Clustering and Projected Clustering with Adaptive Neighbors

Feiping Nie
Department of Computer Science and Engineering
University of Texas, Arlington
Texas, USA
feipingnie@gmail.com

Xiaoqian Wang
Department of Computer Science and Engineering
University of Texas, Arlington
Texas, USA
xqwang1991@gmail.com

Heng Huang*
Department of Computer Science and Engineering
University of Texas, Arlington
Texas, USA
heng@uta.edu

ABSTRACT

Many clustering methods partition the data groups based on the input data similarity matrix. Thus, the clustering results highly depend on the data similarity learning. Because the similarity measurement and data clustering are often conducted in two separated steps, the learned data similarity may not be the optimal one for data clustering and lead to the suboptimal results. In this paper, we propose a novel clustering model to learn the data similarity matrix and clustering structure simultaneously. Our new model learns the data similarity matrix by assigning the adaptive and optimal neighbors for each data point based on the local distances. Meanwhile, the new rank constraint is imposed to the Laplacian matrix of the data similarity matrix, such that the connected components in the resulted similarity matrix are exactly equal to the cluster number. We derive an efficient algorithm to optimize the proposed challenging problem, and show the theoretical analysis on the connections between our method and the K-means clustering, and spectral clustering. We also further extend the new clustering model for the projected clustering to handle the high-dimensional data. Extensive empirical results on both synthetic data and real-world benchmark data sets show that our new clustering methods consistently outperforms the related clustering approaches.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications-Data Mining

General Terms

Algorithms

Keywords

Clustering; Block diagonal similarity matrix; Adaptive neighbors; Clustering with dimensionality reduction

1. INTRODUCTION

Clustering, which partitions the data points into different groups such that the objects in the same group have high similarity to each other, is one of the most fundamental topic in data mining. The clustering technique has been playing an outstanding role in data mining applications. In the past decades, many clustering algorithms have been proposed, such as hierarchical clustering [10], K-means clustering [13], spectral clustering [15], spectral embedded clustering [19], support vector clustering [1], maximum margin clustering [22], initialization independent clustering [18], multi-view clustering [21, 3, 4], *etc.*

Due to the efficiency and simpleness, the most popularly used clustering method is the K-means clustering algorithm, which aims to learn c cluster centroids that minimize the within cluster data distances. The spectral clustering method [15] does a low-dimension embedding of the affinity matrix between samples, followed by a K-means clustering in the low dimensional space. Because the data graph and manifold information are utilized in the clustering model, the graph based clustering methods (such as normalized cut [20], ratio cut [7]) usually show better clustering performance than K-means method. Such methods especially work well for a small number of clusters. More recently, the Nonnegative Matrix Factorization (NMF) has been used as the relaxation technique for clustering with excellent performance [12, 16]. Although the graph-based clustering methods have good performance, they partition the data based on the fixed data graph such that the clustering results are sensitive to the input affinity matrix.

In this paper, we propose to solve the clustering problem from a new point of view and learn the data similarity matrix by assigning the adaptive and optimal neighbors for each data point based on the local connectivity. Our main assumption is that the data points with a smaller distance should have a larger probability to be neighbors. More important, we impose the rank constraint on the Laplacian matrix of the learned similarity matrix to achieve the ideal neighbors assignment, such that the connected components in the data are exact the cluster number and each connected component corresponds to one cluster. Our new model learns the data similarity matrix and cluster structure simultaneously to achieve the optimal clustering results. We derive a novel and efficient algorithm to solve this challenging problem, and show the theoretical analysis on the connections between our method and K-means clustering, and spectral clustering. Moreover, we extend the proposed

[*] Corresponding author.

KDD'14, August 24–27, 2014, New York, New York, USA.
Copyright © 2014 ACM 978-1-4503-2956-9/14/08…$15.00.
http://dx.doi.org/10.1145/2623330.2623726 .

clustering model for the projected clustering to handle the high-dimensional data. Extensive experiments have been conducted on both synthetic data and real-world benchmark data sets. All empirical results show that our new clustering models consistently outperform the related clustering methods.

Notations: Throughout the paper, all the matrices are written as uppercase. For matrix M, the i-th row (with transpose) and the (i,j)-th element of M are denoted by m_i and m_{ij}, respectively. The trace of matrix M is denoted by $Tr(M)$. The L2-norm of vector v is denoted by $\|v\|_2$, the Frobenius norm of matrix M is denoted by $\|M\|_F$. An identity matrix is denoted by I, and $\mathbf{1}$ denotes a column vector with all the elements as one. For vector v and matrix M, $v \geq 0$ and $M \geq 0$ mean all the elements of M and v are equal to or larger than zero.

2. CLUSTERING WITH ADAPTIVE NEIGHBORS

Exploring the local connectivity of data is a successful strategy for clustering task. Given a data set $\{x_1, x_2, ..., x_n\}$, we denote $X \in \mathbb{R}^{n \times d}$ as the data matrix. The neighbors of $x_i \in \mathbb{R}^{d \times 1}$ can be defined as the k-nearest data points in the data set to x_i. In this paper, we consider the probabilistic neighbors, and use the Euclidean distance as the distance measure for simplicity.

For the i-th data point x_i, all the data points $\{x_1, x_2, ..., x_n\}$ can be connected to x_i as a neighbor with probability s_{ij}. Usually, a smaller distance $\|x_i - x_j\|_2^2$ should be assigned a larger probability s_{ij}, so a natural method to determine the probabilities $s_{ij}|_{j=1}^n$ is solving the following problem:

$$\min_{s_i^T \mathbf{1}=1, 0 \leq s_i \leq 1} \sum_{j=1}^n \|x_i - x_j\|_2^2 s_{ij} \tag{1}$$

where $s_i \in \mathbb{R}^{n \times 1}$ is a vector with the j-th element as s_{ij}.

However, the problem (1) has a trivial solution, only the nearest data point can be the neighbor of x_i with probability 1 and all the other data points can not be the neighbors of x_i. On the other hand, if we solve the following problem without involving any distance information in the data:

$$\min_{s_i^T \mathbf{1}=1, 0 \leq s_i \leq 1} \sum_{j=1}^n s_{ij}^2 \tag{2}$$

the optimal solution is that all the data points can be the neighbors of x_i with the same probability $\frac{1}{n}$, which can be seen as a prior in the neighbors assignment.

Combining Eq.(1) and (2), we can solve the following problem:

$$\min_{s_i^T \mathbf{1}=1, 0 \leq s_i \leq 1} \sum_{j=1}^n \left(\|x_i - x_j\|_2^2 s_{ij} + \gamma s_{ij}^2 \right) \tag{3}$$

The second term in Eq.(3) is a regularization and γ is the regularization parameter. Denote $d_{ij}^x = \|x_i - x_j\|_2^2$, and denote $d_i^x \in \mathbb{R}^{n \times 1}$ as a vector with the j-th element as d_{ij}^x, then the problem (3) can be written in vector form as

$$\min_{s_i^T \mathbf{1}=1, 0 \leq s_i \leq 1} \left\| s_i + \frac{1}{2\gamma} d_i^x \right\|_2^2 \tag{4}$$

We will see in Subsection 2.4 that this problem can be solved with a closed form solution.

For each data point x_i, we can use Eq.(3) to assign its neighbors. Therefore, we can solve the following problem to assign the neighbors for all the data points:

$$\min_{\forall i, s_i^T \mathbf{1}=1, 0 \leq s_i \leq 1} \sum_{i,j=1}^n \left(\|x_i - x_j\|_2^2 s_{ij} + \gamma s_{ij}^2 \right) \tag{5}$$

In the clustering task to partition the data into c clusters, an ideal neighbors assignment is that the connected components in the data are exact c. Usually the neighbors assignment with Eq.(5) can not reach the ideal case for any value of γ. In most cases, all the data points are connected as just one connected component. In order to achieve the ideal neighbors assignment, the probabilities $s_{ij}|_{i,j=1}^n$ in the problem (5) should be constrained such that the neighbors assignment becomes an adaptive process to make the connected components are exact c. It seems look like an impossible goal since this kind of structured constraint on the similarities is fundamental but also very difficult to handle. In this paper, we will propose a novel but very simple method to achieve this goal.

The matrix $S \in \mathbb{R}^{n \times n}$ obtained in the neighbors assignment can be seen as a similarity matrix of the graph with the n data points as the nodes. Suppose each node i is assigned a function value as $f_i \in \mathbb{R}^{c \times 1}$, then it can be verified that

$$\sum_{i,j=1}^n \|f_i - f_j\|_2^2 s_{ij} = 2Tr(F^T L_S F) \tag{6}$$

where $F \in \mathbb{R}^{n \times c}$ with the i-th row formed by f_i, $L_S = D_S - \frac{S^T + S}{2}$ is called Laplacian matrix in graph theory, the degree matrix $D_S \in \mathbb{R}^{n \times n}$ is defined as a diagonal matrix where the i-th diagonal element is $\sum_j (s_{ij} + s_{ji})/2$.

If the similarity matrix S is nonnegative, then the Laplacian matrix has an important property as follows [14, 5].

THEOREM 1. *The multiplicity c of the eigenvalue 0 of the Laplacian matrix L_S is equal to the number of connected components in the graph with the similarity matrix S.*

Theorem 1 indicates that if $rank(L_S) = n - c$, then the neighbors assignment is an ideal assignment and we already partition the data points into c clusters based on S, without having to perform K-means or other discretization procedures. Motivated by Theorem 1, we add an additional constraint $rank(L_S) = n - c$ into the problem (5) to achieve the ideal neighbors assignment with clear clustering structure. Thus, our new clustering model is to solve:

$$\begin{aligned} J_{opt} = \min_S & \sum_{i,j=1}^n \left(\|x_i - x_j\|_2^2 s_{ij} + \gamma s_{ij}^2 \right) \\ s.t. \quad & \forall i, s_i^T \mathbf{1} = 1, 0 \leq s_i \leq 1, rank(L_S) = n - c \end{aligned} \tag{7}$$

It is difficult to solve the problem (7). Because $L_S = D_S - \frac{S^T + S}{2}$ and D_S also depends on S, the constraint $rank(L_S) = n - k$ is not easy to tackle. In the next subsection, we will propose a novel and efficient algorithm to solve this challenging problem.

2.1 Optimization Algorithm Solving Problem (7)

Suppose $\sigma_i(L_S)$ is the i-th smallest eigenvalue of L_S, we know $\sigma_i(L_S) \geq 0$ since L_S is positive semi-definite. It can be seen that the problem (7) is equivalent to the following problem for a large enough value of λ:

$$\begin{aligned}\min_{S} &\sum_{i,j=1}^{n} \left(\|x_i - x_j\|_2^2 s_{ij} + \gamma s_{ij}^2\right) + 2\lambda \sum_{i=1}^{c} \sigma_i(L_S) \\ s.t. &\quad \forall i, s_i^T \mathbf{1} = 1, 0 \le s_i \le 1\end{aligned} \tag{8}$$

When λ is large enough, note that $\sigma_i(L_S) \ge 0$ for every i, then the optimal solution S to the problem (8) will make the second term $\sum_{i=1}^{c} \sigma_i(L_S)$ to be zero, and thus the constraint $rank(L_S) = n - c$ in the problem (7) could be satisfied.

According to the Ky Fan's Theorem [6], we have

$$\sum_{i=1}^{c} \sigma_i(L_S) = \min_{F \in \mathbb{R}^{n \times c}, F^T F = I} Tr(F^T L_S F) \tag{9}$$

Therefore, the problem (8) is further equivalent to the following problem

$$\begin{aligned}\min_{S,F} &\sum_{i,j=1}^{n} \left(\|x_i - x_j\|_2^2 s_{ij} + \gamma s_{ij}^2\right) + 2\lambda Tr(F^T L_S F) \\ s.t. &\quad \forall i, s_i^T \mathbf{1} = 1, 0 \le s_i \le 1, F \in R^{n \times c}, F^T F = I\end{aligned} \tag{10}$$

Compared with the original problem (7), the problem (10) is much easier to solve. We can apply the alternative optimization approach to solve it.

When S is fixed, the problem (10) becomes

$$\min_{F \in \mathbb{R}^{n \times c}, F^T F = I} Tr(F^T L_S F) \tag{11}$$

The optimal solution F to the problem (11) is formed by the c eigenvectors of L_S corresponding to the c smallest eigenvalues.

When F is fixed, the problem (10) becomes

$$\begin{aligned}\min_{S} &\sum_{i,j=1}^{n} \left(\|x_i - x_j\|_2^2 s_{ij} + \gamma s_{ij}^2\right) + 2\lambda Tr(F^T L_S F) \\ s.t. &\quad \forall i, s_i^T \mathbf{1} = 1, 0 \le s_i \le 1\end{aligned} \tag{12}$$

According to Eq.(6), the problem (12) can be rewritten as

$$\begin{aligned}\min_{S} &\sum_{i,j=1}^{n} \left(\|x_i - x_j\|_2^2 s_{ij} + \gamma s_{ij}^2 + \lambda \|f_i - f_j\|_2^2 s_{ij}\right) \\ s.t. &\quad \forall i, s_i^T \mathbf{1} = 1, 0 \le s_i \le 1\end{aligned} \tag{13}$$

Note that the problem (13) is independent between different i, so we can solve the following problem individually for each i:

$$\begin{aligned}\min_{s_i} &\sum_{j=1}^{n} \left(\|x_i - x_j\|_2^2 s_{ij} + \gamma s_{ij}^2 + \lambda \|f_i - f_j\|_2^2 s_{ij}\right) \\ s.t. &\quad s_i^T \mathbf{1} = 1, 0 \le s_i \le 1\end{aligned} \tag{14}$$

Denote $d_{ij}^x = \|x_i - x_j\|_2^2$ and $d_{ij}^f = \|f_i - f_j\|_2^2$, and denote $d_i \in \mathbb{R}^{n \times 1}$ as a vector with the j-th element as $d_{ij} = d_{ij}^x + \lambda d_{ij}^f$, then the problem (14) can be written in vector form as

$$\min_{s_i^T \mathbf{1} = 1, 0 \le s_i \le 1} \left\| s_i + \frac{1}{2\gamma} d_i \right\|_2^2 \tag{15}$$

We will see in Subsection 2.4 that this problem can be solved with a closed form solution.

The detailed algorithm to solve the problem (7) is summarized in Algorithm 1.

Algorithm 1 Algorithm to solve problem (7).

```
input Data matrix $X \in \mathbb{R}^{n \times d}$, cluster number $c$, parameter
    $\gamma$, a large enough $\lambda^1$.
output $S \in \mathbb{R}^{n \times n}$ with exact $c$ connected components.
  Initialize $S$ by the optimal solution to the problem (5).
  while not converge do
    1. Update $F$, which is formed by the $c$ eigenvectors
    of $L_S = D_S - \frac{S^T + S}{2}$ corresponding to the $c$ smallest
    eigenvalues.
    2. For each $i$, update the $i$-th row of $S$ by solving the
    problem (15), where $d_i \in \mathbb{R}^{n \times 1}$ is a vector with the $j$-th
    element as $d_{ij} = \|x_i - x_j\|_2^2 + \lambda \|f_i - f_j\|_2^2$.
  end while
```

2.2 Connection to K-means Clustering

Denote the centering matrix by

$$H = I - \frac{1}{n}\mathbf{1}\mathbf{1}^T, \tag{16}$$

and denote $D^x \in \mathbb{R}^{n \times n}$ as a distance matrix where the (i,j)-th element is $d_{ij}^x = \|x_i - x_j\|_2^2$. To analyze the connection of Algorithm 1 to K-means, we first need the following lemma:

LEMMA 1. *$HD^xH = -2HXX^TH$*

PROOF. Since $d_{ij}^x = \|x_i - x_j\|_2^2 = x_i^T x_i + x_j^T x_j - 2x_i^T x_j$, we have $D^x = Diag(XX^T)\mathbf{1}\mathbf{1}^T + \mathbf{1}\mathbf{1}^T Diag(XX^T) - 2XX^T$, where $Diag(XX^T)$ is a diagonal matrix with the diagonal elements of XX^T. Note that $H\mathbf{1} = \mathbf{1}^T H = \mathbf{0}$ according to the definition of H, we have $HD^xH = -2HXX^TH$. □

Theorem 2 reveals that Algorithm 1 is exactly solving the K-means problem when $\gamma \to \infty$.

THEOREM 2. *When $\gamma \to \infty$, the problem (7) is equivalent to the problem of K-means.*

PROOF. The problem (7) can be written in matrix form as

$$\min_{S\mathbf{1} = \mathbf{1}, S \ge 0, rank(L_S) = n - c} Tr(S^T D^x) + \gamma \|S\|_F^2 \tag{17}$$

Due to the constraint $rank(L_S) = n - c$, the solution S has exact c components (that is, S is block diagonal with proper permutation). Suppose the i-th component of S is $S_i \in \mathbb{R}^{n_i \times n_i}$, where n_i is the number of data points in the component, then solving problem (17) is to solve the following problem for each i:

$$\min_{S_i\mathbf{1} = \mathbf{1}, S_i \ge 0} Tr(S_i^T D_i^x) + \gamma \|S_i\|_F^2 \tag{18}$$

When $\gamma \to \infty$, then the problem (18) becomes

$$\min_{S_i\mathbf{1} = \mathbf{1}, S_i \ge 0} \|S_i\|_F^2 \tag{19}$$

The optimal solution to the problem (19) is that all the elements of S_i are equal to $\frac{1}{n_i}$.

Therefore, the optimal solution S to the problem (17) should be the following form when $\gamma \to \infty$:

$$s_{ij} = \begin{cases} \frac{1}{n_k} & x_i, x_j \text{ are in the same component } k \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

We denote the solution set that satisfies the form in Eq.(20) by $\mathcal{V}$. Note that for any possible partition of the c components such that S has the form as in Eq.(20), $\|S\|_F^2$ has

[1] In practice, to accelerate the procedure, the λ can be determined during the iteration. We can initialize $\lambda = \gamma$, then increase λ if the connected components of S is smaller than c and decrease λ if it is greater than c in each iteration.

the same value, i.e., $\|S\|_F^2 = c$. Therefore, the problem (17) becomes

$$\min_{S \in \mathcal{V}} Tr(S^T D^x) \tag{21}$$

According to the constraint of S in Eq.(21), S is symmetric and $S\mathbf{1} = \mathbf{1}^T S = \mathbf{1}$. So $Tr(HD^x HS) = Tr(D^x S) - \frac{1}{n}\mathbf{1}^T D^x \mathbf{1}$ and thus the problem (21) is equivalent to the following problem:

$$\min_{S \in \mathcal{V}} Tr(HD^x HS) \tag{22}$$

Define the label matrix $Y \in \mathbb{R}^{n \times c}$, where the (i,j)-th element is

$$y_{ij} = \begin{cases} \frac{1}{\sqrt{n_k}} & x_i \text{ belongs to the } k\text{-th compoment} \\ 0 & \textit{otherwise} \end{cases} \tag{23}$$

Then according to Eq.(22) and Lemma 1, we have

$$\begin{aligned} & \min_{S \in \mathcal{V}} Tr(HD^x HS) \\ \Leftrightarrow & \max_{S \in \mathcal{V}} Tr(HXX^T HS) \\ \Leftrightarrow & \max_{S \in \mathcal{V}} Tr(X^T HSHX) \\ \Leftrightarrow & \min_{S \in \mathcal{V}} Tr(X^T H(I-S)HX) \\ \Leftrightarrow & \min_{Y} Tr(X^T H(I-YY^T)HX) \\ \Leftrightarrow & \min_{Y} Tr(S_w) \end{aligned} \tag{24}$$

which is exactly the problem of K-means. The matrix S_w is called within-class scatter matrix in classical Linear Discriminant Analysis (LDA) when the labels Y of data are given. K-means is to find the optimal labels Y such that the trace of the within-class scatter matrix $Tr(S_w)$ is minimized. □

We will see in the next subsection that the proposed method in Algorithm 1 is closely related to spectral clustering. Therefore, although the new algorithm is to solve the K-means problem (which can only partition data with spherical shape) when $\gamma \to \infty$, it can partition data with arbitrary shape when γ is not very large. We will also see in the experiments that this new algorithm can find much better solution to the K-means problem even when γ is not very large.

2.3 Connection to Spectral Clustering

Given a graph with the similarity S, spectral clustering is to solve the following problem:

$$\min_{F \in R^{n \times c}, F^T F = I} Tr(F^T L_S F) \tag{25}$$

The optimal solution is the spectral decomposition of the Laplacian matrix L_S, i.e., the optimal solution F is formed by the c eigenvectors of L_S corresponding to the c smallest eigenvalues, as in Eq.(11).

Usually, given a similarity S, the obtained F can not be directly used for clustering since the graph with S does not has exact c connected components. K-means or other discretization procedures should be performed on F to obtain the final clustering results [9].

In the convergence of Algorithm 1, we also obtain an optimal solution F to the problem (25), the difference is that the similarity S is also learned by the algorithm. Note that in the convergence, the last term $2\lambda Tr(F^T L_S F)$ in the problem (10) will approximate zero, the learned S is mainly achieved by solving problem (5).

Thanks to the constraint $rank(L_S) = n - c$, the graph with the learned S will has exact c connected components. Therefore, the optimal solution F, which is formed by the c eigenvectors of L_S corresponding to the c smallest eigenvalues, can be written as

$$F = YQ \tag{26}$$

where $Y \in \mathbb{R}^{n \times c}$ is the label matrix defined in Eq.(23) and $Q \in \mathbb{R}^{c \times c}$ is an arbitrary orthogonal matrix. That is to say, we can use the obtained F directly to get the final clustering result, without the K-means or other discretization procedures as traditional spectral clustering has to do.

In another viewpoint, it can be seen that the proposed algorithm learns the similarity matrix S and the label matrix F simultaneously, while traditional spectral clustering only learns the F given the similarity matrix S. Therefore, the new algorithm could achieve better performance in practice since it also learns an adaptive graph for clustering.

2.4 Determine The Value of γ

In practice, regularization parameter is difficult to tune since its value could be from zero to infinite. In this subsection, we present an effective method to determine the regularization parameter γ in the problem (7).

For each i, the objective function in the problem (7) is equal to the one in the problem (4). The Lagrangian function of problem (4) is

$$\mathcal{L}(s_i, \eta, \beta_i) = \frac{1}{2}\left\| s_i + \frac{d_i^x}{2\gamma_i} \right\|_2^2 - \eta(s_i^T \mathbf{1} - 1) - \beta_i^T s_i \tag{27}$$

where η and $\beta_i \geq \mathbf{0}$ are the Lagrangian multipliers.

According to the KKT condition [2], it can be verified that the optimal solution s_i should be

$$s_{ij} = (-\frac{d_{ij}^x}{2\gamma_i} + \eta)_+ \tag{28}$$

In practice, usually we could achieve better performance if we focus on the locality of data. Therefore, it is preferred to learn a sparse s_i, i.e., only the k nearest neighbors of x_i could have chance to connect to x_i. Another benefit of learning a sparse similarity matrix S is that the computation burden can be alleviated largely for subsequent processing.

Without loss of generality, suppose $d_{i1}^x, d_{i2}^x, ..., d_{in}^x$ are ordered from small to large. If the optimal s_i has only k nonzero elements, then according to Eq.(28), we know $s_{ik} > 0$ and $s_{i,k+1} = 0$. Therefore, we have

$$\begin{cases} -\frac{d_{ik}^x}{2\gamma_i} + \eta > 0 \\ -\frac{d_{i,k+1}^x}{2\gamma_i} + \eta \leq 0 \end{cases} \tag{29}$$

According to Eq.(28) and the constraint $s_i^T \mathbf{1} = 1$, we have

$$\begin{aligned} & \sum_{j=1}^{k} (-\tfrac{d_{ij}^x}{2\gamma_i} + \eta) = 1 \\ & \Rightarrow \eta = \tfrac{1}{k} + \tfrac{1}{2k\gamma_i}\sum_{j=1}^{k} d_{ij}^x \end{aligned} \tag{30}$$

So we have the following inequality for γ_i according to Eq.(29) and Eq.(30):

$$\frac{k}{2}d_{ik}^x - \frac{1}{2}\sum_{j=1}^{k} d_{ij}^x < \gamma_i \leq \frac{k}{2}d_{i,k+1}^x - \frac{1}{2}\sum_{j=1}^{k} d_{ij}^x \tag{31}$$

Therefore, in order to obtain an optimal solution s_i to the problem (4) that has exact k nonzero values, we could set

γ_i to be

$$\gamma_i = \frac{k}{2} d_{i,k+1}^x - \frac{1}{2} \sum_{j=1}^{k} d_{ij}^x \tag{32}$$

The overall γ could be set to the mean of $\gamma_1, \gamma_2, ..., \gamma_n$. That is, we could set the γ to be

$$\gamma = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{k}{2} d_{i,k+1}^x - \frac{1}{2} \sum_{j=1}^{k} d_{ij}^x \right) \tag{33}$$

The number of neighbors k is much easier to tune than the regularization parameter γ since k is an integer and has explicit meaning.

3. PROJECTED CLUSTERING WITH ADAPTIVE NEIGHBORS

Clustering high-dimensional data is an important and challenging problem in practice. In this section, we propose a projected clustering approach with the adaptive neighbors to solve this problem. The goal is to find an optimal subspace on which the adaptive neighboring is performed such that there are exact c connected components in the data.

Denote the total scatter matrix by $S_t = X^T H X$, where H is the centering matrix defined in Eq.(16). Suppose we are to learn a projection matrix $W \in \mathbb{R}^{d \times m}$. First, we constrain the subspace with $W^T S_t W = I$ such that the data on the subspace are statistically uncorrelated. As in Eq.(5), we assign the neighbors for each data by solving the following problem:

$$\begin{array}{l} \min\limits_{S,W} \sum\limits_{i,j=1}^{n} \left(\left\| W^T x_i - W^T x_j \right\|_2^2 s_{ij} + \gamma s_{ij}^2 \right) \\ s.t. \quad \forall i, s_i^T \mathbf{1} = 1, 0 \le s_i \le 1, W^T S_t W = I \end{array} \tag{34}$$

Similarly, to make the neighbors assignment be adaptive such that the connected components in the data are exact c, we impose the constraint on S with $rank(L_S) = n - c$. Therefore, we propose the following problem for learning the projection W and the clustering simultaneously:

$$\begin{array}{l} \min\limits_{S,W} \sum\limits_{i,j=1}^{n} \left(\left\| W^T x_i - W^T x_j \right\|_2^2 s_{ij} + \gamma s_{ij}^2 \right) \\ s.t. \quad \forall i, s_i^T \mathbf{1} = 1, 0 \le s_i \le 1, W^T S_t W = I, \\ \qquad rank(L_S) = n - c \end{array} \tag{35}$$

3.1 Optimization Algorithm for Problem (35)

Using the same trick as in Subsection 2.1, we know that solving problem (35) is equivalent to solving the following problem

$$\begin{array}{l} \min\limits_{S,W,F} \sum\limits_{i,j=1}^{n} \left(\left\| W^T x_i - W^T x_j \right\|_2^2 s_{ij} + \gamma s_{ij}^2 \right) \\ \qquad +2\lambda Tr(F^T L_S F) \\ s.t. \quad \forall i, s_i^T \mathbf{1} = 1, 0 \le s_i \le 1, W^T S_t W = I, \\ \qquad F \in R^{n \times c}, F^T F = I \end{array} \tag{36}$$

We can also apply the alternative optimization approach to solve this problem.

When S is fixed, the problem (36) becomes the problem (11), and the optimal solution F is formed by the c eigenvectors of L_S corresponding to the c smallest eigenvalues.

When F is fixed, the problem (36) becomes

$$\begin{array}{l} \min\limits_{S,W,F} \sum\limits_{i,j=1}^{n} \left(\left\| W^T x_i - W^T x_j \right\|_2^2 s_{ij} + \gamma s_{ij}^2 \right) \\ \qquad +2\lambda Tr(F^T L_S F) \\ s.t. \quad \forall i, s_i^T \mathbf{1} = 1, 0 \le s_i \le 1, W^T S_t W = I \end{array} \tag{37}$$

In problem (37), if S is fixed, then the problem becomes

$$\min_{W^T S_t W = I} \sum_{i,j=1}^{n} \left\| W^T x_i - W^T x_j \right\|_2^2 s_{ij} \tag{38}$$

which can be rewritten as the following problem according to Eq.(6):

$$\min_{W^T S_t W = I} Tr(W^T X^T L_S X W) \tag{39}$$

The optimal solution W to the problem (39) is formed by the m eigenvectors of $S_t^{-1} X^T L_S X$ corresponding to the m smallest eigenvalues (we assume the null space of the data X is removed, i.e., S_t is invertible).

In problem (37), if W is fixed, then according to Eq.(6) again, the problem (37) can be rewritten as

$$\begin{array}{l} \min\limits_{S} \sum\limits_{i,j=1}^{n} \left(\left\| W^T x_i - W^T x_j \right\|_2^2 s_{ij} + \gamma s_{ij}^2 \right) \\ \qquad +\lambda \sum\limits_{i,j=1}^{n} \left\| f_i - f_j \right\|_2^2 s_{ij} \\ s.t. \quad \forall i, s_i^T \mathbf{1} = 1, 0 \le s_i \le 1 \end{array} \tag{40}$$

Note that the problem (40) is independent between different i, so we can solve the following problem individually for each i:

$$\begin{array}{l} \min\limits_{s_i} \sum\limits_{j=1}^{n} \left(\left\| W^T x_i - W^T x_j \right\|_2^2 s_{ij} + \gamma s_{ij}^2 \right) \\ \qquad +\lambda \sum\limits_{j=1}^{n} \left\| f_i - f_j \right\|_2^2 s_{ij} \\ s.t. \quad s_i^T \mathbf{1} = 1, 0 \le s_i \le 1 \end{array} \tag{41}$$

Denote $d_{ij}^{wx} = \left\| W^T x_i - W^T x_j \right\|_2^2$ and $d_{ij}^f = \left\| f_i - f_j \right\|_2^2$, and denote $d_i^w \in \mathbb{R}^{n \times 1}$ as a vector with the j-th element as $d_{ij}^w = d_{ij}^{wx} + \lambda d_{ij}^f$, then the problem (41) can be written in vector form as

$$\min_{s_i^T \mathbf{1} = 1, 0 \le s_i \le 1} \left\| s_i + \frac{1}{2\gamma} d_i^w \right\|_2^2 \tag{42}$$

which is the same problem as in Eq.(15) and can be solved with a closed form solution.

The detailed algorithm to solve the problem (35) is summarized in Algorithm 2. We can also use Eq.(33) to determine the regularization parameter γ.

Algorithm 2 Algorithm to solve problem (35).

```
input Data matrix $X \in \mathbb{R}^{n\times d}$, cluster number $c$, reduced dimension $m$, parameter $\gamma$, a large enough $\lambda$.
output $S \in \mathbb{R}^{n\times n}$ with exact $c$ connected components, projection $W \in \mathbb{R}^{d\times m}$.
  Initialize $S$ by the optimal solution to the problem (3).
  while not converge do
    1. Update $L_S = D_S - \frac{S^T+S}{2}$, where $D_S \in \mathbb{R}^{n\times n}$ is a diagonal matrix with the $i$-th diagonal element as $\sum_j (s_{ij}+s_{ji})/2$.
    2. Update $F$, whose columns are the $c$ eigenvectors of $L_S$ corresponding to the $c$ smallest eigenvalues.
    2. Update $W$, whose columns are the $m$ eigenvectors of $S_t^{-1}X^T L_S X$ corresponding to the $m$ smallest eigenvalues.
    3. For each $i$, update the $i$-th row of $S$ by solving the problem (42), where $d_i^w \in \mathbb{R}^{n\times 1}$ is a vector with the $j$-th element as $d_{ij}^w = \left\|W^T x_i - W^T x_j\right\|_2^2 + \lambda \left\|f_i - f_j\right\|_2^2$.
  end while
```

3.2 Connection to LDA

In this subsection, we show that when $\gamma \to \infty$, the proposed method in Algorithm 2 is to solve the LDA problem with the labels also being optimized at the same time. The result is summarized in the following theorem:

THEOREM 3. *When $\gamma \to \infty$, the problem (35) is equivalent to the problem of LDA, in which the labels are also variables to be optimized.*

PROOF. The problem (35) can be written in matrix form as

$$\min_{\substack{S\mathbf{1}=\mathbf{1}, S\geq 0, W^T S_t W=I,\\ rank(L_S)=n-c}} Tr(S^T D^{wx}) + \gamma \left\|S\right\|_F^2 \tag{43}$$

where $D^{wx} \in \mathbb{R}^{n\times n}$ is a distance matrix with the (i,j)-th element as $d_{ij}^{wx} = \left\|W^T x_i - W^T x_j\right\|_2^2$. Due to the constraint $rank(L_S) = n - c$, the solution S has exact c components. Suppose the i-th component of S is $S_i \in \mathbb{R}^{n_i\times n_i}$, where n_i is the number of data points in the component, then solving problem (43) is to solve the following problem for each i:

$$\min_{S_i\mathbf{1}=\mathbf{1}, S_i\geq 0, W^T S_t W=I} Tr(S_i^T D_i^{wx}) + \gamma \left\|S_i\right\|_F^2 \tag{44}$$

When $\gamma \to \infty$, then the problem (44) becomes

$$\min_{S_i\mathbf{1}=\mathbf{1}, S_i\geq 0} \left\|S_i\right\|_F^2 \tag{45}$$

The optimal solution to the problem (45) is that all the elements of S_i are equal to $\frac{1}{n_i}$.

With similar derivation as in Subsection 2.2, we know the problem (43) is equivalent to the following problem when $\gamma \to \infty$:

$$\min_{S\in\mathcal{V}, W^T S_t W=I} Tr(HD^{wx}HS) \tag{46}$$

where $\mathcal{V}$ is the solution set of S that satisfies the form in Eq.(20).

Then according to Eq.(46) and Lemma 1, we have

$$\begin{aligned}
&\min_{S\in\mathcal{V}, W^T S_t W=I} Tr(HD^{wx}HS)\\
=&\max_{S\in\mathcal{V}, W^T S_t W=I} Tr(HXWW^T X^T HS)\\
=&\max_{S\in\mathcal{V}, W^T S_t W=I} Tr(W^T X^T HSHXW)\\
=&\min_{S\in\mathcal{V}, W^T S_t W=I} Tr(W^T X^T H(I-S)HXW)\\
=&\min_{Y, W^T S_t W=I} Tr(W^T X^T H(I-YY^T)HXW)\\
=&\min_{Y, W^T S_t W=I} Tr(W^T S_w W)
\end{aligned} \tag{47}$$

where Y is the label matrix defined as in Eq.(23). If the label matrix Y is given, then the optimal solution to the problem (47) is equal to the solution of LDA. Therefore, when $\gamma \to \infty$, the proposed method in Algorithm 2 is to solve the LDA problem, but the labels are also unknown and are to be optimized at the same time. □

When γ is not very large, the matrix $X^T L_S X$ in Eq.(39) can be viewed as local within-class scatter matrix. In this case, the Algorithm 2 can be viewed as a unsupervised version of local scatter matrices based LDA methods, which are designed for handling multimodal non-Gaussian data [17].

4. EXPERIMENTS

In this part, we will show the performance of the proposed clustering method (Algorithm 1) and the projected clustering method (Algorithm 2) on both synthetic data and real data sets. For simplicity, we denote our clustering method as CAN (Clustering with Adaptive Neighbors), and our projected clustering method as PCAN (Projected Clustering with Adaptive Neighbors) in the following context.

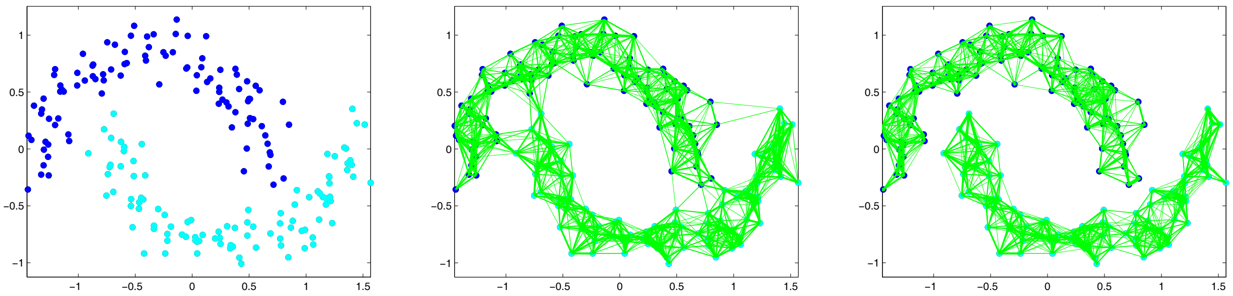
4.1 Experiments on Synthetic Data

4.1.1 Experiments to Evaluate CAN Method

We use two synthetic data sets to measure the clustering efficiency of CAN. Also, we find that CAN is able to obtain an excellent initialization index vector for K-Means, which decreases its objective value and promotes its clustering accuracy to a large extent.

The first toy data set we used is a randomly generated two-moon data. In this data set, there are two clusters of data distributed in the two-moon shape. Our goal is to construct a new similarity matrix to divide the data points into exact two clusters. From Fig. 1 we can easily find the clustering efficiency of the proposed CAN method. In this figure, we set the color of the two clusters to be cyan and blue separately and let the width of the connecting line denote the learned similarity of two corresponding points. In the initial similarity matrix learned by Eq.(5), several pairs of points from different clusters are connected. Whereas, after the learning by Eq.(7), there is not even a single line between these two clusters, which means that the proposed clustering method CAN can successfully partition the original data into two classes.

The second toy data set is a randomly generated multi-cluster data. In this data set, there are 196 clusters distributed in a spherical way. Firstly we run K-Means for 10000 times and report the result with the minimal K-means objective value in the independently 10000 times run. Then we run CAN once to generate an clustering result and use it as initialization for K-means. The comparison results are



(a) Original Data (b) Learnt Graph by Eq.(5) (c) Learnt Graph by Eq.(7) (CAN)

Figure 1: Learned graph by CAN on the two-moon synthetic data.

shown in Tab. 1 and Fig. 2. It is apparent that even after 10000 times run, the minimal K-means objective value and clustering accuracy obtained by K-means are far behind the result obtained by PCAN with only one run.

Methods	Acc%(min_obj)	Min_obj
K-Means	69.95	318.29
CAN	**98.98**	**106.12**

Table 1: Clustering accuracy and minimal K-Means objective value on multi-cluster synthetic data sets.

4.1.2 *Experiments to Evaluate PCAN Method*

For PCAN method, not only did we measure its clustering efficiency but we also tested its projection ability. The first toy data set for PCAN is a randomly generated three-ring data. In this data set, there are five dimensions, among which the data in the first two dimensions are distributed in a three-circle shape while the data in the other three dimensions are noises.

Since only the first two dimensions contain useful information, it would be important if the method could find a subspace which contains only important dimensions. We compare the PCAN method with two popular dimension reduction methods, Principal Component Analysis (PCA) [11] and Locality Preserving Projections(LPP) [8]. From Fig. 3 we can see the projection results of these three methods. Apparently, PCA and LPP are not powerful enough to find a good subspace for this projection task. In contrast, the proposed method PCAN successfully finds a subspace which is almost the same as the subspace formed by the first two significant dimensions, and also accomplishes the clustering task.

The second toy data set for PCAN is a randomly generated two-Gaussian data. In this data set, there are two clusters of data which obeys the Gaussian distribution. Our goal is to find a good projection direction which helps to partition the two clusters apart. Still, we compare PCAN with PCA and LPP. The comparison results are displayed in Fig. 4. Seen from Fig. 4, we can find out that when these two clusters are far from each other, all these three methods could easily find a good projection direction. However, as the distance between these two clusters lower down, PCA becomes inefficient. As the two clusters become more close, LPP also loses its way to achieve the projection goal. However, the PCAN method always behave well. The reason for this phenomenon is that PCA focus on the global structure thus fail immediately after the two clusters become close. While LPP pays more attention to the local structure, thus could achieve good performance when the two clusters are relatively close. But when the distance of these two clusters are fairly small, LPP is not capable any more. But our method, PCAN, lays more emphasis on the discriminative structure and thus keeps its projection quality all the time.

4.2 Experiments on Real Benchmark Datasets

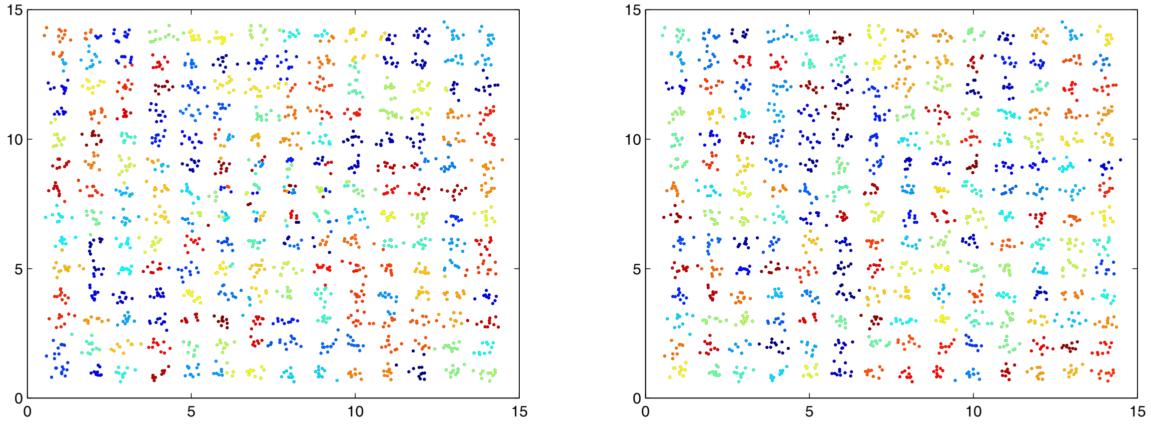
4.2.1 *Experiments on Clustering*

We evaluated the proposed clustering methods on 15 benchmark datasets: Stock, Pathbased, Movements, Wine, Compound, Spiral, Yeast, Glass, Ecoli, UmistData, COIL25, JAFFE, USPS, Palm and MSRA, among which three are shape data sets, six are biological data sets from UCI Machine Learning Repository and the other six are image data sets. The descriptions of these 15 datasets are summarized in Tab. 2.

We compared our clustering methods CAN and PCAN with K-means, Ratio Cut, Normalized Cut and NMF methods.

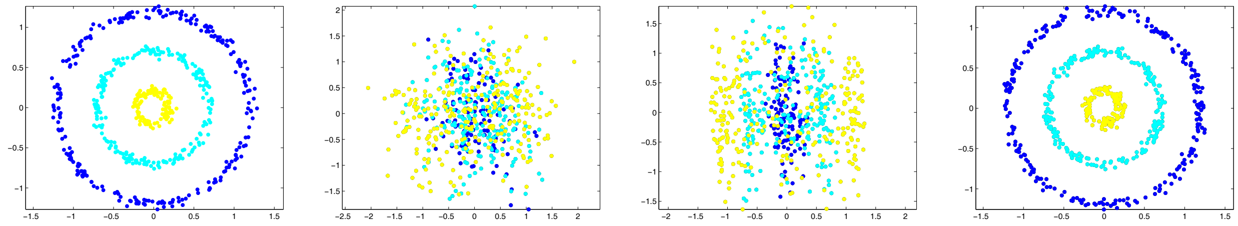
In the clustering experiment, we set the number of clusters to be the ground truth in each data set. For those methods calling for an input of an affinity matrix, like Ratio Cut, Normalized Cut and NMF, the graph is constructed with the self-tune Gaussian method [23]. For the methods involving K-means, including K-means, Ratio Cut and Normalized Cut, we use the same initialization and repeat 100 times independently to perform K-means for discretization. For these methods, we record the average performance, standard deviation and the performance corresponding to the best K-Means objective value. As for NMF, CAN and PCAN, we run only once and record the results.

The evaluation of different methods is based on two widely used clustering metrics: accuracy and NMI (normalized mutual information). From Tab. 3 and Tab. 4, we can come into the conclusion that our proposed methods CAN and PCAN outperform other methods on most of the benchmark data sets. We always get a better accuracy and NMI under different circumstances. In addition, the results of other methods are dependent of the initialization, while ours are always stable with a certain setting.



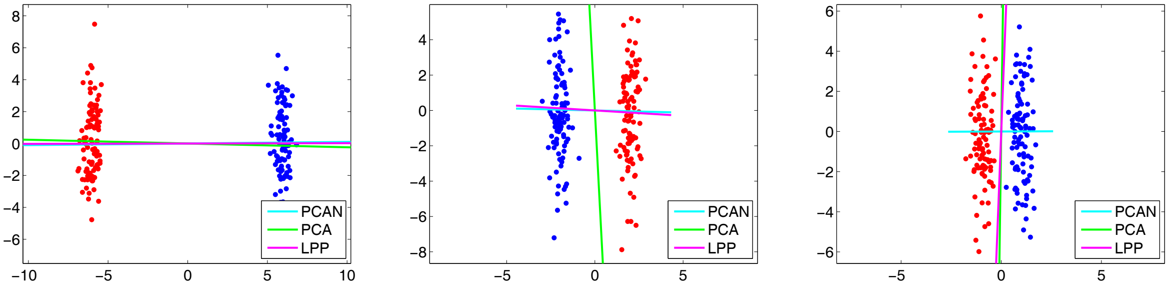
(a) K-Means Result for 196 Clusters (b) CAN Result for 196 Clusters

Figure 2: Clustering results on multi-cluster synthetic data sets.



(a) The First Two Dimensions(b) Learnt Subspace by PCA (c) Learnt Subspace by LPP(d) Learnt Subspace by PCAN

Figure 3: Results on the three-ring synthetic data.



(a) Clusters far away (b) Clusters relatively close (c) Clusters fairly close

Figure 4: Results on the two-Gaussian synthetic data.

4.2.2 Experiments on Projection

In this experiments, we evaluated the projection ability of the proposed PCAN method on 6 benchmark data sets with high dimension: Movements, UmistData, Coil20, Jaffe50, Palm and MSRA50. Similar to that in the synthetic data experiments, we compared PCAN with PCA and LPP methods.

The affinity matrix for LPP is constructed with the self-tune Gaussian method [23]. In order to measure the quality of dimension reduction, we let the three methods learn a projection matrix first and perform K-Means on the projected data for projection ability measurement. For each method, the K-Means method is repeated for 100 times so as to compute the average clustering accuracy and standard deviation.

The results are reported in Fig. 5 and Tab. 5, from which we can see the superiority of the PCAN method for projection task. Moreover, the PCAN method is able to project the original data onto a subspace with quite small dimensions(c-1), where c denotes the number of clusters in each data set. Such a subspace with low dimensions projected by our method would even outperform that obtained by PCA and LPP with higher dimensions. It is worthy to noting that, if we apply PCAN method for only projection task, the c in

	K-Means		Ratio Cut		Normalized Cut		NMF	CAN	PCAN
	%(min_obj)	Average%	%(min_obj)	Average%	%(min_obj)	Average%			
Umist	42.61	43.31±2.66	66.09	60.44±3.81	62.26	59.51±3.50	62.26	**77.57**	68.17
Coil20	71.67	56.54±5.35	77.50	69.42±4.60	79.31	70.30±4.66	70.42	**90.14**	83.33
Jaffe50	91.55	73.70±8.56	96.71	84.78±8.39	96.71	81.63±8.89	96.71	96.71	**100.00**
USPS	65.21	64.27±3.08	69.20	67.59±5.11	69.53	68.43±5.10	67.37	**78.96**	63.81
Palm	72.40	68.43±2.86	65.00	61.35±1.84	63.50	60.97±2.01	61.40	84.85	**88.85**
MSRA50	57.14	53.50±5.00	52.47	49.28±2.67	52.47	47.90±3.53	50.69	**57.87**	**57.87**
Stock	66.74	74.02±11.73	57.68	55.23±4.03	57.68	55.39±5.06	56.21	67.79	**77.16**
Pathbased	74.33	74.34±0.03	77.67	77.67±0.00	77.67	77.67±0.00	78.00	**87.00**	**87.00**
Movements	48.33	44.04±2.27	48.33	46.01±2.29	45.56	44.49±1.90	43.33	**49.17**	**49.17**
Spiral	34.29	34.56±0.26	99.68	96.92±10.10	99.68	96.03±11.82	91.03	**100.00**	**100.00**
Wine	95.51	94.65±0.53	95.51	95.44±0.54	94.94	94.99±4.58	94.94	97.19	**100.00**
Compound	69.42	65.49±9.36	53.63	52.94±3.93	53.13	52.67±3.80	52.38	**80.20**	79.70
Yeast	40.50	38.00±2.13	41.44	38.11±2.18	40.03	36.99±2.81	35.65	**50.27**	50.07
Glass	43.46	45.57±3.51	37.85	38.28±2.27	37.85	38.26±3.00	37.85	**50.00**	49.53
Ecoli	62.50	57.10±6.07	59.23	54.08±5.30	57.44	53.10±4.22	54.17	83.04	**83.33**

Table 3: Clustering Accuracy on Real Datasets

	K-Means		Ratio Cut		Normalized Cut		NMF	CAN	PCAN
	%(min_obj)	Average%	%(min_obj)	Average%	%(min_obj)	Average%			
Umist	66.61	64.44±1.77	81.64	77.78±1.79	78.15	77.26±1.88	78.97	**88.52**	85.60
Coil20	80.66	73.45±2.61	87.10	84.01±1.94	89.17	84.42±1.93	81.22	**94.60**	89.10
Jaffe50	91.75	82.44±5.48	96.23	90.24±4.29	96.23	89.43±4.16	96.23	96.23	**100.00**
USPS	62.99	62.07±1.64	75.80	73.95±2.41	75.78	73.97±2.40	74.15	**80.46**	68.93
Palm	89.42	88.59±1.08	87.17	85.68±0.64	86.53	85.59±0.62	85.73	94.33	**95.14**
MSRA50	66.89	62.79±3.72	59.97	58.59±2.37	59.84	57.38±3.10	63.49	**70.32**	70.10
Stock	73.70	**76.35±6.22**	62.57	56.56±5.69	62.57	56.20±6.40	60.29	74.89	68.30
Pathbased	51.28	51.30±0.04	55.16	55.16±0.00	55.16	55.16±0.00	52.51	**75.63**	**75.63**
Movements	60.43	57.20±1.89	64.76	61.89±2.33	61.39	59.85±2.05	58.95	**64.07**	60.84
Spiral	0.03	0.05±0.02	98.35	94.98±12.34	98.35	94.19±13.95	75.95	**100.00**	**100.00**
Wine	84.89	82.41±1.44	84.47	84.37±1.25	83.24	84.02±3.90	83.24	88.97	**100.00**
Compound	69.68	69.60±5.68	73.37	70.71±4.44	73.34	70.29±4.39	73.26	**79.27**	78.65
Yeast	26.15	25.19±1.06	27.95	24.94±1.37	25.63	23.88±1.35	23.66	30.30	**30.55**
Glass	32.95	33.13±2.60	29.37	29.10±3.05	28.00	28.58±2.49	28.70	26.91	**33.82**
Ecoli	53.34	53.04±3.02	51.26	48.96±3.11	51.26	49.78±2.32	47.12	72.20	**72.44**

Table 4: Clustering NMI on Real Datasets

Data sets	Num of Instances	Dimensions	Classes
Stock	950	10	5
Pathbased	300	2	3
Movements	360	90	15
Spiral	312	2	3
Wine	178	13	3
Compound	399	2	6
Yeast	1484	8	10
Glass	219	9	6
Ecoli	367	7	8
Umist	165	3456	15
Coil20	1440	1024	20
Jaffe50	213	1024	10
USPS	1854	256	10
Palm	2000	256	100
MSRA50	1799	1024	12

Table 2: Descriptions of 15 Benchmark Datasets

	PCA	LPP	PCAN
Palm	68.66±2.55	78.77±2.94	**80.03±2.96**
MSRA50	54.65±4.37	52.93±4.31	**62.31±3.68**
Movements	44.46±2.48	47.47±2.55	**55.99±3.06**
Jaffe50	77.45±7.30	79.41±9.61	**81.78±10.15**
Umist	43.85±2.28	64.55±4.66	**67.39±4.53**
Coil20	59.39±4.63	73.32±5.10	**74.93±5.43**

Table 5: The best results with optimal dimensions.

PCAN can be viewed as a parameter and thus we can reduce data into arbitrary dimensions. In this experiment, we set the c in PCAN to the class number of data for simplicity.

5. CONCLUSIONS

In this paper, we proposed a novel clustering model to simultaneously learn the data similarity matrix and the clustering structure. In our new clustering method, the data similarity matrix is learned by assigning the adaptive neighbors for each data point based on the local connectivity. The new rank constraint is imposed on the Laplacian matrix of the data similarity matrix to create the clustering structure in the similarity matrix as several disconnected components. We derived an efficient algorithm to optimize the proposed challenging problem, and proved the theoretical connections between our method and other clustering approaches. We further extended the proposed clustering model for the projected clustering on high-dimensional data. Extensive experiments have been conducted on both synthetic data and 15

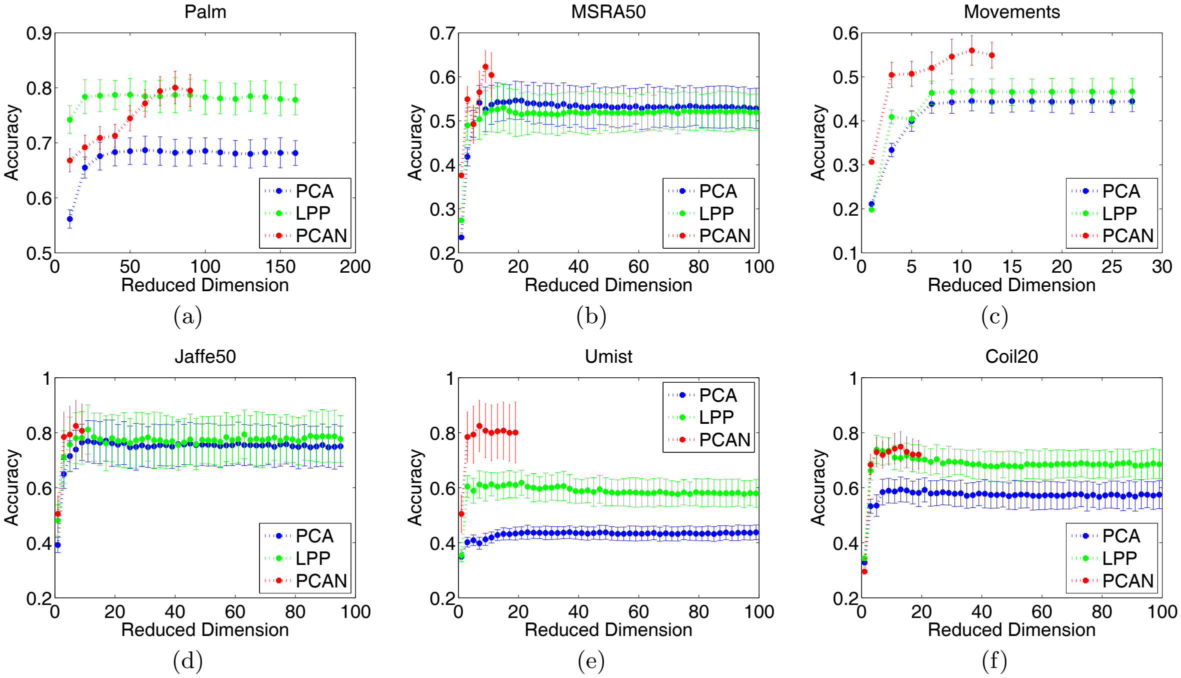


Figure 5: Results of projection methods on 6 benchmark data sets.

real-world benchmark data sets to demonstrate the superior performance of our models.

6. ACKNOWLEDGMENTS

This research was partially supported by NSF-IIS 1117965, NSF-IIS 1302675, NSF-IIS 1344152, NSF-DBI 1356628.

7. REFERENCES

[1] A. Ben-Hur, D. Horn, H. Siegelmann, and V. Vapnik. Support vector clustering. 2:125–137, 2001.

[2] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[3] X. Cai, F. Nie, and H. Huang. Multi-view k-means clustering on big data. *23rd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2598–2604, 2013.

[4] X. Cai, F. Nie, H. Huang, and F. Kamangar. Heterogeneous image features integration via multi-modal spectral clustering. In *The 24th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2011)*, pages 1977–1984, 2011.

[5] F. R. K. Chung. *Spectral Graph Theory*. CBMS Regional Conference Series in Mathematics, No. 92, American Mathematical Society, February 1997.

[6] K. Fan. On a theorem of weyl concerning eigenvalues of linear transformations. i. 35(11):652–655, 1949.

[7] L. W. Hagen and A. B. Kahng. New spectral methods for ratio cut partitioning and clustering. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 11(9):1074–1085, 1992.

[8] X. He and P. Niyogi. Locality preserving projections. In *NIPS*, 2003.

[9] J. Huang, F. Nie, and H. Huang. Spectral rotation versus k-means in spectral clustering. In *AAAI*, 2013.

[10] S. C. Johnson. Hierarchical clustering schemes. *Psychometrika*, 2:241–254, 1967.

[11] I. T. Jolliffe. *Principal Component Analysis,2nd Edition*. Springer-Verlag, New York, 2002.

[12] T. Li and C. H. Q. Ding. The relationships among various nonnegative matrix factorization methods for clustering. In *ICDM*, pages 362–371, 2006.

[13] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297. Berkeley, University of California Press, 1967.

[14] B. Mohar. The laplacian spectrum of graphs. In *Graph Theory, Combinatorics, and Applications*, pages 871–898. Wiley, 1991.

[15] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems (NIPS)*, 14:849–856, 2002.

[16] F. Nie, C. H. Q. Ding, D. Luo, and H. Huang. Improved minmax cut graph clustering with nonnegative relaxation. In *ECML/PKDD*, pages 451–466, 2010.

[17] F. Nie, S. Xiang, and C. Zhang. Neighborhood minmax projections. In *IJCAI*, pages 993–998, 2007.

[18] F. Nie, D. Xu, and X. Li. Initialization independent clustering with actively self-training method. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 42(1):17–27, 2012.

[19] F. Nie, Z. Zeng, I. W. Tsang, D. Xu, and C. Zhang. Spectral embedded clustering: A framework for in-sample and out-of-sample spectral clustering. *IEEE Transactions on Neural Networks*, 22(11):1796–1808, 2011.

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

[21] H. Wang, F. Nie, and H. Huang. Multi-view clustering and feature learning via structured sparsity. *The 30th International Conference on Machine Learning (ICML 2013)*, pages 352–360, 2013.

[22] L. Xu, J. Neufeld, B. Larson, and D. Schuurmans. Maximum margin clustering. Cambridge, MA, 2005. MIT Press.

[23] L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. In *NIPS*, 2004.