Noise Robust Spectral Clustering

Zhenguo Li[1], Jianzhuang Liu[1], Shifeng Chen[1], and Xiaoou Tang[1,2]
[1]Dept. of Information Engineering
The Chinese University of Hong Kong
{zgli5,jzliu,sfchen5}@ie.cuhk.edu.hk
[2]Microsoft Research Asia
Beijing, China
xitang@microsoft.com

Abstract

This paper aims to introduce the robustness against noise into the spectral clustering algorithm. First, we propose a warping model to map the data into a new space on the basis of regularization. During the warping, each point spreads smoothly its spatial information to other points. After the warping, empirical studies show that the clusters become relatively compact and well separated, including the noise cluster that is formed by the noise points. In this new space, the number of clusters can be estimated by eigenvalue analysis. We further apply the spectral mapping to the data to obtain a low-dimensional data representation. Finally, the K-means algorithm is used to perform clustering. The proposed method is superior to previous spectral clustering methods in that (i) it is robust against noise because the noise points are grouped into one new cluster; (ii) the number of clusters and the parameters of the algorithm are determined automatically. Experimental results on synthetic and real data have demonstrated this superiority.

1. Introduction

Clustering is an important research topic in computer vision and machine learning. Many algorithms have been developed with different motivations. Recently, spectral clustering has been proposed [9] [8] and attracted much attention [6], [15], [16], [1], [4], [17], [7]. In spectral clustering, one makes use of the eigenvectors of the normalized graph Laplacian to reveal the cluster structures of the data. Two notable methods came from Shi and Malik [9] and Ng et al. [8]. Impressively good results have been demonstrated in spectral clustering and it is considered as a most promising clustering technique [14]. In addition, many theoretical studies have been done on spectral clustering with relations to random walks, normalized cut, matrix perturbation theory, and diffusion map. However, several main issues remain to be solved in the framework of spectral clustering: (i) how to choose the scaling parameter automatically; (ii) how to find the number of clusters automatically; (iii) how to be robust against noise, or furthermore, how to recognize the noise so as to remove them; and (iv) how to incorporate available side information to improve the clustering performance.

Most recently, Zelnik-Manor and Perona [17] improved Ng et al.'s spectral clustering algorithm by addressing the first two issues. They used a *local scale* scheme to compute the affinity matrix and exploited the structure of the eigenvectors of the normalized graph Laplacian to infer the number of clusters. This method works well on a noise-free data set, but it has two main disadvantages. First, the estimation of the number of clusters may get trapped into local minima, giving wrong results. Second, it fails to obtain satisfactory clustering results when significant noise is present even with the correct number of clusters manually set.

In this paper, we focus on issues (ii) and (iii). Specifically, we consider the problem of robust clustering where the data may contain significant noise, and the number of clusters is unknown. We also present a simple strategy to determine the parameters including the scaling parameters automatically.

Real-world data often contain noise. This poses challenges to the clustering community: (i) how to obtain correct clustering from noisy data; or further, (ii) how to obtain correct clustering from noisy data and remove the noise. While the former considers only data partition regardless of the noise, the latter performs both clustering and denoising simultaneously. In other words, the former outputs noisy clusters and the latter gives noise-free clusters. Obviously, the latter is more challenging and of more practical significance. Indeed, most of the existing clustering algorithms including the spectral clustering often fail even in the former problem, let alone the latter, due to the difficulty that the real data distribution is masked and distorted by noise.

In this paper, we consider the latter clustering problem, and at the same time, determine the number of clusters and the parameters of the algorithm automatically. Our work is motivated by the transductive inference in [19], [20], and [18] on semi-supervised learning and the work in [10] and

978-1-4244-1631-8/07/$25.00 ©2007 IEEE

[21] on graph kernels, and built upon the spectral clustering in [8]. We find that the main reason leading to the failure of the spectral clustering on noisy data is that the block structure of the affinity matrix is destroyed by noise. So an intuitive and natural way to this tough clustering problem is to reshape the noisy data set so that the block structure of the new affinity matrix can be recovered, followed by the spectral clustering. We propose a data warping model to map the data into a new space, where the number of clusters is estimated and then the spectral clustering is applied.

2. Spectral Clustering and Graph Kernels

In this section, we briefly review the spectral clustering in [8] and the work in [10] and [21] on graph kernels. Let $G = (V, W)$ be an undirected weighted graph with the set of nodes V consisting of the given data points $\{\mathbf{x}_i \in \mathbb{R}^d | i = 1, 2, \cdots, n\}$, and $W = [w_{ij}]_{n\times n}$ a symmetric matrix with w_{ij} being the weight (affinity) of the edge connecting nodes $\mathbf{x}_i$ and $\mathbf{x}_j$. Commonly, the affinities w_{ij} are chosen as

$$w_{ij} = \begin{cases} e^{-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma^2} & i \neq j \\ 0 & i = j \end{cases}, \quad (1)$$

where σ is a scaling parameter. The graph Laplacian L of G is defined as $L = I - W$, where I is the identity matrix, and the normalized graph Laplacian $\bar{L}$ of G is defined as

$$\bar{L} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2}, \quad (2)$$

where $D = [d_{ij}]_{n\times n}$ is a diagonal matrix with $d_{ii} = \sum_j w_{ij}$. W is called the affinity matrix, and $\bar{W} = D^{-1/2} W D^{-1/2}$ the normalized affinity matrix.

Let $\{(\boldsymbol{\nu}_i, \lambda_i) | i = 1, 2, \cdots, n\}$ be the eigensystem of $\bar{L}$ where $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$. Let k be the number of clusters and $X = [\boldsymbol{\nu}_1, \boldsymbol{\nu}_2, \cdots, \boldsymbol{\nu}_k]$. Each row of $X = [\nu_{ij}]_{n\times k}$ is further normalized to have unit length, resulting in a new matrix $\bar{X} = [\bar{\nu}_{ij}]_{n\times k}$ with $\bar{\nu}_{ij} = \nu_{ij}/(\sum_{j=1}^k \nu_{ij}^2)^{1/2}$. Then the low-dimensional data representation in [8] can be viewed as a mapping

$$\varphi_{\bar{L}} : V \longrightarrow \mathbb{R}^d, \ \ \mathbf{x}_i \mapsto \bar{X}(i, \cdot)^T, \quad (3)$$

where $\bar{X}(i, \cdot)$ denotes the i^{th} row vector of $\bar{X}$. We call $\varphi_{\bar{L}}$ the *spectral mapping* or *spectral embedding* with respect to $\bar{L}$. The spectral clustering then applies the K-means to this representation. The success of the spectral clustering is due to the fact that after the spectral mapping, the block structure of the data is amplified to a great extent.

Let $L^2(V)$ denote the Hilbert space of real-valued functions $f : V \to \mathbb{R}$, which assigns a real value $f(\mathbf{x}_i)$ to a node $\mathbf{x}_i \in V$, endowed with the standard inner product. A function f can also be represented as a vector $\mathbf{f} = (f_1, f_2, \cdots, f_n)^T$ where $f_i = f(\mathbf{x}_i)$, $i = 1, 2, \cdots, n$. The normalized graph Laplacian $\bar{L}$ can be naturally viewed as a linear operator $\bar{L} : L^2(V) \to L^2(V)$ with $(\bar{L}\mathbf{f})_i = f_i - \sum_j \frac{w_{ij}}{\sqrt{d_{ii} d_{jj}}} f_j$ and

$$< \mathbf{f}, \bar{L}\mathbf{f} >= \frac{1}{2} \sum_{i,j=1}^n w_{ij} (\frac{f_i}{\sqrt{d_{ii}}} - \frac{f_j}{\sqrt{d_{jj}}})^2 \geq 0, \quad (4)$$

where the inequality holds because the entries of W are non-negative. In (4), the semi-norm $< \mathbf{f}, \bar{L}\mathbf{f} >$ on $L^2(V)$ induced by $\bar{L}$ penalizes the large change of the function between two nodes linked with a large weight, implying that it may serve as a smoothing regularizer.

The work in [10] and [21] generalizes $\bar{L}$ to a family of regularization operators:

$$\bar{L}_r = \sum_{i=1}^n r(\lambda_i) \boldsymbol{\nu}_i \boldsymbol{\nu}_i^T, \quad (5)$$

where $r(\cdot)$ is a real-valued function and should be chosen such that (i) $r(\cdot)$ is non-negative in $[0, 2]$ since $\bar{L}_r$ is required to be positive semi-definite; (ii) $r(\cdot)$ is non-decreasing in $[0, 2]$ since rather uneven functions on the graph should be penalized strongly.

Let $K_r = \bar{L}_r^{-1}$ where $\bar{L}_r^{-1}$ denotes the inverse of $\bar{L}_r$ if it is non-singular or the pseudo-inverse of $\bar{L}_r$ if it is singular. Then K_r is exactly the reproducing kernel of a Hilbert space consisting of all the images of the functions in $L^2(V)$ under the linear operator $\bar{L}_r$, endowed with the inner product $< \mathbf{f}, \mathbf{g} >_{K_r} := < \mathbf{f}, \bar{L}_r \mathbf{g} >, \mathbf{f}, \mathbf{g} \in L^2(V)$ [10]. K_r is called a graph kernel with respect to $r(\cdot)$.

This construction of graph kernels includes some previously well-known kernels as special cases. For instance, when $r(\lambda) = 1 + t^2\lambda$, $K_r = (I + t^2\bar{L})^{-1}$, which is the regularized Laplacian kernel [19]; when $r(\lambda) = e^{\frac{t^2}{2}\lambda}$, $K_r = e^{-\frac{t^2}{2}\bar{L}}$, which is the diffusion kernel [5]; when $r(\lambda) = (a - \lambda)^{-p}$ with $a > 2$, $K_r = (aI - \bar{L})^p$, which is the p-step random walk kernel [2]. If $r(\lambda) = \lambda$, $\bar{L}_r$ turns out to be the normalized graph Laplacian $\bar{L}$ and $K_r = L^{-1}$.

The algorithmic flows of the spectral clustering in [8] is illustrated in Fig. 1 where it consists of modules 1, 2, 3, 7, and 8. Our work improves the spectral clustering by removing the module 3 and introducing modules 4, 5, and 6, which are presented in Sections 3, 4, and 5, respectively. In module 4, a data warping model is proposed to map the data into a new space where a new data graph is built (module 5) and the number of clusters is estimated (module 6).

3. Data Warping

Given a data set of n points $V = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}, \mathbf{x}_i \in \mathbb{R}^d$, $i = 1, 2, \cdots, n$, we want to learn a mapping f to map V into $\mathbb{R}^n$,

$$f : \mathbf{x}_i \mapsto \mathbf{y}_i \in \mathbb{R}^n, \ \ i = 1, 2, \cdots, n, \quad (6)$$

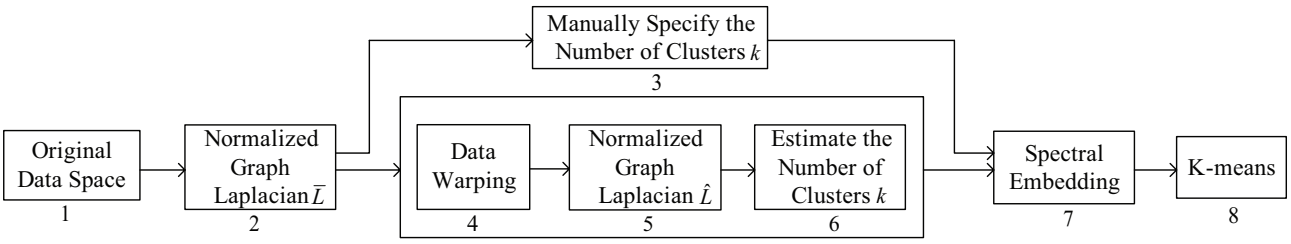

Figure 1. The algorithmic flows of the spectral clustering in [8] and our algorithm. The spectral clustering consists of modules 1, 2, 3, 7, and 8, while our algorithm comprises modules 1, 2, 4, 5, 6, 7, and 8.

such that *each cluster of an arbitrary shape is mapped as a relatively compact cluster, the noise points are mapped also to form a relatively compact cluster, and different clusters become well separated.* For easy understanding, we also consider $\mathbf{y}_i$ and $\mathbf{y}_j$ are connected with a weight that equals w_{ij}, the weight of the edge connecting $\mathbf{x}_i$ and $\mathbf{x}_j$ in G, $i,\ j = 1, 2, \cdots, n$. To this end, we consider the following regularization framework:

$$\Omega(Y) = \|Y - I\|_F^2 + \alpha \mathrm{tr}(Y^T K_r^{-1} Y), \tag{7}$$

where I is the identity matrix, $\|\cdot\|_F$ denotes the Frobenious norm of a matrix[1], $\mathrm{tr}(\cdot)$ denotes the trace of a matrix, α is a positive regularization parameter controlling the trade-off between the two terms, K_r is some graph kernel as discussed in the last section, K_r^{-1} denotes the inverse of K_r if it is non-singular or the pseudo-inverse of K_r if it is singular, and $Y = [\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_n]^T$ which is a representation of the data after the warping that we need to find. The minimization of $\Omega(Y)$ to obtain Y can be considered as a transductive process, called the transductive warping:

$$Y^* = \arg\min_Y \Omega(Y). \tag{8}$$

Next we give insight into this regularization functional.

Let $\mathbf{e}_i \in \mathbb{R}^n$ be a vector with the i^{th} entry being 1 and 0's elsewhere, and $\mathbf{z}_i$ be the i^{th} column vector of Y. Then (7) becomes

$$\Omega(Y) = \sum_{i=1}^{n} \left(\|\mathbf{z}_i - \mathbf{e}_i\|^2 + \alpha < \mathbf{z}_i, \bar{L}_r \mathbf{z}_i > \right). \tag{9}$$

Therefore, by the construction of $\bar{L}_r$ as described in (5), the second term in (7) actually acts as a regularizer that encourages smoothness on the $\mathbf{y}_i$'s that are strongly connected (i.e., connected with larger weights), which is explained as follows. In fact, $\mathbf{z}_i = (y_{1i}, y_{2i}, \cdots, y_{ni})^T$ consists of the i^{th} components (i.e., the i^{th} coordinates in $\mathbb{R}^n$) of $\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_n$. If the weight w_{kl} between $\mathbf{y}_k$ and $\mathbf{y}_l$ is large, y_{ki} and y_{li} will be forced close to each other, which is similar to the case in (4). In other words, $\mathbf{z}_i$ should not change too much between y_{ki} and y_{li} if $\mathbf{y}_k$ and $\mathbf{y}_l$ are strongly connected. In the ideal case, the points in the same cluster in $\mathbb{R}^d$ are warped into a single point in $\mathbb{R}^n$, and different clusters are warped into different points.

Based on (9), the transductive warping can be interpreted in more detail as follows. First, $\mathbf{x}_i$ is mapped to a vector $\mathbf{y}_i = I(i,\cdot)^T$ in $\mathbb{R}^n$ (see the first term in (7)), where $I(i,\cdot)$ denotes the i^{th} row vector of the identity matrix I, and $I(i,\cdot)^T$, $i = 1, \cdots, n$, form the canonical coordinate system of $\mathbb{R}^n$. Then the i^{th} component of $\mathbf{y}_i$, which is initially 1, is *spreading smoothly* to the i^{th} components of all $\mathbf{y}_j,\ j = 1, 2, \cdots, n,\ j \neq i$, which are initially 0's. In this sense, the two terms in (7) are called the fitting term and the smoothness term, respectively. After the transductive warping, the second term ensures that the points sharing the same cluster with the i^{th} point will have relatively large and similar values for their i^{th} components, while other points will have their i^{th} components close to zero, for $i = 1, 2, \cdots, n$. Consequently, these clusters become compact and well separated from each other, and they are not close to the origin in $\mathbb{R}^n$ compared with the *noise cluster* that is formed by the noise points.

Due to randomly distributed nature of noise points in $\mathbb{R}^d$, the majority of the edges connected to each noise point in G are weak (i.e., the majority of the weights of these edges are small). After the transductive warping, most of the coordinates of a noise point in $\mathbb{R}^n$ are thus small, and the noise points are close to the origin, forming a compact cluster. We call it the noise cluster, and in contrast, we call other clusters the ordinary clusters.

It is easy to show that the functional (7) is strictly convex, implying that only one minimum exists. Now we find this minimum in the following. Taking the derivative of $\Omega(Y)$ with respect to Y and setting it to zero yields,

$$\frac{\partial \Omega(Y)}{\partial Y} = 2(Y - I) + 2\alpha K_r^{-1} Y = 0, \tag{10}$$

which results in $(I + \alpha K_r^{-1})Y = I$. Since $I + \alpha K_r^{-1}$ is nonsingular, we have the minimum of $\Omega(Y)$:

$$Y^* = (I + \alpha K_r^{-1})^{-1}. \tag{11}$$

Then, the transductive warping actually results in this mapping:

$$\phi_{K_r} :\ V \to \mathbb{R}^n,\ \ \mathbf{x}_i \mapsto \bar{Y}^*(i,\cdot)^T, \tag{12}$$

[1]The Frobenious norm of a matrix $Q = [q_{ij}]$ is $\|Q\|_F^2 = \sum_{ij} q_{ij}^2$.

where $\bar{Y}^*$ is a matrix obtained from Y^* by scaling linearly the features in each dimension (i.e., each column of Y^*) onto $[0, 1]$, and $\bar{Y}^*(i, \cdot)$ denotes the i^{th} row vector of $\bar{Y}^*$. Comparing (12) and (3), we can see the difference between ϕ_{K_r} and $\varphi_{\bar{L}}$. The spectral mapping maps $\mathbf{x}_i$ to a space of dimension equal to the number of clusters, while the transductive warping maps $\mathbf{x}_i$ to a space of dimension equal to the number of points in the data set. Unlike the former, the latter does not need to know the number of clusters in advance. Instead, this number of clusters is found in $\mathbb{R}^n$, which will be discussed in Section 5.

4. Spectral Embedding

Since a low-dimensional representation of data are important in many applications, we further employ the spectral mapping to map the data from $\mathbb{R}^n$ to a low-dimensional space of dimension equal to the number of the clusters (including the noise cluster). Finding this number will be addressed in the next section. Assuming this number is known and denoted by k, next we derive this spectral mapping.

Let $\hat{L}$ be the normalized graph Laplacian corresponding to the data after warping in $\mathbb{R}^n$. Let $\{(\boldsymbol{\mu}_i, \xi_i) \mid i = 1, 2, \cdots, n\}$ be the eigensystem of $\hat{L}$ where $\xi_1 \leq \xi_2 \leq \cdots \leq \xi_n$ and $F = [\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \cdots, \boldsymbol{\mu}_k]$. Each row of $F = [\mu_{ij}]_{n\times k}$ is normalized to have unit length, resulting in a new matrix $\bar{F} = [\bar{\mu}_{ij}]_{n\times k}$ with $\bar{\mu}_{ij} = \mu_{ij}/(\sum_{j=1}^{k} \mu_{ij}^2)^{1/2}$. Then the spectral mapping with respect to $\hat{L}$ is given by

$$\varphi_{\hat{L}} : \hat{V} \longrightarrow \mathbb{R}^k, \;\; \mathbf{y}_i \mapsto \bar{F}(i, \cdot)^T, \tag{13}$$

where $\bar{F}(i, \cdot)$ denotes the i^{th} row vector of $\bar{F}$.

By combining the transductive warping (12) and the spectral mapping (13), we have the following *tight transductive warping*:

$$\begin{aligned} \Phi_{K_r} \triangleq \varphi_{\hat{L}} \circ \phi_{K_r} : \; & S \longrightarrow \hat{V} \longrightarrow \mathbb{R}^k \\ & \mathbf{x}_i \longmapsto \mathbf{y}_i \longmapsto \bar{F}(i, \cdot)^T. \end{aligned} \tag{14}$$

5. Finding the Number of Clusters

This section explores how to find the number of clusters (including the noise cluster) from the data after warping. We achieve this goal by analyzing the eigenvalues of the normalized graph Laplacian $\hat{L}$. First, we define two terms in Definition 1.

Definition 1. *An affinity matrix $W = \{w_{ij}\}$ of a data set is called ideal if (i) $w_{ij} = 1$, $i \neq j$, if point i and point j are in the same cluster; (ii) $w_{ij} = 0$ if point i and point j are in different clusters; and (iii) $w_{ii} = 0$. A normalized graph Laplacian corresponding to an ideal affinity matrix is also called ideal.*

In what follows, we first consider the eigenvalue distribution of an ideal normalized graph Laplacian, then extend to the general case using the matrix perturbation theory. The main result for the ideal case is stated in the following theorem.

Theorem 1. *Let $\hat{L}$ be an ideal normalized graph Laplacian of a data set which consists of k clusters of size $n_1, n_2, \cdots, n_k$, respectively. Then $\hat{L}$ has eigenvalues $0, \frac{n_1}{n_1-1}, \frac{n_2}{n_2-1}, \cdots, \frac{n_k}{n_k-1}$ of algebraic multiplicities $k, n_1 - 1, n_2 - 1, \cdots, n_k - 1$, respectively.*

Before proving this theorem, we presents two lemmas.

Lemma 1. *Let $\hat{L}$ be the ideal normalized graph Laplacian of a data set consisting of only one cluster of size m. Then $\hat{L}$ has eigenvalue 0, and eigenvalue $\frac{m}{m-1}$ of algebraic multiplicity $m - 1$.*

Proof. Let W be the ideal affinity matrix of this data set. Since there is only one cluster in it, by Definition 1, the elements of W are all 1's except the diagonal elements that are 0's. So W can be expressed as $W = E - I$ where E is a matrix whose elements are all 1's, and I is the identity matrix. It can be shown that E has eigenvalue 0 of algebraic multiplicity $m - 1$, and eigenvalue m. Therefore W has eigenvalue -1 of algebraic multiplicity $m - 1$, and eigenvalue $m-1$. Since $\hat{L} = I - D^{-1/2} W D^{-1/2} = I - \frac{1}{m-1} W$, $\hat{L}$ has eigenvalue 0, and eigenvalue $\frac{m}{m-1}$ of algebraic multiplicity $m - 1$. □

Lemma 2. *Let $\hat{L}$ be an ideal normalized graph Laplacian of a data set consisting of k clusters of size $n_1, n_2, \cdots, n_k$ respectively and ordered in such a way that all the points belonging to the first cluster appear first, all the points belonging to the second cluster appear second, etc. Then $\hat{L}$ has eigenvalues $0, \frac{n_1}{n_1-1}, \frac{n_2}{n_2-1}, \cdots, \frac{n_k}{n_k-1}$ of multiplicities $k, n_1 - 1, n_2 - 1, \cdots, n_k - 1$, respectively.*

Proof. Note that $\hat{L}$ is block diagonal

$$\hat{L} = \text{diag}[\hat{L}_1, \hat{L}_2, \cdots, \hat{L}_k] \tag{15}$$

whose i^{th} block $\hat{L}_i$ is the ideal normalized graph Laplacian of cluster i. From Lemma 1, this lemma follows immediately from the fact that the eigenvalues of a block diagonal matrix are simply the set of eigenvalues of the individual blocks. □

Proof of Theorem 1. Given a data set, clearly there exists a permutation f to arrange the data in the same order as in Lemma 2. Let $\hat{L}$ and $\hat{L}'$ be the normalized graph Laplacians of the original and the one after the permutation, respectively. Then from the group theory, $\hat{L}$ and $\hat{L}'$ are related by

$$\hat{L}' = P \hat{L} P^{-1} \tag{16}$$

where P is the permutation matrix associated with the permutation f[2]. So $\hat{L}$ and $\hat{L}'$ are similar and thus share the same eigenvalues. □

Three important observations on the eigenvalue distribution of an ideal normalized graph Laplacian can be obtained from Theorem 1: (i) the smallest eigenvalue is 0 whose algebraic multiplicity equals the number of clusters k, and the $(k+1)^{th}$ smallest eigenvalue is strictly larger than 1; (ii) the largest gap (strictly larger than 1) is located between the k^{th} and $(k+1)^{th}$ smallest eigenvalues; (iii) the second largest gap is much less than the largest one.

Generally, practical cases are different from this ideal one. However, since the clusters after warping are relatively compact and well separated, we can choose a suitable scaling parameter β such that $\hat{w}_{ij}$ is close to 1 if $\mathbf{y}_i$ and $\mathbf{y}_j$ are in the same cluster, $\hat{w}_{ij}$ is close to 0 if $\mathbf{y}_i$ and $\mathbf{y}_j$ are in different clusters, and set $\hat{w}_{ii} = 0$. In this sense, the practical normalized graph Laplacian $\hat{L}$ can be considered as a perturbation of an ideal normalized graph Laplacian. The following theorem from the matrix perturbation theory ensures that the eigenvalues of $\hat{L}$ remains close to the eigenvalues of the ideal normalized graph Laplacian as long as the perturbation is sufficiently small [11].

Theorem 2. *Let A and A' be two real symmetric matrices of size $n \times n$ with eigenvalues $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ and $\lambda'_1 \leq \lambda'_2 \leq \cdots \leq \lambda'_n$, respectively. Then*

$$\sqrt{\sum_{i=1}^{n}(\lambda'_i - \lambda_i)^2} \leq \|A' - A\|_F, \tag{17}$$

where $\|\cdot\|_F$ is the Frobenius norm of a matrix.

According to these two theorems, it is expected that there is a significant and largest gap between the k^{th} and $(k+1)^{th}$ smallest eigenvalues of $\hat{L}$ provided that suitable parameters including the scaling parameter β are selected. This is also confirmed by our experiments in Section 7, where we describe a simple strategy to search for such parameters that maximize this gap.

Based on the above discussion, we propose to estimate the number of clusters k^* by

$$k^* = \arg\max_k \{\lambda_{k+1}(\hat{L}) - \lambda_k(\hat{L})\}, \tag{18}$$

where $\lambda_i(\hat{L})$ denotes the i^{th} smallest eigenvalue of $\hat{L}$.

[2]A permutation matrix is a $(0,1)$-matrix that has exactly one entry 1 in each row and each column and 0's elsewhere. Permutation matrices are the matrix representation of permutations. A permutation matrix is always invertible.

6. The Clustering Algorithm

Based on previous analysis, we develop an algorithm for robust spectral clustering which is listed in Algorithm 1. In step 3, we simply choose $K_r = \bar{L}^{-1}$ for all our experiments reported in Section 7. Three parameters need to be determined in this algorithm: the two scaling parameters σ and β (steps 1 and 5), and the regularization parameter α (step 4). A scheme for choosing them is presented in Section 7. The main computation in Algorithm 1 is in steps 4 and 7, which involve taking the matrix inverse (see (11)) and the eigenvalue decomposition of $\hat{L}$, respectively. So the time complexity of Algorithm 1 is $O(n^3)$, the same as that of the spectral clustering in [8].

Algorithm 1 Robust Spectral Clustering Algorithm

1: Construct the affinity matrix $W = [w_{ij}]_{n\times n}$ with $w_{ij} = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma^2)$ if $i \neq j$ and $w_{ii} = 0$.
2: Construct the diagonal matrix D whose $(i,i)^{th}$ entry is the sum of the entries in W's i^{th} row, and form the normalized graph Laplacian $\bar{L} = I - D^{-1/2}WD^{-1/2}$.
3: Choose a graph kernel K_r.
4: Compute the transductive warping ϕ_{K_r}.
5: Construct the affinity matrix $\hat{W} = [\hat{w}_{ij}]_{n\times n}$ with $\hat{w}_{ij} = \exp(-\|\phi_{K_r}(\mathbf{x}_i) - \phi_{K_r}(\mathbf{x}_j)\|^2/2\beta^2)$ if $i \neq j$ and $\hat{w}_{ii} = 0$.
6: Construct the diagonal matrix $\hat{D}$ whose $(i,i)^{th}$ entry is the sum of the entries in $\hat{W}$'s i^{th} row, and form the normalized graph Laplacian $\hat{L} = I - \hat{D}^{-1/2}\hat{W}\hat{D}^{-1/2}$.
7: Compute all the eigenvalues of $\hat{L}$.
8: Find the number of clusters k by locating the largest gap of the eigenvalues of $\hat{L}$.
9: Compute the tight transductive warping Φ.
10: Perform clustering on $\{\Phi(\mathbf{x}_i) \mid i = 1, 2, \cdots, n\}$ by the K-means algorithm.
11: Assign the point $\mathbf{x}_i$ to cluster j if $\Phi(\mathbf{x}_i)$ is in cluster j, $i = 1, 2, \cdots, n$.

7. Experimental Results

In this section, we compare the proposed robust spectral clustering (RSC) algorithm with two closely related clustering algorithms, the spectral clustering (SC) algorithm in [8] as well as the self-tuning spectral clustering (STSC) algorithm in [17] on a number of synthetic and real data sets.

7.1. Performance Evaluation

To evaluate the performances of different algorithms, we compare the clustering results with the ground true data labels. We adopt the *normalized mutual information* (NMI) as the performance measure since it is widely used for the evaluation of clustering algorithms [12]. For two random

variable $\mathbf{X}$ and $\mathbf{Y}$, the NMI is defined as:

$$NMI(\mathbf{X},\mathbf{Y}) = \frac{I(\mathbf{X},\mathbf{Y})}{\sqrt{H(\mathbf{X})H(\mathbf{Y})}} \tag{19}$$

where $I(\mathbf{X},\mathbf{Y})$ is the mutual information between $\mathbf{X}$ and $\mathbf{Y}$, and $H(\mathbf{X})$ and $H(\mathbf{Y})$ are the entropies of $\mathbf{X}$ and $\mathbf{Y}$, respectively. Note that $0 \leq NMI(\mathbf{X},\mathbf{Y}) \leq 1$ and $NMI(\mathbf{X},\mathbf{Y}) = 1$ when $\mathbf{X} = \mathbf{Y}$.

Let $\{S_1, S_2, \cdots, S_c\}$ be the true classes of a data set of size n, where c is the true number of clusters, and the number of points in S_i is $|S_i| = n_i$, $i = 1, 2, \cdots, c$. Let $\{S'_1, S'_2, \cdots, S'_k\}$ be a clustering result of this data set produced by an algorithm, where $|S'_i| = n'_i$, and k is the obtained number of clusters. Then the NMI of this clustering result can be explicitly expressed as

$$NMI = \frac{\sum_{i=1}^{c}\sum_{j=1}^{k} n_{ij}\log\left(\frac{nn_{ij}}{n_i n'_j}\right)}{\sqrt{(\sum_{i=1}^{c} n_i \log\frac{n_i}{n})(\sum_{j=1}^{k} n'_j \log\frac{n'_j}{n})}}, \tag{20}$$

where $n_{ij} = |S_i \cap S'_j|$. The larger the NMI, the better the clustering result. In the performance evaluation, we treat the noise points as a new cluster.

7.2. Parameter Selection

Three parameters in the RSC need to be determined: the two scaling parameters σ and β, and the regularization parameter α. According to the analysis in Section 5, a significant gap can be expected among the eigenvalues of $\hat{L}$. This motivates us to choose σ, β, and α to maximize this gap. Empirically, we find that the performance of the RSC is not sensitive to α as long as it is large enough to ensure the sufficiency of transduction. So we simply set it to 10000 for all the experiments reported here.

Besides, it is a common experience that the scaling parameter has a significant impact on the performance in spectral clustering. Although several schemes have been proposed to address this problem [13], it is still difficult to find a single one that can handle all types of cluster shapes and sizes, especially when noise is present. Since the scaling parameter actually plays a role as an effective neighborhood radius for each point in building the data graph, to maximize the gap among the eigenvalues of $\hat{L}$, we choose σ and β such that $2\sigma^2 \in \{16\bar{a}^2, 8\bar{a}^2, 4\bar{a}^2, \bar{a}^2, \frac{1}{4}\bar{a}^2, \frac{1}{8}\bar{a}^2\ \frac{1}{16}\bar{a}^2\}$ and $2\beta^2 \in \{16\bar{b}^2, 8\bar{b}^2, 4\bar{b}^2, \bar{b}^2, \frac{1}{4}\bar{b}^2, \frac{1}{8}\bar{b}^2\ \frac{1}{16}\bar{b}^2\}$, respectively, where $\bar{a}$ (or $\bar{b}$) is the average of the distances from each point $\mathbf{x}_i$ (or $\mathbf{y}_i$) to its k^{th} nearest neighbor, and k is set to 10 in the experiments, i.e.,

$$(\sigma^*, \beta^*) = \arg\max_{\sigma,\beta} g_{\hat{L}}(\sigma, \beta), \tag{21}$$

where $g_{\hat{L}}(\sigma,\beta)$ is the largest gap between two successive eigenvalues of $\hat{L}$ corresponding to σ and β (the eigenvalues are ordered non-decreasingly).

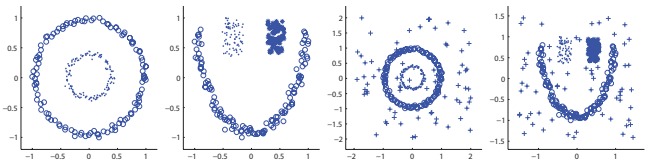

Figure 2. Four data sets with the ground truth clusters denoted by different markers. (a) Two Circles. (b) Face Contour. (c) Noisy Two Circles. (d) Noisy Face Contour. The data sets in (c) and (d) are formed by adding 30% uniformly distributed noise to the ones in (a) and (b), respectively.

Table 1. Clustering results on the synthetic data measured by NMI.

	Two Circles	Face Contour	Noisy Two Circles	Noisy Face Contour
RSC	1	1	0.9433	0.9785
SC	1	1	0.7419	0.6865
STSC	1	1	0.7238	0.6977

For the SC, two parameters need to be set: the scaling parameter and the number of clusters. The number of clusters is manually set to the correct number of clusters. For the scaling parameter, we try a number of values and pick the optimal one.

7.3. Clustering Results on Synthetic Data

We conduct experiments on four synthetic data sets depicted in Fig. 2. The first two (called Two Circles and Face Contour) comprise of 254 and 266 points, and the latter two (called Noisy Two Circles and Noisy Face Contour) are formed by adding 30% uniformly distributed noise to the formers, respectively. The quantitative results measured by the NMI are summarized in Table 1. From Table 1, we can see the three algorithms all give correct results for the two noise-free data sets, while the RSC demonstrates a substantial advantage over the SC and STSC on the two noisy sets. The clustering results for the noisy sets can also be visualized in Fig. 3. Note that although the STSC finds the correct number of clusters (=4) on the Noisy Face Contour, it gives unsatisfactory clustering result (see Fig. 3(f)).

To systematically test the sensitivities of different algorithms against noise, we form a series of noisy data sets by adding uniformly distributed noise of different levels $(5\%, 10\%, 20\%, \cdots, 100\%)$ to the Two Circles. Three of them are shown in Fig. 4. The numerical clustering results on these noisy data sets are depicted in Fig. 5(a), where we can see that the RSC is consistently superior to the SC and STSC on all these noisy data sets.

Figs. 6(a) and (b) show the distance matrices of the Noisy Two Circles (see Fig. 2(c)) before and after the warping, respectively, where the data are ordered (according to the ground truth cluster labels) in such a way that all the points belonging to the first cluster appear first, all the points belonging to the second cluster appear second, etc., and the

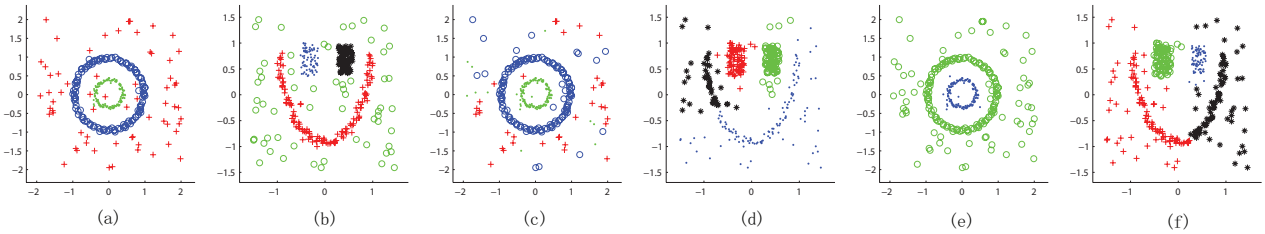

Figure 3. Clustering results on the synthetic data where different markers and colors denote different clusters. (a)(b) Results by the RSC. (c)(d) Results by the SC. (e)(f) Results by the STSC.

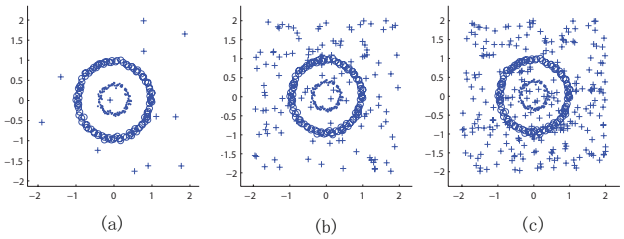

Figure 4. (a)(b)(c) Three noisy Two Circles formed by adding 5%, 50%, and 100% uniformly distributed noise to the Two Circles.

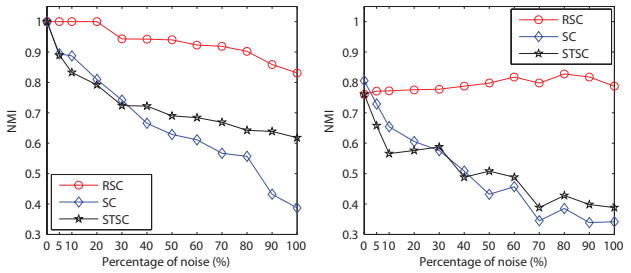

Figure 5. (a) Clustering results on a series of noisy Two Circles. (b) Clustering results on a series of noisy Iris.

noise points appear in the end. We can see from the distance matrix shown in Fig. 6(a) that the present noise significantly destroys the block structure of the distance matrix of the original data, while after the warping, the block structure of the distance matrix is recovered very well, meaning that after the warping, different clusters (including the noise cluster) in the new space $\mathbb{R}^n$ become relatively compact and well separated.

Fig. 6(c) shows the eigenvalues of $\hat{L}$ for the Noisy Two Circles corresponding to $\sigma = 0.0173$ and $\beta = 2.4354$ which are determined automatically. We can see that there is a significant gap, and clearly the number of clusters is identified as 3 according to (18).

7.4. Clustering Results on Real Data

In this section, we conduct experiments on the following 6 real data sets:

- Iris: from the UCI repository comprising 150 instances of 3 clusters where each instance is described by 4 attributes.
- Noisy Iris: formed by adding 30% uniformly distributed noise to the Iris.
- Pendigit: a subset of Pen-Based Handwriting data set from the UCI repository, comprising 2000 instances of 10 clusters, where each instance is described by 16 attributes.
- USPS-01: a subset of digits 0 and 1 from the USPS data set, comprising 1000 instances, 500 per digit, where each instance is described by 256 attributes.
- Noisy USPS-01: formed by adding 160 instances from digits 2–9, 20 per digit (treated as noise instances), to the USPS-01.
- USPS-3568: a subset of digits 3, 5, 6, and 8 from the USPS data set, comprising 2000 instances, 500 per digit.

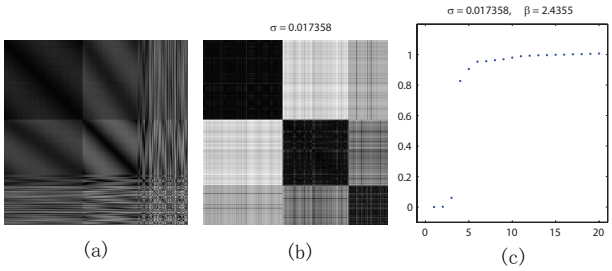

Figure 6. (a) Distance matrix for the Noisy Two Circles. (b) Distance matrix for the Noisy Two Circles after the warping where the scaling parameter σ is determined automatically. (c) The found largest gap of the eigenvalues of $\hat{L}$ for the Noisy Two Circles where the scaling parameters σ and β are determined automatically. The first 20 smallest eigenvalues are shown.

In the Noisy USPS-01 data set, each of digits 2–9 is with only 20 instances, a very small fraction compared with 500 digit 0 and 500 digit 1. So it is reasonable that we consider them as outliers (noise) in this data set.

The clustering results are summarized in Table 2. For the Iris data set, both RSC and STSC identify 2 clusters, because two of the three ground truth clusters touch each other [3]. We set 3 as the number of clusters for the SC, which results in a little larger NMI (0.8058) than the NMI (0.7612) obtained by the RSC and STSC. If we also set 3 as the number of clusters for the RSC and STSC, we will obtain NMIs 0.8135 and 0.5799, respectively. For the Noisy Iris, the RSC performs best although it identifies 3 clusters,

Table 2. Clustering results on the real data sets measured by NMI.

	Iris	Noisy Iris	Pendigit	USPS-01	Noisy USPS-01	USPS-3568
RSC	0.7612	0.7779	0.5899	1	1	0.8294
SC	0.8058	0.5759	0.4902	1	0.6234	0.7825
STSC	0.7612	0.5881	0.5107	1	0.7439	0.4869

still corresponding to a merging of two clusters. In this case, the STSC finds only 2 clusters. For the Pendigit, the RSC and STSC obtain respectively 12 and 9 clusters, different from the true number of clusters 10. In this case, the RSC performs a little better than the other two. All the three algorithms give correct partitions of the USPS-01, while only the RSC obtains correct results on Noisy USPS-01 where the STSC identifies 2 clusters. Although the STSC finds 4 clusters on the USPS-3568, its performance is unsatisfactory compared with those by the SC and RSC, where the RSC identifies 5 clusters. From these results, we can see that the proposed RSC gives comparable results on the data sets Iris, Pendigt, USPS-01, and USPS-3568 but performs much better on the noisy data sets, Noisy Iris and Noisy USPS-01.

We also form a series of noisy data sets by adding uniformly distributed noise of different levels $(5\%, 10\%, 20\%, \cdots, 100\%)$ to the Iris. The results on these noisy data sets are shown in Fig. 5(b). We can see that the RSC consistently outperforms the SC and STSC on all these noisy data sets.

8. Conclusions

We have developed a robust spectral clustering algorithm to cluster a data set where the clusters may be of different shapes and sizes, and noise, if present, is grouped into one new cluster. Most of the existing clustering algorithms including the spectral clustering often fail in this problem. Our algorithm is motivated by the transductive inference in semi-supervised learning and built upon the spectral clustering. The number of clusters and the parameters of the algorithm are estimated automatically. A data warping model is proposed to map the data into a new space. After the warping, each cluster becomes relatively compact and different clusters are well separated, including the noise cluster that is formed by the noise points. In this space, the number of clusters is estimated and the spectral clustering algorithm is applied. Experimental results on four synthetic data sets and six real data sets show that the proposed algorithm significantly outperforms the spectral clustering and the self-tuning spectral clustering on noisy data sets, and gives comparable results on noise-free data sets.

References

[1] M. Brand and K. Huang. A unifying theorem for spectral embedding and clustering. In *AISTATS*, 2003. 1

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

[3] A. Fred and A. Jain. Robust data clustering. In *CVPR*, pages 128–133, 2003. 7

[4] R. Kannan, S. Vempala, and A. Vetta. On clusterings: Good, bad and spectral. *J. ACM*, 51(3):497–515, 2004. 1

[5] R. Kondor and J. Lafferty. Diffusion kernels on graphs and other discrete input spaces. In *ICML*, 2002. 2

[6] M. Meila and J. Shi. Learning segmentation by random walks. In *NIPS*, pages 873–879, 2000. 1

[7] B. Nadler, S. Lafon, R. Coifman, and I. Kevrekidis. Diffusion maps, spectral clustering and eigenfunctions of fokker-planck operators. In *NIPS*, pages 955–962, 2006. 1

[8] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, pages 849–856, 2001. 1, 2, 3, 5

[9] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. PAMI*, 22(8):888–905, 2000. 1

[10] A. Smola and R. Kondor. Kernels and regularization on graphs. In *COLT*, 2003. 1, 2

[11] G. W. Stewart and J.-G. Sun. *Matrix Perturbation Theory*. Academic Press, San Diego, 1990. 5

[12] A. Strehl and J. Ghosh. Cluster ensembles- A knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3(3):583–617, 2003. 5

[13] H. Valizadegan and R. Jin. Generalized maximum margin clustering and unsupervised kernel learning. In *NIPS*, 2007. 6

[14] D. Verma and M. Meila. A comparison of spectral clustering algorithms. Technical Report 03-05-01, University of Washington Department of Computer Science, 2003. 1

[15] Y. Weiss. Segmentation using eigenvectors: A unifying view. In *ICCV*, pages 975–982, 1999. 1

[16] S. X. Yu and J. Shi. Multiclass spectral clustering. In *ICCV*, pages 313–319, 2003. 1

[17] L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. In *NIPS*, pages 1601–1608, 2005. 1, 5

[18] T. Zhang and R. Ando. Analysis of spectral kernel design based semi-supervised learning. In *NIPS*, volume 18, pages 1601–1608. 1

[19] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *NIPS*, 2004. 1, 2

[20] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, 2003. 1

[21] X. Zhu, J. Kandola, Z. Ghahramani, and J. Lafferty. Nonparametric transforms of graph kernels for semi-supervised learning. In *NIPS*, pages 1641–1648, 2005. 2