Structured Doubly Stochastic Matrix for Graph Based Clustering

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

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

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

ABSTRACT

As one of the most significant machine learning topics, clustering has been extensively employed in various kinds of area. Its prevalent application in scientific research as well as industrial practice has drawn high attention in this day and age. A multitude of clustering methods have been developed, among which the graph based clustering method using the affinity matrix has been laid great emphasis on. Recent research work used the doubly stochastic matrix to normalize the input affinity matrix and enhance the graph based clustering models. Although the doubly stochastic matrix can improve the clustering performance, the clustering structure in the doubly stochastic matrix is not clear as expected. Thus, post-processing step is required to extract the final clustering results, which may not be optimal. To address this problem, in this paper, we propose a novel convex model to learn the structured doubly stochastic matrix by imposing low-rank constraint on the graph Laplacian matrix. Our new structured doubly stochastic matrix can explicitly uncover the clustering structure and encode the probabilities of pair-wise data points to be connected, such that the clustering results are enhanced. An efficient optimization algorithm is derived to solve our new objective. Also, we provide theoretical discussions that when the input differs, our method possesses interesting connections with K-means and spectral graph cut models respectively. We conduct experiments on both synthetic and benchmark datasets to validate the performance of our proposed method. The empirical results demonstrate that our model provides an approach to better solving the K-mean clustering problem. By using the cluster indicator provided by our model as initialization, K-means converges to a smaller objective function value with better clustering performance. Moreover, we compare the clustering performance of our model with spectral clustering and related double stochastic model. On all datasets, our method performs equally or better than the related methods.

CCS Concepts

•**Theory of computation** → **Unsupervised learning and clustering;**

Keywords

Doubly Stochastic Matrix; Graph Laplacian; K-means Clustering; Spectral Clustering

1. INTRODUCTION

Graph based learning is one of the central research topics in machine learning. These models use similarity graph as input and perform learning tasks over the graph, such as spectral clustering [17], manifold based dimensional reduction [2, 12], graph-based semi-supervised learning [30, 31], *etc*. Because the input data graph is often associated to the affinity matrix (the pair-wise similarities between data points), graph based learning methods show promising performance and have been introduced to many applications in data mining [20, 19, 18, 1, 6]. The graph-based learning methods depend on the input affinity matrix, thus the quality of the input data graph is crucial in achieving the final superior learning solutions.

To enhance the learning results, graph based learning methods often need pre-processing on the affinity matrix. The doubly stochastic matrix (also called bistochastic matrix) was utilized to normalize the affinity matrix and showed promising clustering results [29]. A doubly stochastic matrix $S \in \Re^{n \times n}$ is a square matrix and all elements in S satisfy:

$$s_{ij} \geq 0, \sum_{j=1}^{n} s_{ij} = 1, \sum_{i=1}^{n} s_{ij} = 1, 1 \leq i, j \leq n\,. \quad (1)$$

Some previous works have been proposed to learn the best doubly stochastic approximation to a given affinity matrix [29, 26, 11]. Imposing the double stochastic constraints can properly normalize the affinity matrix such that the data graph is more suitable for clustering tasks. However, even with using the doubly stochastic matrix, the final clustering structure is still not obvious in the data graph. The graph based clustering methods often use K-means algorithm to post-process the clustering results to get the clustering indicators, thus the final clustering results are dependent on and sensitive to the initializations.

To address these challenges, in this paper, we propose a novel model to learn the structured doubly stochastic matrix, which encodes the probability of each pair of data points to be connected. We explicitly constrain the rank of the graph Laplacian and guarantee the number of connected components in the graph to be exactly k, *i.e.* the number of clusters. Thus, the clustering results

[*]To whom all correspondence should be addressed. This work was partially supported by US NSF-IIS 1117965, NSF-IIS 1302675, NSF-IIS 1344152, NSF-DBI 1356628, NIH R01 AG049371.

KDD '16, August 13-17, 2016, San Francisco, CA, USA
© 2016 ACM. ISBN 978-1-4503-4232-2/16/08…$15.00
DOI: http://dx.doi.org/10.1145/2939672.2939805

can be directly obtained without the post-processing step, such that the clustering results are superior and stable. To solve the proposed new objective, we introduce an efficient optimization algorithm. Meanwhile, we also provide theoretical analysis on the connection between our method and K-means clustering as well as spectral clustering, respectively.

We conduct extensive experiments on both synthetic and benchmark datasets to evaluate the performance of our method. We find that our method provides a good initialization for K-means clustering such that a smaller objective function value can be achieved for the K-means problem. Also, we compare the clustering performance of our model with other methods. On all datasets, our method performs equally or better than related methods.

Notations: Throughout this paper, matrices are all written as uppercase letters while vectors as bold lower case letters. For a matrix $M \in \Re^{d\times n}$, its i-th row, j-th column and ij-th element are denoted by $\mathbf{m}^i$, $\mathbf{m}_j$ and m_{ij} respectively. The Frobenius norm of M is defined as $\|M\|_F = \sqrt{\sum\limits_{i=1}^{d}\sum\limits_{j=1}^{n} m_{ij}^2}$. The trace norm (also known as the nuclear norm) is defined as $\|M\|_* = \sum\limits_{i=1}^{min\{d,n\}} \sigma_i$, where σ_i is the i-th singular value of M. For a vector $\mathbf{v} \in \Re^n$, when $p \neq 0$, its ℓ_p-norm $\|\mathbf{v}\|_p$ is defined as $\|\mathbf{v}\|_p = (\sum\limits_{i=1}^{n} |v_i|^p)^{\frac{1}{p}}$. Specially, $\mathbf{1}$ represents a vector whose elements are 1 consistently and I stands for the identity matrix.

2. CLUSTERING WITH NEW STRUCTURED DOUBLY STOCHASTIC MATRIX

To achieve good clustering results, doubly stochastic matrix is usually utilized to approximate the given affinity matrix such that the clustering structure in data graph can be maintained. However, final clustering structures are still not obvious in the data graph and post-processing step has to be employed, which leads the clustering results to be not optimal. To address this problem and learn a more powerful doubly stochastic matrix, we propose a new structured doubly stochastic model to capture clustering structure and encode probabilistic data similarity.

2.1 New Structured Doubly Stochastic Matrix

The ideal clustering structure of a graph with n data points is to have exactly k connected components, where k is the number of clusters. If we reshuffle the n data points in the similarity matrix of the ideal graph such that points in the same connected components are arranged together, then the k connected components form k blocks ranging along the diagonal of the similarity matrix. With such ideal structure we can immediately obtain clustering indicators without post-processing steps. Thus, we propose a novel method to learn the new structured doubly stochastic matrix which has such ideal clustering structure with exactly k blocks. From [15, 8], we know that each connected component in the graph W corresponds to an eigenvalue 0 in the Laplacian matrix $L_W = D_W - W$. D_W is the degree matrix of graph W defined as $D_W = Diag(\sum\limits_j w_{ij})$, where $Diag(\mathbf{a})$ denotes a diagonal matrix with diagonal elements formed by the entries of vector $\mathbf{a}$. If W has exactly k blocks, then k eigenvalues of L_W are zeros, *i.e.* the rank of L_W is equal to $n - k$.

Therefore, given the affinity matrix W, we propose to learn a new structured doubly stochastic matrix $M \in \Re^{n\times n}$ such that its Laplacian matrix $L_M = D_M - M$ is restricted to be $rank(L_M) = n - k$. With this constraint, the learnt M has the clustering block structure along the diagonal with proper permutation, based on which we can directly partition the data points into k clusters.

Moreover, to make M encode the probability of each pair of data points to be connected in the graph, we normalize $M\mathbf{1} = \mathbf{1}$. As a result, the entire probability for a point to connect with others is 1. Consequently, we have $D_M = I$. Meanwhile, since M represents the probability of each pair of data points to be connected, we naturally expect the learnt M is symmetric and nonnegative. These constraints validate the doubly stochastic property of matrix M.

Under our new constraints, we learn a structured doubly stochastic matrix M to best approximate the affinity matrix W by solving:

$$\begin{aligned} &\min_{M} \|M - W\|_F^2 \\ s.t. \quad & M \geq 0, M = M^T, M\mathbf{1} = \mathbf{1}, rank(L_M) = n - k\,. \end{aligned} \tag{2}$$

Theoretically, in the ideal case the probability of a certain point to correlate with points in the same cluster should be the same. That is, suppose there are n_i points in the i-th cluster, for any two points p_s and p_n in the i-th cluster, the probability of p_s and p_n to be connected is $m_{sn} = \frac{1}{n_i}$. Consistently, for point p_s, we have $m_{ss} = \frac{1}{n_i}$. Toward this end we add another term $r\|M\|_F^2$, where according to Lemma 1 a large enough parameter r forces the elements in each block of matrix M to be the same.

LEMMA 1. *In the following problem:*

$$\begin{aligned} &\min_{M} \|M - W\|_F^2 + r\|M\|_F^2 \\ s.t. \quad & M \geq 0, M = M^T, M\mathbf{1} = \mathbf{1}, rank(L_M) = n - k\,, \end{aligned}$$

if the value of r tends to be infinity, the matrix M is block diagonal with elements in each block to be the same. The number of blocks in matrix M is k.

Proof: If r tends to infinity, the following optimization problem

$$\begin{aligned} &\min_{M} \|M - W\|_F^2 + r\|M\|_F^2 \\ s.t. \quad & M \geq 0, M = M^T, M\mathbf{1} = \mathbf{1}, rank(L_M) = n - k \end{aligned} \tag{3}$$

is equivalent to

$$\begin{aligned} &\min_{M} \|M\|_F^2 \\ s.t. \quad & M \geq 0, M = M^T, M\mathbf{1} = \mathbf{1}, rank(L_M) = n - k\,. \end{aligned} \tag{4}$$

According to the previous discussion, with proper rearrangement of rows and columns of M, the constraint $rank(L_M) = n - k$ require the structure of M to be k blocks diagonally arranged like:

$$\begin{bmatrix} M_1 & & & 0 \\ & M_2 & & \\ & & \ddots & \\ 0 & & & M_k \end{bmatrix}.$$

Meanwhile, let's we denote a random row (or column) of one block in M as $\mathbf{m}^T$(or $\mathbf{m}$). Since $\sum\limits_i m_i^2 \geq \frac{(\sum_i m_i)^2}{length(\mathbf{m})}$, and only when all m_i are of the same value makes the left and right sides be equal. Thus, the solution to minimize Problem (4) is that M is a block diagonal matrix with elements in each block to be the same.

□

In addition, to highlight the constraint that M has exactly k blocks, we add another constraint as $Tr(M) = k$ and have:

$$\begin{aligned} &\min_{M} \|M - W\|_F^2 + r\|M\|_F^2 \\ s.t. \quad & M \geq 0, M = M^T, M\mathbf{1} = \mathbf{1}, \\ & rank(L_M) = n - k, Tr(M) = k\,. \end{aligned} \tag{5}$$

The constraint $rank(L_M) = n - k$ makes Problem (5) nonconvex. Therefore, we use the trace norm of L_M as a relaxation form of $rank(L_M)$. Our final objective is to solve:

$$\begin{aligned} J_{opt} = \min_{M} \|M - W\|_F^2 + \gamma \|L_M\|_* + r \|M\|_F^2 \\ s.t. \quad M \geq 0, M = M^T, M\mathbf{1} = \mathbf{1}, Tr(M) = k\,. \end{aligned} \quad (6)$$

It is not trivial to solve our new objective J_{opt} in Eq. (6). We will propose a novel algorithm to solve the new objective. Before that, we first show the interesting connection between our model and spectral clustering.

2.2 Connections to Spectral Graph Cut Models

THEOREM 1. *In Problem (5), if $r \to \infty$ and W is doubly stochastic, then Problem (5) is equivalent to spectral clustering.*

Proof: As illustrated in [25], given a graph G with n points and its affinity matrix $W \in \Re^{n \times n}$, the definition of graph cut is:

$$\begin{aligned} J &= \sum_{1 \leq p \leq q \leq k} \frac{s(C_p, C_q)}{\rho(C_p)} + \frac{s(C_p, C_q)}{\rho(C_q)} \\ &= \sum_{l=1}^{k} \frac{s(C_l, \bar{C}_l)}{\rho(C_l)}\,, \end{aligned} \quad (7)$$

where k is the number of clusters, C_l is the l-th cluster and $\bar{C}_l$ is the complement subset of cluster C_l in graph G, $s(M, N) = \sum\limits_{m \in M} \sum\limits_{n \in N} W_{mn}$.

For Ratio Cut, the function $\rho(C_l)$ in Eq. (7) is defined as:

$$\rho_{RCut}(C_l) = |C_l|\,. \quad (8)$$

We introduce an indicator vector $\mathbf{q}_l \in \Re^n (l = 1, 2, \cdots, k)$, such that the i-th element of $\mathbf{q}_l$ equals to 1 if the i-th point in graph G belongs to the l-th cluster, and 0 otherwise.

With the indicator vector $\mathbf{q}_l$, we have:

$$s(C_l, \bar{C}_l) = \sum_{i \in C_l} \sum_{j \in \bar{C}_l} W_{ij} = \mathbf{q}_l^T (D_W - W) \mathbf{q}_l\,.$$

Along with Eq. (8), we can rewrite the cut function of Ratio Cut in Eq. (7) as:

$$J_{RCut} = \sum_{l=1}^{k} \frac{\mathbf{q}_l^T (D_W - W) \mathbf{q}_l}{\mathbf{q}_l^T \mathbf{q}_l}\,. \quad (9)$$

Define matrix $G \in \Re^{n \times k}$ such that $\mathbf{g}_l = \mathbf{q}_l$. We introduce an indicator matrix F:

$$F = G(G^T G)^{-\frac{1}{2}}\,, \quad (10)$$

Assume $x_i \in C_i$ and $x_j \in C_j$. For the F matrix in Eq. (10), we can observe that if $C_i = C_j$, then $f_{ij} = \frac{1}{\sqrt{n_i}}$; otherwise $f_{ij} = 0$, where n_i denotes the number of data points in the i-th cluster.

Thus, the cut function (9) can be written as:

$$J_{RCut} = Tr(F^T (D_W - W) F)\,. \quad (11)$$

If W is doubly stochastic, then $D_W = I$. In this case, Normalized Cut is equivalent to Ratio Cut and tackles the following problem:

$$\begin{aligned} &\min_F Tr(F^T (I - W) F) \\ \Longrightarrow\ &\max_F Tr(F^T W F) \\ \Longrightarrow\ &\min_F \left\| F F^T - W \right\|_F^2\,. \end{aligned} \quad (12)$$

Let $M = FF^T$, the Problem (12) becomes:

$$\min_M \|M - W\|_F^2\,. \quad (13)$$

We can directly find that $m_{ij} = \frac{1}{n_i}$ when $C_i = C_j$; while $m_{ij} = 0$ when $C_i \neq C_j$, where $x_i \in C_i$ and $x_j \in C_j$.

Thus, the spectral clustering problem is to find a matrix M minimizing Problem (13). From the definition of M, we can obtain some properties of M that $M \geq 0$, $M^T = M$ and $M\mathbf{1} = \mathbf{1}$, thus it is doubly stochastic. As for $Tr(M)$, we have:

$$Tr(M) = Tr(FF^T) = Tr(G(G^T G)^{-1} G) = k\,. \quad (14)$$

□

In practice, matrix W may not always be doubly stochastic. Given affinity matrix W_0, we can learn a doubly stochastic matrix W as the initialization by solving:

$$\min_{W \geq 0, W = W^T, W\mathbf{1} = \mathbf{1}} \|W - W_0\|_F^2\,. \quad (15)$$

The above problem is the same as the problem solved in [29] and also a special case of our proposed objective in Eq. (6) when both parameter γ and r are set as 0.

Previous method in [29] can only learn a doubly stochastic matrix without clear clustering structure. After adding the regularization terms, our new objective can achieve a better doubly stochastic matrix with clear clustering structure to improve the clustering results.

3. OPTIMIZATION ALGORITHM

We use the Augmented Lagrange Multiplier (ALM) optimization strategy [4] to solve our new objective J_{opt} in Eq. (6).

Here we introduce a slack variable L that $L = I - M$, then Problem (6) can be rewritten as:

$$\begin{aligned} &\min_M \|M - W\|_F^2 + \gamma \|L\|_* + r \|M\|_F^2 \\ s.t. \quad &M \geq 0, M = M^T, M\mathbf{1} = \mathbf{1}, Tr(M) = k, \\ &I - M = L\,, \end{aligned} \quad (16)$$

and Problem (16) is equivalent to:

$$\begin{aligned} &\min_M \|M - W\|_F^2 + \gamma \|L\|_* \\ &+ \frac{\mu}{2} \left\| I - M - L + \frac{1}{\mu}\Lambda \right\|_F^2 + r \|M\|_F^2 \\ s.t. \quad &M \geq 0, M = M^T, M\mathbf{1} = \mathbf{1}, Tr(M) = k\,, \end{aligned} \quad (17)$$

where $\Lambda \in \Re^{n \times n}$ is the Lagrange multiplier and μ is the penalty parameter for Eq. (17).

Compared with Problem (6), Problem (17) is easier to solve since the trace norm term $\gamma \|L\|_*$ is now independent to M. We introduce an efficient alternating algorithm to tackle Problem (17).

The first step is fixing L and solving M, thus Problem (17) becomes:

$$\begin{aligned} &\min_M \|M - W\|_F^2 + \frac{\mu}{2} \left\| I - M - L + \frac{1}{\mu}\Lambda \right\|_F^2 \\ &+ r \|M\|_F^2 \\ s.t. \quad &M \geq 0, M = M^T, M\mathbf{1} = \mathbf{1}, Tr(M) = k\,. \end{aligned} \quad (18)$$

Let

$$T = \frac{1}{\mu + 2r}(2W + \mu(I - L + \frac{1}{\mu}\Lambda))\,,$$

the Problem (18) can be rewritten as:

$$\begin{aligned} &\min_{M} \|M - T\|_F^2 \\ s.t. \quad & M \geq 0, M = M^T, M\mathbf{1} = \mathbf{1}, Tr(M) = k\,. \end{aligned} \tag{19}$$

Since the constraint on $Tr(M)$ is only concerned with the diagonal elements, Problem (19) can be divided into two subproblems:

$$\min_{M} \|M - T\|_F^2\,,\ s.t.\ M = M^T, M\mathbf{1} = \mathbf{1}\,, \tag{20}$$

and

$$\min_{M} \|M - T\|_F^2\,,\ s.t.\ M \geq 0, \mathbf{m}^T\mathbf{1} = k\,, \tag{21}$$

where $\mathbf{m} = diag(M)$ and $diag(M)$ denotes a vector formed by the diagonal elements of M.

Our strategy is to solve two subproblems, Problem (20) and Problem (21) alternately, and let their solutions project mutually. In each iteration, we solve Problem (20) first and let its solution M_1 to be the T matrix in Problem (21), afterwards we solve Problem (21) and let its solution M_2 play the role of matrix T in Problem (20). We solve these two problems alternately and iteratively until M converges.

According to Von Neumann's successive projection lemma [16], this mutual projection strategy we use will converge to the cross of two subspaces formed by Problems (20) and (21). The lemma theoretically ensures that the solution of the alternate projection strategy ultimately converges onto the global optimal solution of Problem (19).

According to Lemma 2 in Appendix A, the optimal solution of Problem (20) is as follows:

$$M = K + \frac{n + \mathbf{1}^T K \mathbf{1}}{n^2}\mathbf{1}\mathbf{1}^T - \frac{1}{n}K\mathbf{1}\mathbf{1}^T - \frac{1}{n}\mathbf{1}\mathbf{1}^T K\,, \tag{22}$$

where $K = \frac{T+T^T}{2}$.

As far as Problem (21) is concerned, firstly we let matrix T in Problem (21) equal to the solution of M to Problem (20) (shown in Eq. (22)). Then according to Lemma 3 in Appendix B, the optimal solution of Problem (21) is:

$$M = T_+, \quad \mathbf{m} = (\mathbf{t} - \lambda\mathbf{1})_+\,. \tag{23}$$

Alternately we solve Problems (20) and (21) till M converges onto its global optimal solution.

The second step is fixing M and solving L, then Problem (17) becomes:

$$\min_{L} \gamma \|L\|_* + \frac{\mu}{2}\left\|I - M - L + \frac{1}{\mu}\Lambda\right\|_F^2\,. \tag{24}$$

Let $N = (I - M + \frac{1}{\mu}\Lambda)$, Problem (24) becomes

$$\min_{L} \frac{\gamma}{\mu}\|L\|_* + \frac{1}{2}\|L - N\|_F^2\,. \tag{25}$$

According to [5], the solution of Problem (25) is:

$$L = UDiag((\sigma_i - \frac{\gamma}{\mu})_+)V^T\,, \tag{26}$$

where the singular value decomposition of N is $N = U\Sigma V^T$. $Diag((\sigma_i - \frac{\gamma}{\mu})_+)$ is a diagonal matrix with i-th diagonal element as $(\sigma_i - \frac{\gamma}{\mu})_+$.

Our algorithm to solve the new objective is summarized in Algorithm 1.

Convergence and Complexity Analysis: The convergence of ALM algorithm was proved and discussed in previous papers. Please refer to the literature therein [3, 22]. Because our new objective is convex, our algorithm converges to the global optimum.

Algorithm 1 Proposed Algorithm

```
Input:
  The given affinity matrix $W \in \Re^{n\times n}$;
  The number of clusters $k$;
Output:
  The learnt similarity matrix $M^*$;
  Initialization:
  Let the count number of iteration $t = 0$. Randomly initialize
  matrix $L^{(0)} \in \Re^{n\times n}$ and set the Lagrange multiplier matrix
  $\Lambda^{(0)} = \mathbf{0} \in \Re^{n\times n}$. Set the penalty parameter $\mu^{(0)} = 0.1$,
  and the increment step parameter $\rho > 1$;
  Preprocessing:
  Solve Problem (15) to pre-process $W$ and get a doubly stochastic
  matrix $M^{(0)}$. Let $W = M^{(0)}$.
  while not converge do
    1. Update $M^{(t+1)}$ using Eq. (22) Eq. (23) alternatively via the
    successive projection strategy;
    2. Update $L^{(t+1)}$ by Eq. (26);
    3. Update $\Lambda^{(t+1)} = \Lambda^{(t)} + \mu^{(t)}(I - M^{(t+1)} - L^{(t+1)})$;
    4. Update $\mu^{(t+1)} = \rho\mu^{(t)}$;
    5. Update $t = t + 1$;
  end while
  Return: $M^*$;
```

In Algorithm 1, the slowest step is Step 2 for updating L. It requires $O(n^3)$ time to implement the singular vector decomposition, where n is the number of samples in the dataset. This time complexity is comparable to that of spectral clustering.

4. CONNECTIONS TO K-MEANS CLUSTERING

Here in this section, we will further discuss the connection between our model and the K-means clustering problem.

THEOREM 2. *In Problem (5), if $r \to \infty$ and $W = X^T X$, then Problem (5) is equivalent to K-means clustering.*

Proof: Given a set of data points $X = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n] \in \Re^{d\times n}$, the K-means clustering problem is meant to partition X into k $(1 \leq k \leq n)$ clusters $C = \{C_1,\ C_2,\ ...,\ C_k\}$ such that the sum of the within cluster variance is minimized [13]. That is to say, the objective function of the K-means problem is:

$$\min_{C} \sum_{i=1}^{k} \sum_{\mathbf{x}_j \in \mathbf{s}_i} \|\mathbf{x}_j - \mu_i\|^2 \tag{27}$$

where μ_i is the mean of data points belonging to C_i.

If we introduce two matrix $U \in \Re^{d\times k}$ and $G \in \Re^{n\times k}$, where $U = [\mu_1,\ \mu_2,\ ...,\ \mu_k]$ and G indicates the clustering indices , then Eq. (27) can be reformulated as:

$$\begin{aligned} &\min_{G\in Ind, U} \left\|X - UG^T\right\|_F^2 \\ \Longleftrightarrow &\min_{G\in Ind, U} Tr(GU^T UG^T) - 2Tr(X^T UG^T) \end{aligned} \tag{28}$$

Since the solution of U w.r.t. X and G is $U = XG(G^TG)^{-1}$, we have $Tr(GU^TUG^T) = Tr(X^TUG^T)$, thus Eq. (28) can be written as:

$$\begin{aligned} & \max_{G\in Ind,U} Tr(X^TUG^T) \\ \Longleftrightarrow & \max_{G\in Ind} Tr((G^TG)^{-\frac{1}{2}}G^T(X^TX)G(G^TG)^{-\frac{1}{2}}) \\ \Longleftrightarrow & \max_{F} Tr(F^T(X^TX)F) \end{aligned} \tag{29}$$

where $F = G(G^TG)^{-\frac{1}{2}}$.

Note that the $Tr(FF^TFF^T) = Tr(FF^T) = Tr(G(G^TG)^{-1}G) = k$, so Problem (29) is equivalent to:

$$\min_{F} \left\| FF^T - X^TX \right\|_F^2 \tag{30}$$

Let $M = FF^T$, then Problem (30) can be rewritten as:

$$\min_{M\in\mathcal{D}} \left\| M - X^TX \right\|_F^2 \tag{31}$$

where $M \in \mathcal{D}$ indicates some constraints on the M matrix. So the K-means clustering problem is to find a matrix M meeting some requirements such that M can minimize Problem (31).

Let's take further observe the properties of M.

From the definition of M, where $M = FF^T$, we can directly find that $m_{ij} = \frac{1}{n_i}$, if $\mathbf{x}_i$ and $\mathbf{x}_j$ belongs to the same cluster; and $m_{ij} = 0$ otherwise. Also, it's apparent that $M \geq 0$, $M^T = M$ and $M\mathbf{1} = \mathbf{1}$, that is to say, M is doubly stochastic. Moreover, we have $Tr(M) = Tr(FF^T) = k$. □

5. EXPERIMENTAL RESULTS

Our structured doubly stochastic model (SDS) can uncover the clustering structure and directly provide the clustering results, thus the clustering performance using doubly stochastic matrix can be enhanced. In this section we evaluate the clustering performance of our method on both synthetic and benchmark datasets, and compare them to the related doubly stochastic model and spectral clustering methods.

Moreover, according to the discussion in the previous section, our model possesses interesting connection with K-means clustering, we also conduct experiments to test whether our model provides an approach to better solving the K-means clustering problem.

5.1 Experiments on Clustering

In this subsection, we conduct clustering experiments on our method and several related method. Our goal is to test whether the structured doubly stochastic matrix learned in our model is beneficial to improve the clustering performance under different circumstances.

5.1.1 Experimental Settings on Clustering

To evaluate the clustering performance of our method, we compare with spectral clustering, *i.e,* Ratio Cut and Normalized Cut, as well as the doubly stochastic normalization (DSN) method [29].

All comparing methods require an affinity matrix as the input. We construct the input affinity matrix with the self-tune Gaussian method [7], where the number of neighbors is set to be 5 and the value of σ is self-tuned. Moreover, we let the input matrix W of our method to be initialized as shown in Eq. (15) such that W is doubly stochastic. In the experiment, we set the number of clusters to be the ground truth in each dataset. In our method, we set parameter μ = 0.1, ρ = 1.1 and r to be tuned in the range of $\{10^0,\ 10^{0.5},\ ...,\ 10^5\}$.

For all methods requiring K-means as the post-processing step, including Ratio Cut, Normalized Cut and DSN, we give them the same 100 random initializations and compute their respective best initialization vector w.r.t K-means objective function value. Since their performance is unstable with different initialization, we only report their respective best results in the 100 times repetition. For DSN method, we set the number of iteration as 3000 so as to get a good doubly stochastic matrix for clustering.

All experiments are conducted on a Windows system with Intel Core i7-3770 Processor (8M Cache, 3.40 GHz).

The evaluation of different methods is based on two clustering metrics: accuracy and NMI (Normalized Mutual Information).

Accuracy is the percentage of the correctly assigned labels. NMI is short for the normalized mutual information. Let L denote the real label vector in a certain dataset, while L' denotes the predicted one, then

$$NMI(L,L^{'}) = \frac{I(L,L^{'})}{\max(H(L),H(L'))}\,, \tag{32}$$

where $I(L,L^{'})$ is the mutual information between L and $L^{'}$:

$$I(L,L^{'}) = \sum_{l_i\in L}\sum_{l_j^{'}\in L'} p(l_i,l_j^{'}) \log\frac{p(l_i,l_j^{'})}{p(l_i)p(l_j^{'})}\,, \tag{33}$$

and $H(L)$ is the entropy of L:

$$H(L) = -\sum_{i=1}^{n} p(l_i)\log p(l_i)\,. \tag{34}$$

5.1.2 Clustering Experiments on Synthetic Data

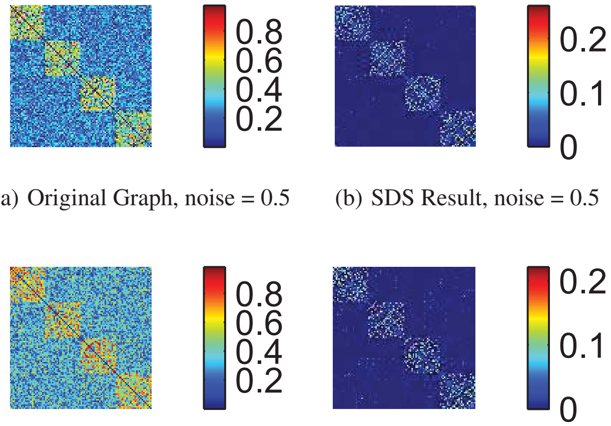
First of all, we conduct clustering experiments on the synthetic data as a sanitary check. The synthetic dataset is a 100×100 matrix with four 25×25 block matrices diagonally arranged. The data within each block denotes the probability of two corresponding points from one same cluster to be connected; while the data outside all the blocks denotes the probability of pair-wise data points from different clusters to be connected, *i.e.,* noise (which should be 0 in the ideal clustering data). The probability values within each block are randomly generated in the range of $(0,\ 1)$; while the noise data is randomly generated in the range of $(0,\ c)$, where c is set to be 0.5 and 0.6 respectively. What's more, to make this clustering task more challenging, we randomly pick out 25 noise data and set their value to be 1.

Fig. 1 shows the original random matrix and corresponding clustering results of SDS. We can notice that our model performs well in this task. In our approach, we successfully learn a structured doubly stochastic matrix with explicit block structure, which divides the data into exactly four clusters. After adding high-level disturbance in the random data, our method still effectively recovers the clustering structure, which indicates the robustness of our model.

When the noise ratio is 0.5, our method works out an almost perfect structured doubly stochastic matrix with four clear blocks. As the noise increases, the block structure in the original data blurs, but our model is still able to detect the intrinsic cluster structure from the data.

5.1.3 Clustering Experiments on Benchmark Datasets

We evaluated the proposed double stochastic method on 7 benchmark datasets: AR [14], FERET [21], Yale [9], ORL [23], Carcino-



(a) Original Graph, noise = 0.5

(b) SDS Result, noise = 0.5

(c) Original Graph, noise = 0.6

(d) SDS Result, noise = 0.6

Figure 1: Illustration of our clustering results on the block diagonal synthetic data with different settings of noise. On the left column shows the graph structure of the original data generated in the experiment. Figures on the right denote the structure of the doubly stochastic matrix obtained in our SDS model.

Table 1: Descriptions of benchmark datasets used in our experiments.

Datasets	Number of Instances	Dimensions	Classes
AR	840	768	120
FERET	1400	1296	200
Yale	2414	1024	38
ORL	400	1024	40
Carcinomas	174	9182	11
SRBCTML	83	2308	4
LEUML	72	3571	2

mas [24, 28], SRBCTML [10] and LEUML[1]. The detailed description of these datasets is summarized in Table 1.

The clustering performance comparison is summarized in Table 2. Results in Table 2 suggest that our method works very well on real benchmark datasets. Our SDS method maintains a high potential to outperform other methods on these distinct datasets. The theoretical proof in the methodology section indicates the connection between our our model and spectral clustering problem, while the experimental results here verify SDS's better clustering performance. This suggests SDS has the ability to better solve the spectral clustering problem. Compared with the up-to-date method, our method gains an obvious advantage over DSN. DSN only holds constraints on the doubly stochastic property of the learned graph but not its cluster structure, thus cannot get the optimal clustering results. On the contrary, our model learns a novel doubly stochastic matrix with explicit block structure, which performs better in clustering.

5.1.4 Clustering Results Analysis

To further analyze the clustering performance, we draw the graph learned from different methods and compare their structure. We compare the graphs represented by the doubly stochastic matrix learned in DSN and SDS, respectively, and also, we display the graph constructed via self-tune Gaussian method, which is the input for spectral clustering. To explicitly view the graph structure, here we use the two datasets with relatively small number of classes and samples as the example, *i.e.,* LEUML and SRBCTML. We present the graphs in Fig. 2. In each graph, row and columns are reshuffled such that samples from the same cluster are put together, which makes the cluster structure more clear to observe in the graph. For LEUML and SRBCTML, the r value we set for SDS is 10^0 and $10^{0.5}$, respectively. We can notice that the graph learned by SDS maintains the most clear block structure, and elements in each cluster tend to have similar value. These observations coincide with our theoretical analysis. Especially in the LEUML dataset, the doubly stochastic matrix learned by DSN is quite noisy, which leads to bad clustering performance shown in Table 2. Whereas, due to the low-rank constraint on the graph Laplacian matrix, the doubly stochastic matrix learned in our SDS model has more clear block structure, which accounts for the better clustering results obtained from SDS.

5.2 Experiments on K-means Task

In the K-means clustering problem, a "better" solution signifies a smaller objective function value as well as a higher clustering accuracy. Since K-means problem is non-convex, the quality of initialization is crucial in performing K-means clustering. In this subsection, we conduct experiments on both synthetic and real benchmark datasets to demonstrate the contribution of our method in better solving the K-means problem.

5.2.1 Experimental Settings on K-means Task

The experimental settings are similar to the settings in the clustering experiments. The different part is that in this section, we assign the clustering indicator obtained in our method as an initialization for K-means and see if the K-means clustering problem can be better solved with our initialization. For our method, we use $W = X^T X$ as the input matrix.

Still, the number of clusters is set to be the ground truth in each dataset. When implementing K-means clustering, unless specified otherwise, the following settings are adopted: we use 100 random initializations and record the average as well as best result w.r.t. K-means objective function value in the 100 times repetition.

The evaluation is based on three metrics: accuracy, NMI and the K-means objective function value.

5.2.2 K-means Experiments on Synthetic Data

In the synthetic experiment, our toy data is a randomly generated multi-cluster matrix. Data points in each cluster are sampled i.i.d. from the Gaussian distribution $\mathcal{N}(0, 1)$. In our experiment, we set the number of clusters to be 100, number of samples to be 1000, while the dimensionality to be $\{2, 50, 1000\}$ respectively. Our goal is to partition these clusters apart with K-means method. In the beginning we run K-Means for 10000 times and record the minimum K-means objective value and the corresponding clustering accuracy. Then we run our method once by setting the input matrix to be $W = X^T X$ and use the obtained clustering results as an initialization index vector for K-means and compute the same metrics. Comparison results are summarized in Table 3, which indicates apparent superiority of our method over K-means. It shows that even after 10000 times run, the minimum K-means objective value and clustering accuracy obtained by K-means are still far behind the result obtained by our method with just one run. This verifies that our method is able to better solve the K-means problem.

5.2.3 K-means Experiments on Benchmark Datasets

Still, we evaluate our model on the 7 benchmark datasets shown

[1] http://www2.stat.duke.edu/courses/Spring01/sta293b/datasets.html

Table 2: Experimental results comparison of clustering on benchmark datasets.

		Ratio Cut	Normalized Cut	DSN	SDS
ACCURACY	AR	0.358	0.358	0.382	**0.404**
	FERET	0.249	0.255	0.279	**0.280**
	Yale	0.387	0.396	0.439	**0.448**
	ORL	0.653	0.625	0.605	**0.663**
	Carcinomas	**0.724**	0.695	0.690	0.718
	SRBCTML	0.434	0.434	0.410	**0.446**
	LEUML	0.903	0.903	0.542	**0.917**
NMI		Ratio Cut	Normalized Cut	DSN	SDS
	AR	0.677	0.700	0.705	**0.706**
	FERET	0.647	0.674	0.682	**0.683**
	Yale	0.561	0.570	0.604	**0.605**
	ORL	0.799	0.794	0.783	**0.814**
	Carcinomas	**0.719**	0.697	0.707	0.712
	SRBCTML	0.169	0.160	0.132	**0.187**
	LEUML	0.547	0.547	0.079	**0.585**

Table 3: K-means objective function value and clustering results comparison on synthetic datasets.

	K-means Min_obj	K-means Min_obj with SDS Initialization	K-means Accuracy (min_obj)	K-means Accuracy with SDS Initialization
d = 2	1.60	**1.15**	0.713	**0.822**
d = 50	420.08	**0.14**	0.861	**1.000**
d = 1000	10292.00	**2.7305**	0.863	**1.000**

Table 4: Experimental results comparison for K-means problem on benchmark datasets.

		K-means Min_obj	K-means Average	K-means with SDS Initialization
K-means Objective Function Value	AR	7982.58	8395.09 ± 124.98	**7562.62**
	FERET	28684.82	29101.97 ± 215.69	**26420.99**
	Yale	39578.65	40398.72 ± 337.14	**39381.64**
	ORL	6223.80	6633.50±156.62	**5943.80**
	Carcinomas	47160.00	48682.00±692.63	**46787.00**
	SRBCTML	**5747.30**	5982.00±161.18	**5747.30**
	LEUML	**11364.00**	11398.00±78.78	**11364.00**
Accuracy		K-means Min_obj	K-means Average	SDS
	AR	0.310	0.285 ± 0.011	**0.343**
	FERET	0.206	0.200 ± 0.005	**0.234**
	Yale	0.111	0.110 ± 0.006	**0.114**
	ORL	0.568	0.490±0.031	**0.638**
	Carcinomas	0.672	0.571±0.050	**0.695**
	SRBCTML	0.374	0.446±0.070	**0.458**
	LEUML	0.708	0.740±0.058	**0.736**
NMI		K-means Min_obj	K-means Average	SDS
	AR	0.640	0.621 ± 0.009	**0.688**
	FERET	0.598	0.581 ± 0.006	**0.638**
	Yale	0.170	0.163 ± 0.009	**0.178**
	ORL	0.754	0.711±0.018	**0.781**
	Carcinomas	0.648	0.591±0.042	**0.704**
	SRBCTML	0.106	0.187±0.081	**0.261**
	LEUML	0.182	0.225±0.112	**0.237**

in Table 1. We summarize the K-means performance comparison of K-means clustering and our method in Table 4. From Table 4, we can notice that our method improves the performance of K-means on real benchmark datasets. On all dataset, our SDS method performs equally or even better than K-means clustering. These results demonstrates that our model makes a good way to better solve the K-means clustering problem. By adopting the cluster indicator learned in our model as the initialization, not only is the K-means objective function value reduced, but the clustering performance is also boosted to a large extent.

5.3 Experiments on Convergence Analysis

In this subsection, we analyze the influence of parameter r in Eq. (6) to the convergence of our algorithm. To save space, we just take two datasets, Carcinomas and ORL, as an example. We apply our method to these benchmark datasets with three different r values (*i.e.,* 10, 10^3 and 10^5) and record the objective value of our model in each iteration.

The convergence results are presented in Fig. 3. We can notice that no matter what the r value is, our model always converges within about 80 iterations, which indicates the fast convergence of our algorithm.

6. CONCLUSIONS

In this paper, we proposed a novel structured doubly stochastic model with rank constraint on the graph Laplacian matrix. The doubly stochastic matrix learned in our model possesses explicit clustering structure, from which we can immediately partition data points into k connected components, where k is the number of clusters. The doubly stochastic property guarantees the effectiveness of the learnt similarity matrix while the rank constraint on the graph Laplacian matrix enhances the clustering ability. The quality of the learnt graph was verified by extensive experimental results, which suggested the feasibility of our model. What's more, we theoretically and empirically proved that our method made its own contribution in better solving the K-means and spectral clustering problem.

Appendix A

LEMMA 2. *The following gives the global optimal solution to Problem (20):*

$$M = K + \frac{n + \boldsymbol{1}^T K \boldsymbol{1}}{n^2}\boldsymbol{1}\boldsymbol{1}^T - \frac{1}{n}K\boldsymbol{1}\boldsymbol{1}^T - \frac{1}{n}\boldsymbol{1}\boldsymbol{1}^T K,$$
$$K = \frac{T+T^T}{2}\,.$$

Proof: With the Lagrangian function, Problem (20) can be rewritten as:

$$\min_M \frac{1}{2}\|M-T\|_F^2 - (\lambda^T(M\mathbf{1}-\mathbf{1})) - Tr(\Lambda^T(M^T-M))\,, \quad (35)$$

where $\Lambda \in \Re^{n\times n}$ and $\lambda \in \Re^n$ are Lagrange multipliers.

Taking derivative w.r.t. M and set it to 0, we have:

$$M - T - (\Lambda^T - \Lambda) - \lambda\mathbf{1}^T = 0\,. \quad (36)$$

Compute transpose on both sides of Eq. (36), we have:

$$M - T^T + (\Lambda^T - \Lambda) - \mathbf{1}\lambda^T = 0\,. \quad (37)$$

Subtracting Eq. (36) from Eq. (37), we get:

$$T - T^T + 2(\Lambda^T - \Lambda) + \lambda\mathbf{1}^T - \mathbf{1}\lambda^T = 0$$
$$\Longrightarrow -(\Lambda^T - \Lambda) = \tfrac{1}{2}(T - T^T + \lambda\mathbf{1}^T - \mathbf{1}\lambda^T)\,. \quad (38)$$

Combining Eq. (36) with Eq. (38), we further get:

$$2(M-T) + (T - T^T + \lambda\mathbf{1}^T - \mathbf{1}\lambda^T) - 2\lambda\mathbf{1}^T = 0$$
$$\Longrightarrow 2M - T - T^T - \lambda\mathbf{1}^T - \mathbf{1}\lambda^T = 0\,. \quad (39)$$

Multiply the vector $\mathbf{1}$ on both sides of Eq. (39), we have:

$$2\mathbf{1} - T\mathbf{1} - T^T\mathbf{1} - n\lambda - \mathbf{1}\lambda^T\mathbf{1} = 0\,. \quad (40)$$

Since $\lambda^T\mathbf{1}$ is a number, it is apparent that $(\lambda^T\mathbf{1})^T = \lambda^T\mathbf{1}$, thus:

$$2\mathbf{1} - T\mathbf{1} - T^T\mathbf{1} - n\lambda - \mathbf{1}\mathbf{1}^T\lambda = 0\,. \quad (41)$$

From Eq. (41) we can obtain the solution of λ as follows:

$$\lambda = (\mathbf{1}\mathbf{1}^T + nI)^{-1}(2\mathbf{1} - T\mathbf{1} - T^T\mathbf{1})\,. \quad (42)$$

To enhance the computing speed, we compute the inverse term in Eq. (42) by means of the Woodbury formula [27]:

$$(A+UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1}+VA^{-1}U)^{-1}VA^{-1}\,. \quad (43)$$

Thus the inverse term in Eq. (42) can be rewritten as:

$$(\mathbf{1}\mathbf{1}^T + nI)^{-1} = (-\frac{1}{2n^2}\mathbf{1}\mathbf{1}^T + \frac{1}{n}I)\,. \quad (44)$$

From Eq. (44), we can rewrite Eq. (42) as follows:

$$\begin{aligned} \lambda &= (-\frac{1}{2n^2}\mathbf{1}\mathbf{1}^T + \frac{1}{n}I)(2\mathbf{1} - T\mathbf{1} - T^T\mathbf{1}) \\ &= \frac{1}{n}\mathbf{1} - (-\frac{1}{2n^2}\mathbf{1}\mathbf{1}^T + \frac{1}{n}I)(T+T^T)\mathbf{1}\,. \quad (45) \end{aligned}$$

Plugging the solution of λ in Eq. (45) to Eq. (39), we get:

$$\begin{aligned} 2M &= T + T^T + \lambda\mathbf{1}^T + \mathbf{1}\lambda^T \\ &= T + T^T + \frac{2}{n}\mathbf{1}\mathbf{1}^T - (-\frac{1}{2n^2}\mathbf{1}\mathbf{1}^T + \frac{1}{n}I)(T+T^T)\mathbf{1}\mathbf{1}^T \\ &\quad -\mathbf{1}\mathbf{1}^T(T+T^T)(-\frac{1}{2n^2}\mathbf{1}\mathbf{1}^T + \frac{1}{n}I) \\ &= T + T^T + \frac{2}{n}\mathbf{1}\mathbf{1}^T + \frac{\mathbf{1}^T T^T\mathbf{1}}{n^2}\mathbf{1}\mathbf{1}^T + \frac{\mathbf{1}^T T\mathbf{1}}{n^2}\mathbf{1}\mathbf{1}^T \\ &\quad -\frac{1}{n}T\mathbf{1}\mathbf{1}^T - \frac{1}{n}T^T\mathbf{1}\mathbf{1}^T - \frac{1}{n}\mathbf{1}\mathbf{1}^T T^T - \frac{1}{n}\mathbf{1}\mathbf{1}^T T \end{aligned}$$

Let $K = \frac{T+T^T}{2}$, then we can rewrite the above equation as:

$$M = K + \frac{n + \mathbf{1}^T K\mathbf{1}}{n^2}\mathbf{1}\mathbf{1}^T - \frac{1}{n}K\mathbf{1}\mathbf{1}^T - \frac{1}{n}\mathbf{1}\mathbf{1}^T K\,, \quad (46)$$

which is the global optimal solution of Problem (20). Obviously, M is symmetric and

$$M\mathbf{1} = K\mathbf{1} + \frac{n + \mathbf{1}^T K\mathbf{1}}{n}\mathbf{1} - K\mathbf{1} - \frac{1}{n}\mathbf{1}\mathbf{1}^T K\mathbf{1} = \mathbf{1}\,, \quad (47)$$

which means that the solution of M in Eq. (46) meets the requirements of Problem (20).

Specially, when T is symmetric, the solution of M is:

$$M = T + \frac{n + \mathbf{1}^T T\mathbf{1}}{n^2}\mathbf{1}\mathbf{1}^T - \frac{1}{n}T\mathbf{1}\mathbf{1}^T - \frac{1}{n}\mathbf{1}\mathbf{1}^T T\,. \quad (48)$$

□

Appendix B

LEMMA 3. *The following gives the global optimal solution to Problem (21):*

$$M = T_+,$$
$$\mathbf{m} = (\mathbf{t} - \lambda\boldsymbol{1})_+\,.$$

where $\mathbf{m} = diag(M)$.

Proof: Require $M = T_+$, then Problem (21) could be rewritten as follows:

$$\min_{\mathbf{m}} \|\mathbf{m} - \mathbf{t}\|_2^2 \quad (49)$$
$$s.t. \quad \mathbf{m} \geq 0, \mathbf{m}^T\mathbf{1} = k\,, \quad (50)$$

where $\mathbf{m} = diag(S)$ and $\mathbf{t} = diag(T)$.

Similarly, we can use the Lagrangian function to solve Problem (49) and define:

$$G(\mathbf{m}, \lambda, \eta) = \frac{1}{2} \|\mathbf{m} - \mathbf{t}\|_2^2 - \lambda(\mathbf{m}^T\mathbf{1} - k) - \eta\mathbf{m}^T , \tag{51}$$

where $\lambda \in \Re^n$ and $\eta \in \Re^n$ are Lagrange multipliers.

Taking derivative w.r.t. $\mathbf{m}$ and set it to 0, then we can solve $\mathbf{m}$ in Problem (51) as follows:

$$\begin{aligned} & \mathbf{m} - \mathbf{t} - \lambda\mathbf{1} - \eta = 0 \\ \Longrightarrow \quad & \mathbf{m} = (\mathbf{t} - \lambda\mathbf{1})_+ . \end{aligned} \tag{52}$$

□

7. REFERENCES

[1] R. Angelova and G. Weikum. Graph-based text classification: Learn from your neighbors. *annual international ACM SIGIR conference on Research and development in information retrieval*, pages 485–492, 2006.

[2] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[3] D. P. Bertsekas. *Constrained optimization and lagrange multiplier methods*. Athena Scientific, 1996.

[4] D. P. Bertsekas et al. Augmented lagrangian and differentiable exact penalty methods. 1981.

[5] J.-F. Cai, E. J. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

[6] A. Celikyilmaz, M. Thint, and Z. Huang. A graph-based semi-supervised learning for question-answering. *the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 719–727, 2009.

[7] W.-Y. Chen, Y. Song, H. Bai, C.-J. Lin, and E. Y. Chang. Parallel spectral clustering in distributed systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(3):568–586, 2011.

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

[9] A. Georghiades, P. Belhumeur, and D. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 23(6):643–660, 2001.

[10] J. Khan, J. S. Wei, M. Ringner, L. H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C. R. Antonescu, C. Peterson, et al. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature medicine*, 7(6):673–679, 2001.

[11] D. Luo, C. Ding, and H. Huang. Forging The Graphs: A Low Rank and Positive Semidefinite Graph Learning Approach. *Advances in Neural Information Processing Systems (NIPS)*, pages 2969–2977, 2012.

[12] D. Luo, C. Ding, F. Nie, and H. Huang. Cauchy graph embedding. *International Conference on Machine Learning*, pages 553–560, 2011.

[13] J. MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA., 1967.

[14] A. Martinez and R. Benavente. The ar face database. Technical report, CVC Technical report, 1998.

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

[16] J. V. Neumann. *Functional Operators*, volume 2. 1950.

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

[18] F. Nie, H. Wang, H. Huang, and C. Ding. Unsupervised and semi-supervised learning via l1-norm graph. In *IEEE Conference on Computer Vision*, pages 2268–2273, 2011.

[19] F. Nie, X. Wang, and H. Huang. Clustering and projected clustering via adaptive neighbor assignment. *The 20th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2014)*, pages 977–986, 2014.

[20] F. Nie, X. Wang, M. Jordan, and H. Huang. The constrained laplacian rank algorithm for graph-based clustering. *Thirtieth AAAI Conference on Artificial Intelligence (AAAI 2016)*, pages 1969–1976, 2016.

[21] P. Phillips, H. Wechsler, J. Huang, and P. Rauss. The FERET database and evaluation procedure for face-recognition algorithms. *Image and Vision Computing*, 16(5):295–306, 1998.

[22] M. J. D. Powell. *A method for nonlinear constraints in minimization problems*. In R. Fletcher, editor, Optimization. Academic Press, London and New York, 1969.

[23] F. S. Samaria and A. C. Harter. Parameterisation of a stochastic model for human face identification. In *Applications of Computer Vision, 1994., Proceedings of the Second IEEE Workshop on*, pages 138–142. IEEE, 1994.

[24] A. I. Su, J. B. Welsh, L. M. Sapinoso, S. G. Kern, P. Dimitrov, H. Lapp, P. G. Schultz, S. M. Powell, C. A. Moskaluk, H. F. Frierson, and G. M. Hampton. Molecular Classification of Human Carcinomas by Use of Gene Expression Signatures. *Cancer Res.*, 61(20):7388–7393, 2001.

[25] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[26] F. Wang, P. Li, and A. Konig. Learning a bi-stochastic data similarity matrix. *IEEE International Conference on Data Mining*, pages 551–560, 2010.

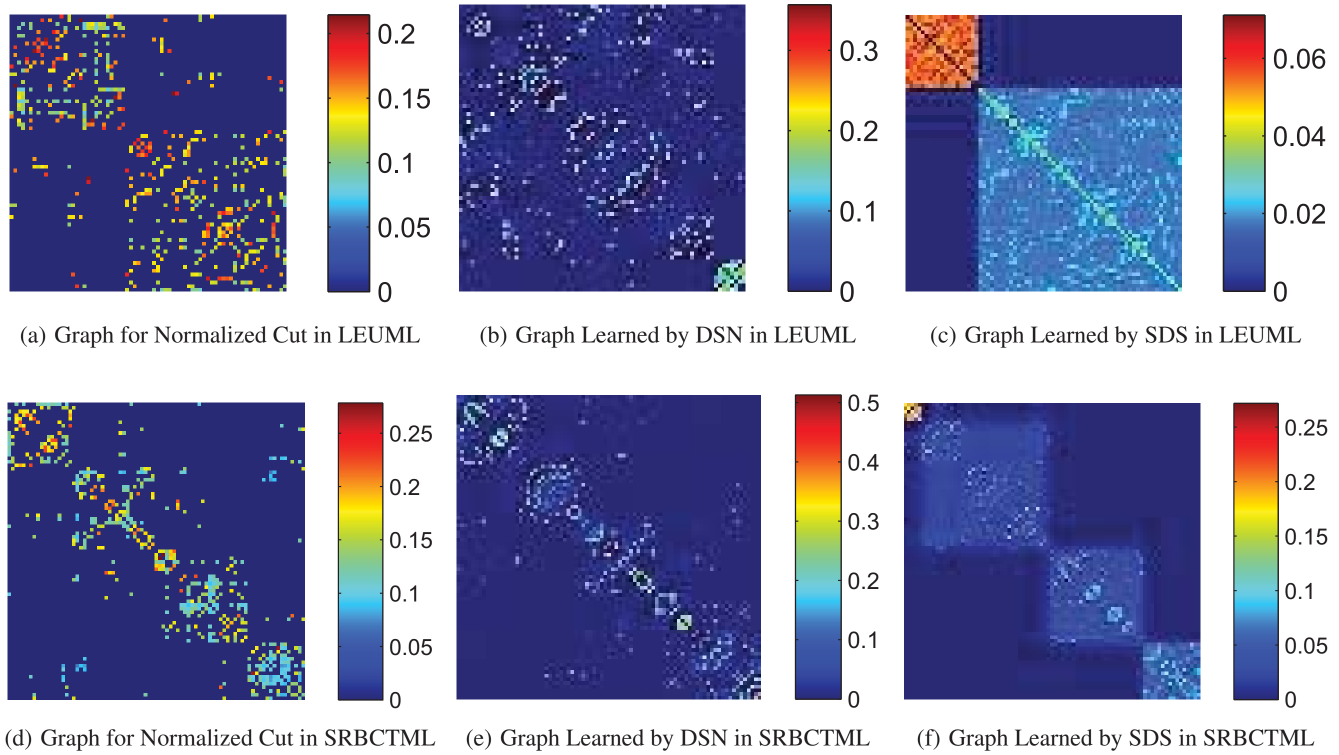
[27] M. A. Woodbury. *Inverting modified matrices*. 1950.

[28] K. Yang, Z. Cai, J. Li, and G. Lin. A stable gene selection in microarray data analysis. *BMC bioinformatics*, 7(1):228, 2006.

[29] R. Zass and A. Shashua. Doubly stochastic normalization for spectral clustering. In *NIPS*, pages 1569–1576, 2006.

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

[31] X. Zhu, Z. Ghahramani, and J. D. Lafferty. Semi-supervised learning using Gaussian fields and harmonic functions. In *ICML*, pages 912–919, 2003.



(a) Graph for Normalized Cut in LEUML

(b) Graph Learned by DSN in LEUML

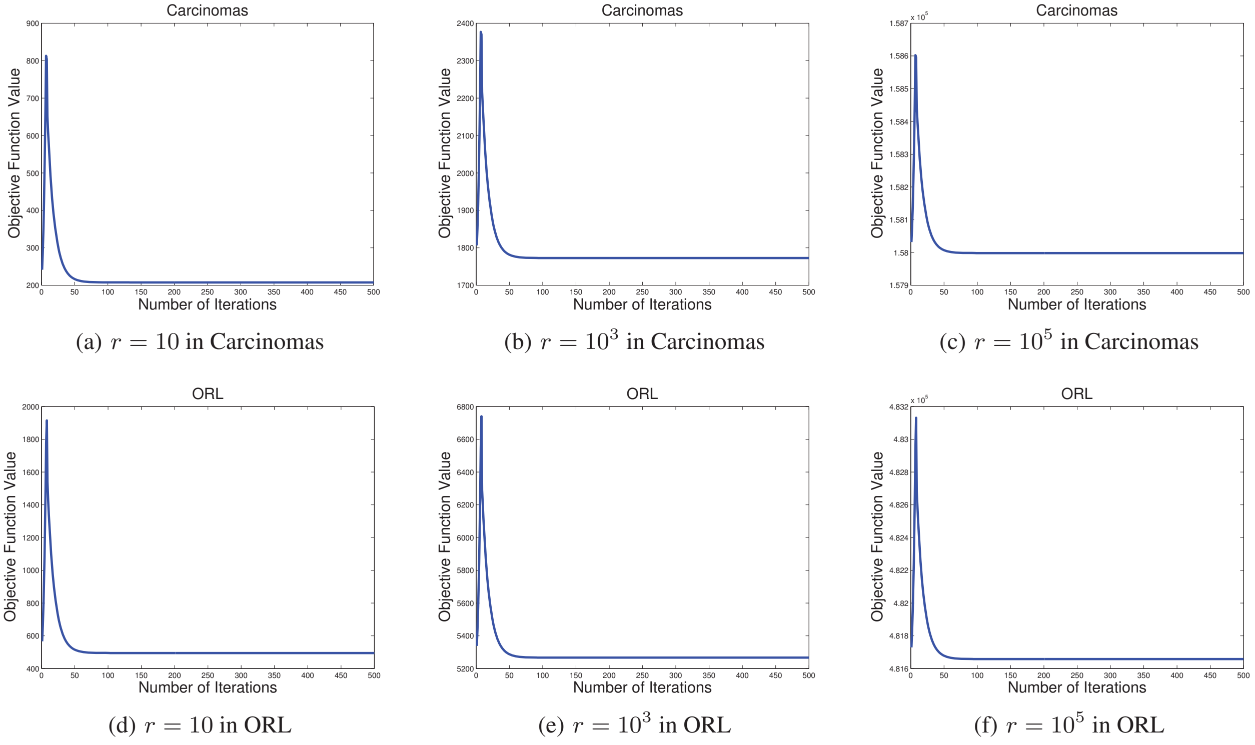
(c) Graph Learned by SDS in LEUML

(d) Graph for Normalized Cut in SRBCTML

(e) Graph Learned by DSN in SRBCTML

(f) Graph Learned by SDS in SRBCTML

Figure 2: Illustration of the graph learned from different methods on LEUML and SRBCTML datasets. Rows and columns of the graph are reshuffled respectively such that data points belonging to the same cluster are put together.



(a) $r = 10$ in Carcinomas

(b) $r = 10^3$ in Carcinomas

(c) $r = 10^5$ in Carcinomas

(d) $r = 10$ in ORL

(e) $r = 10^3$ in ORL

(f) $r = 10^5$ in ORL

Figure 3: Objective function value of Eq. (6) with different r parameters in each iteration on Carcinomas and ORL datasets.