The Constrained Laplacian Rank Algorithm for Graph-Based Clustering

Feiping Nie[1], **Xiaoqian Wang**[1], **Michael I. Jordan**[2], **Heng Huang**[1*]
[1]Department of Computer Science and Engineering, University of Texas, Arlington
[2]Departments of EECS and Statistics, University of California, Berkeley
feipingnie@gmail.com, xqwang1991@gmail.com, jordan@cs.berkeley.edu, heng@uta.edu

Abstract

Graph-based clustering methods perform clustering on a fixed input data graph. If this initial construction is of low quality then the resulting clustering may also be of low quality. Moreover, existing graph-based clustering methods require post-processing on the data graph to extract the clustering indicators. We address both of these drawbacks by allowing the data graph itself to be adjusted as part of the clustering procedure. In particular, our Constrained Laplacian Rank (CLR) method learns a graph with exactly k connected components (where k is the number of clusters). We develop two versions of this method, based upon the L1-norm and the L2-norm, which yield two new graph-based clustering objectives. We derive optimization algorithms to solve these objectives. Experimental results on synthetic datasets and real-world benchmark datasets exhibit the effectiveness of this new graph-based clustering method.

Introduction

State-of-the art clustering methods are often based on graphical representations of the relationships among data points. For example, spectral clustering (Ng, Jordan, and Weiss 2001), normalized cut (Shi and Malik 2000) and ratio cut (Hagen and Kahng 1992) all transform the data into a weighted, undirected graph based on pairwise similarities. Clustering is then accomplished by spectral or graph-theoretic optimization procedures. See (Ding and He 2005; Li and Ding 2006) for a discussion of the relations among these graph-based methods, and also the connections to non-negative matrix factorization. All of these methods involve a two-stage process in which an data graph is formed from the data, and then various optimization procedures are invoked on this fixed input data graph. A disadvantage of this two-stage process is that the final clustering structures are not represented explicitly in the data graph (e.g., graph-cut methods often use K-means algorithm to post-process the results to get the clustering indicators); also, the clustering results are dependent on the quality of the input data graph (*i.e.*, they are sensitive to the particular graph construction methods). It seems plausible that a strategy in which the optimization phase is allowed to change the data graph could have advantages relative to the two-phase strategy.

In this paper we propose a novel graph-based clustering model that learns a graph with exactly k connected components (where k is the number of clusters). In our new model, instead of fixing the input data graph associated to the affinity matrix, we learn a new data similarity matrix that is a block diagonal matrix and has exactly k connected components—the k clusters. Thus, our new data similarity matrix is directly useful for the clustering task; the clustering results can be immediately obtained without requiring any post-processing to extract the clustering indicators. To achieve such ideal clustering structures, we impose a rank constraint on the Laplacian graph of the new data similarity matrix, thereby guaranteeing the existence of exactly k connected components. Considering both L2-norm and L1-norm objectives, we propose two new clustering objectives and derive optimization algorithms to solve them. We also introduce a novel graph-construction method to initialize the graph associated with the affinity matrix.

We conduct empirical studies on simulated datasets and seven real-world benchmark datasets to validate our proposed methods. The experimental results are promising—we find that our new graph-based clustering method consistently outperforms other related methods in most cases.

Notation: Throughout the paper, all the matrices are written as uppercase. For a matrix M, the i-th row and the ij-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 and the L1 norm of matrix M are denoted by $\|M\|_F$ and $\|M\|_1$, respectively.

New Clustering Formulations

Graph-based clustering approaches typically optimize their objectives based on a given data graph associated with an affinity matrix $A \in \mathbb{R}^{n\times n}$ (which can be symmetric or non-symmetric), where n is the number of nodes (data points) in the graph. There are two drawbacks with these approaches: (1) the clustering performance is sensitive to the quality of the data graph construction; (2) the cluster structures are not

*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.
Copyright © 2016, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

explicit in the clustering results and a post-processing step is needed to uncover the clustering indicators.

To address these two challenges, we aim to learn a new data graph S based on the given data graph A such that the new data graph is more suitable for the clustering task. In our strategy, we propose to learn a new data graph S that has exactly k connected components, where k is the number of clusters.

In order to formulate a clustering objective based on this strategy, we start from the following theorem. If the affinity matrix A is nonnegative, then the Laplacian matrix $L_A = D_A - (A^T + A)/2$, where the degree matrix $D_A \in \mathbb{R}^{n\times n}$ is defined as a diagonal matrix whose i-th diagonal element is $\sum_j (a_{ij} + a_{ji})/2$, has the following important property (Mohar 1991; Chung 1997):

Theorem 1 *The multiplicity k of the eigenvalue zero of the Laplacian matrix L_A is equal to the number of connected components in the graph associated with A.*

Given a graph with affinity matrix A, Theorem 1 indicates that if $rank(L_A) = n - k$, then the graph is an ideal graph based on which we already partition the data points into k clusters, without the need of performing K-means or other discretization procedures as is necessary with traditional graph-based clustering methods such as spectral clustering.

Motivated by Theorem 1, given an initial affinity matrix $A \in \mathbb{R}^{n\times n}$, we learn a similarity matrix $S \in \mathbb{R}^{n\times n}$ such that the corresponding Laplacian matrix $L_S = D_S - (S^T + S)/2$ is constrained to be $rank(L_S) = n - k$. Under this constraint, the learned S is block diagonal with proper permutation, and thus we can directly partition the data points into k clusters based on S (Nie, Wang, and Huang 2014). To avoid the case that some rows of S are all zeros, we further constrain the S such that the sum of each row of S is one. Under these constraints, we learn that S that best approximates the initial affinity matrix A. Considering two different distances, the L2-norm and the L1-norm, between the given affinity matrix A and the learned similarity matrix S, we define the *Constrained Laplacian Rank* (CLR) for graph-based clustering as the solution to the following optimization problem:

$$J_{\text{CLR\_L2}} = \min_{\sum_j s_{ij}=1, s_{ij}\geq 0, rank(L_S)=n-k} \|S - A\|_F^2 \quad (1)$$

$$J_{\text{CLR\_L1}} = \min_{\sum_j s_{ij}=1, s_{ij}\geq 0, rank(L_S)=n-k} \|S - A\|_1. \quad (2)$$

These problems seem very difficult to solve since $L_S = D_S - (S^T + S)/2$, and D_S also depends on S, and the constraint $rank(L_S) = n-k$ is a complex nonlinear constraint. In the next section, we will propose novel and efficient algorithms to solve these problems.

Optimization Algorithms

Optimization Algorithm for Solving $J_{\text{CLR\_L2}}$ in Eq. (1)

Let $\sigma_i(L_S)$ denote the i-th smallest eigenvalue of L_S. Note that $\sigma_i(L_S) \geq 0$ because L_S is positive semidefinite. The problem (1) is equivalent to the following problem for a large enough value of λ:

$$\min_{\sum_j s_{ij}=1, s_{ij}\geq 0} \|S - A\|_F^2 + 2\lambda \sum_{i=1}^{k} \sigma_i(L_S). \quad (3)$$

When λ is large enough, note that $\sigma_i(L_S) \geq 0$ for every i, thus the optimal solution S to the problem (3) will make the second term $\sum_{i=1}^{k} \sigma_i(L_S)$ equal to zero and thus the constraint $rank(L_S) = n - k$ in the problem (1) will be satisfied.

According to Ky Fan's Theorem (Fan 1949), we have

$$\sum_{i=1}^{k} \sigma_i(L_S) = \min_{F\in\mathbb{R}^{n\times k}, F^TF=I} Tr(F^T L_S F). \quad (4)$$

Therefore, the problem (3) is further equivalent to the following problem:

$$\begin{aligned} &\min_{S,F} \|S - A\|_F^2 + 2\lambda Tr(F^T L_S F) \\ &s.t. \quad \textstyle\sum_j s_{ij} = 1, s_{ij} \geq 0, F \in \mathbb{R}^{n\times k}, F^TF = I. \end{aligned} \quad (5)$$

Compared with the original problem (1), the problem (5) is much easier to solve.

When S is fixed, the problem (5) becomes

$$\min_{F\in\mathbb{R}^{n\times k}, F^TF=I} Tr(F^T L_S F). \quad (6)$$

The optimal solution of F is formed by the k eigenvectors of L_S corresponding to the k smallest eigenvalues.

When F is fixed, the problem (5) becomes

$$\min_{\sum_j s_{ij}=1, s_{ij}\geq 0} \sum_{i,j} (s_{ij} - a_{ij})^2 + \lambda \sum_{i,j} \|f_i - f_j\|_2^2 s_{ij}. \quad (7)$$

Note that the problem (7) is independent for different i, so we can solve the following problem separately for each i:

$$\min_{\sum_j s_{ij}=1, s_{ij}\geq 0} \sum_{j} (s_{ij} - a_{ij})^2 + \lambda \sum_{j} \|f_i - f_j\|_2^2 s_{ij}. \quad (8)$$

Denoting $v_{ij} = \|f_i - f_j\|_2^2$, and denoting v_i as a vector with the j-th element equal to v_{ij} (and similarly for s_i and a_i), the problem (8) can be written in vector form as

$$\min_{s_i^T \mathbf{1}=1, s_i\geq \mathbf{0}} \left\| s_i - (a_i - \frac{\lambda}{2} v_i) \right\|_2^2. \quad (9)$$

This problem can be solved with a closed form solution as in Eq. (30), or solved by an efficient iterative algorithm (Huang, Nie, and Huang 2015).

In Algorithm 1 we provide a detailed algorithm for solving the problem (1). In this algorithm, we only update the m nearest similarities for each data point in S and thus the complexity of updating S and updating F (which only requires computing the top k eigenvectors of a very sparse matrix) is thereby reduced significantly. Further work, however, will be needed to makis this technique practicable on very large-scale data sets.

Algorithm 1 Algorithm to solve $J_{\text{CLR\_L2}}$ in Eq. (1).

input $A \in \mathbb{R}^{n\times n}$, cluster number k, a large enough λ.
output $S \in \mathbb{R}^{n\times n}$ with exactly k connected components.
Initialize $F \in \mathbb{R}^{n\times k}$, which is formed by the k eigenvectors of $L_A = D_A - \frac{A^T+A}{2}$ corresponding to the k smallest eigenvalues.
while not converge **do**
1. For each i, update the i-th row of S by solving the problem (9), where the j-th element of v_i is $v_{ij} = \|f_i - f_j\|_2^2$.
2. Update F, which is formed by the k eigenvectors of $L_S = D_S - \frac{S^T+S}{2}$ corresponding to the k smallest eigenvalues.
end while

Optimization Algorithm for Solving $J_{\text{CLR\_L1}}$ in Eq. (2)

Similarly, the problem (2) is equivalent to the following problem for a large enough value of λ:

$$\min_{\sum_j s_{ij}=1, s_{ij}\geq 0} \|S-A\|_1 + 2\lambda \sum_{i=1}^{k} \sigma_i(L_S), \quad (10)$$

and the problem (10) is further equivalent to the following problem:

$$\begin{aligned} &\min_{S,F} \|S-A\|_1 + 2\lambda Tr(F^T L_S F) \\ &s.t. \quad \textstyle\sum_j s_{ij} = 1, s_{ij} \geq 0, F \in \mathbb{R}^{n\times k}, F^T F = I. \end{aligned} \quad (11)$$

This problem can also be solved by the alternative optimization approach.

For fixed S, the matrix F is updated as in Eq. (6). For fixed F, the problem (11) becomes

$$\min_{\sum_j s_{ij}=1, s_{ij}\geq 0} \sum_{i,j} |s_{ij} - a_{ij}| + \lambda \sum_{i,j} \|f_i - f_j\|_2^2 s_{ij}.$$

Note that the above problem is independent between different i, so we can solve the following problem separately for each i:

$$\min_{\sum_j s_{ij}=1, s_{ij}\geq 0} \sum_{j} |s_{ij} - a_{ij}| + \lambda \sum_{j} \|f_i - f_j\|_2^2 s_{ij}. \quad (12)$$

Similarly to Eqs. (8) and (9), the problem (12) can be written in vector form as:

$$\min_{s_i^T \mathbf{1}=1, s_i \geq \mathbf{0}} \|s_i - a_i\|_1 + \lambda s_i^T v_i. \quad (13)$$

Using the iterative reweighted method, the problem (13) can be solved by iteratively solving the following problem:

$$\min_{s_i^T \mathbf{1}=1, s_i \geq \mathbf{0}} Tr(s_i - a_i)^T U (s_i - a_i) + \lambda s_i^T v_i, \quad (14)$$

where U is a diagonal matrix with the j-th diagonal element equal to $\frac{1}{2|\tilde{s}_{ij} - a_{ij}|}$, and $\tilde{s}_{ij}$ is the current solution. It has been proved that this iterative method decreases the objective of problem (13) in each iteration and will converge to the optimal solution to the problem (13) (Nie et al. 2010).

The problem (14) can be simplified to

$$\min_{s_i^T \mathbf{1}=1, s_i \geq \mathbf{0}} \frac{1}{2} s_i^T U s_i - s_i^T (U a_i - \frac{\lambda}{2} v_i). \quad (15)$$

Let $p_i = U a_i - \frac{\lambda}{2} v_i$, so for each i, we need to solve the following problem

$$\min_{s_i^T \mathbf{1}=1, s_i \geq \mathbf{0}} \frac{1}{2} s_i^T U s_i - s_i^T p_i. \quad (16)$$

This problem can be solved efficiently. The Lagrangian function of problem (16) is

$$\mathcal{L}(s_i, \eta, \alpha_i) = \frac{1}{2} s_i^T U s_i - s_i^T p_i - \eta(s_i^T \mathbf{1} - 1) - \alpha_i^T s_i, \quad (17)$$

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

Taking the derivative of Eq. (17) w.r.t. s_i and setting to zero, we have

$$U s_i - p_i - \eta \mathbf{1} - \alpha_i = \mathbf{0}. \quad (18)$$

Then for the j-th element of s_i, we have

$$u_{ii} s_{ij} - p_{ij} - \eta - \alpha_{ij} = 0. \quad (19)$$

Note that $s_{ij}\alpha_{ij} = 0$ according to the KKT condition, then from Eq. (19) we have:

$$s_{ij} = (\frac{1}{u_{ii}}\eta + \frac{1}{u_{ii}} p_{ij})_+, \quad (20)$$

where $(v)_+ = max(0, v)$. We define the following function w.r.t. η

$$g_i(\eta) = \sum_i (\frac{\eta}{u_{ii}} + \frac{p_{ij}}{u_{ii}})_+ - 1. \quad (21)$$

Then according to Eqs. (20)-(21), and the constraint $s_i^T \mathbf{1} = 1$, we have the following equation:

$$g_i(\eta) = 0. \quad (22)$$

Therefore, the value of η is the root of function $g_i(x)$. Note that $g_i(x)$ is a piecewise linear and monotonically increasing function, thus the root can be easily obtained by Newton's method. After computing η, the optimal solution to the problem (16) can be obtained by Eq. (20).

Based on the above analysis, the detailed procedure for solving Eq. (2) is summarized in Algorithm 2.

Algorithm 2 Algorithm to solve $J_{\text{CLR\_L1}}$ in Eq. (2).

input $A \in \mathbb{R}^{n\times n}$, cluster number k, a large enough λ.
output $S \in \mathbb{R}^{n\times n}$ with exactly k connected components.
Initialize $F \in \mathbb{R}^{n\times k}$, which is formed by the k eigenvectors of $L_A = D_A - \frac{A^T+A}{2}$ corresponding to the k smallest eigenvalues.
while not converge **do**
1. For each i, update the i-th row of S by solving the problem (16), where U is a diagonal matrix with the j-th diagonal element as $\frac{1}{2|s_{ij} - a_{ij}|}$ and $p_i = U a_i - \frac{\lambda}{2} v_i$, the j-th element of v_i is $v_{ij} = \|f_i - f_j\|_2^2$.
2. Update F, which is formed by the k eigenvectors of $L_S = D_S - \frac{S^T+S}{2}$ corresponding to the k smallest eigenvalues.
end while

Learning An Initial Graph

In the proposed algorithms, an initial graph affinity matrix $A \in \mathbb{R}^{n\times n}$ is required to be given before learning the normalized and block-diagonal similarity matrix $S \in \mathbb{R}^{n\times n}$. We propose an approach to initialize graph A. Since we are to learn a nonnegative and normalized similarity matrix S such that the sum of each row of S is equal to one, it is desirable for the initial graph A to have the same constraint. If we do not have any information about the data, we can set all the affinities of A to the same value, which could be seen as a prior. Under these nonnegativity and normalization constraints, minimizing the L2-norm of each row of A will result in the affinities with the same value. Therefore, we can use the L2-norm of each row of A as the regularization to learn the affinity values of A.

Given the data points $\{x_1, ..., x_n\}$, it is desirable to learn the affinity values of A such that smaller distance $\|x_i - x_j\|_2^2$ between data points x_i and x_j corresponds to a larger affinity value a_{ij}. In addition, we simply set $a_{ii} = 0$. We propose to solve the following problem:

$$\min_{a_i^T\mathbf{1}=1, a_i\geq\mathbf{0}, a_{ii}=0} \sum_{j=1}^{n} \|x_i - x_j\|_2^2 a_{ij} + \gamma \sum_{j=1}^{n} a_{ij}^2. \quad (23)$$

In many cases, we prefer a sparse affinity matrix A for efficiency and higher performance. Therefore, we learn the affinities with the maximal γ such that the optimal solution a_i to the problem (23) has exactly m nonzero values; i.e., the L0-norm of a_i is constrained to be m. To this end, we solve the following problem:

$$\max_{\gamma, \|\hat{a}_i\|_0=m} \gamma, \quad (24)$$

where $\hat{a}$ is the optimal solution to the problem (23).

Let us define

$$e_{ij} = \|x_i - x_j\|_2^2 \quad (25)$$

and denote e_i as a vector with the j-th element as e_{ij}, then the problem (23) can be simplified as

$$\min_{a_i^T\mathbf{1}=1, a_i\geq\mathbf{0}, a_{ii}=0} \frac{1}{2}\left\|a_i + \frac{e_i}{2\gamma}\right\|_2^2. \quad (26)$$

The Lagrangian function of problem (26) is

$$\mathcal{L}(a_i, \eta, \beta_i) = \frac{1}{2}\left\|a_i + \frac{e_i}{2\gamma}\right\|_2^2 - \eta(a_i^T\mathbf{1} - 1) - \beta_i^T a_i, \quad (27)$$

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

The optimal solution $\hat{a}$ should satisfy that the derivative of Eq. (27) w.r.t. a_i is equal to zero, so we have

$$\hat{a}_i + \frac{e_i}{2\gamma} - \eta\mathbf{1} - \beta_i = \mathbf{0}. \quad (28)$$

Then for the j-th element of $\hat{a}_i$, we have

$$\hat{a}_{ij} + \frac{e_{ij}}{2\gamma} - \eta - \beta_{ij} = 0. \quad (29)$$

Noting that $a_{ij}\beta_{ij} = 0$ according to the KKT condition, from Eq. (29) we have

$$\hat{a}_{ij} = (-\frac{e_{ij}}{2\gamma} + \eta)_+. \quad (30)$$

Without loss of generality, suppose $e_{i1}, e_{i2}, ..., e_{in}$ are ordered from small to large. In order to impose $\hat{a}_{ii} = 0$, we always let e_{ii} place this value last despite having $e_{ii} = 0$. According to the constraint $\|\hat{a}_i\|_0 = m$ in problem (24), we know $\hat{a}_{im} > 0$ and $\hat{a}_{i,m+1} = 0$. Therefore, we have

$$-\frac{e_{im}}{2\gamma} + \eta > 0, \quad \text{and} \quad -\frac{e_{i,m+1}}{2\gamma} + \eta \leq 0\,. \quad (31)$$

According to Eq. (30) and the constraint $a_i^T\mathbf{1} = 1$ in problem (23), we have

$$\sum_{j=1}^{m}(-\frac{e_{ij}}{2\gamma} + \eta) = 1 \Rightarrow \eta = \frac{1}{m} + \frac{1}{2m\gamma}\sum_{j=1}^{m} e_{ij}\,. \quad (32)$$

So we have the following inequality for γ according to Eq. (31) and Eq. (32):

$$\frac{m}{2}e_{im} - \frac{1}{2}\sum_{j=1}^{m} e_{ij} < \gamma \leq \frac{m}{2}e_{i,m+1} - \frac{1}{2}\sum_{j=1}^{m} e_{ij}. \quad (33)$$

Therefore, to obtain the optimal solution $\hat{a}_i$ to the problem (23) that has exactly m nonzero values, the maximal γ is

$$\gamma = \frac{m}{2}e_{i,m+1} - \frac{1}{2}\sum_{j=1}^{m} e_{ij}. \quad (34)$$

According to Eqs. (30), (32) and (34), we get the optimal affinities $\hat{a}_{ij}$ as follows:

$$\hat{a}_{ij} = \begin{cases} \frac{e_{i,m+1} - e_{ij}}{m e_{i,m+1} - \sum_{h=1}^{m} e_{ih}} & j \leq m \\ 0 & j > m \end{cases}. \quad (35)$$

The affinities $\hat{a}_{ij}$ computed by Eq. (35) have the following advantages:

(1). Eq. (35) only involves the basic operations of addition, subtraction, multiplication and division. Methods such as LLE (Roweis and Saul 2000) and sparse coding which are often can be used to compute the affinities require computations of Gaussian functions and other more operations that make them less efficient than the current method.

(2). The learned matrix $\hat{A}$ is naturally sparse. A sparse graph is computationally efficient for graph-based learning tasks such as clustering and semi-supervised classification.

(3). The affinities are distance consistent. This property is guaranteed from the motivation of this method. If the distance between x_i and x_j is smaller than the distance between x_i and x_k, then the affinity $\hat{a}_{ij}$ computed by Eq. (35) is larger than the affinity $\hat{a}_{ik}$. In LLE and sparse coding this property is not guaranteed.

(4). The affinities are scale invariant. If the data points $\{x_1, ..., x_n\}$ are scaled by an arbitrary scalar t, i.e., let x_i be $t \cdot x_i$ for each i, then e_{ij} is changed to be $t \cdot e_{ij}$ for each i, j, but the affinities $\hat{a}_{ij}$ computed by Eq. (35) will not be changed. While in Gaussian function, the affinities will be changed in this case, which makes the parameter difficult to tune.

(5). Computing the affinities by Eq. (35) only involves one parameter: the number of neighbors m. This parameter is an integer, which is easy to tune. In most cases, $m < 10$ is

likely to yield reasonable results. This property is important since the tuning of hyperparameters remains a difficult and open problem in clustering. In graph-based clustering (and more generally in semi-supervised learning), there are few labeled data points and thus traditional supervised hyperparameter tuning techniques such as cross validation can not be used.

Connection to Normalized Cut

In this section, we show that the proposed problem (1) is closely connected to the Normalized Cut problem in (Shi and Malik 2000).

We add a regularization term to S in Eq. (1), and solve the following problem for graph-based clustering:

$$\min_{\sum_j s_{ij}=1, s_{ij}\geq 0, rank(L_S)=n-k} \|S-A\|_F^2 + \gamma \|S\|_F^2, \quad (36)$$

where A is a doubly stochastic matrix.

Theorem 2 *When $\gamma \to \infty$, the problem (36) is equivalent to the Normalized Cut problem.*

PROOF. The problem (36) can be written as

$$\min_{S\mathbf{1}=\mathbf{1}, S\geq 0, rank(L_S)=n-k} -2Tr(S^T A) + (1+\gamma)\|S\|_F^2 \ . \quad (37)$$

Due to the constraint $rank(L_S) = n-k$, the solution S has exactly k 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 (37) is to solve the following problem for each i:

$$\min_{S_i\mathbf{1}=\mathbf{1}, S_i\geq 0} -2Tr(S_i^T A_i) + (1+\gamma)\|S_i\|_F^2 . \quad (38)$$

When $\gamma \to \infty$, then the problem (38) becomes

$$\min_{S_i\mathbf{1}=\mathbf{1}, S_i\geq 0} \|S_i\|_F^2 . \quad (39)$$

The optimal solution to the problem (39) is that all the elements of S_i are equal to $\frac{1}{n_i}$.

Therefore, the optimal solution S to the problem (37) should be the following form when $\gamma \to \infty$:

$$s_{ij} = \begin{cases} \frac{1}{n_p} & x_i, x_j \text{ are in the same component } p \\ 0 & \text{otherwise.} \end{cases} \quad (40)$$

We denote the solution set that satisfies the form in Eq. (40) by $\mathcal{V}$. Note that for any possible partition of the k components such that S has the form in Eq. (40), $\|S\|_F^2$ has the same value, *i.e.*, $\|S\|_F^2 = k$. Therefore, the problem (37) or (36) becomes

$$\min_{S\in\mathcal{V}} \|S-A\|_F^2 . \quad (41)$$

It can be easily verified that when A is a doubly stochastic matrix, $\|S-A\|_F^2$ is exactly the Ratio Cut under the partition with S. Note that Normalized Cut is equal to Ratio Cut if A is doubly stochastic, thus the problem (36) is equivalent to the Normalized Cut problem when $\gamma \to \infty$.

□

Data sets	Num of Instances	Dimensions	Classes
Yeast	1484	8	10
Abalone	4177	8	29
COIL20	1440	1024	20
COIL100	7200	1024	100
AR	840	768	120
XM2VTS	1180	1024	295
Umist	165	3456	15

Table 1: Descriptions of seven benchmark datasets.

Experiments

In this section, we explore the performance of our clustering methods on both synthetic and real benchmark data sets. For simplicity, we denote our Constrained Laplacian Rank L1-norm clustering method as CLR_L1, and our Frobenius norm clustering method as CLR_L2.

Block Diagonal Synthetic Data

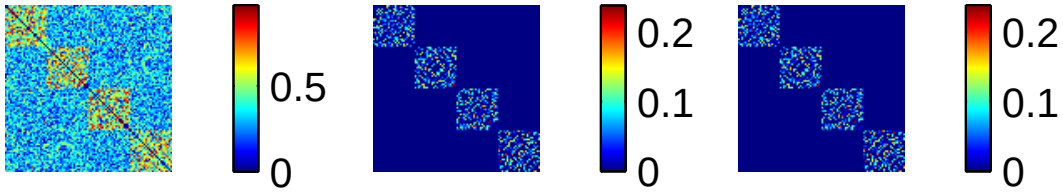
The first synthetic dataset we used is a 100×100 matrix with four 25×25 block matrices diagonally arranged. The data within each block denotes the affinity of two corresponding points in one cluster, while the data outside all blocks denotes noise. The affinity data within each block is randomly generated in the range of 0 and 1, while the noise data is randomly generated in the range of 0 and c, where c is set as 0.6, 0.7 and 0.8 respectively. Moreover, to make this clustering task more challenging, we randomly pick out 25 noise data points and set their values to be 1.

Fig. 1 shows the original random matrix and the clustering results under different settings. We can notice that both CLR_L1 and CLR_L2 exhibit good performance in this clustering task.

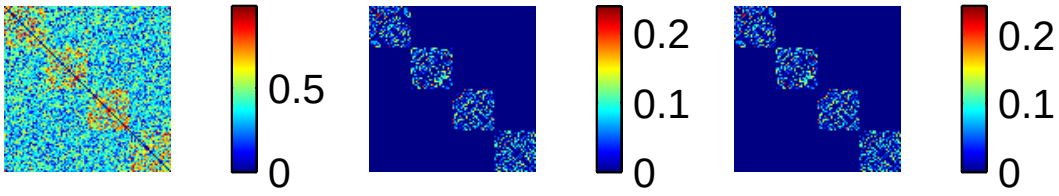
We also compared the clustering accuracy with other graph-based clustering methods, among which the Normalized Cut (NCut) performed best. When noise = 0.6, the clustering accuracy of NCut, CLR_L1, CLR_L2 are all 100%. When noise = 0.7, the clustering accuracy of NCut, CLR_L1, CLR_L2 are 99%, 100%, 100%, respectively. When noise = 0.8, the clustering accuracy of NCut, CLR_L1, CLR_L2 are 85%, 99%, 99%, respectively.

Two-Moon Synthetic Data

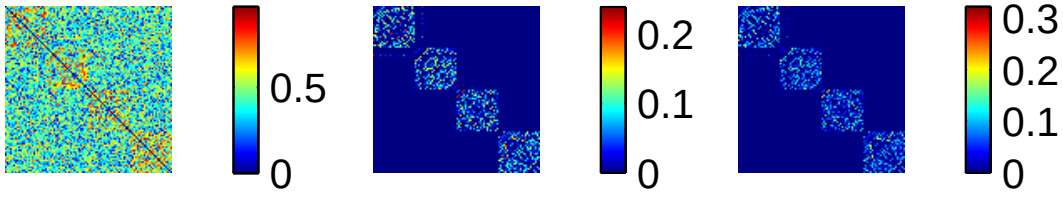
The second toy data set we used is a randomly generated two-moon matrix. In this test, there are two clusters of data distributed in the moon shape. Each cluster has a volume of 100 samples and the noise percentage is set to be 0.13. Our goal is to recompute the similarity matrix such that the number of connected components in the learned similarity matrix is exactly two. We tested with CLR_L1 and CLR_L2 methods and obtain good results on both of them. From Fig. 2 we can easily observe the effectiveness of our proposed methods. In this figure, we set the color of the two clusters to be red and blue respectively and let the width of connecting lines denote the affinity of two corresponding points. In the original matrix, there are several pairs of connected points from different clusters. However, after the computation, there is not even a single line between the two clusters,



(a) Original Graph, noise = 0.6 (b) CLR_L1 Result, noise = 0.6 (c) CLR_L2 Result, noise = 0.6



(d) Original Graph, noise = 0.7 (e) CLR_L1 Result, noise = 0.7 (f) CLR_L2 Result, noise = 0.7

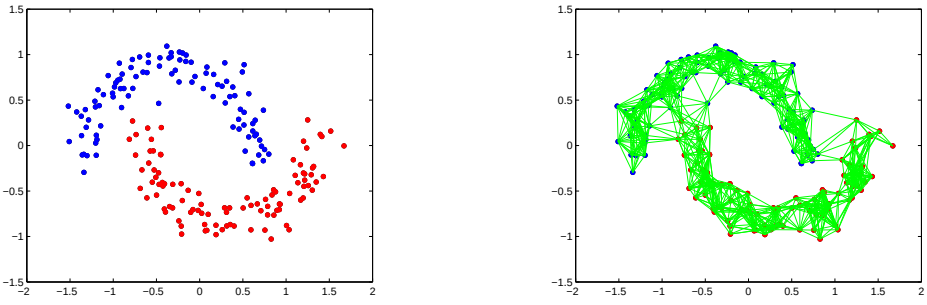


(g) Original Graph, noise = 0.8 (h) CLR_L1 Result, noise = 0.8 (i) CLR_L2 Result, noise = 0.8

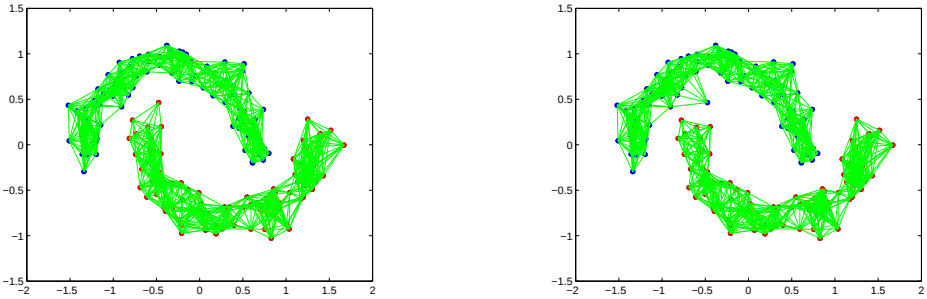
Figure 1: Clustering results on the block diagonal synthetic data by CLR_L1 and CLR_L2 methods.

		Yeast	Abalone	COIL20	COIL100	AR	XM2VTS	Umist
ACC	K-means	0.4063	0.148	0.6382	0.5153	0.2798	0.4814	0.4052
	RCut	0.4144	0.1427	0.7958	0.6304	0.3619	0.5873	0.6557
	NCut	0.4003	0.1547	0.7944	0.6535	0.3643	0.661	0.6365
	NMF	0.4009	0.1568	0.7104	0.5332	0.3833	0.6873	0.5757
	CLR_L1	0.4158	**0.2025**	0.8535	**0.8122**	**0.4202**	**0.722**	**0.7287**
	CLR_L2	**0.4872**	0.1968	**0.8736**	0.8035	**0.4202**	**0.722**	**0.7287**
		Yeast	Abalone	COIL20	COIL100	AR	XM2VTS	Umist
NMI	K-means	0.2619	0.1504	0.7794	0.753	0.6195	0.8065	0.6367
	RCut	**0.2795**	0.1532	0.8894	0.8435	0.6841	0.8078	0.8148
	NCut	0.2536	0.1568	0.8877	0.8473	0.6986	0.8883	0.8009
	NMF	0.2521	0.1482	0.8404	0.7581	**0.7026**	0.8929	0.7595
	CLR_L1	0.1946	0.1619	**0.945**	0.9401	0.6242	0.8942	**0.8634**
	CLR_L2	0.2622	**0.1715**	**0.945**	**0.9407**	0.5909	**0.8951**	**0.8634**

Table 2: Experimental results on real benchmark datasets.



(a) Original Points (b) Original Connected Graph



(c) CLR_L1 Result (d) CLR_L2 Result

Figure 2: Clustering results on the two-moon synthetic data by CLR_L1 and CLR_L2 methods.

which indicates our proposed clustering methods have successfully partitioned the original data into two classes.

Experimental Results on Real Benchmark Datasets

We also evaluated the proposed clustering methods on 7 benchmark datasets: Yeast (Asuncion and Newman 2007), Abalone (Asuncion and Newman 2007), COIL20 (Nene, Nayar, and Murase 1996b), COIL100 (Nene, Nayar, and Murase 1996a), AR (Martinez 1998), XM2VTS (XM2VTS) and UMIST (Graham and Allinson 1998), the first two of which are bioinformatics datasets from UCI Machine Learning Repository while the latter five are image datasets. The descriptions of these 7 datasets are summarized in Table 1.

We compared our clustering methods with K-means, Ratio Cut (RCut), Normalized Cut (NCut) and NMF methods. For RCut and NCut methods, we utilized the widely used self-tune Gaussian method (Zelnik-Manor and Perona 2004) to construct the affinity matrix (the value of σ is self-tuned). For both self-tune Gaussian and our method, we set the number of neighbors, m, to be five for the affinity matrix construction. As for our clustering method, we determined the value of λ in a heuristic way to accelerate the procedure: first set λ with a small value, then in each iteration, we computed the number of zero eigenvalues in L_S, if it was larger than k, we divided λ by two; if smaller we multiplied λ by two; otherwise we stopped the iteration. Moreover, we set the number of clusters to be the ground truth in each dataset for all methods. The standard clustering accuracy (ACC) and normalized mutual information (NMI) metrics were used to evaluate all clustering methods.

For all the methods involving K-means, including K-means, RCut and NCut methods, we used the same initialization and repeated 50 times to compute their respective best initialization vector in terms of objective value of K-means. For the four compared methods, since their performance is unstable with different initializations, we only report their respective best results (in terms of objective value of K-means) over the 50 repetitions. As for our methods, we ran only once with the initialization described in Eq. (35). Table 2 shows the clustering results of each method.

From Table 2, we conclude that our proposed methods outperform the competing methods on most of the benchmark datasets. Our proposed clustering methods CLR_L1 and CLR_L2 learn the data similarity matrix as part of the clustering task, and we believe that this confers robustness on the procedure in addition to accuracy improvements.

Conclusions

In this paper, we proposed a novel graph-based clustering model to learn a new data graph with exactly k connected components, which is an ideal structure for clustering. This differs from existing graph-based approaches which fixed the input data graph (associated with an affinity matrix). We instead learned a new block diagonal data similarity matrix such that the clustering results can be immediately obtained without requiring any post-processing to extract the clustering indicators. Considering both L2 norm and L1 norm distances, we proposed two new clustering objectives and derived optimization algorithms to solve them. Empirical results on two synthetic datasets and seven real benchmark datasets showed our methods outperform competing clustering approaches.

References

Asuncion, A., and Newman, D. 2007. *UCI Machine Learning Repository*.

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

Ding, C. H. Q., and He, X. 2005. On the equivalence of nonnegative matrix factorization and spectral clustering. In *SDM*.

Fan, K. 1949. On a theorem of Weyl concerning eigenvalues of linear transformations. 35(11):652–655.

Graham, D. B., and Allinson, N. M. 1998. Characterizing virtual eigensignatures for general-purpose face recognition. *NATO ASI Series F, Computer and Systems Sciences* 163:446–456.

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

Huang, J.; Nie, F.; and Huang, H. 2015. A new simplex sparse learning model to measure data similarity for clustering. In *Proceedings of the 24th International Conference on Artificial Intelligence*, 3569–3575.

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

Martinez, A. 1998. The AR face database. *CVC Technical Report* 24.

Mohar, B. 1991. The Laplacian spectrum of graphs. In *Graph Theory, Combinatorics, and Applications*, 871–898. Wiley.

Nene, S. A.; Nayar, S. K.; and Murase, H. 1996a. *Columbia object image library (COIL-100), Technical Report CUCS-006-96*.

Nene, S. A.; Nayar, S. K.; and Murase, H. 1996b. *Columbia object image library (COIL-20), Technical Report CUCS-005-96*.

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

Nie, F.; Huang, H.; Cai, X.; and Ding, C. 2010. Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization. In *NIPS*.

Nie, F.; Wang, X.; and Huang, H. 2014. Clustering and projected clustering with adaptive neighbors. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 977–986.

Roweis, S. T., and Saul, L. 2000. Nonlinear dimensionality reduction by locally linear embedding. *Science* 290:2323–2326.

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

XM2VTS. http://www.ee.surrey.ac.uk/cvssp/xm2vtsdb/.

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