Community Detection and Link Prediction via Cluster-driven Low-rank Matrix Completion

Junming Shao, **Zhong Zhang**, **Zhongjing Yu**, **Jun Wang**, **Yi Zhao** and **Qinli Yang**
Data Mining Lab, University of Electronic Science and Technology of China
{junmshao, qinli.yang}@uestc.edu.cn

Abstract

Community detection and link prediction are highly dependent since knowing cluster structure as a priori will help identify missing links, and in return, clustering on networks with supplemented missing links will improve community detection performance. In this paper, we propose a Cluster-driven Low-rank Matrix Completion (CLMC), for performing community detection and link prediction simultaneously in a unified framework. To this end, CLMC decomposes the adjacent matrix of a target network as three additive matrices: clustering matrix, noise matrix and supplement matrix. The community-structure and low-rank constraints are imposed on the clustering matrix, such that the noisy edges between communities are removed and the resulting matrix is an ideal block-diagonal matrix. Missing edges are further learned via low-rank matrix completion. Extensive experiments show that CLMC achieves state-of-the-art performance.

1 Introduction

Community structure is ubiquitous in real-world networks, and many community detection algorithms have been proposed, e.g., NCut [Shi and Malik, 2000], MCL [Van Dongen, 2000], Modularity [Newman, 2006], Attractor [Shao *et al.*, 2015], FUSE [Ye *et al.*, 2016], ORSC[Shao *et al.*, 2017], MAPPR [Yin *et al.*, 2017], Dcut [Shao *et al.*, 2018], to mention a few. Although community detection has achieved great success during the past decade, finding intrinsic community structure in networks is still challenging. For example, most existing community detection approaches usually work well if (but only if) the communities are well separated. With the increase of inter-cluster edges, the performance of established approaches tends to decrease since they significantly hamper community separation. In addition, many connections in real-world networks are missing. Such problem, further aggravates the difficulty to find high-quality communities. Therefore, identifying inter-cluster edges and supplementing missing edges offer a promising way to enhance the performance of community detection and link prediction simultaneously.

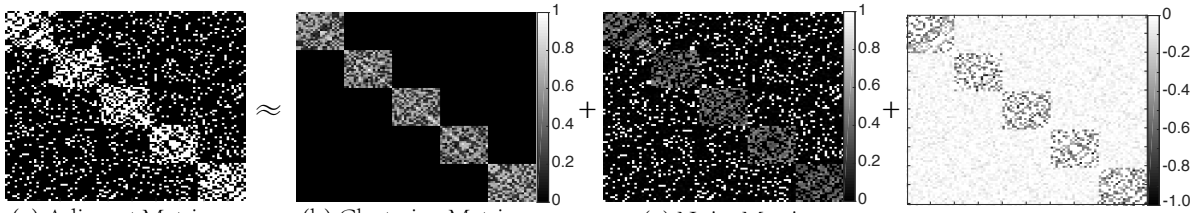
In recent years, matrix completion becomes an important tool in machine learning, and has been extensively studied in recommender systems and computer vision [Liu *et al.*, 2013; Natarajan and Dhillon, 2014; Kang *et al.*, 2016]. Given an observed matrix A, matrix completion constructs a new matrix $\hat{A}$ that approximates A at its unobserved entries. Since there are infinite number of matrices that perfectly agree with the observed entries of A, some additional assumptions are usually imposed such as low-rank [Candès and Recht, 2009; Liu *et al.*, 2013; Hastie *et al.*, 2015; Yang *et al.*, 2018] and sparsity [Lu *et al.*, 2016; Fan and Chow, 2017].

Motivated by aforementioned problems and existing matrix completion techniques, in this paper, we propose a Cluster-driven Low-rank Matrix Completion, called CLMC, which aims to learn an ideal similarity matrix to perform graph clustering and a supplement matrix to conduct link prediction simultaneously in a unified framework. The basic idea of CLMC is to decompose the original data matrix into three additive matrices: the first corresponds to an ideal clustering matrix (Figure 1(b)), which is expected to be a block-diagonal matrix where the inter-cluster edges are removed and meanwhile, the missing or trend connections are supplemented in each community. The second corresponds to the inter-cluster edges, which is viewed as a noisy matrix (Figure 1(c)). The third one, called supplement matrix, is used to characterize the missing or trend connections (Figure 1(d)). With the resulting matrices, we can perform community detection on the clustering matrix, and, conduct link prediction with the supplement matrix.

CLMC has several attractive benefits, most importantly:

- **Reciprocal Learning.** Instead of treat community detection and link prediction as two separated problems, CLMC learns an ideal similarity matrix to perform graph clustering and a supplement matrix to conduct link prediction simultaneously. As a result, CLMC allows finding high-quality communities since the noisy connections are well filtered and the missing connections are supplemented. In return, the resulting good graph clustering further boosts the link prediction performance.

- **Robustness**: Since CLMC examines clustering part, noise part and missing part simultaneously, it is thus quite insensitive to noisy or missing edges.



(a) Adjacent Matrix (b) Clustering Matrix (c) Noise Matrix (d) Supplement Matrix

Figure 1: The illustration of cluster-driven low-rank matrix completion. Here the original adjacent matrix is decomposed to three additive matrices, where clustering matrix is expected to be a block-diagonal matrix, noise matrix contains the inter-cluster connectivity (i.e., noisy edges) and supplement matrix characterizes the missing or trend edges. Note that the white dot indicates connection in (a)(b)(c) and black dot means no connection, while in (d), white dots indicate zeroes and black dots indicate **negatives values** of entries in supplement matrix (i.e., missing connections).

2 Proposed Approach

2.1 Low-rank Matrix Completion

Given an observed matrix A, if the number of known entries in A is sufficiently large (i.e., $p \geq Crn^{6/5}\log n$) and the entries are uniformly distributed, we can complete the data matrix by solving the following optimization problem [Candès and Recht, 2009].

$$\begin{aligned} \min \quad & \text{rank}(\hat{A}) \\ \text{s.t.} \quad & \hat{A}_{ij} = A_{ij} \ \ \forall (i,j) \in \Omega, \end{aligned} \tag{1}$$

where $\hat{A}$ is the completed matrix, $\text{rank}(\hat{A})$ is the rank of matrix $\hat{A}$, and Ω is the set of indices of observed entries. Although the optimization problem is NP-hard, it can be approximated to optimize the *nuclear norm* [Candès and Recht, 2009; Keshavan *et al.*, 2010] as follows.

$$\begin{aligned} \min \quad & ||\hat{A}||_* \\ \text{s.t.} \quad & \hat{A}_{ij} = A_{ij} \ \ \forall (i,j) \in \Omega, \end{aligned} \tag{2}$$

where $||\cdot||_*$ denotes the nuclear norm of a matrix.

2.2 Cluster-driven Low-rank Matrix Completion

As aforementioned, the key idea of our algorithm is to learn an ideal clustering matrix by removing noisy edges and completing missing edges in each community. To this end, the observed data matrix A can be divided into three parts: clustering matrix, noise matrix and supplement matrix. Here, we borrow the idea from low-rank matrix completion to handle missing edges. However, since the low-rank assumption of the recovered matrix $\hat{A}$ may not hold in real-world applications, unlike existing approaches, we impose the low-rank constraint on the clustering matrix instead. As a result, we first formulate our objective function as follows.

$$\begin{aligned} \min \quad & ||\hat{C}||_* \\ \text{s.t.} \quad & A = \hat{C} + \hat{R} + E, \ \pi_\Omega(E) = 0, \hat{C} \geq 0, \hat{R} \geq 0 \end{aligned} \tag{3}$$

Where $\hat{C}$, $\hat{R}$ and E are the clustering matrix, noise matrix and supplement matrix, respectively. $\pi_\Omega : \mathcal{R}^{m\times n} \rightarrow \mathcal{R}^{m\times n}$ is a linear operator that keeps the entries in Ω unchanged and sets those outside Ω zeroes.

Since we expect $\hat{C}$ to be an ideal block-diagonal matrix where the inter-cluster edges are removed, and meanwhile, the missing or trend connections are supplemented in each community, a cluster structure constraint needs to be added to make $\hat{C}$ have exactly k connected components, where k is the number of clusters. Let $L_{\hat{C}} = D - \hat{C}$ be the Laplacian matrix associated with $\hat{C}$, and D is the diagonal matrix where the ii-th diagonal element is $\sum_j \hat{C}_{ij}$. If $\hat{C}$ is non-negative, according to the property of Laplacian matrix, we have the following theorem [Mohar *et al.*, 1991; Chung, 1997].

THEOREM 1 *The number k of the eigenvalue 0 of the Laplacian matrix $L_{\hat{C}}$ is equal to the number of connected components in the graph associated with $\hat{C}$.*

According to Theorem 1, the ideal clustering matrix can be obtained by imposing $rank(L_{\hat{C}}) = n - k$ constraint. Suppose that λ_i is the i-th smallest eigenvalue of $L_{\hat{C}}$, the constraint can be approximated to the following problem for a large positive value of α_1.

$$\min \quad \alpha_1 \sum_{i=1}^{k} \lambda_i \tag{4}$$

When α_1 is large enough, $\sum_{i=1}^{k} \lambda_i$ will be driven to be close to zero, which results in $rank(L_{\hat{C}}) = n - k$. In addition, according to the Ky Fan's Theorem [Fan, 1949], we have

$$\sum_{i=1}^{k} \lambda_i = \min_{F \in \mathbb{R}^{n\times k}, F^T F = I} tr(F^T L_{\hat{C}} F) \tag{5}$$

By adding the clustering constraint, our problem is further written as follows.

$$\begin{aligned} \min \quad & ||\hat{C}||_* + \alpha_1 tr(F^T L_{\hat{C}} F) \\ \text{s.t.} \quad & A = \hat{C} + \hat{R} + E, \ \pi_\Omega(E) = 0 \\ & \hat{C} \geq 0, \hat{R} \geq 0, F^T F = I \end{aligned} \tag{6}$$

As for noise matrix, we assume that it is sparse and the ℓ_1-norm constraint is imposed. As a result, we have our final objective function as follows.

$$\begin{aligned} \min \quad & ||\hat{C}||_* + \alpha_1 tr(F^T L_{\hat{C}} F) + \alpha_2 ||\hat{R}||_1 \\ \text{s.t.} \quad & A = \hat{C} + \hat{R} + E, \ \pi_\Omega(E) = 0 \\ & \hat{C} \geq 0, \hat{R} \geq 0, F^T F = I \end{aligned} \tag{7}$$

2.3 Optimization

In the following, we will introduce an iterative strategy to solve our optimization problem.

Update F By fixing $\hat{C}$, $\hat{R}$ and E, the problem (7) is equivalent to optimizing the following objective function:

$$\min_{F} tr(F^T L_{\hat{C}} F) \qquad s.t.\ F \in R^{n\times k}, F^T F = I \quad (8)$$

The optimal solution F to the problem is formed by the k eigenvectors of $L_{\hat{C}}$ corresponding to the k smallest eigenvalues.

Update $\hat{C}$, $\hat{R}$ and E By fixing F, we can obtain $\hat{C}$, $\hat{R}$ and E by optimizing the following objective function.

$$\min_{\hat{C},\hat{R},E} ||\hat{C}||_* + \alpha_1 tr(F^T L_{\hat{C}} F) + \alpha_2 ||\hat{R}||_1 \quad (9)$$

$$\text{s.t. } A = \hat{C} + \hat{R} + E, \pi_\Omega(E) = 0, \hat{C} \geq 0, \hat{R} \geq 0, Z \geq 0$$

Generally, it is a non-trivial task to obtain $\hat{C}$ directly. Therefore, we introduce an auxiliary variable Z, and optimize the following objective function.

$$\min_{\hat{C}} ||\hat{C}||_* + \alpha_1 tr(F^T L_Z F) + \alpha_2 ||\hat{R}||_1 \quad (10)$$

$$\text{s.t. } Z = \hat{C}, A = \hat{C} + \hat{R} + E, \pi_\Omega(E) = 0, \hat{C} \geq 0, \hat{R} \geq 0, Z \geq 0$$

where L_Z is the Laplacian matrix of Z, $L_Z = D_Z - Z$.

By fixing the others, we can update $\hat{C}$ as follows. First, we introduce the partial augmented Lagrange multipliers and incorporate the equality constraints into the objective function, the problem is transformed as follows.

$$\min_{\hat{C}} ||\hat{C}||_* + \alpha_1 tr(F^T L F) + \alpha_2 ||\hat{R}||_1 + tr(Y_1^T (Z - \hat{C})) + \frac{\mu}{2}||Z-\hat{C}||_F^2 + tr(Y_2^T (A-\hat{C}-\hat{R}-E)) + \frac{\mu}{2}||A-\hat{C}-\hat{R}-E||_F^2 \quad (11)$$

By fixing the others, we can update Z by optimizing the following objective function.

$$\min_{Z} \alpha_1 tr(F^T L_Z F) + tr(Y_1^T (Z - \hat{C})) + \frac{\mu}{2}||Z - \hat{C}||_F^2 \quad (12)$$

Since Eq.(12) is convex and smooth, the closed-form of Z can be derived as follows.

$$Z = \hat{C} - \frac{\alpha_1 H + Y_1}{\mu} \quad (13)$$

where f_i is the i-th row of $F \in R^{n\times k}$, and $H_{ij} = ||f_i - f_j||_F^2$. Since Z is non-negative, during the whole optimization process, Z keeps to be non-negative by $Z = Z_+$, where Z_+ is the positive part of Z, namely, $Z_+ = max(0, Z)$. This is the same to $\hat{C}$ and $\hat{R}$.

By fixing the others, we can update $\hat{C}$ by optimizing the following objective function.

$$\min_{\hat{C}} ||\hat{C}||_* + tr(Y_1^T (Z - \hat{C})) + \frac{\mu}{2}||Z - \hat{C}||_F^2 + tr\big(Y_2^T (A - \hat{C} - \hat{R} - E)\big) + \frac{\mu}{2}||A - \hat{C} - \hat{R} - E||_F^2 \quad (14)$$

By further reformulating the equation (14), the problem is equivalent to optimizing the following well-known objective function [Cai *et al.*, 2010].

$$\min_{\hat{C}} \frac{1}{2\mu}||\hat{C}||_* + \frac{1}{2}||\hat{C} - (\frac{1}{2\mu}(Y_1 + Y_2) + \frac{1}{2}(A + Z - \hat{R} - E))||_F^2 \quad (15)$$

According to the singular value thresholding (SVT) algorithm [Cai *et al.*, 2010], we can obtain $\hat{C}$ as follows.

$$\hat{C} = D_{\frac{1}{2\mu}}\Big((\frac{1}{2\mu}(Y_1 + Y_2) + \frac{1}{2}(A + Z - \hat{R} - E)\Big) \quad (16)$$

where the soft-thresholding operator D_μ is defined as $D_\mu(X) := U D_\mu(\Sigma) V^T$, $D_\mu(\Sigma) = diag(\delta_i - \mu)_+$.

By fixing the others, we can update $\hat{R}$ by optimizing the following objective function.

$$\min_{\hat{R}} \alpha_2 ||\hat{R}||_1 + tr\big(Y_2^T (A - \hat{C} - \hat{R} - E)\big) + \frac{\mu}{2}||A - \hat{C} - \hat{R} - E||_F^2 \quad (17)$$

The problem is equivalent to optimizing the following well-known objective function.

$$\min_{\hat{R}} \frac{\alpha_2}{\mu}||\hat{R}||_1 + \frac{1}{2}||\hat{R} - (Y_2/\mu + (A - \hat{C} - E))||_F^2 \quad (18)$$

According to the shrinkage operator [Hale *et al.*, 2007], we can obtain the $\hat{R}$ as follows.

Algorithm 1 Solving Problem (7)

Input: Data matrix A, parameters: α_1, α_2
Output: $\hat{C}, \hat{R}, E, F$
Initialize $Z = \hat{C} = \mathbf{0}$, $\hat{R} = \mathbf{0}$, $Y_1^0 = \mathbf{0}$, $Y_2^0 = \mathbf{0}$, $\mu = 10^{-6}$, $max_\mu = 10^{10}$, $\rho = 1.5$, $\epsilon = 10^{-8}$
while not converge **do**
  Fix the others and update F by solving problem (8),
    $F = \arg\min_{F^T F = I} tr(F^T L_{\hat{C}} F)$;
  Fix the others and update Z by solving problem (12),
    $Z = \hat{C} - \Big(H + Y_1\Big)/\mu$
  Fix the others and update $\hat{C}$ by solving problem (14),
    $\hat{C} = D_{\frac{1}{2\mu}}\Big(\frac{1}{2\mu}(Y_1 + Y_2) + \frac{1}{2}(A + Z - \hat{R})\Big)$
  Fix the others and update $\hat{R}$ by solving the problem (17),
    $\hat{R} = S_{\frac{\alpha_2}{\mu}}\Big((Y_2/\mu + (A - \hat{C})\Big)$
  Fix the others and update E by solving problem (20),
    $E = \pi_\Omega(A - \hat{C} - \hat{R} + \mu^{-1} Y_2)$
  Update the Lagrange multipliers
    $Y_1 = Y_1 + \mu(Z - \hat{C})$
    $Y_2 = Y_2 + \pi_\Omega\Big(\mu(A - \hat{C} - \hat{R} - E)\Big)$
  Update the parameter μ by $\mu = \min(\rho\mu, max_\mu)$
  Check the convergence conditions
    $||A - \hat{C} - \hat{R} - E||_F^2 < \epsilon$ and $||Z - \hat{C}||_F^2 < \epsilon$
end while

Algorithm 2 Clustering and Link Prediction via CLMC

1: **Input:** Data matrix A, α_1, α_2, number of clusters k
2: Obtain four matrices matrices, $\hat{C}$, $\hat{R}$, E and F by solving Problem (7) via Algorithm 1.
3: Use k-means to cluster the affinity matrix F into k groups.
4: Construct the recovered data matrix $\hat{A} = A - \frac{E+E^T}{2}$.
5: Perform link prediction with $\hat{A}$.

$$\hat{R} = S_{\frac{\alpha_2}{\mu}}\Big((Y_2/\mu + (A - \hat{C} - E)\Big) \tag{19}$$

where the shrinkage operator S_μ is defined as $S_\mu(X) := X - \mu sgn(X) + sgn(X) \odot [\mu - |X|]_+$.

By fixing the others, we can update E by optimizing the following objective function.

$$\min_E tr\big(Y_2^T(A - \hat{C} - \hat{R} - E)\big) + \frac{\mu}{2}||A - \hat{C} - \hat{R} - E||_F^2 \tag{20}$$

To solve the problem, we can update E as follows.

$$E = \pi_\Omega(A - \hat{C} - \hat{R} + Y_2/\mu) \tag{21}$$

Furthermore, the multipliers Y_1 and Y_2 are updated directly by

$$Y_1 = Y_1 + \mu(Z - \hat{C}) \tag{22}$$

$$Y_2 = Y_2 + \pi_\Omega\Big(\mu(A - \hat{C} - \hat{R} - E)\Big) \tag{23}$$

Finally, Algorithm 1 gives the pseudocode of solving Problem (7).

2.4 Graph Clustering and Link Prediction

After solving Problem (7), we can derive three matrices, including a clustering matrix $\hat{C}$, a noise matrix $\hat{R}$ and a completion matrix E, plus the clustering indicator matrix F. We utilize F as a similarity matrix, and then use the k-means algorithm to produce the final clustering results. Moreover, since E is a completion matrix, and we can use it to perform link prediction. Specifically, we can compute the completed matrix $\hat{A} = A - \frac{E+E^T}{2}$. It is important to note that community detection and link prediction work on $\hat{F}$ and $\hat{A}$, respectively, instead of the original data matrix A. Beyond, the two mining tasks are also mutually enhanced. On the one hand, with supplementing missing and/or trend edges while removing noisy edges, it makes clustering be more easy and effective. On the other hand, a good clustering will result in a better matrix completion to support link prediction. Finally, Algorithm 2 gives the pseudocode of graph clustering and link prediction via CLMC.

3 Experiment

We use ten publicly available real-world datasets to evaluate the performance of CLMC, including Karate, Football, Polbooks, Politics-ie, Olympics, Twitter, USAir, Celegans, PB and NS.

3.1 Proof of Concept

Here we start with several synthetic networks featuring distinct characteristics to prove the concept of CLMC. To make the synthetic networks to be more consistent with real-world networks, the LFR benchmark networks [Lancichinetti *et al.*, 2008] have been applied, where the distributions of degree and community size of networks can be easily controlled.

Community Detection with Noise

First, we evaluate how well our clustering algorithm allows detecting communities by varying their inter-cluster edges. We fix node average degree and community size, and change the *mixing parameter* μ from 0.1 to 0.6 to generate a set of networks with different inter-cluster edges. μ is defined as the fraction of links of each node outside its community, which is used to control the difficulty of community separation. With the increase of *mixing parameter*, the performance (measured by NMI) of all eight approaches is shown in Figure 2. We can observe that CLMC, Attractor, Ncut and MCL almost achieve perfect clustering with the *mixing parameter* up to 0.5. The performance begins to decrease with more and more inter-edges added into the network, and CLMC tends to be more robust to these noisy edges. For Modularity, PIC and MCL, the performance is not comparable with other five algorithms on these networks.

Community Detection with Missing Edges

We further evaluate how the algorithms respond to networks with different average degrees, which can be regarded as missing edges in communities. Here we fix the inter-cluster edges, and change the average degree from 5 to 40, and examine the graph clustering performance. Figure 3 shows CLMC allows perfect clustering when the average degree

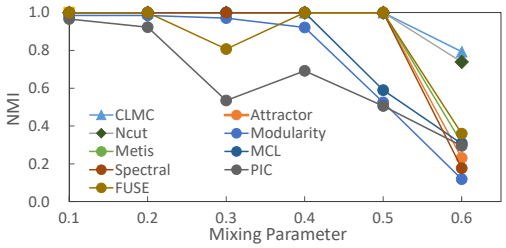


Figure 2: The performance of different algorithms on the LFR benchmark networks by varying the number of inter-cluster edges.

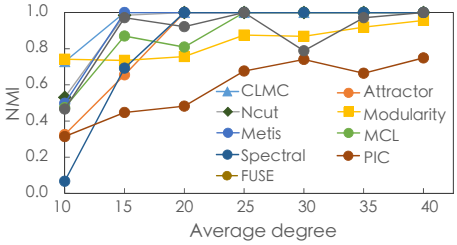


Figure 3: The performance of different algorithms on the LFR benchmark networks by varying community density using the average degree.

Algorithms	Football			Karate			Polbooks			Politics-ie			Olympics			Twitter		
	NMI	ARI	Pur.	NMI	ARI	Pur.	NMI	ARI	Pur.	NMI	ARI	Pur.	NMI	ARI	Pur.	NMI	ARI	Pur.
CLMC	**0.942**	**0.916**	**0.930**	**1.000**	**1.000**	**1.000**	**0.633**	0.757	**0.895**	**0.970**	**0.963**	**0.986**	**0.984**	**0.989**	0.997	**0.909**	**0.838**	**0.923**
Attractor	0.923	0.897	**0.930**	0.859	0.939	**1.000**	0.559	0.680	0.857	0.905	0.942	0.955	0.551	0.417	0.881	0.487	0.127	0.911
Ncut	0.923	0.897	**0.930**	0.833	0.882	0.971	0.565	0.675	0.848	0.940	0.931	0.973	0.959	0.978	0.990	0.824	0.610	0.822
Modularity	0.596	0.474	0.574	0.577	0.680	0.971	0.508	0.638	0.838	0.783	0.739	0.796	0.797	0.858	0.891	0.385	0.177	0.279
Metis	0.393	0.095	0.339	0.836	0.882	0.970	0.502	0.516	0.781	0.738	0.639	0.756	0.698	0.612	0.878	0.427	0.127	0.352
MCL	0.923	0.897	**0.930**	0.833	0.882	0.970	0.455	0.594	0.857	0.605	0.209	0.462	0.612	0.528	**1.000**	0.318	0.061	0.911
Spectral	0.923	0.897	**0.930**	0.426	0.403	0.824	0.629	**0.758**	0.886	0.189	0.077	0.344	0.013	0.005	0.480	0.378	0.067	0.405
PIC	0.556	0.279	0.548	0.833	0.882	0.971	0.583	0.680	0.838	0.642	0.589	0.719	0.834	0.857	0.897	0.415	0.114	0.352
FUSE	0.897	0.856	0.896	0.634	0.669	0.912	0.526	0.612	0.819	0.909	0.886	0.946	0.798	0.829	0.917	0.816	0.578	0.802

Table 1: The performance of different community detection algorithms on real-world data sets. Here the evaluation matrices, NMI, ARI and Purity are reported. We use bold font to mark the best results and underline the second best.

	Football			Karate			Polbooks			Politics-ie			Olympics			Twitter		
	NMI	ARI	Pur.	NMI	ARI	Pur.	NMI	ARI	Pur.	NMI	ARI	Pur.	NMI	ARI	Pur.	NMI	ARI	Pur.
CLMC	**0.941**	**0.915**	0.930	**0.836**	**0.882**	**0.971**	0.573	**0.700**	**0.867**	**0.979**	**0.974**	**0.991**	**0.986**	**0.994**	**0.997**	**0.873**	**0.778**	**0.862**
Attractor	0.920	0.909	**0.939**	0.593	0.702	**0.971**	0.515	0.600	0.848	0.886	0.812	0.914	0.570	0.431	0.758	0.820	0.611	0.773
Ncut	0.930	0.906	0.930	0.833	0.882	**0.971**	0.548	0.646	0.838	0.921	0.908	0.964	**0.986**	**0.994**	**0.997**	0.790	0.573	0.769
Modularity	0.656	0.522	0.617	0.577	0.680	**0.971**	0.496	0.610	0.838	0.783	0.739	0.796	0.831	0.904	0.891	0.365	0.211	0.340
Metis	0.428	0.148	0.435	**0.836**	0.882	**0.971**	0.506	0.519	0.800	0.775	0.673	0.837	0.678	0.598	0.871	0.426	0.119	0.344
MCL	0.919	0.896	0.930	**0.836**	0.882	**0.971**	**0.587**	0.675	0.848	0.846	0.726	0.882	0.844	0.857	0.957	0.694	0.286	0.733
Spectral	0.921	0.899	0.930	0.181	0.110	0.676	0.551	0.681	0.857	0.189	0.077	0.344	0.013	0.005	0.480	0.272	0.040	0.328
PIC	0.523	0.241	0.522	0.181	0.110	0.676	0.583	0.682	0.838	0.614	0.532	0.692	0.797	0.817	0.894	0.396	0.092	0.328
FUSE	0.886	0.831	0.896	0.496	0.572	0.882	0.585	0.642	0.829	0.885	0.865	0.932	0.784	0.801	0.907	0.770	0.603	0.741

Table 2: The performance of different community detection algorithms on real-world data sets with missing edges of 10%. Here NMI, ARI and Purity are reported. We use bold font to mark the best results and underline the second best.

ranges from 15 to 40. For Spectral, Metis, FUSE and Modularity, they cannot produce promising results.

3.2 Community Detection Evaluation

Here, we compare CLMC to eight representatives of community detection algorithms: Ncut, Modularity, Metis, MCL, Spectral, PIC, FUSE and Attractor. **Ncut** [Shi and Malik, 2000] is a well-known algorithm for graph clustering by optimizing the *normalized cut* criterion. **Modularity** [Newman, 2006] is a popular community detection algorithm based on the *modularity* measure. **Metis** [Karypis and Kumar, 1998] is a typical clustering approach for large networks via multilevel partitioning and parallelized implementation. **MCL** [Van Dongen, 2000] is an algorithm widely used in life sciences based on the simulation of (stochastic) flow in graphs. **Spectral** is a typical clustering algorithm which allows finding arbitrarily shaped clusters [Lin and Cohen, 2010]. **PIC** is a power-iteration-based clustering method based on spectral clustering [Ng *et al.*, 2002]. **FUSE** is a new spectral clustering approach by employing ensemble learning and ICA [Ye *et al.*, 2016]. **Attractor** is a model-based community detection approach based on distance dynamics [Shao *et al.*, 2015].

For all experiments, Ncut, Modularity, Metis, Spectral, PIC and FUSE specify the number of clusters ($k = |C|$), where $|C|$ is the true number of classes. For MCL, we perform parameter tuning on the range (1.0, 4.0) with step size of 0.2. For Attractor, the cohesion parameter varies from 0.1 to 1.0 with stepsize of 0.1. For CLMC, α_1 takes the values from $(0.001, 0.1, 1)$ and α_2 takes the values from $(0.1, 1, 10)$. For all algorithms, the best results are recorded. In addition, since the performance of Spectral, PIC, FUSE and CLMC depend on the initialization with k-means clustering, we run them for 10 times, and the best results are reported. For networks whose communities are already known, the performance is measured by three widely used evaluation measures: Normalized Mutual Information (NMI) [Strehl and Ghosh, 2003], Adjusted Rand Index (ARI) [Rand, 1971] and Cluster Purity.

Table 1 summarizes the performance of different community detection algorithms on six real-world data sets. From the table, we can observe that CLMC achieves good performance on all data sets. On the Karate data, it even produces perfect clustering. The success of CLMC on community detection lies in the removal of noisy edges and the completion of missing/trend connections. Table 2 further reports the performance of different algorithms on the real-world data sets with 10% missing edges, and we can see that CLMC outperforms the state-of-the-art algorithms.

3.3 Link Prediction Evaluation

As an other function, CLMC also supports link prediction with the supplement matrix. Here we further evaluate its link prediction performance.

	CLMC	CN	Jaccard	HPI	HDI	LHN	AA	RA	PA	LP	Katz	LHNII	ACT	RWR	SimRank	TS	MF
Football	**0.930**	0.835	0.848	0.846	0.848	0.850	0.838	0.838	0.266	0.853	0.851	0.870	0.601	0.875	0.893	0.619	0.751
Karate	**0.918**	0.705	0.617	0.717	0.606	0.604	0.744	0.754	0.720	0.721	0.714	0.492	0.670	0.902	0.652	0.749	0.729
Polbooks	**0.931**	0.891	0.871	0.888	0.859	0.836	0.900	0.901	0.703	0.912	0.911	0.821	0.748	0.925	0.871	0.256	0.832
Politics-ie	**0.960**	0.943	0.940	0.936	0.932	0.898	0.947	0.951	0.721	0.940	0.940	0.836	0.749	0.952	0.904	0.382	0.921
Olympics	**0.936**	0.907	0.906	0.880	0.892	0.794	0.913	0.924	0.798	0.905	0.904	0.681	0.802	0.920	0.807	0.495	0.923
Twitter	**0.917**	0.896	0.889	0.893	0.874	0.828	0.907	0.913	0.702	0.904	0.904	0.741	0.711	0.910	0.842	0.494	0.804
USAir	**0.965**	0.929	0.893	0.867	0.887	0.767	0.940	0.946	0.877	0.925	0.922	0.619	0.894	0.943	0.782	0.594	0.921
Celegans	**0.909**	0.851	0.798	0.810	0.787	0.732	0.869	0.873	0.750	0.854	0.853	0.602	0.742	0.901	0.764	0.447	0.829
PB	**0.943**	0.918	0.873	0.852	0.869	0.762	0.921	0.922	0.901	0.929	0.929	0.639	0.884	0.934	0.774	0.492	0.928
NS	**0.971**	0.932	0.932	0.932	0.932	0.931	0.932	0.932	0.673	0.937	0.934	0.966	0.596	0.932	0.931	0.862	0.637

Table 3: The performance of seventeen link prediction algorithms on ten data sets. Here the AUC values are reported, and we use bold font to mark the best results and underline the second best.

Here, we compare CLMC with sixteen link prediction algorithms, including neighor-based methods: common neighbors (CN), Jaccard, Hub Promoted Index (HPI), Hub Depressed Index (HDI), Leicht-Holme-Newman Index (LHN1 and LHN2), Adamic-Adar (AA), resource allocation (RA), preferential attachment (PA), path-based methods: Katz, local path index, average commute time (ACT), random walk with restart (RWR), transfer similarity (TS), SimRank, and latent-based method: matrix factorization (MF). For details, please refer to the link prediction survey paper [Martínez *et al.*, 2017]. For Katz, the damping factor is searched from 0.001 and 0.01. The free parameter ϕ is searched in $\{0.90, 0.95, 0.99\}$. For RWR, the probability parameter c is searched from 0.85 and 0.95. For MF, the number of latent factors is searched in $\{5, 10, 15, 20, 50\}$. The best results on all data sets are recorded. Moreover, the standard metric: the area under the ROC curve (AUC) [Hanley and McNeil, 1982], is used to measure the link prediction performance.

For all data sets, existing links are randomly split into: a training set (90%) and a test set (10%). After that, all link prediction approaches perform experiments on training test, and validate its performance on test set with AUC. Table 3 summarizes the link prediction performances of different algorithms on ten data sets with 10% missing connections. From Table 3, we can observe that CLMC shows its superiority over all comparing link prediction algorithms. The potential reason is that the community detection and link prediction are mutually enhanced. The clustering constraint will make more intra-connections to be supplemented.

3.4 Parameter Analysis

For CLMC, it involves two parameters α_1 and α_2. Figure 4 shows the different community detection performance by varying the values of α_1 and α_2. From the figure, we can see that CLMC can achieve a relatively stable good performance (indicating by the yellow blocks) when α_1 ranges from 0.001 to 1, and α_2 ranges from 0.01 to 10 on the four data sets.

4 Conclusions

In this paper, we consider community detection and link prediction simultaneously in a unified framework with cluster-driven low-rank matrix completion. Instead of working on the original network directly, the proposed framework learns a new ideal clustering matrix to perform graph clustering and a supplement matrix to conduct link prediction. Experimental results show that the proposed CLMC achieves better results on both tasks of community detection and link prediction over many state-of-the-art approaches.

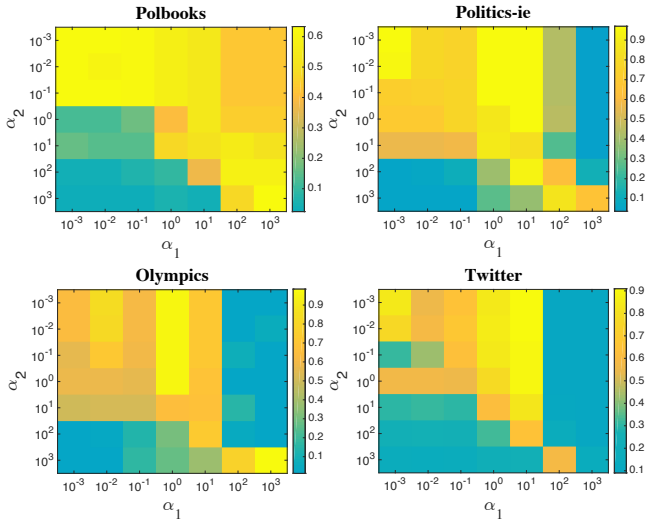


Figure 4: The performance of CLMC with differernt α_1 and α_2 in a wide range of grids in term of NMI.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (61403062, 61433014, 41601025), Science-Technology Foundation for Young Scientist of Sichuan Province (2016JQ0007), Fok Ying-Tong Education Foundation for Young Teachers in the Higher Education Institutions of China (161062) and National key research and development program (2016YFB0502300).

References

[Cai *et al.*, 2010] Jian-Feng Cai, Emmanuel J Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

[Candès and Recht, 2009] Emmanuel J Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717, 2009.

[Chung, 1997] Fan RK Chung. *Spectral graph theory*. Number 92. American Mathematical Soc., 1997.

[Fan and Chow, 2017] Jicong Fan and Tommy WS Chow. Matrix completion by least-square, low-rank, and sparse self-representations. *Pattern Recognition*, 71:290–305, 2017.

[Fan, 1949] Ky Fan. On a theorem of weyl concerning eigenvalues of linear transformations i. *Proceedings of the National Academy of Sciences*, 35(11):652–655, 1949.

[Hale *et al.*, 2007] Elaine T Hale, Wotao Yin, and Yin Zhang. A fixed-point continuation method for l1-regularized minimization with applications to compressed sensing. *CAAM TR07-07, Rice University*, 43:44, 2007.

[Hanley and McNeil, 1982] James A Hanley and Barbara J McNeil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36, 1982.

[Hastie *et al.*, 2015] Trevor Hastie, Rahul Mazumder, Jason D Lee, and Reza Zadeh. Matrix completion and low-rank svd via fast alternating least squares. *The Journal of Machine Learning Research*, 16(1):3367–3402, 2015.

[Kang *et al.*, 2016] Zhao Kang, Chong Peng, and Qiang Cheng. Top-n recommender system via matrix completion. In *AAAI*, pages 179–185, 2016.

[Karypis and Kumar, 1998] George Karypis and Vipin Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on scientific Computing*, 20(1):359–392, 1998.

[Keshavan *et al.*, 2010] Raghunandan H Keshavan, Andrea Montanari, and Sewoong Oh. Matrix completion from noisy entries. *Journal of Machine Learning Research*, 11(Jul):2057–2078, 2010.

[Lancichinetti *et al.*, 2008] Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. Benchmark graphs for testing community detection algorithms. *Physical review E*, 78(4):046110, 2008.

[Lin and Cohen, 2010] Frank Lin and William W Cohen. Power iteration clustering. 2010.

[Liu *et al.*, 2013] Guangcan Liu, Zhouchen Lin, Shuicheng Yan, Ju Sun, Yong Yu, and Yi Ma. Robust recovery of subspace structures by low-rank representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):171–184, 2013.

[Lu *et al.*, 2016] Jin Lu, Guannan Liang, Jiangwen Sun, and Jinbo Bi. A sparse interactive model for matrix completion with side information. In *Advances in neural information processing systems*, pages 4071–4079, 2016.

[Martínez *et al.*, 2017] Víctor Martínez, Fernando Berzal, and Juan-Carlos Cubero. A survey of link prediction in complex networks. *ACM Computing Surveys (CSUR)*, 49(4):69, 2017.

[Mohar *et al.*, 1991] Bojan Mohar, Y Alavi, G Chartrand, and OR Oellermann. The laplacian spectrum of graphs. *Graph theory, combinatorics, and applications*, 2(871-898):12, 1991.

[Natarajan and Dhillon, 2014] Nagarajan Natarajan and Inderjit S Dhillon. Inductive matrix completion for predicting gene–disease associations. *Bioinformatics*, 30(12):i60–i68, 2014.

[Newman, 2006] Mark EJ Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006.

[Ng *et al.*, 2002] Andrew Y Ng, Michael I Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*, pages 849–856, 2002.

[Rand, 1971] William M Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850, 1971.

[Shao *et al.*, 2015] Junming Shao, Zhichao Han, Qinli Yang, and Tao Zhou. Community detection based on distance dynamics. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1075–1084. ACM, 2015.

[Shao *et al.*, 2017] Junming Shao, Xinzuo Wang, Qinli Yang, Claudia Plant, and Christian Böhm. Synchronization-based scalable subspace clustering of high-dimensional data. *Knowledge and Information Systems*, 52(1):83–111, 2017.

[Shao *et al.*, 2018] Junming Shao, Qinli Yang, Zhong Zhang, Jinhu Liu, and Stefan Kramer. Graph clustering with local density-cut. In *International Conference on Database Systems for Advanced Applications*, pages 187–202. Springer, 2018.

[Shi and Malik, 2000] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE TPAMI*, 22(8):888–905, 2000.

[Strehl and Ghosh, 2003] Alexander Strehl and Joydeep Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *The Journal of Machine Learning Research*, 3:583–617, 2003.

[Van Dongen, 2000] Stijn Van Dongen. A cluster algorithm for graphs. *Report-Information systems*, (10):1–40, 2000.

[Yang *et al.*, 2018] Linxiao Yang, Jun Fang, Huiping Duan, Hongbin Li, and Bing Zeng. Fast low-rank bayesian matrix completion with hierarchical gaussian prior models. *IEEE Transactions on Signal Processing*, 66(11):2804–2817, 2018.

[Ye *et al.*, 2016] Wei Ye, Sebastian Goebl, Claudia Plant, and Christian Böhm. Fuse: Full spectral clustering. In *ACM SIGKDD*, pages 1985–1994, 2016.

[Yin *et al.*, 2017] Hao Yin, Austin R Benson, Jure Leskovec, and David F Gleich. Local higher-order graph clustering. In *ACM SIGKDD*, pages 555–564, 2017.