Integrated KL (K-means - Laplacian) Clustering: A New Clustering Approach by Combining Attribute Data and Pairwise Relations

Fei Wang* Chris Ding† Tao Li‡

January 21, 2009

Abstract

Most datasets in real applications come in from multiple sources. As a result, we often have attributes information about data objects and various pairwise relations (similarity) between data objects. Traditional clustering algorithms use either data attributes only or pairwise similarity only. We propose to combine K-means clustering on data attributes and normalized cut spectral clustering on pairwise relations. We show that these two methods can be coherently integrated together to make use of different data sources to obtain good clustering results. We also show that our integrated KL (K-means - Laplacian) clustering method can be naturally extended to semi-supervised clustering, data embedding and metric learning. Finally the experimental results on benchmark data sets are presented to show the effectiveness of our method.

1 Introduction

The problem of clustering data arises in many disciplines and has a wide range of applications. Intuitively, clustering is the problem of partitioning a finite set of points in a multi-dimensional space into classes (called clusters) so that (i) the points belonging to the same class are *similar* and (ii) the points belonging to different classes are *dissimilar*.

Conventional clustering techniques assume that the input data objects are homogeneous and not relational. As a result, a data object is usually represented as a fixed-length vector of attribute values. For example, a document is represented as a vector of term values (e.g., TF-IDF weights) using the vector space model. Objects are grouped into clusters based on their attribute values: two objects are similar if they have similar attributed values. In many traditional applications, conventional clustering techniques are sufficient [1].

Recently, with the advancement of science and technology, especially the popularization of Internet, the majority of data routinely captured by business and organizations are rich in structure and relational in nature. In particular, many data sets include relation information as well as independent object attributes, possibly from different data sources.

Relation information provides graph structure in the data and induces pairwise similarity between objects while attribute values provide inherent characteristic information about the data objects [35]. For example, a webpage can be represented as a vector of term values. In addition, there are hyper-links between webpages. If webpage i links to webpage j, this indicates a similarity relationship between two webpages [16, 23]. In other words, links confer a relationship between two webpages in the same way that similar attribute values indicate a relationship. As another example, in bioinformatics applications, gene expression profiles provides attribute information about the molecular aspects of genes while protein-protein interaction reveals the composition of protein complex and induce pairwise relationships among genes. In such situations, both relation information and attribute values can be used to cluster data objects: conventional clustering algorithms such as K-means can identify group of similar data objects based on their attribute values and graph-based partitioning approaches such as spectral clustering can identify highly connected components from the graph structure or using pairwise similarity.

Although both the relation information and attribute values can be used independently to cluster data objects, clustering algorithms that make use of them simultaneously should be able to generate more meaningful clustering structures. Recently, many clustering algorithms have been developed using both the attribute and relation information [22, 13]. For instance, Neville et al. adapted graph partitioning algorithms to incorporate both relationship structure and attribute information by weighting the existing relational graph with an attribute similarity metric [23]. He et al. [16] combined the similarity matrices of attributes and relations

*School of Computer Science, Florida International University
†Comp. Sci & Eng. Dept, University of Texas ar Arlington
‡School of Computer Science, Florida International University

and used a spectral graph partitioning algorithm for webpage clustering. Bhattacharya and Getoor [5] used the linear combination of graph similarity and attribute similarity for relational clustering for entity resolution in graphs. Taskar [29] proposed to use probabilistic models to cluster relational data with attributes and relationships. Yin et al. [40] proposed a relational table clustering algorithm by producing a single object type that is a compound of features from other objects.

Most of the existing relational clustering algorithms can be categorized into the following two types: (1) **Feature Integration**: This approach enlarges the feature representation to incorporate all data and produces a unified feature space. In particular, the relation information is viewed as additional features/attributes. The advantage of feature integration is that the unified feature representation is often more informative and also allows many different data mining methods to be applied and systematically compared. One disadvantage is the increased learning complexity and difficulty as the data dimension becomes large. (2) **Kernel Integration**: The data is kept in their original form and they are integrated at the similarity computation or the Kernel level [19]. In other words, graph similarity and attribute similarity are combined directly. Different weights can be used for different types of similarity. One drawback of the kernel integration is that it does not fully explore the correlation between the attribute information and the relation information.

In this paper, we propose a new clustering framework by combining K-means clustering on attribute values and spectral clustering on relation information. Our proposed framework can be viewed as a kind of semantic integration, which avoids the limitations of feature integration and it also implicitly learns the correlation structure between attribute information and relation information. The rest of the paper is organized as follows: Section 2 introduces our integrated KL clustering method by combining K-means clustering on data attributes and spectral clustering on pairwise relations; Section 3 extends our KL clustering method to semi-supervised clustering, data embedding and metric learning; Section 4 presents experimental results on benchmark datasets; and finally Section 5 concludes.

2 Clustering by Integrating Attribute and Pairwise Relations

Given a data matrix $\mathbf{X} \in \mathbb{R}^{d\times n}$ (each column corresponds to a data point) together with their pairwise relationship matrix $\mathbf{W} \in \mathbb{R}^{n\times n}$ such that $\mathbf{W}_{ij}$ represents the relationship between $\mathbf{x}_i$ and $\mathbf{x}_j$. The problem is how to obtain good clustering result by incorporating both the attribute information $\mathbf{X}$ and the pairwise relation information $\mathbf{W}$.

K-means [12] and spectral clustering [27] are two types of representative clustering methods. The inputs of the K-means algorithm is the attribute data set $\mathbf{X}$, and the inputs of the Spectral Clustering method is the relationship matrix $\mathbf{W}$. We can construct a clustering objective as

$$J = \alpha J_{K-means} + (1-\alpha) J_{Ncut} \tag{2.1}$$

where $\alpha > 0$ is a tradeoff parameter. Therefore we can seek clustering results to minimize the above criterion.

2.1 An Intuitive Solution

Here we first present a simple and intuitive algorithm to solve the integrated KL clustering problem.

Denote the *scaled cluster membership matrix* $\mathbf{H} \in \mathbb{R}^{n\times C}$ as

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \cdot \begin{bmatrix} n_1 & & & \\ & \ddots & & \\ & & & n_C \end{bmatrix}^{-1/2}$$

such that

$$\mathbf{H}_{ij} = \begin{cases} 1/\sqrt{n_j}, & if\ \mathbf{x}_i \in \pi_j \\ 0, & otherwise \end{cases} \tag{2.2}$$

where π_j represents the j-th cluster and n_j is the size of π_j, and C is the total number of clusters. Clearly $\mathbf{H}^T\mathbf{H} = \mathbf{I}$. Then through some derivations we can obtain that [12][42]

$$J_{K-means} = \|\mathbf{X}\|_F^2 - tr(\mathbf{H}^T\mathbf{X}^T\mathbf{X}\mathbf{H})$$

and

$$J_{Ncut} = tr(\mathbf{H}^T\widehat{\mathbf{L}}\mathbf{H})$$

where $\widehat{\mathbf{L}} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$ is the *Normalized Laplacian matrix* with $\mathbf{D} = diag(\sum_j \mathbf{W}_{1j}, \sum_j \mathbf{W}_{2j}, \cdots, \sum_j \mathbf{W}_{nj})$ being the *degree matrix*. Then

$$J = \alpha\|\mathbf{X}\|_F^2 - \alpha tr(\mathbf{H}^T\mathbf{X}^T\mathbf{X}\mathbf{H}) + (1-\alpha)tr(\mathbf{H}^T\widehat{\mathbf{L}}\mathbf{H})$$

Since $\|\mathbf{X}\|_F^2$ is a constant, then the minimization of the above criterion is equivalent to solve the following optimization problem

$$\begin{aligned} \min_{\mathbf{H}} \quad & tr(\mathbf{H}^T[(1-\alpha)\widehat{\mathbf{L}} - \alpha\mathbf{X}^T\mathbf{X}]\mathbf{H}) \\ s.t. \quad & \mathbf{H}^T\mathbf{H} = \mathbf{I} \end{aligned} \tag{2.3}$$

Note here we relax the constraint on $\mathbf{H}$ as in traditional clustering approaches [42]. Following the Ky Fan theorem [42], we can derive that the solutions to the above problem are the eigenvectors of the matrix $(1-\alpha)\widehat{\mathbf{L}}-\alpha\mathbf{X}^T\mathbf{X}$ corresponding to its smallest C eigenvalues. In practice, we can treat the C eigenvectors as the C-dimensional embedding of the n data points, and then applying the K-means algorithm to cluster them into C clusters [24].

The problem with the above formulation is that there is a hyperparameter α which controls the relevant importance of K-means and spectral clustering. However in practice it is hard to set an optimal α. Although we can apply gradient based methods, they may easily get trapped in a local optimum which will make the results suboptimal.

2.2 A Trace Quotient Formulation We seek a clustering formulation which does not contain the hyperparameter α.

Now let us revisit the formulation in Eq.(2.3), which is actually a trace **difference** formulation composed of two terms:

$$\min_{\mathbf{H}} Tr(\mathbf{H}^T\widehat{\mathbf{L}}\mathbf{H}), \quad \max_{\mathbf{H}} Tr[\mathbf{H}^T(\mathbf{X}^T\mathbf{X})\mathbf{H}], \tag{2.4}$$

- The first term, $tr(\mathbf{H}^T\widehat{\mathbf{L}})\mathbf{H})$ reflects the cost of "cutting" the data graph into different groups, in other words, it reflects the *between-cluster scatterness* of the data set after clustering. The smaller this value, the better the clusters are discriminated with each other.
- The second term, $tr(\mathbf{H}^T(\mathbf{X}^T\mathbf{X})\mathbf{H})$ reflect how compact the clusters are. The larger this value, the compacter the clusters are.

The above analysis shows that our algorithm here has a close relationship with the *Linear Discriminant Analysis* (*LDA*) [12] formulation, which is trace quotient rather than trace difference. Therefore we can use a similar formulation to our problem as

$$\begin{aligned} &\max_{\mathbf{H}} \quad \frac{tr(\mathbf{H}^T\mathbf{X}^T\mathbf{X}\mathbf{H})}{tr(\mathbf{H}^T\widehat{\mathbf{L}}\mathbf{H})} \\ &s.t. \quad \mathbf{H}^T\mathbf{H}=\mathbf{I} \end{aligned} \tag{2.5}$$

One advantage of the above formulation is that there is no hyperparameter to control the relevant importance of the two trace terms. However, the constraint $\mathbf{H}^T\mathbf{H}=\mathbf{I}$ strictly restricts that the data clusters are non-overlap (since the cluster indicator vectors for every two different clusters are orthogonal to each other, which indicates that each data point can only belong to one cluster). Therefore we remove such constraint to allow overlapping clusters, then our problem becomes

$$\max_{\mathbf{H}} \frac{tr(\mathbf{H}^T\mathbf{X}^T\mathbf{X}\mathbf{H})}{tr(\mathbf{H}^T\widehat{\mathbf{L}}\mathbf{H})} \tag{2.6}$$

2.3 A Quotient Trace Formulation Generally the trace quotient formulation Eq.(2.6) is not easy to solve, and people can resort to gradient descent methods, which may result in an iterative process with heavy computational burden. A common tradeoff is to solve the following quotient trace problem instead:

$$\max_{\mathbf{H}} tr\left((\mathbf{H}^T\widehat{\mathbf{L}}\mathbf{H})^{-1}(\mathbf{H}^T\mathbf{X}^T\mathbf{X}\mathbf{H})\right). \tag{2.7}$$

To solve the above optimization problem, we have the following theorem:

THEOREM 2.1. *Let $\mathbf{H}^* \in \mathbb{R}^{n\times K}$ be the optimal solution to problem 2.7, then $\mathbf{H}^*$ is composed by the largest K eigenvectors of the matrix $\widehat{\mathbf{L}}^+\mathbf{X}^T\mathbf{X}$*

Here $\widehat{\mathbf{L}}^+$ denotes the pseudo inverse of $\widehat{\mathbf{L}}$.

Proof. Since for $\forall \mathbf{a} \in \mathbb{R}^{n\times 1}$, we have

$$\mathbf{a}^T\widehat{\mathbf{L}}\mathbf{a} = \sum_{ij}\mathbf{W}_{ij}\left(\frac{\mathbf{a}_i}{\sqrt{\mathbf{D}_{ii}}}-\frac{\mathbf{a}_i}{\sqrt{\mathbf{D}_{ii}}}\right)^2 \geq 0$$

Thus $\widehat{\mathbf{L}}$ is positive semi-definite. Let

$$\widehat{\mathbf{L}} = \mathbf{U}\begin{bmatrix}\mathbf{\Sigma} & \\ & 0\end{bmatrix}\mathbf{U}^T$$

be the eigenvalue decomposition of $\mathbf{L}$ with the diagonal line of $\mathbf{\Sigma}$ being the positive eigenvalues of $\widehat{\mathbf{L}}$, then

$$\mathbf{L} = \mathbf{U}_1\mathbf{\Sigma}\mathbf{U}_1^T$$

where $\mathbf{U}_1$ is composed of the eigenvectors of $\mathbf{L}$ corresponding to its positive eigenvalues. Let

$$\mathbf{F} = \widehat{\mathbf{L}}^{1/2}\mathbf{H} = (\mathbf{U}_1\mathbf{\Sigma}^{1/2}\mathbf{U}_1^T)\mathbf{H}$$

then

$$\mathbf{F}^T\mathbf{F} = \mathbf{H}^T\widehat{\mathbf{L}}\mathbf{H}$$

Note that $\mathbf{F}^T\mathbf{F}$ is also positive semi-definite thus we can similarly decompose it as

$$\mathbf{F}^T\mathbf{F} = \mathbf{V}\begin{bmatrix}\mathbf{\Sigma}_F & \\ & 0\end{bmatrix}\mathbf{V}^T = \mathbf{V}_1\mathbf{\Sigma}_F\mathbf{V}_1^T$$

where $\mathbf{\Sigma}_F$ is a diagonal matrix with the positive eigenvalues on its diagonal line, and $\mathbf{V}_1$ is composed of the corresponding eigenvectors. Then we can define

$$(\mathbf{F}^T\mathbf{F})^{-1/2} = \mathbf{V}_1\mathbf{\Sigma}^{-1/2}\mathbf{V}_1^T$$

So

$$\begin{aligned} J(\mathbf{H}) &= tr\left((\mathbf{H}^T\widehat{\mathbf{L}}\mathbf{H})^{-1}(\mathbf{H}^T\mathbf{X}^T\mathbf{X}\mathbf{H})\right) \\ &= tr\left((\mathbf{F}^T\mathbf{F})^{-1/2}(\mathbf{H}^T\mathbf{X}^T\mathbf{X}\mathbf{H})(\mathbf{F}^T\mathbf{F})^{-1/2}\right) \\ &= tr\left((\mathbf{F}^T\mathbf{F})^{-1/2}(\mathbf{F}^T\widehat{\mathbf{L}}^{-1/2}\mathbf{X}^T\mathbf{X}\widehat{\mathbf{L}}^{-1/2}\mathbf{F})(\mathbf{F}^T\mathbf{F})^{-1/2}\right) \\ &= tr\left(\mathbf{Q}^T\widehat{\mathbf{L}}^{-1/2}\mathbf{X}^T\mathbf{X}\widehat{\mathbf{L}}^{-1/2}\mathbf{Q}\right) \end{aligned}$$

where $\mathbf{Q} = \mathbf{F}(\mathbf{F}^T\mathbf{F})^{-1/2}$ subject to

$$\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$$

Following the Ky-Fan theorem we know that the solution to the above problem is just

$$\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_K]$$

where $\mathbf{q}_i$ is the eigenvector of $\widehat{\mathbf{L}}^{-1/2}\mathbf{X}^T\mathbf{X}\widehat{\mathbf{L}}^{-1/2}$ corresponding to its i-th largest eigenvalues. That is,

$$\widehat{\mathbf{L}}^{-1/2}\mathbf{X}^T\mathbf{X}\widehat{\mathbf{L}}^{-1/2}\mathbf{Q} = \mathbf{Q}\mathbf{\Lambda}$$

where $\mathbf{\Lambda}$ is the eigenvalue matrix of $\widehat{\mathbf{L}}^{-1/2}\mathbf{X}^T\mathbf{X}\widehat{\mathbf{L}}^{-1/2}$, and

$$J(\mathbf{H}) = \sum_j \mathbf{\Lambda}_{jj}$$

Therefore to maximize $J(\mathbf{H})$, we should select $\mathbf{Q}$ constituted by the eigenvectors of $\widehat{\mathbf{L}}^{-1/2}\mathbf{X}^T\mathbf{X}\widehat{\mathbf{L}}^{-1/2}$ corresponding to its largest K eigenvalues. Furthermore, let $\mathbf{R} = \widehat{\mathbf{L}}^{-1/2}\mathbf{Q}$, then

- $\mathbf{R}^T\mathbf{X}^T\mathbf{X}\mathbf{R} = \mathbf{\Lambda}$
- $\mathbf{R}^T\mathbf{L}\mathbf{R} = \mathbf{I}$

So $\mathbf{R}$ can simultaneously diagonalize $\mathbf{X}^T\mathbf{X}$ and $\mathbf{L}$, which makes

$$J(\mathbf{H}) = J(\mathbf{R})$$

and

$$\begin{aligned} &\widehat{\mathbf{L}}^{-1/2}\mathbf{X}^T\mathbf{X}\widehat{\mathbf{L}}^{-1/2}\mathbf{Q} = \widehat{\mathbf{L}}^{-1/2}\mathbf{X}^T\mathbf{X}\mathbf{R} = \widehat{\mathbf{L}}^{1/2}\mathbf{R}\mathbf{\Lambda} \\ &\Longrightarrow \mathbf{X}^T\mathbf{X}\mathbf{R} = \widehat{\mathbf{L}}\mathbf{R}\mathbf{\Lambda} \end{aligned}$$

Therefore $\mathbf{R}$ are constituted by the eigenvectors of $\mathbf{L}^+\mathbf{X}^T\mathbf{X}$, where

$$\widehat{\mathbf{L}}^+ = \mathbf{U}_1\mathbf{\Sigma}^{-1}\mathbf{U}_1^T$$

The maximization of $J(\mathbf{H})$ is equivalent to the maximization of $J(\mathbf{R})$, and the optimal

$$\mathbf{H}^* = \mathbf{R}^* = [\mathbf{r}_1, \mathbf{r}_2, \cdots, \mathbf{r}_K]$$

where $\mathbf{r}_i$ is the eigenvector of $\widehat{\mathbf{L}}^+\mathbf{X}^T\mathbf{X}$ corresponding to its i-th largest eigenvalue. □

3 Discussions and Extensions

In the previous section we have introduced a novel clustering method that can make use of both attribute and relation information, which might be obtained from different sources. In this section we provide several extensions of the proposed algorithm and discuss their relationships with traditional approaches.

3.1 Semi-supervised Clustering Semi-supervised clustering refers to a class of clustering methods making use of some prior knowledge. Typically, the knowledge that indicates the two points belong to the same class is referred to as *must-link constraints* $\mathcal{M}$, and the knowledge that indicates the two points belong to different classes is referred to as *cannot-link constraints* $\mathcal{C}$. This type of information can be incorporated into traditional partitional clustering algorithms by adapting the objective function to include penalties for violated constraints. For instance, the *Pairwise Constrained KMeans* (*PCKM*) algorithm [2] modifies the standard sum of squared errors function in traditional *kmeans* to take into account both object-centroid distortions in a clustering $\pi = \{\pi_1, \pi_2, \cdots, \pi_C\}$ and any associated constraint violations, *i.e.*

$$J_{s-km} = J_{K-means} + \sum_{\substack{\mathbf{x}_i,\mathbf{x}_j\in\mathcal{M} \\ s.t.\ l_i\neq l_j}} \theta_{ij} + \sum_{\substack{\mathbf{x}_i,\mathbf{x}_j\in\mathcal{C} \\ s.t.\ l_i=l_j}} \tilde{\theta}_{ij},$$

where $\{\theta_{ij} \geqslant 0\}$ represent the penalties for violating the must-link constraints, and $\{\tilde{\theta}_{ij} \geqslant 0\}$ denote the penalties for violating the cannot-link constraints. Following [18], we can change the penalties of violations in the constraints in $\mathcal{M}$ into the *awards* as

$$\begin{aligned} J_{s-km} &= J_{K-means} - \sum_{\substack{\mathbf{x}_i,\mathbf{x}_j\in\mathcal{M} \\ s.t.\ l_i=l_j}} \theta_{ij} + \sum_{\substack{\mathbf{x}_i,\mathbf{x}_j\in\mathcal{C} \\ s.t.\ l_i=l_j}} \tilde{\theta}_{ij} \\ &= J_{K-means} + \sum_c \sum_{i,j} \mathbf{H}_{ic}\mathbf{H}_{jc}\mathbf{\Theta}_{ij} \end{aligned}$$

where

$$\mathbf{\Theta}_{ij} = \begin{cases} \sqrt{n_i n_j}\tilde{\theta}_{ij}, & (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{C} \\ -\sqrt{n_i n_j}\theta_{ij}, & (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M} \\ 0, & otherwise \end{cases} \tag{3.8}$$

where n_i is the size of π_i, n_j is the size of π_j. Therefore

$$J_{s-km} = \|\mathbf{X}\|_F^2 - tr(\mathbf{H}^T\mathbf{X}^T\mathbf{X}\mathbf{H}) + tr(\mathbf{H}^T\mathbf{\Theta}\mathbf{H})$$

Hence we can solve the semi-supervised clustering problem by solving

$$\min_{\mathbf{H}} tr\left((\mathbf{H}^T\widehat{\mathbf{L}}\mathbf{H})^{-1}(\mathbf{H}^T(\mathbf{X}^T\mathbf{X} - \mathbf{\Theta})\mathbf{H}\right) \tag{3.9}$$

From theorem 2.1 we know that the optimal $\mathbf{H}^*$ can be solved by the eigenvectors of $\widehat{\mathbf{H}}^+(\mathbf{X}^T\mathbf{X} - \mathbf{\Theta})$ corresponding to its largest K eigenvalues.

However, this formulation is, although straightforward, not very natural since the prior knowledge on $\mathcal{M}$ and $\mathcal{C}$ are also relational information of the data set. Therefore it would be more reasonable to incorporate such information into the spectral clustering part. Thus we can construct a *semi-supervised spectral clustering* objective as

$$J_{spectral} = tr\left(\mathbf{H}^T(\widehat{\mathbf{L}} + \mathbf{\Theta})\mathbf{H}\right)$$

Then we can solve the semi-supervised clustering problem by solving

$$\min_{\mathbf{H}} tr\left((\mathbf{H}^T(\widehat{\mathbf{L}} + \mathbf{\Theta})\mathbf{H})^{-1}(\mathbf{H}^T\mathbf{X}^T\mathbf{X}\mathbf{H})\right) \tag{3.10}$$

Then the optimal $\mathbf{H}^*$ can be solve by the eigenvectors of $(\widehat{\mathbf{L}} + \mathbf{\Theta})^+(\mathbf{X}^T\mathbf{X})$ corresponding to its largest K eigenvalues.

3.2 Embedding by Integrating Multiple Information Sources Another problem that is closely related to clustering is *embedding*, which seeks to project the data set into low-dimensional spaces such that the data can be better visualized/discriminated. Most of the state-of-the-art embedding methods use either attribute data information (such as *Principal Component Analysis* (*PCA*) [17], *Locally Linear Embedding* (*LLE*) [25] and *Isomap*) [30] or pairwise distance information (such as *Multidimensional Scaling*) [9]. It is known [42, 11]) that the spectral relaxation of K-means clustering gives identically the PCA. However, there are not many developments on how to embedding the data points by integrating different information sources. In the following, we introduce an embedding method that is closely related to our KL clustering method.

Two most popular principles used in data embedding are:

- Maximizing the data variances in the embedded space as in *PCA*, this may retain most information contained in the data set [43][37][32].
- Maximizing the smoothness of the data set with respect to their intrinsic manifold in their embedded space. Usually this can be implemented by preserving the localities (*i.e.* pairwise relationships) contained in the data set.

Assuming the data set has been centralized, then we can compute the data covariance matrix as $\mathbf{C} = \mathbf{X}^T\mathbf{X}$. Considering linear embeddings, the variance maximization principle aims to find the projection directions $\mathbf{P} \in \mathbb{R}^{d\times K}$ by maximizing

$$\max J_{PCA} = tr(\mathbf{P}^T\mathbf{X}\mathbf{X}^T\mathbf{P})$$

For the second criterion, if we know some pairwise relationships $\mathbf{W}$ and we want the embedding to preserve such relationships, then we can use the following graph-Laplacian based criterion [4]

$$\min J_{Lap} = tr(\mathbf{P}^T\mathbf{X}\widehat{\mathbf{L}}\mathbf{X}^T\mathbf{P})$$

where $\widehat{\mathbf{L}}$ is just the normalized Laplacian matrix as we introduced in the previous section. The smaller J_{Lap} is, the better the data locality is preserved, and the smoother the embeddings would be with respect to the intrinsic data manifold. Therefore a natural choice would be to maximize the following criterion to get the optimal $\mathbf{P}$

$$\max_{\mathbf{P}} tr\left(\left(\mathbf{P}^T\mathbf{X}\widehat{\mathbf{L}}\mathbf{X}^T\mathbf{P}\right)^{-1}\mathbf{P}^T\mathbf{X}\mathbf{X}^T\mathbf{P}\right) \tag{3.11}$$

Using theorem 2.1 we know that the optimal $\mathbf{P}$ can be obtained by the eigenvectors of $(\mathbf{X}^T\widehat{\mathbf{L}}\mathbf{X})^+\mathbf{X}^T\mathbf{X}$ corresponding to its largest K eigenvalues.

Comparing Eq.(3.11) with Eq.(2.7), we observe that the two expressions are very similar: their nominators are exactly the same, and the denominators only differ in the multiplication of $\mathbf{X}^T$ and $\mathbf{X}$ in both sides of $\widehat{\mathbf{L}}$. In fact, the mathematical formulation of spectral clustering [27] and Laplacian eigenmaps [4] are also similar. However, their solutions have different physical meanings: the solution to spectral clustering is the cluster membership matrix, while the solution to Laplacian embedding is the low dimensional data embeddings. Therefore, the result of spectral clustering can also be viewed as the data embeddings in K-dimensional space and this implicitly explains why we can apply the K-means algorithm to discover the data clusters in the embedded space [24]. Moreover, if we *linearize* Laplacian embedding, we can obtain exactly the same formulations as the denominator in Eq.(3.11).

3.3 Distance Metric Learning Distance metric learning is also an important problem in data mining and machine learning. As we know that for vectorized data, Euclidean distance is the most commonly used distance measure for comparing the difference between pairwise data points. However, the Euclidean distance has a homogeneous assumption on all the dimensions. Therefore in the last decades people began to seek for a proper Mahalanobis distance to compare pairwise points. More concretely, the Mahalanobis distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ is defined as [26][36][38]

$$d_{\mathcal{M}}(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T\mathbf{A}(\mathbf{x}_i - \mathbf{x}_j)} \tag{3.12}$$

Table 1: Descriptions of the document datasets

Datasets	# doc	# classes
CSTR	476	4
WebKB4	4199	4
Reuters	2900	10
WebACE	2340	20
Newsgroup4	3970	4

where $\mathbf{A}$ is a $d \times d$ square matrix. To ensure that $d_{\mathbf{A}}(x, y)$ be a metric, $d(x, y)$ should satisfy the symmetry, non-negativity and triangle inequality, i.e, $\mathbf{A}$ must be *symmetric* and *positive semi-definite*.

If we decompose $\mathbf{A}$ as $\mathbf{A} = \mathbf{P}\mathbf{P}^T$ using Cholesky decomposition, then

$$\begin{aligned} d_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) &= \sqrt{(\mathbf{P}^T(\mathbf{x}_i - \mathbf{x}_j))^T \mathbf{P}^T(\mathbf{x}_i - \mathbf{x}_j)} \\ &= \|\mathbf{P}^T(\mathbf{x}_i - \mathbf{x}_j)\|_F \end{aligned} \tag{3.13}$$

where $\|\cdot\|_F$ is used to denote the Frobenius norm. Therefore, if we treat $\mathbf{P}$ as a projection matrix as in last section, then the Mahalanobis distance is just the Euclidean distance in the projected space. In this sense, learning a good Mahalanobis distance is equivalent to find a good embedding space. Thus the *Embedding by Intergrating Multiple Information Sources* (*EIMIS*) can also be applied to distance metric learning.

4 Experiments

4.1 Clustering We use a variety of document datasets, most of which are frequently used in the data mining research. These datasets include

- **CSTR**. This is the dataset of the abstracts of technical reports (TRs) published in the Department of Computer Science at a research university. The dataset contained 476 abstracts, which were divided into four research areas: Natural Language Processing(NLP), Robotics/Vision, Systems, and Theory.
- **WebKB**. The WebKB dataset contains webpages gathered from university computer science departments. There are about 8280 documents and they are divided into 7 categories: student, faculty, staff, course, project, department and other. The raw text is about 27MB. Among these 7 categories, student, faculty, course and project are four most populous entity-representing categories. The associated subset is typically called WebKB4.
- **Reuters**. The Reuters-21578 Text Categorization Test collection contains documents collected from the Reuters newswire in 1987. It is a standard text categorization benchmark and contains 135 categories. In our experiments, we use a subset of the data collection which includes the 10 most frequent categories among the 135 topics and we call it Reuters-top 10.
- **WebACE**. The WebACE dataset was from WebACE project and has been used for document clustering [14][8]. The WebACE dataset contains 2340 documents consisting news articles from Reuters new service via the Web in October 1997. These documents are divided into 20 classes.
- **News4**. The News4 dataset used in our experiments are selected from the famous 20-newsgroups dataset[1]. The topic *rec* containing *autos*, *motorcycles*, *baseball* and *hockey* was selected from the version 20news-18828. The News4 dataset contains 3970 document vectors.

Table 1 summarizes the characteristics of the datasets.

To pre-process the datasets, we remove the stop words using a standard stop list, all HTML tags are skipped and all header fields except subject and organization of the posted articles are ignored. In all our experiments, we select the top 1000 words by mutual information with class labels and represent all documents in TF-IDF form. Finally all documents are normalized to unit form.

In the experiments, we set the number of clusters equal to the true number of classes C for all the clustering algorithms. To evaluate their performance, we compare the clusters generated by these algorithms with the true classes by computing the following two performance measures.

- **Clustering Accuracy** (***Acc***). The first performance measure is the *Clustering Accuracy*, which discovers the one-to-one relationship between clusters and classes and measures the extent to which each cluster contained data points from the corresponding class. It sums up the whole matching degree between all pair class-clusters. Clustering accuracy can be computed as:

$$Acc = \frac{1}{N} \max \left(\sum_{\mathcal{C}_k, \mathcal{L}_m} T(\mathcal{C}_k, \mathcal{L}_m) \right), \tag{4.14}$$

where $\mathcal{C}_k$ denotes the k-th cluster in the final results, and $\mathcal{L}_m$ is the true m-th class. $T(\mathcal{C}_k, \mathcal{L}_m)$ is the number of entities which belong to class m

[1] http://people.csail.mit.edu/jrennie/20Newsgroups/

are assigned to cluster k. Accuracy computes the maximum sum of $T(\mathcal{C}_k, \mathcal{L}_m)$ for all pairs of clusters and classes, and these pairs have no overlaps. The greater clustering accuracy means the better clustering performance.

- **Normalized Mutual Information** (***NMI***). Another evaluation metric we adopt here is the *Normalized Mutual Information NMI* [28], which is widely used for determining the quality of clusters. For two random variable $\mathbf{X}$ and $\mathbf{Y}$, the *NMI* is defined as:

$$NMI(\mathbf{X}, \mathbf{Y}) = \frac{I(\mathbf{X}, \mathbf{Y})}{\sqrt{H(\mathbf{X})H(\mathbf{Y})}}, \tag{4.15}$$

where $I(\mathbf{X}, \mathbf{Y})$ is the mutual information between $\mathbf{X}$ and $\mathbf{Y}$, while $H(\mathbf{X})$ and $H(\mathbf{Y})$ are the entropies of X and Y respectively. One can see that $NMI(\mathbf{X}, \mathbf{X}) = 1$, which is the maximal possible value of *NMI*. Given a clustering result, the *NMI* in Eq.(4.15) is estimated as

$$NMI = \frac{\sum_{k=1}^{C}\sum_{m=1}^{C} n_{k,m} log\left(\frac{n \cdot n_{k,m}}{n_k \hat{n}_m}\right)}{\sqrt{\left(\sum_{k=1}^{C} n_k log\frac{n_k}{n}\right)\left(\sum_{m=1}^{C} \hat{n}_m log\frac{\hat{n}_m}{n}\right)}}, \tag{4.16}$$

where n_k denotes the number of data contained in the cluster $\mathcal{C}_k$ $(1 \leqslant k \leqslant C)$, $\hat{n}_m$ is the number of data belonging to the m-th class $(1 \leqslant m \leqslant C)$, and $n_{k,m}$ denotes the number of data that are in the intersection between the cluster $\mathcal{C}_k$ and the m-th class. The value calculated in Eq.(4.16) is used as a performance measure for the given clustering result. The larger this value, the better the clustering performance.

We have conducted comprehensive performance evaluations by testing our method and comparing it with 8 other representative data clustering methods using the same data corpora. The algorithms that we evaluated are listed below.

1. Traditional k-means (KM) [12].
2. Spherical k-means (SKM). The implementation is based on [10].
3. Gaussian Mixture Model (GMM). The implementation is based on [21].
4. Spectral Clustering with Normalized Cuts (Ncut). The implementation is based on [41], and the variance of the Gaussian similarity is determined by five-fold cross validation.

Table 4: Descriptions of the datasets

Datasets	Sizes	Classes	Dimensions
Balance	625	3	4
Iris	150	3	4
Ionosphere	351	2	34
Soybean	562	19	35
Wine	178	3	13
Sonar	208	2	60

5. Nonnegative Matrix Factorization (NMF). The implementation is based on [39].

For our integrated KL clustering, (*IKL*), we construct the pairwise relationship matrix by $\mathbf{W}_{ij} = \exp\left(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/(2\sigma^2)\right)$, where σ is a manually defined scale parameter which is equivalent to the medium value of all pairwise document distance. The clustering accuracies comparison results are shown in table 2, and the normalized mutual information comparison results are summarized in table 3.

4.2 Semi-supervised Clustering The data sets used in our experiments including six UCI data sets [7]. In the following we will briefly introduce the basic information of those data sets and Table 4 summarizes the basic information of those data sets.

- **Balance**. This data set was generated to model psychological experimental results. There are totally 625 examples that can be classified as having the balance scale tip to the right, tip to the left, or be balanced.
- **Iris**. The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant.
- **Ionosphere**. It is a collection of the radar signals belonging to two classes. The data set contains 351 objects in total, which are all 34-dimensional.
- **Soybean**. It is collected from the Michalski's famous soybean disease databases, which contains 562 instances from 19 classes.
- **Wine**. The purpose of this data set is to use chemical analysis for determining the origin of wines. It contains 178 instances from 3 classes.
- **Sonar**. This is the data set used by Gorman and Sejnowski in their study of the classification of sonar signals using a neural network, which contains 208 instances from 2 classes.

In our experiments, the constraints were generated as follows: for each constraint, we picked out one pair

Table 2: Clustering accuracies of the various methods

	CSTR	WebKB4	Reuters	WebACE	News4
KM	0.4256	0.3888	0.4448	0.4001	0.3527
SKM	0.4690	0.4318	0.5025	0.4458	0.3912
GMM	0.4487	0.4271	0.4897	0.4521	0.3844
NMF	0.5713	0.4418	0.4947	0.4761	0.4213
Ncut	0.5435	0.4521	0.4896	0.4513	0.4189
IKL	**0.5931**	**0.4977**	**0.5134**	**0.5188**	**0.4697**

Table 3: Normalized mutual information results of the various methods

	CSTR	WebKB4	Reuters	WebACE	News4
KM	0.3675	0.3023	0.4012	0.3864	0.3318
SKM	0.4027	0.4155	0.4587	0.4003	0.4085
GMM	0.4034	0.4093	0.4356	0.4209	0.3994
NMF	0.5235	0.4517	0.4402	0.4359	0.4130
Ncut	0.4833	0.4497	0.4392	0.4289	0.4231
IKL	**0.5673**	**0.4748**	**0.4833**	**0.4695**	**0.4453**

of data points randomly from the input data sets (the labels of which were available for evaluation purpose but unavailable for clustering). If the labels of this pair of points were the same, then we generated a must link. If the labels were different, a cannot link was generated. The amounts of constraints were determined by the size of input data. In all the experiments, the penalties for violating the must-link and cannot link constraints are set to 1 manually, and the results were averaged over 50 trials to eliminate the difference caused by constraints.

For comparison, we also implemented
1) the *constrained kmeans* (*CKmeans*) algorithm [31],
2) the *MPC-Kmeans* (*MPCKmeans*) [6] algorithm,
3) the *PMF* algorithm [34] and
4) the *CMM* algorithm [33].
We denote two methods presented in this paper as
5) SKMSC: semi-supervised clustering with penalization on K-means objective, Eq.(3.9)
6) SSCKM: semi-supervised clustering with penalization on spectral clustering objective, Eq.(3.10)

The F-score [20] is used to evaluate the performance of each algorithm.

Figure 1 shows the F-scores(in percentages) of the four algorithms on the six UCI data sets under different amounts of constraints respectively. We can find that the our SSCKM and SKMSC algorithms perform consistently better than (at leat competitive with) existing methods.

4.3 Face Recognition In this section we will present the experimental results of applying our method to face recognition. First let's briefly introduce the basic information of each data set:

- **Yale**[2]. Contains 165 grayscale images of 15 individuals. There are 11 images per subject, one per different facial expression or configuration: center-light, w/glasses, happy, left-light, w/no glasses, normal, right-light, sad, sleepy, surprised, and wink.
- **ORL**[3]. Contains 10 different images of each of 40 distinct subjects. For some subjects, the images were taken at different times, varying the lighting, facial expressions (open / closed eyes, smiling / not smiling) and facial details (glasses / no glasses). All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position (with tolerance for some side movement).
- **PIE**[4]. Contains 41,368 images of 68 people, each person under 13 different poses, 43 different illumination conditions, and with 4 different expressions. In our experiments, we only use a subset containing 5 near frontal poses (C05, C07, C09, C27, C29) and all the images under different illuminations and expressions. So, there are 170 images for each individual.

In our experiments, all the face images are resize to 32x32. Besides our method, we also implement the following methods for comparisons:

[2] http://cvc.yale.edu/projects/yalefaces/yalefaces.html
[3] http://www.uk.research.att.com/facedatabase.html
[4] http://www.ri.cmu.edu/projects/project_418.html

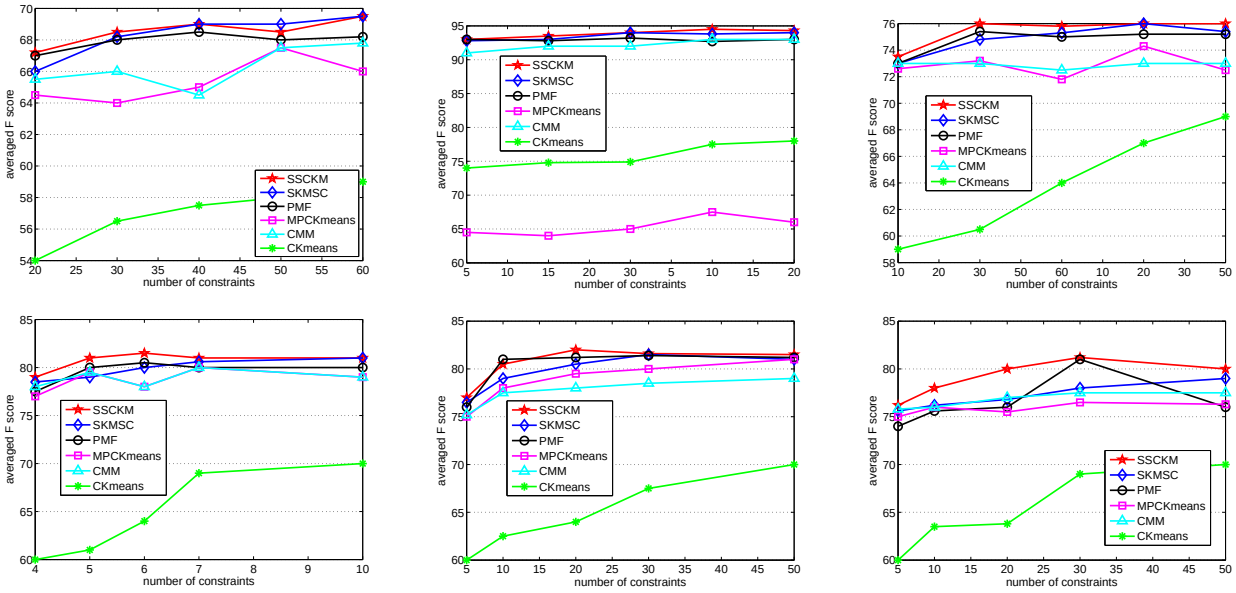


Figure 1: F-score comparisons of different methods for semi-supervised clustering. Dataset: Top row from left: balance, iris, ionsphere; bottom row from left: soybean, wine, sonar.

- Nearest Neighbor Classifier (NN): This method is implemented as the baseline method for comparison, where all the computations are performed in the original data space.
- Eigenface: The face images are first projected by PCA and then the recognition procedure is implemented in the projected space with the NN classifier. The projected dimension is set using exhaustive search by 5-fold cross validation.
- Fisherface: The implementation is the same as in [3].
- Laplacianface: The implementation is the same as in [15].

For our method (which is denoted as EIMIS), we first use the training data to train the projection directions P of Eq.(3.11) and then use them to project the whole data sets, and the recognition process will be implemented in the projected space using the NN classifier. The relation matrix $\mathbf{W}$ is computed by Gaussian function as $\mathbf{W}_{ij} = \exp\left(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/(2\sigma^2)\right)$, and the optimal σ is set by 5 fold cross validation.

In our experiments, we first randomly select a certain number of face images from each subject for each data set. Those selected images will be used as training set and the remaining images will be used for testing. The recognition error averaged over 50 independent runs are shown in Fig.2. For all the figures, the x-axis represents the number of randomly selected training faces, and the y-axis represents the averaged recognition error. From the figures we can clearly see the effectiveness of our method.

5 Conclusions

In this paper, We propose a new integrated clustering approach by combine K-means clustering on data attributes and spectral clustering on pairwise relations. This is an effective way to make use of multiple information sources. This KL clustering does not require extra parameters and can be extended to semi-supervised clustering, data embedding and metric learning.

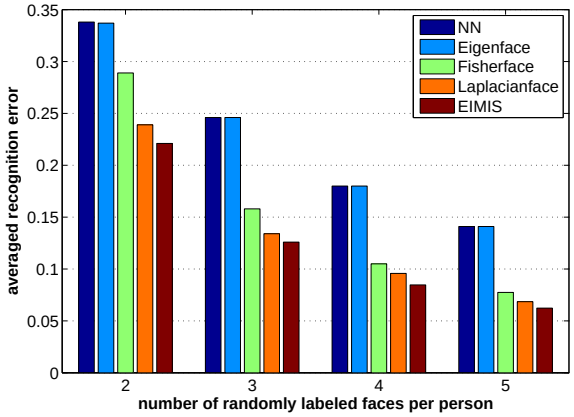
We perform extensive experiments on 4 document sets, 6 UCI data sets, and 3 image data sets, for clustering, classification and semi-supervised learning tasks. Our methods consistently outperform existing methods.

Acknowledgements

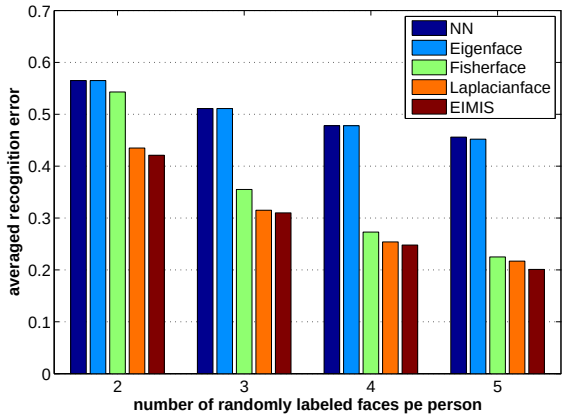
The work of F. Wang and T. Li is partially supported by NSF grants IIS-0546280, DMS-0844513 and CCF-0830659. The work of C. Ding is partially supported by NSF grants DMS-0844497 and CCF-0830780.

References

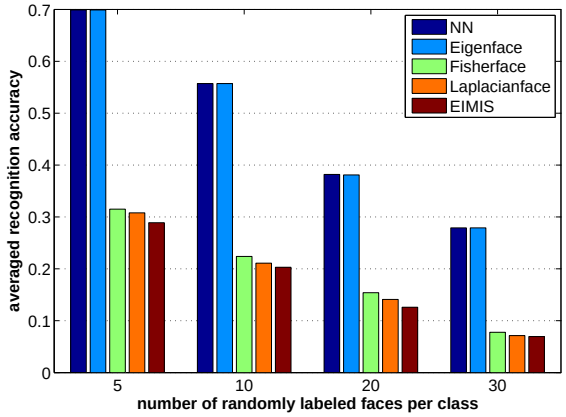
[1] Adam Anthony and Marie DesJardins. Open problems in relational data clustering. In *Proceedings of the*



(a) ORL



(b) Yale



(c) PIE

Figure 2: Face recognition results on 3 datasets. The x-axis gives the number of images randomly selected from each class (person) for training. The y-axis gives the averaged recognition error.

ICML Workshop on Open Problems in Stastistical Relational Learning, 2006.

[2] S. Basu, M. Bilenko, and R. J. Mooney. A probablistic framework for semi-supervised clustering. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 59–68, 2004.

[3] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:711–720, 1997.

[4] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15:1373–1396, 2003.

[5] Indrajit Bhattacharya and Lise Getoor. Entity resolution in graph data. Technical report, University of Maryland, College Park, October 2005.

[6] M. Bilenko, S. Basu, and R. J. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *Proc. of International Conference on Machine Learning*, 2004.

[7] C. Blake, E. Keogh, , and C. J. Merz. Uci repository of machine learning databases. *http://www.ics.uci.edu/m̃learn/MLRepository.html*, 1998.

[8] D. Boley. Principal direction divisive partitioning. *Data Mining and Knowledge Discovery*, 2:325–344, 1998.

[9] T. F. Cox, M. A. A. Cox, and B. Raton. Multidimensional scaling. *Technometrics*, 45(2):182, 2003.

[10] I. S. Dhillon and D. S. Modha. Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42(1):143–175, 2001.

[11] C. Ding and X. He. K-means clustering and principal component analysis. *Int'l Conf. Machine Learning (ICML)*, 2004.

[12] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2000.

[13] Sašo Džeroski. Multi-relational data mining: an introduction. *SIGKDD Explor. Newsl.*, 5(1):1–16, 2003.

[14] E. Han, D. Boley, M. Gini, R. Gross, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, and J. Moore. Webace: a web agent for document categorization and exploration. In *Proc. of the 2nd International Conference on Autonomous Agents*, pages 408–415. ACM Press, 1998.

[15] X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang. Face recognition using laplacianfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):328–340, 2005.

[16] Xiaofeng He, Chris H. Q. Ding, Hongyuan Zha, and Horst D. Simon. Automatic topic identification using webpage clustering. In *ICDM '01: Proceedings of the 2001 IEEE International Conference on Data Mining*, pages 195–202, Washington, DC, USA, 2001. IEEE Computer Society.

[17] I. T. Jolliffe. *Principal Component Analysis*. Springer Verlag, 1986.

[18] B. Kulis, S. Basu, I. Dhillon, and R. J. Mooney. Semi-supervised graph clustering: a kernel approach. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 457–464, 2005.

[19] G.R. Lanckriet, N. Cristianini, P.L. Bartlett, L.E. Ghaoui, and M.I. Jordan. Learning the kernel matrix with semi-definite programming. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 323–330, 2006.

[20] B. Larsen and C. Aone. Fast and effective text mining using linear-time document clustering. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 16–22, 1999.

[21] X. Liu, Y. Gong, W. Xu, and S. Zhu. Document clustering with cluster refinement and model selection capabilities. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 191–198. ACM Press, 2002.

[22] Bo Long, Xiaoyun Wu, Zhongfei (Mark) Zhang, and Philip S. Yu. Unsupervised learning on k-partite graphs. In *Proceedings of ACM SIGKDD*, pages 317–326, 2006.

[23] J. Neville, M. Adler, and D. Jensen. Clustering relational data using attribute and link information. 2003.

[24] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, pages 849–856. MIT Press, 2001.

[25] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[26] N. Shental and D. Weinshall. Learning distance functions using equivalence relations. In *In Proceedings of the Twentieth International Conference on Machine Learning*, pages 11–18, 2003.

[27] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:888–905, 2000.

[28] A. Strehl, J. Ghosh, and C. Cardie. Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617, 2002.

[29] Ben Taskar, Eran Segal, and Daphne Koller. Probabilistic classification and clustering in relational data. In *In Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, pages 870–878, 2001.

[30] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[31] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl. Constrained k-means clustering with background knowledge. In *Proc. of Internation Conference on Machine Learning*, 2001.

[32] F. Wang, S. Chen, T. Li, and C. Zhang. Semi-supervised metric learning by maximizing constraint margin. In *ACM 17th Conference on Information and Knowledge Management*, 2008.

[33] F. Wang, S. Chen, T. Li, and C. Zhang. Semi-supervised metric learning by maximizing constraint margin. In *Proceedings of the ACM 17th Conference on Information and Knowledge Management*, 2008.

[34] F. Wang, T. Li, and C. Zhang. Semi-supervised clustering via matrix factorization. In *Proceedings of The 8th SIAM Conference on Data Mining*, 2008.

[35] F. Wang and C. Zhang. Label propagation through linear neighborhoods. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 985–992, 2006.

[36] K. Q. Weinberger, J. Blitzer, and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in Neural Information Processing Systems 18*. MIT Press, 2006.

[37] K. Q. Weinberger and L. K. Saul. Unsupervised learning of image manifolds by semidefinite programming. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages 988–995, 2004.

[38] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In *Advances in Neural Information Processing Systems 15*, pages 505–512. MIT Press, 2003.

[39] W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 267–273. ACM Press, 2003.

[40] Xiaoxin Yin, Jiawei Han, and Philip S. Yu. Cross-relational clustering with user's guidance. In *KDD '05: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 344–353, 2005.

[41] S. X. Yu and J. Shi. Multiclass spectral clustering. In *In International Conference on Computer Vision*, pages 313–319, 2003.

[42] H. Zha, X. He, C. Ding, H. Simon, and M. Gu. Spectral relaxation for k-means clustering. In *Advances in Neural Information Processing Systems*, pages 1057–1064. MIT Press, 2001.

[43] D. Zhang and Z. Zhou S. Chen. Semi-supervised dimensionality reduction. In *Proceedings of the 7th SIAM International Conference on Data Mining*, 2007.