.

For the Nyström SC method, we set the same $\sigma = 2\sigma_0$, where $\sigma_0 = 1/A$, with A being the mean value of the square distance between the in-sample data as suggested in [18] and $r = \{-3, -2, -1, 0, 1, 2, 3\}$. For the in-sample clustering, the best clustering results from the best parameters for SEC, KSEC, and ESEC are reported in Table 2. For ESEC, we use the RBF kernel as the hidden node function and a grid search of the number of hidden nodes L on $\{10, 20, \dots, 900, 1000\}$ is conducted to seek for the optimal result by using fivefold cross-validation. By means of the optimal parameters in in-sample setting, the results for the out-of-sample clustering are obtained and reported in Table 3.

It should be noted that the results of all clustering methods rely on the initialization. To get statistical results

TABLE 3: Time cost comparison of KM, SC, SEC, KSEC, and ESEC for the in-sample clustering on ten datasets.

Dataset	KM	SC	SEC	KSEC	ESEC
Iris	0.285	0.267	0.296	0.336	0.294
Glass	0.319	0.333	0.304	0.492	0.455
Wine	0.434	0.342	0.260	0.313	0.300
WPBC	0.471	0.475	0.394	0.599	0.319
SpectfHeart	0.662	0.416	0.408	0.744	0.357
USPS	0.316	9.840	10.228	26.031	10.602
Yale	0.602	0.723	1.206	0.982	0.571
COIL20	0.839	74.387	64.485	106.438	83.458
Isolet	0.799	4.479	5.407	5.190	0.893
ORL	0.438	0.919	1.901	1.664	1.149

for different parameters and random partitions, all clustering algorithms are independently repeated 50 times, and we report the mean clustering result and standard deviation using the best parameters on the seen and unseen data. In the experiments, we set the number of clusters as the number of classes c in each dataset. The clustering accuracy (ACC) (refer to [19] for its definition) and time cost are used to evaluate the clustering performance.

4.2. In-Sample Clustering Experiments. To compare clustering performances of various clustering algorithms, we report the in-sample clustering results on all the datasets in Table 2. As can be seen from Table 2, SC outperforms KM for most of the low dimensionality datasets, such as Iris, Glass, Wine, WPBC, and SpectfHeart. But, it might become worse on the high dimensionality datasets, such as Yale and Isolet. This is due to the fact that SC prefers the datasets that have a clear manifold structure in a low-dimensional space. If this assumption does not hold, it even performs worse than the KM algorithm. The performance of SEC is better than KM and SC on Glass, WPBC, USPS, and Isolet. Hence, it does not achieve overwhelming advantages for in-sample clustering on all datasets. One possible explanation is that SEC improves SC by introducing the linear embedding functions, which is only applicable to the data with linear or approximately linear structures. KSEC and ESEC significantly outperform KM, SC, and SEC in most cases. KSEC and ESEC all achieve 5 best in-sample clustering results among all datasets. It should

TABLE 4: Performance comparison of different algorithms for the out-of-sample clustering on ten datasets.

Dataset	Nyström SC	KM	SEC	KSEC	ESEC
Iris	73.6 ± 1.8	75.6 ± 1.9	63.0 ± 1.9	77.1 ± 2.7	79.8 ± 2.1
Glass	49.7 ± 0.9	57.4 ± 1.2	58.6 ± 1.6	58.9 ± 2.4	63.1 ± 2.5
Wine	45.2 ± 0.9	54.6 ± 2.1	49.7 ± 1.9	61.2 ± 2.3	63.7 ± 1.9
WPBC	52.2 ± 2.8	64.7 ± 1.5	73.5 ± 1.5	74.3 ± 1.6	76.2 ± 1.4
SpectfHeart	51.4 ± 1.9	66.8 ± 2.9	53.5 ± 1.7	72.2 ± 2.3	70.8 ± 2.6
USPS	48.3 ± 4.5	56.0 ± 2.2	61.5 ± 2.4	65.1 ± 1.6	62.3 ± 2.8
Yale	20.7 ± 2.3	23.0 ± 1.6	35.9 ± 1.7	42.5 ± 1.6	45.8 ± 1.9
COIL20	63.9 ± 2.4	61.6 ± 3.3	68.9 ± 2.5	72.4 ± 2.7	70.2 ± 3.1
Isolet	34.8 ± 2.7	39.2 ± 1.2	45.3 ± 2.9	66.0 ± 1.8	68.4 ± 1.5
ORL	43.5 ± 2.1	47.6 ± 1.8	55.2 ± 1.5	60.7 ± 1.9	58.7 ± 1.7

be noted that KSEC and ESEC also have superior clustering performance for low dimensionality datasets, since they all introduce regularization terms into SC and can be considered as regularized SC. Compared with KSEC, ESEC achieves better or at least comparable results, which demonstrates that the proposed ESEC method is effective on all the datasets and has the ability to handle the datasets that do not have a clear manifold structure in a low-dimensional space. The running time of all algorithms is listed in Table 3. It is shown that ESEC runs much faster than KSEC, which is consistent with the theoretical analysis, and the running time of ESEC and KSEC is lower than that of KM, SC, and SEC for most of the datasets. Overall, compared with other methods, the proposed ESEC method has better or comparative in-sample performance at much faster training speed.

In Figure 1, we further analyze the sensitivity of the in-sample clustering performances of SEC, KSEC, and ESEC with respect to the parameter μ . We can see from Figure 1 that ESEC prefers a large value for μ on Yale and ORL, and its performance on these datasets is relatively stable when μ is set as a large value. While ESEC and KSEC favor a small value of μ for COIL20 and Isolet. We can observe that ESEC outperforms KSEC and SEC in a wide range of μ ; that is, the clustering accuracy of ESEC is less sensitive to the parameter for most of the datasets when compared with SEC and KSEC.

4.3. Out-of-Sample Clustering Experiments. We also study the performances of KM, Nyström SC, SEC, KSEC, and ESEC for the out-of-sample extension. Table 4 shows the clustering accuracies of these methods for the out-of-sample clustering on all the datasets. The optimal parameters of SEC, KSEC, and ESEC are determined by cross-validation from the in-sample clustering. From Table 4, it can be seen that SEC, KSEC, and ESEC significantly outperform the Nyström method for out-of-sample clustering. The reason is that the Nyström method utilizes Nyström extension to evaluate the similarity matrix between the unseen data, which might be inaccurate or even has a serious deviation. However, our proposed framework aims at minimizing the error between the cluster assignment matrix and the low-dimensional embedding of the data, which is feasible for handling real-world data. Thus, ESEC has the natural ability of solving out-of-sample extension problems. In addition, KM is sharply degraded on Yale and

ORL compared with the corresponding results in Table 2. This is due to the fact that the unseen face data has the large variation compared to the seen data. On the other hand, the clustering accuracies of ESEC are comparable to the in-sample testing results, which validates that ESEC has better generalization performance. ESEC achieves 6 best clustering results among all ten testing results and has comparable results on the rest of the datasets when compared to KSEC. Consequently, the proposed ESEC algorithm provides a new way to cope with the out-of-sample data in clustering tasks.

5. Conclusion

In this paper, we propose a general spectral embedded clustering framework based on the objective function of SC, from which SEC, KSEC, and ESEC can all be derived by using different embedding functions. By virtue of ELM, the fast spectral nonlinearly embedded clustering algorithm (ESEC) is proposed, which can naturally solve the out-of-sample extension problem for the clustering tasks. Experimental results on benchmark datasets validate the effectiveness and efficiency of the proposed ESEC method for both in-sample and out-of-sample clustering. In the future, we intend to develop a new semisupervised clustering framework by incorporating pair constraints into the present framework and propose some semisupervised clustering algorithms based on spectral nonlinearly embedded clustering models.

Notations

- $\mathbb{R}^d$: The input d -dimensional Euclidean space
- $\mathbb{B}$: The output 0-1 binary space
- n : The number of total training data points
- c : The number of classes that the samples belong to
- $\mathbf{X}$: $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n] \in \mathbb{R}^{d \times n}$ is the training data matrix
- $\mathbf{Y}$: $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_n]^T \in \mathbb{B}^{n \times c}$ is the 0-1 class assignment matrix; $\mathbf{y}_i \in \mathbb{B}^{c \times 1}$ is the label vector of $\mathbf{x}_i$, and all components of $\mathbf{y}_i$ are 0s except one being 1
- $\phi(\cdot)$: $\phi(\mathbf{X}) = (\psi_1(\mathbf{X}), \dots, \psi_c(\mathbf{X})) \in \mathbb{R}^{n \times c}$ is the embedding vector function

FIGURE 1: Clustering accuracy of SEC, KSEC, and ESEC with different μ for the in-sample clustering on the six datasets. (a) Iris. (b) USPS. (c) Yale. (d) COIL20. (e) Isolet. (f) ORL.

$k(\mathbf{x}, \mathbf{y})$: Kernel function of variables $\mathbf{x}$ and $\mathbf{y}$
 $\mathbf{K}$: Kernel matrix $\mathbf{K} = \{k(\mathbf{x}_i, \mathbf{x}_j)\} \in \mathbb{R}^{n \times n}$
 α : $\alpha = [\alpha_1, \dots, \alpha_n]^T \in \mathbb{R}^{n \times c}$; Its columns are the coefficients of kernel functions to represent the embedding function $\phi(\cdot)$
 $\text{tr}(\mathbf{M})$: The trace of the matrix $\mathbf{M}$, that is, the sum of the diagonal elements of the matrix $\mathbf{M}$.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (no. 61403394) and the Fundamental Research Funds for the Central Universities (no. 2014QNA46).

References

- [1] A. P. Benavent, F. E. Ruiz, and J. M. Sáez, "Learning Gaussian mixture models with entropy-based criteria," *IEEE Transactions on Neural Networks*, vol. 20, no. 11, pp. 1756–1771, 2009.
- [2] K. Zhang and J. T. Kwok, "Simplifying mixture models through function approximation," *IEEE Transactions on Neural Networks*, vol. 21, no. 4, pp. 644–658, 2010.
- [3] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.
- [4] M. Filippone, F. Camastra, F. Masulli, and S. Rovetta, "A survey of kernel and spectral methods for clustering," *Pattern Recognition*, vol. 41, no. 1, pp. 176–190, 2008.
- [5] L. Xu, J. Neufeld, B. Larson, and D. Schuurmans, "Maximum margin clustering," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS '05)*, pp. 1537–1544, Vancouver, Canada, 2005.
- [6] K. Zhang, I. W. Tsang, and J. T. Kwok, "Maximum margin clustering made practical," *IEEE Transactions on Neural Networks*, vol. 20, no. 4, pp. 583–596, 2009.
- [7] Y. Li, I. W. Tsang, J. T. Kwok, and Z. Zhou, "Tighter and convex maximum margin clustering," in *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pp. 344–351, Clearwater Beach, Fla, USA, 2009.
- [8] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, 2003.
- [9] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: a geometric framework for learning from labeled and unlabeled examples," *Journal of Machine Learning Research*, vol. 7, pp. 2399–2434, 2006.
- [10] F. Nie, Z. Zeng, I. W. Tsang, D. Xu, and C. Zhang, "Spectral embedded clustering: a framework for in-sample and out-of-sample spectral clustering," *IEEE Transactions on Neural Networks*, vol. 22, no. 11, pp. 1796–1808, 2011.
- [11] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, "Spectral grouping using the Nyström method," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 214–225, 2004.
- [12] Y. Bengio, J.-F. Paiement, P. Vincent, O. Delalleau, N. L. Roux, and M. Ouimet, "Out-of-sample extensions for LLE, Isomap, MDS, eigenmaps, and spectral clustering," in *Proceedings of the 17th Annual Conference on Neural Information Processing Systems (NIPS '03)*, pp. 126–133, Whistler, Canada, December 2003.
- [13] S. X. Yu and J. Shi, "Multiclass spectral clustering," in *Proceedings of the 9th IEEE International Conference on Computer Vision*, pp. 313–319, Beijing, China, October 2003.
- [14] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: analysis and an algorithm," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS '01)*, pp. 849–856, Vancouver, Canada, 2001.
- [15] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 2, pp. 513–529, 2011.
- [16] J. Ye, Z. Zhao, and M. Wu, "Discriminative K -means for clustering," in *Proceedings of the Neural Information Processing Systems*, pp. 1649–1656, Vancouver, Canada, 2007.
- [17] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *Proceedings of the 18th Annual Conference on Neural Information Processing Systems (NIPS '04)*, pp. 1601–1608, Vancouver, Canada, December 2004.
- [18] L. Duan, D. Xu, I. W. Tsang, and J. Luo, "Visual event recognition in videos by learning from web data," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 1959–1966, San Francisco, Calif, USA, June 2010.
- [19] W. Chen and G. Feng, "Spectral clustering: a semi-supervised approach," *Neurocomputing*, vol. 77, no. 1, pp. 229–242, 2012.

Hindawi

Submit your manuscripts at
<http://www.hindawi.com>