Constrained Clustering by Spectral Kernel Learning

Zhenguo Li[1,2] and Jianzhuang Liu[1,2]
[1]Dept. of Information Engineering, The Chinese University of Hong Kong, Hong Kong
[2]Multimedia Lab, Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences
{zgli,jzliu}@ie.cuhk.edu.hk

Abstract

Clustering performance can often be greatly improved by leveraging side information. In this paper, we consider constrained clustering with pairwise constraints, which specify some pairs of objects from the same cluster or not. The main idea is to design a kernel to respect both the proximity structure of the data and the given pairwise constraints. We propose a spectral kernel learning framework and formulate it as a convex quadratic program, which can be optimally solved efficiently. Our framework enjoys several desirable features: 1) it is applicable to multi-class problems; 2) it can handle both must-link and cannot-link constraints; 3) it can propagate pairwise constraints effectively; 4) it is scalable to large-scale problems; and 5) it can handle weighted pairwise constraints. Extensive experiments have demonstrated the superiority of the proposed approach.

1. Introduction

1.1. Challenges in Clustering

Clustering is a fundamental problem in pattern recognition, whose goal is to group similar objects of a data set into clusters. Representative algorithms include k-means, Gaussian mixture models, spectral clustering, and linkage clustering. Typically, clustering is conducted in an unsupervised way, and thus the performance depends heavily on the data features[1]. Due to the unsupervised nature, clusters obtained by a clustering algorithm may not necessarily correspond to the semantic categories of the data. This phenomenon is called *semantic gap* [14]. Furthermore, real-world data may admit various semantic concepts on category. For example, persons may differ in identity, pose, with or without glasses, or gender. In other words, a data set may possess multiple natural clusterings. Depending on specific applications, we may desire one or another, but a clustering algorithm always produces one certain clustering. This dilemma is called *clustering ambiguity* [14]. Clearly, unsupervised clustering is hard to address these practical issues, which leads to the idea of constrained clustering.

1.2. Constrained Clustering

In constrained clustering, one resorts to side information to guide clustering. Popular side information includes pairwise constraints [22], relative comparisons [5], and cluster sizes [25]. A pairwise constraint specifies two objects from the same cluster or not, known as the must-link and the cannot-link. Such pairwise relationship can be easily collected from domain knowledge, class labels, or the user. A relative comparison states that object A is more similar to B than to C. In this paper, we focus on pairwise constraints, as in most of the literature. Conceptually, pairwise constraints can reduce the semantic gap and remove the clustering ambiguity to some extent.

In clustering with pairwise constraints, one faces two sources of similarity information of a data set, the feature similarity and the pairwise constraints, and the task is to combine the two to find a consistent partition of the data. One line of research aims to adapt particular clustering algorithms, where one either changes the clustering process of the algorithms like k-means [23], Gaussian mixtures [19], and linkage [9], or modifies the optimization problems such as Normalized Cuts [26, 4, 25]. These methods, however, are subject to the limitations of the base algorithms [23, 19, 9], limited to two-class problems [4, 25], or deal with the must-link only [26].

Another line of research proposes to derive a new similarity measure of the data. To this end, one can train a distance metric such as Mahalanobis distance, which corresponds to a linear transformation of the data features [24]. A more general idea is to seek a similarity matrix directly [8, 11, 15, 13, 14]. With this idea, some methods modify similarities between constrained objects only [8, 11]. One problem with these methods is that the obtained similarity matrix is not a valid kernel matrix generally, thus requiring further justification. Kernel learning methods have also been explored [15, 13, 14]. One desirable property with these methods is that they aim at propagating pair-

[1] The features here refer to vectors or pairwise similarities.

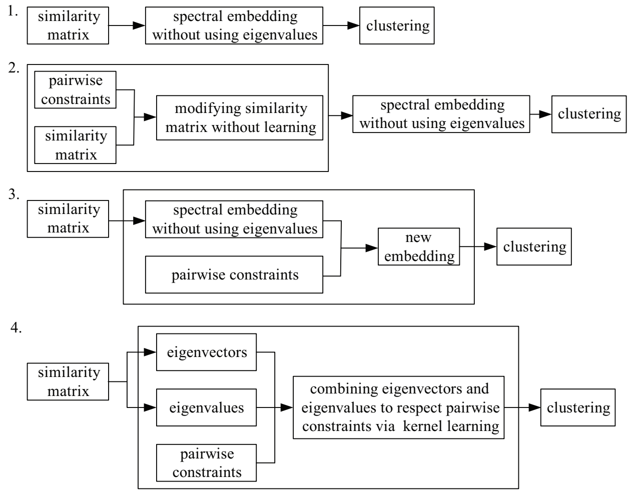

1.
similarity matrix
spectral embedding without using eigenvalues
clustering
2.
pairwise constraints
similarity matrix
modifying similarity matrix without learning
spectral embedding without using eigenvalues
clustering
3.
similarity matrix
spectral embedding without using eigenvalues
pairwise constraints
new embedding
clustering
4.
similarity matrix
eigenvectors
eigenvalues
pairwise constraints
combining eigenvectors and eigenvalues to respect pairwise constraints via kernel learning
clustering

Figure 1. Algorithmic flows for constrained clustering. 1. Spectral clustering [20, 16]. 2. Modifying the similarity matrix without learning [8, 11]. 3. Adapting a spectral embedding via SDP [14]. 4. Adapting a spectral kernel via QP proposed in this paper.

wise constraints to the whole data set, which is important because pairwise constraints are often sparse in practice. However, the one in [15] is confined to two-class problems. Though the method [13] applies to multi-class problems seamlessly, it is computationally expensive for it requires semidefinite programming (SDP) [1] over a full kernel matrix. Instead of learning a full-nonparametric kernel matrix, the method [14] suggests to learn a spectral regularized, semi-parametric kernel matrix. This method is efficient only for problems with not too many clusters, e.g., less than 50.

In this paper, we propose to learn a spectral kernel matrix for constrained clustering. Our framework can be formulated as a quadratic programming (QP) problem that is easy to solve. The algorithmic flows of the proposed and other related methods are shown in Fig. 1. Compared with previous methods, our approach has several attractive features: 1) it is applicable to multi-class problems; 2) it can handle both must-link and cannot-link constraints; 3) it can propagate pairwise constraints effectively; 4) it is scalable to large-scale problems with large numbers of clusters; and 5) it can handle weighted pairwise constraints. The rest of the paper is organized as follows. We review the preliminaries in Section 2 and present the motivation in Section 3. The main framework is proposed in Section 4. Experimental results are reported in Section 5. Section 6 concludes the paper.

2. Spectral Graph Theory

In this section, we review necessary background in spectral graph theory [3] and introduce the notation used in the paper. Suppose the pairwise similarity information of a given data set $\mathcal{X} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ is captured by a nonnegative and symmetric matrix $W = (w_{ij})$, with w_{ij} denoting the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$. W may be sparse for efficiency. In graph notation, we have a graph $\mathcal{G} = \{\mathcal{X}, W\}$ with $\mathcal{X}$ and W being the node set and the weight matrix, respectively. The Laplacian and the normalized Laplacian of $\mathcal{G}$ are respectively defined as $L = D - W$ and $\bar{L} = D^{-1/2}LD^{-1/2}$, where $D = \text{diag}(d_1, ..., d_n)$ with $d_i = \sum_j w_{ij}$. The main subject in spectral graph theory is the study of the properties of the eigenvectors and eigenvalues of L and $\bar{L}$, where the smoothness of the eigenvectors is important.

The smoothness of a function on the graph $f : \mathcal{X} \to \mathcal{R}$ can be measured by [28]

$$\Omega(f) = \frac{1}{2}\sum_{i,j} w_{ij} \left(\frac{f(\mathbf{x}_i)}{\sqrt{d_i}} - \frac{f(\mathbf{x}_j)}{\sqrt{d_j}} \right)^2 = \mathbf{f}^T \bar{L} \mathbf{f}, \quad (1)$$

where $\mathbf{f} = (f(\mathbf{x}_1), ..., f(\mathbf{x}_n))^T$. The smaller is $\Omega(f)$, the smoother is f. This measure penalizes large changes between nodes strongly connected. An unnormalized smoothness measure also frequently used can be found in [29]. Let $(\lambda_i, \mathbf{v}_i)$'s be the eigen-pairs of $\bar{L}$, where $\lambda_1 \leq \cdots \leq \lambda_n$ and $\|\mathbf{v}_i\| = 1$. Note that λ_i's are nonnegative for $\bar{L}$ is positive semidefinite [3]. We have

$$\Omega(\mathbf{v}_i) = \mathbf{v}_i^T \bar{L} \mathbf{v}_i = \lambda_i. \quad (2)$$

This tells that the smoothness of the eigenvector $\mathbf{v}_i$ is measured by its eigenvalue λ_i. A smaller eigenvalue corresponds to a smoother eigenvector.

Recall that spectral clustering [20, 16] uses the smoothest k (the number of clusters) eigenvectors of $\bar{L}$ to reveal the cluster structure of the data. This justifies the main property of spectral clustering that nearby objects are more likely to be partitioned in the same cluster[2]. In this work, we seek a similar clustering capability subject to the given pairwise constraints.

3. Motivation

In this section, we present the idea of spectral kernel learning for clustering with pairwise constraints. We resort to kernel methods and spectral graph theory since they are general frameworks for analyzing pairwise relationship.

In kernel learning, one seeks a feature mapping $\Phi : \mathcal{X} \to \mathcal{F}$ from the input space $\mathcal{X}$ to a feature space $\mathcal{F}$ to reshape the original data. This is equivalent to finding a kernel matrix $K = (k_{ij})$ with $k_{ij} =< \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) >_{\mathcal{F}}$, where $< \cdot, \cdot >_{\mathcal{F}}$ denotes the inner product in the feature space $\mathcal{F}$ [18]. For clustering purposes, it is desired for a feature

[2] This property is well known as the cluster assumption in semi-supervised learning [28].

mapping that maps objects from the same cluster close and maps objects from different clusters well-separated.

Because it is desired that nearby objects are more likely to be partitioned in the same cluster, we like Φ to map nearby objects nearby. In other words, Φ should be smooth on the graph $\mathcal{G}$. By (1), the smoothness of Φ is measured by

$$\Omega(\Phi) = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}\left\|\frac{\Phi(\mathbf{x}_i)}{\sqrt{d_{ii}}} - \frac{\Phi(\mathbf{x}_j)}{\sqrt{d_{jj}}}\right\|_{\mathcal{F}}^2 = \bar{L} \bullet K. \quad (3)$$

Let $(\beta_i, \mathbf{u}_i)$, where $\|\mathbf{u}_i\| = 1$, $i = 1, ..., n$, and $\beta_1 \geq \cdots \geq \beta_n \geq 0$, be the pairs of the eigenvalues and eigenvectors of K. Since $K = \sum_{i=1}^{n} \beta_i \mathbf{u}_i\mathbf{u}_i^T$ and $\Omega(\mathbf{u}_i) = \mathbf{u}_i^T\bar{L}\mathbf{u}_i$, we have

$$\Omega(\Phi) = \bar{L} \bullet K = \sum_{i=1}^{n} \beta_i \mathbf{u}_i^T \bar{L}\mathbf{u}_i = \sum_{i=1}^{n} \beta_i \Omega(\mathbf{u}_i). \quad (4)$$

Thus the smoothness of Φ is fully determined by the eigenvalues of K and the smoothness the eigenvectors of K.

To determine a kernel matrix, it suffices to determine its eigenvalues and eigenvectors. The above analysis suggests us to choose smooth eigenvectors for K. Particularly, a smoother eigenvector should be assigned to a larger eigenvalue in order to decrease the value of $\Omega(\Phi)$ in (4). We argue that a natural choice for the eigenvectors is $\mathbf{u}_i = \mathbf{v}_i$, $i = 1, ..., n$, as $\mathbf{v}_i$'s contain useful information for clustering and $\Omega(\mathbf{v}_1) \leq \cdots \leq \Omega(\mathbf{v}_n)$. The next is to determine the eigenvalues for K. We propose to learn the eigenvalues using pairwise constraints.

Formally, we propose to learn a kernel matrix in the form

$$K = \sum_{i=1}^{n} \beta_i \mathbf{v}_i\mathbf{v}_i^T,\ \beta_1 \geq \cdots \geq \beta_n \geq 0, \quad (5)$$

where $\mathbf{v}_i$ is the i-th smoothest eigenvector of $\bar{L}$. Furthermore, for clustering purposes, rough eigenvectors are less informative and probably carry noise information that is unfavorable to clustering. Thus we may keep the smoothest m ($m \ll n$) eigenvectors only by simply setting $\beta_i = 0$ for $i > m$ to further impose smoothness on Φ.

Kernel matrices constructed from the (normalized) graph Laplacian via adapting the eigenvalues of the graph Laplacian are typically called *spectral kernels* [27] or *graph kernels* [29] in semi-supervised learning. Often used spectral kernels include the regularized Laplacian kernel $K = (I + t^2\bar{L})^{-1}$ [28], the diffusion kernel $K = e^{-\frac{t^2}{2}\bar{L}}$ [10], and the p-step random walk kernel $K = (aI - \bar{L})^p$ [2]. We refer the reader to [27, 21] for the theoretical justification of spectral kernels. Spectral kernels have been used to address unsupervised noise robust spectral clustering [12]. In this paper, we develop a novel and efficient spectral kernel learning framework for combining the low-level feature similarity and the high-level pairwise constraints for constrained clustering.

4. Spectral Kernel Learning

In this section, we present the spectral kernel learning framework for constrained clustering. We use $\mathcal{M} = \{(i, j)\}$ and $\mathcal{C} = \{(i, j)\}$ to denote the sets of must-link and cannot-link constraints, respectively. The goal is to learn the eigenvalues of the spectral kernel so that the kernel respects the pairwise constraints as much as possible. Our main idea is to find the spectral kernel that is closest to an ideal kernel.

4.1. An Ideal Kernel

For k-class problems, the aim is to find the binary indicator matrix $Y = (y_{ij})$ of size $n \times k$ where $y_{ij} = 1$ if $\mathbf{x}_i$ is in the j-th *ground-truth* cluster and $y_{ij} = 0$ otherwise. Denote Φ as the mapping that associates $\mathbf{x}_i$ with the i-th row of Y. Then under this mapping, all the objects are mapped to a unit sphere where objects from the same cluster are mapped to the same point and any two objects from different clusters are mapped to be orthogonal. This is ideal for clustering purposes. We thus consider a mapping with this property as an ideal mapping and the corresponding kernel matrix $K = (k_{ij})$ with $k_{ij} =< \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) >$, i.e., $K = YY^T$, as an ideal kernel. Clearly, clustering becomes trivial with an ideal kernel.

4.2. A Cost Function

Our main idea is to seek a spectral kernel K that is closest to an ideal kernel. Mathematically, we propose to find a spectral kernel K to minimize the following cost function:

$$\begin{aligned}\mathcal{L}(K) = &\sum_{i=1}^{n}(k_{ii} - 1)^2 + \sum_{(i,j)\in\mathcal{M}}(k_{ij} - 1)^2 \\ &+ \sum_{(i,j)\in\mathcal{C}}(k_{ij} - 0)^2. \end{aligned} \quad (6)$$

Let $\mathcal{S} = \{(i, j, t_{ij})\}$ be the set of pairwise constraints, where t_{ij} is a binary variable that takes 1 or 0 to indicate $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to the same cluster or not. It is reasonable to assume $(i, i, t_{ii}) \in \mathcal{S}$, where $t_{ii} = 1$, $i = 1, ..., n$. Then (6) can be rewritten as

$$\mathcal{L}(K) = \sum_{(i,j,t_{ij})\in\mathcal{S}} (k_{ij} - t_{ij})^2. \quad (7)$$

Let $C = (c_{ij})$ be the constraint indicator matrix where $c_{ij} = c_{ji} = 1$ if $(i, j) \in \mathcal{M}$ or $(i, j) \in \mathcal{C}$, $c_{ii} = 1$, $i = 1, ..., n$, and $c_{ij} = 0$ otherwise. Let $T = (t_{ij})$. Then we can write (7) in the following matrix form

$$\mathcal{L}(K) = \|C \circ (K - T)\|_F^2, \quad (8)$$

where $\circ$ denotes element-wise product between two matrices, and $\|\cdot\|_F$ denotes the Frobenius norm.

4.3. Quadratic Programming

The above analysis suggests the optimization problem:

$$\min_{\beta_1,\dots,\beta_m} \|C \circ (K - T)\|_F^2 \tag{9}$$

$$\text{s.t. } K = \sum_{i=1}^{m} \beta_i \mathbf{v}_i \mathbf{v}_i^T \tag{10}$$

$$\beta_1 \geq \cdots \geq \beta_m \geq 0. \tag{11}$$

This is in fact a convex QP problem, as shown below. Let $F = (\mathbf{v}_1, ..., \mathbf{v}_m) = (\mathbf{y}_1, ..., \mathbf{y}_n)^T$ where $\mathbf{y}_i^T$ denotes the i-th row of F, $\Lambda = \text{diag}(\beta_1, ..., \beta_m)$, and $\mathbf{z} = (\beta_1, ..., \beta_m)^T$. Then $k_{ij} = \mathbf{y}_i^T \Lambda \mathbf{y}_j = \mathbf{z}^T(\mathbf{y}_i \circ \mathbf{y}_j)$ and $(k_{ij} - t_{ij})^2 = \mathbf{z}^T \mathbf{y}_{ij} \mathbf{y}_{ij}^T \mathbf{z} - 2t_{ij}\mathbf{y}_{ij}^T \mathbf{z} + t_{ij}^2$, where $\mathbf{y}_{ij} = \mathbf{y}_i \circ \mathbf{y}_j$. Thus

$$\|C \circ (K - T)\|_F^2 = \sum_{(i,j,t_{ij}) \in \mathcal{S}} c_{ij}^2 (k_{ij} - t_{ij})^2 \tag{12}$$

$$= \frac{1}{2}\mathbf{z}^T A \mathbf{z} + \mathbf{b}^T \mathbf{z} + c, \tag{13}$$

where

$$A = 2 \sum_{(i,j,t_{ij}) \in \mathcal{S}} c_{ij}^2 \mathbf{y}_{ij} \mathbf{y}_{ij}^T, \quad \mathbf{b} = -2 \sum_{(i,j,t_{ij}) \in \mathcal{S}} c_{ij}^2 t_{ij} \mathbf{y}_{ij},$$

$$c = \sum_{(i,j,t_{ij}) \in \mathcal{S}} c_{ij}^2 t_{ij}^2. \tag{14}$$

Therefore, the problem (9–11) becomes

$$\min_{\mathbf{z}} \frac{1}{2}\mathbf{z}^T A \mathbf{z} + \mathbf{b}^T \mathbf{z} \tag{15}$$

$$\text{s.t. } \beta_1 \geq \cdots \geq \beta_m \geq 0, \tag{16}$$

where the constant term c is dropped. This is a standard QP problem with m variables and m linear inequalities [1]. Note that A is symmetric and positive definite. Thus this QP problem is convex and can be optimally solved efficiently.

4.4. Weighted Pairwise Constraints

In practice, prior information can be available about how likely a pairwise constraint is believed to be correct. This prior can be used to further improve the learning. One simple way to encode such information is by weighting, i.e., each constraint is assigned a nonnegative weight where the larger is the weight, the more likely it is believed to be correct. A zero weight indicates no prior bias. Our framework can deal with weighted pairwise constraints by simply replacing the c_{ij} in (8) with the associated weight.

4.5. The CCSKL Algorithm

With the solved β_1,...,β_m, the kernel matrix K can be obtained as $K = \sum_{i=1}^{m} \beta_i \mathbf{v}_i \mathbf{v}_i^T = F \Lambda F^T$ where $F = (\mathbf{v}_1, ..., \mathbf{v}_m)$ and $\Lambda = \text{diag}(\beta_1, ..., \beta_m)$. To perform clustering, we can apply kernel k-means to K, or equivalently apply k-means to the rows of $F\Lambda^{\frac{1}{2}}$. We take the latter in the experiments as it is of less space complexity.

The overall procedure is summarized in Algorithm 1, which we call constrained clustering by spectral kernel learning (CCSKL). The smoothest m eigenvectors of $\bar{L}$ can be efficiently obtained with the Lanczos algorithm [7].

Algorithm 1 Constrained Clustering by Spectral Kernel Learning (CCSKL)

```
Input: A data set $\mathcal{X} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$, pairwise constraint sets $\mathcal{M}$ and $\mathcal{C}$, and the number of clusters $k$.
Output: Cluster labels for the objects.
1: Form a sparse symmetric similarity matrix $W = (w_{ij})$.
2: Form the normalized graph Laplacian $\bar{L} = I - D^{-1/2} W D^{-1/2}$, where $I$ is the identity matrix of size $n \times n$ and $D = \text{diag}(d_1, ..., d_n)$ with $d_i = \sum_{j=1}^{n} w_{ij}$.
3: Compute the $m$ eigenvectors $\mathbf{v}_1, ..., \mathbf{v}_m$ of $\bar{L}$ corresponding to the first $m$ smallest eigenvalues. Denote $F = (\mathbf{v}_1, ..., \mathbf{v}_m)$.
4: Solve the QP problem in (15) and (16) for $\beta_1, ..., \beta_m$. Denote $\Lambda = \text{diag}(\beta_1, ..., \beta_m)$.
5: Apply $k$-means to the rows of $F\Lambda^{\frac{1}{2}}$ to form $k$ clusters.
```

5. Experimental Results

In this section, we conduct experiments on real data to evaluate the proposed CCSKL. Three most related algorithms are compared, including Spectral Learning (SL) [8], Semi-Supervised Kernel k-means (SSKK) [11], and Constrained Clustering via Spectral Regularization (CCSR) [14]. All the four methods directly address multi-class problems and deal with both the must-link and the cannot-link constraints, while other related methods are either limited to two-class problems [15, 25] or only the must-link [26], or computationally impractical [13]. The performance of Normalized Cuts[3] (NCuts) [20] is also reported for reference, where no pairwise constraints are used.

5.1. Clustering Evaluation

We use clustering error to evaluate a clustering result, which is a widely used criterion for the evaluation of clustering algorithms. This measure works by best matching a clustering result to the ground-truth cluster labels. Given a permutation mapping map$(\cdot)$ over the cluster labels, the clustering error with respect to map$(\cdot)$ is

$$1 - \frac{1}{n} \sum_{i=1}^{n} \delta(y_i, \text{map}(y_i')), \tag{17}$$

[3] http://www.cis.upenn.edu/∼jshi/software/

where y_i and y'_i are the ground-truth label and the obtained cluster label for object $\mathbf{x}_i$, respectively, $\delta(x, y) = 1$ if $x = y$ and $\delta(x, y) = 0$ otherwise, and n is the number of objects. The clustering error is defined as the minimal error over all possible permutation mappings.

5.2. Parameter Selection

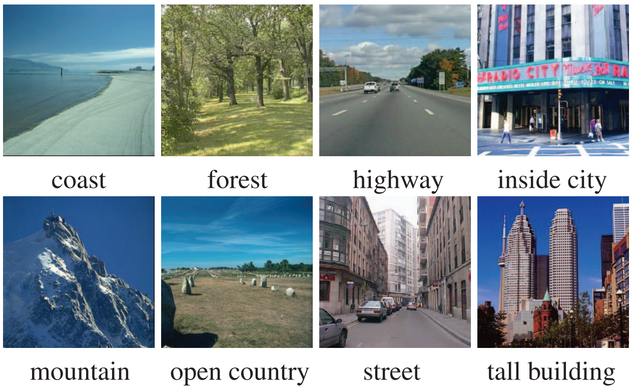
For CCSKL, we use the standard QP solver *quadprog* in MATLAB to solve the QP problem. The number m is set to 20. For SSKK, the normalized cut version is used for it performs best as reported in [11]. As in [11], the constraint penalty is set to $n/(ks)$, where n, k, and s are the numbers of objects, clusters, and pairwise constraints, respectively.

We follow CCSR [14] to construct graphs and generate pairwise constraints. All the algorithms are tested on the same graphs. The weighted 20-NN graph is used, i.e., $w_{ij} = e^{-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma^2}$ if $\mathbf{x}_i$ is among the 20-nearest neighbors of $\mathbf{x}_j$ or vice versa, and $w_{ij} = 0$ otherwise, where σ ranges over the set linspace$(0.1r, r, 5) \bigcup$ linspace$(r, 10r, 5)$ with r being the averaged distance from each object to its 20-th nearest neighbor and linspace(r_1, r_2, t) denoting the set of t linearly equally-spaced numbers between r_1 and r_2. In summary, we form 9 candidate graphs for each data set and select the one that gives the best result.

For each data set, 10 different numbers of pairwise constraints are randomly generated using the ground-truth cluster labels. For each set of pairwise constraints, the result is averaged over 50 realizations of k-means (for CCSKL and SL) or weighted kernel k-means (for SSKK) with different random initializations. For a fixed number of pairwise constraints, the reported result is averaged over 50 realizations of different pairwise constraints. For NCuts, the reported result is averaged over 50 realizations of the discretization procedure.

5.3. Image Data

In this experiment, we test the algorithms on four image databases, USPS[4], MNIST[5], Yale Face Database B (YaleB) [6], and a scene category data set (Scene) [17]. USPS and MNIST contain images of handwritten digits from 0 to 9 of sizes 16 × 16 and 28 × 28, respectively. There are 7291 training examples and 2007 test examples in USPS. MNIST has a training set of 60,000 examples and a test set of 10,000 examples. YaleB contains 5760 single light source images of 10 subjects captured under 576 viewing conditions (9 poses × 64 illumination). We down-sample each image in YaleB to 30 × 40 pixels. The Scene data set was collected by Oliva and Torralba [17], containing 8 categories of natural scenes (see Fig. 2 for some sample images). We use the feature called Spatial Envelope [17] to represent each scene image, although other choices are certainly possible. The feature is a 512-dimensional vector, capturing the dominant spatial structure (naturalness, openness, roughness, expansion and ruggedness) of the scene. For USPS, MNIST, and YaleB, the feature to represent each image is a vector formed by concatenating all the columns of the image intensities. In the experiments, we use all the examples from USPS, YaleB, and Scene, but use only digits 0 to 4 in the test set in MNIST due to its large amount of data. Table 1 summarizes the four data sets used.


coast forest highway inside city

mountain open country street tall building

Figure 2. Example images in the Scene data set from [17].

Table 1. Four image data sets used in the experiment.

	USPS	MNIST	YaleB	Scene
# objects	9298	5139	5760	2688
# dimensions	256	784	1200	512
# clusters	10	5	10	8

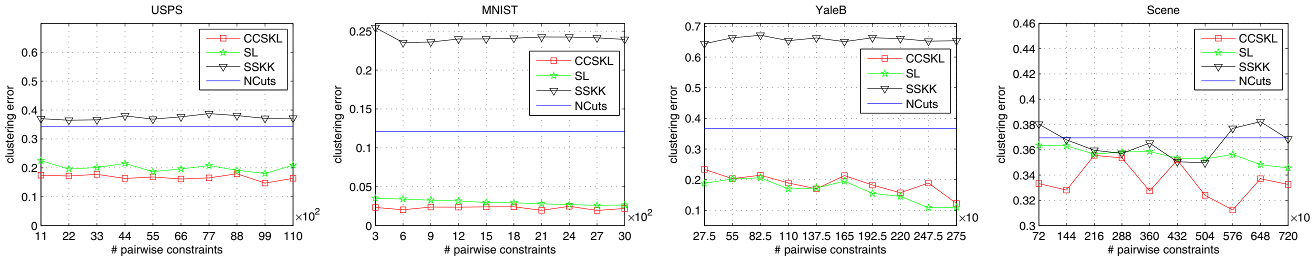
We summarize the results in Fig. 3, from which one can see that CCSKL performs best in most cases. CCSKL and SL consistently obtain better results than NCuts, showing that they do exploit the pairwise constraints effectively. In contrast, SSKK performs unsatisfactorily, even worse than NCuts in some cases. This may be due to the two main components of SSKK [11], i.e., the resultant similarity matrix contains negative entries and the optimization procedure is not guaranteed to converge. The results of CCSKL are comparable to those of CCSR reported in [14]. Note that CCSKL is much more efficient than CCSR computationally, especially for problems with a large number of clusters.

The pairwise constraints used are quite sparse. Even only 2% of the data from each cluster of USPS can generate 14770 constraints, larger than 11000, the largest number of pairwise constraints in the experiments. To visualize the propagation effect of the proposed CCSKL, we show in Fig. 4 the distance matrices of the original data, the spectral embedding, and the spectral kernel of CCSKL. We can see that among all, the block structure of the distance matrices of CCSKL is most significant, meaning that CCSKL does propagate pairwise constraints effectively.

We also compare the execution times of the algorithms.

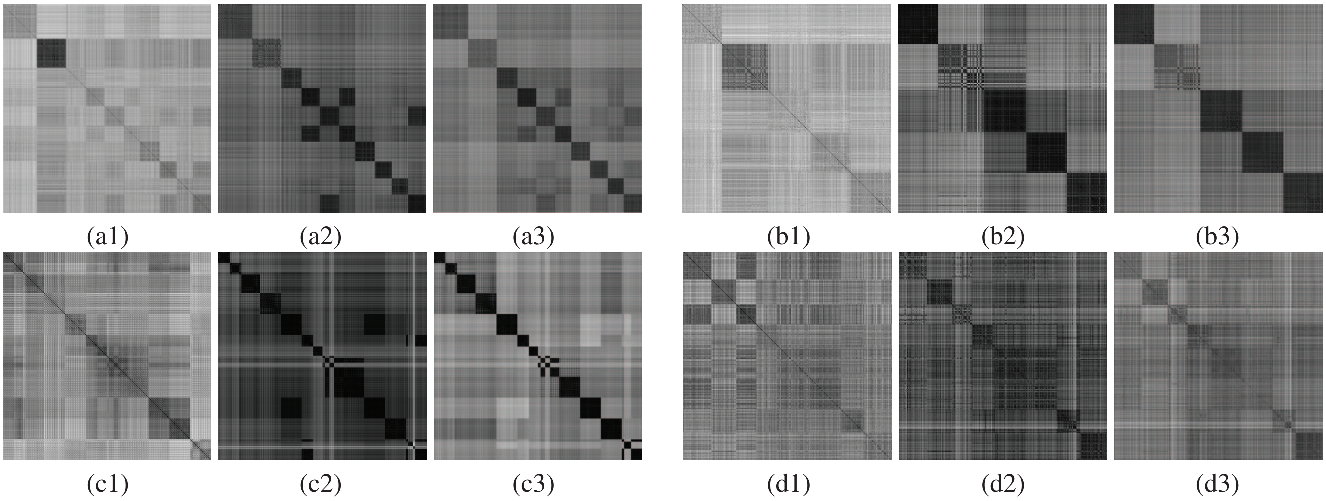
[4] http://www-stat.stanford.edu/ tibs/ElemStatLearn/

[5] http://yann.lecun.com/exdb/mnist/



(a) (b) (c) (d)

Figure 3. Constrained clustering results on the image data: clustering error vs. the number of pairwise constraints.



(a1) (a2) (a3) (b1) (b2) (b3)

(c1) (c2) (c3) (d1) (d2) (d3)

Figure 4. Distance matrices of the original data, the spectral embedding where NCuts obtains the best result, and the embedding of CCSKL where CCSKL gives the best result. For illustration purpose, the data are arranged such that objects within a cluster appear consecutively. The darker is a pixel, the smaller is the distance the pixel represents. (a1)–(a3) Results for USPS. (b1)–(b3) Results for MNIST. (c1)–(c3) Results for YaleB. (d1)–(d3) Results for Scene.

Table 2. Four UCI data sets used in the experiment.

	iris	wdbc	sonar	protein
# objects	150	569	208	116
# dimensions	4	30	60	20
# clusters	3	2	2	6

For example, the times taken by CCSKL, SL, SSKK, NCuts, and CCSR on USPS are about 50, 68, 69, 76, and 120 seconds, respectively, where CCSKL costs only 0.02 seconds to solve the QP problem while CCSR takes 52 seconds to solve the SDP problem [14]. All the algorithms run in MATLAB 7.6.0 (R2008a) on a PC with 3.4 GHz CPU and 4GB RAM.

5.4. UCI Data

We also conduct experiments on four data sets from UCI Machine Learning Repository[6]. UCI data are widely used to evaluate clustering and classification algorithms in machine learning. The four data sets we used in this experiment are described in Table 2. The results are shown in Fig. 5. Again we can see that CCSKL performs best in most of the cases.

6. Conclusions

An efficient framework CCSKL has been proposed for constrained clustering. The task is to combine the feature similarity and the pairwise constraints to derive a consistent partition of the data. The key idea is to train a spectral kernel to respect the pairwise constraints. The spectral kernel is ensured to preserve, to some extent, the smoothness structure of the graph. We formulate the spectral kernel learning framework as a convex QP problem, which is easy to solve optimally. Our approach has several attractive features:

1. It combines feature similarity and pairwise constraints via learning, in contrast to previous methods that modify the similarity matrix without any learning.
2. It is applicable to multi-class problems and handles both must-link and cannot-link constraints, in contrast to previous methods limited to two-class problems or only must-link constraints.

[6] http://archive.ics.uci.edu/ml.

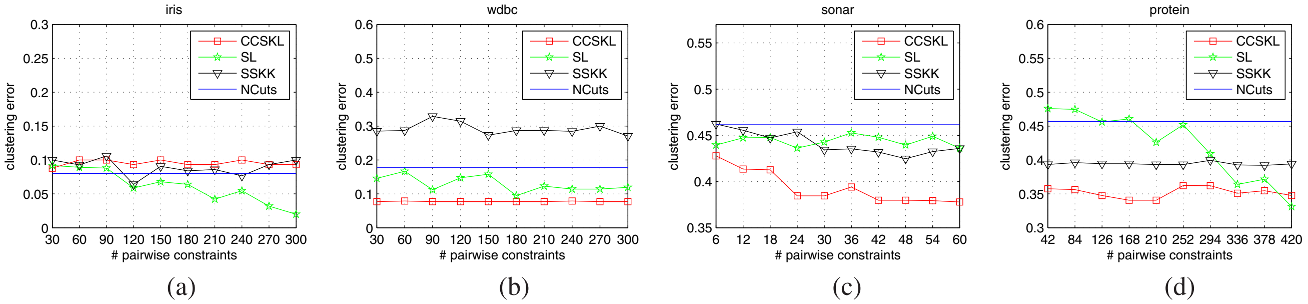


Figure 5. Constrained clustering results on the UCI data: clustering error vs. the number of pairwise constraints.

3. It can propagate pairwise constraints effectively.
4. It is scalable to large-scale problems with many clusters.
5. It can handle weighted pairwise constraints seamlessly.

Experimentally, CCSKL compares favorably with related methods on eight real data sets. Future work should consider automatic construction of graph and selection of m.

Acknowledgements

This work was supported by two grants from the Research Grants Council of the Hong Kong SAR, China (Project No. CUHK 414306 and 415408).

References

[1] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[2] O. Chapelle, J. Weston, and B. Scholkopf. Cluster kernels for semi-supervised learning. In *NIPS*, 2003.

[3] F. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997.

[4] T. Coleman, J. Saunderson, and A. Wirth. Spectral clustering with inconsistent advice. In *ICML*, pages 152–159, 2008.

[5] A. Frome, Y. Singer, F. Sha, and J. Malik. Learning globallyconsistent local distance functions for shape-based image retrieval and classification. In *ICCV*, 2007.

[6] A. Georghiades, P. Belhumeur, and D. Kriegman. From few to many: illumination cone models for face recognitionunder variable lighting and pose. *IEEE Trans. PAMI*, 23(6):643–660, 2001.

[7] G. Golub and C. Van Loan. *Matrix computations*. 1996.

[8] S. Kamvar, D. Klein, and C. Manning. Spectral learning. In *IJCAI*, pages 561–566, 2003.

[9] D. Klein, S. Kamvar, and C. Manning. From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. In *ICML*, pages 307–314, 2002.

[10] R. Kondor and J. Lafferty. Diffusion kernels on graphs and other discrete structures. In *ICML*, pages 315–322, 2002.

[11] B. Kulis, S. Basu, I. Dhillon, and R. Mooney. Semi-supervised graph clustering: A kernel approach. In *ICML*, pages 457–464, 2005.

[12] Z. Li, J. Liu, S. Chen, and X. Tang. Noise robust spectral clustering. In *ICCV*, 2007.

[13] Z. Li, J. Liu, and X. Tang. Pairwise constraint propagation by semidefinite programming for semi-supervised classification. In *ICML*, 2008.

[14] Z. Li, J. Liu, and X. Tang. Constrained clustering via spectral regularization. In *CVPR*, 2009.

[15] Z. Lu and M. Á. Carreira-Perpiñán. Constrained Spectral Clustering through Affinity Propagation. In *CVPR*, 2008.

[16] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, pages 849–856, 2001.

[17] A. Oliva and A. Torralba. Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001.

[18] B. Schölkopf and A. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT Press, 2002.

[19] N. Shental, A. Bar-Hillel, T. Hertz, and D. Weinshall. Computing Gaussian mixture models with EM using equivalence constraints. In *NIPS*, volume 16, pages 465–472, 2004.

[20] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. PAMI*, 22(8):888–905, 2000.

[21] A. Smola and R. Kondor. Kernels and regularization on graphs. In *COLT*, 2003.

[22] K. Wagstaff and C. Cardie. Clustering with instance-level constraints. In *ICML*, pages 1103–1110, 2000.

[23] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl. Constrained k-means clustering with background knowledge. In *ICML*, pages 577–584, 2001.

[24] E. Xing, A. Ng, M. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In *NIPS*, pages 505–512, 2003.

[25] L. Xu, W. Li, and S. D. Fast normalized cut with linear constraints. In *CVPR*, 2009.

[26] S. Yu and J. Shi. Segmentation Given Partial Grouping Constraints. *IEEE Trans. PAMI*, pages 173–180, 2004.

[27] T. Zhang and R. Ando. Analysis of spectral kernel design based semi-supervised learning. *NIPS*, 18:1601, 2006.

[28] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *NIPS*, 2004.

[29] X. Zhu, J. Kandola, Z. Ghahramani, and J. Lafferty. Nonparametric transforms of graph kernels for semi-supervised learning. In *NIPS*, pages 1641–1648, 2005.