Noname manuscript No.
(will be inserted by the editor)

Clustering of Noisy Graphs via Non-negative Matrix Factorization With Sparsity Regularization

Yijie Wang · Xiaoning Qian

Received: date / Accepted: date

Abstract In this paper, we propose a flexible noise tolerant graph clustering formulation based on non-negative matrix factorization (NMF), which solves graph clustering under different settings, such as community detection and block modeling problems for either undirected or directed graphs. Comparing to the existing graph clustering algorithms, many of which do not perform robustly when graphs of interest contain high noise, we improve the noise tolerance in our NMF model by explicitly controlling the sparsity for every decomposed factor. Based on our sparsity regularized formulation, we develop a novel alternating proximal method (APANMF) to solve the challenging optimization problem. Furthermore, we prove that APANMF converges to a stationary point. Experiments on well-known synthetic networks for different graph clustering tasks and real-world networks demonstrate that our APANMF outperforms other state-of-the-art NMF-based graph clustering methods in terms of flexibility and noise tolerance.

Keywords Non-negative Matrix Factorization · Community Detection · Block Modeling · Sparsity Regularization · Proximal Method

1 Introduction

The advanced data profiling technologies have produced a large number of network datasets in different research areas. Without prior knowledge, an important research challenge is to extract useful information carried in those

Yijie Wang
Department of Electrical and Computer Engineering, College Station, Texas A&M University. E-mail: yijie@tamu.edu

Xiaoning Qian
Department of Electrical and Computer Engineering, College Station, Texas A&M University. E-mail: xqian@ece.tamu.edu

network datasets by graph clustering methods, which help visualize and understand the topology or structures of graphs and reduce the dimensionality of network datasets so that further analyses can be carried out for knowledge discovery.

Non-negative Matrix Factorization (NMF) can produce meaningful non-negative representations of the given original datasets [1–4]. Recently, NMF has been successfully applied for graph clustering [5–8]. The authors in [6,5] propose to decompose the adjacency matrix of an undirected graph into symmetric non-negative components to identify communities under the assumption that all clusters consist of highly connected vertices. Further investigation has demonstrated its potential for detecting overlapping clusters in graphs [6, 9,10]. In order to handle more challenging directed graph clustering problems, an asymmetric NMF formulation [7] has been proposed to allow an asymmetric matrix to capture the connectivity profiles among different clusters so that the new formulation can deal with asymmetric adjacency matrices in directed graphs. For more general blockmodel clustering problems, where vertices with similar interaction patterns are considered to play the same role in the graph and hence belong to the same cluster, the authors in [8] propose to tackle the problem with the introduction of an image graph, which presents the latent blockmodel structures of graphs.

One important property of NMF-based graph clustering formulations is that they usually yield a sparse representation of the original graph. With the assumption that the multiplication of factorized components can closely recover original adjacency matrices, when the network datasets are noise free or the blockmodel structures in adjacency matrices are obvious under a low noise level, sparse factorization can be obtained by the existing formulations [5–8]. However, the real-world network datasets are often noisy and such low-noise assumptions may not be satisfied. In fact, without the explicit control of the sparsity of each factorized component, previously proposed methods [5–8] often yield non-sparse results empirically when the noise of network topology is high, which are difficult to interpret for understanding the inherent structures in these networks.

Due to the complexity of general graph clustering problems, the corresponding optimization algorithms for different NMF-based formulations may have different convergence guarantees. For the algorithms SymNMF_MU [6] and ASymNMF [7], the denominators of the corresponding multiplicative algorithms are not well-defined [11] so that SymNMF_MU and ASymNMF may not converge. Additionally, for ASymNMF, the authors in [11] point out that merely a proof of the monotonic decreasing property of the objective function values does not imply the convergence to a stationary point. Hence, SymNMF_MU and ASymNMF may not converge to a stationary point. For the algorithm BNMF [8], there has been no convergence proof provided and the quality of the solutions obtained by BNMF has no theoretical guarantee either. Another algorithm SymNMF_NT [5] does converge to a stationary point. However, it may consume prohibitive amount of memory and does not scale up well with the number of clusters.

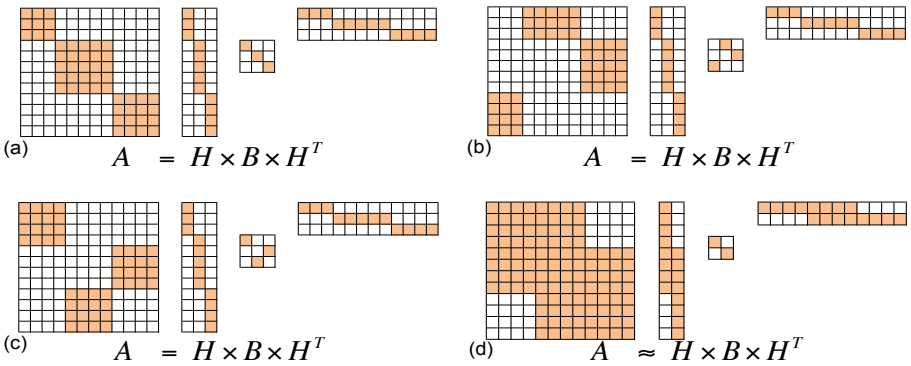

Fig. 1: Graph clustering under different settings: (a) Toy example for community detection for undirected graph. (b) Toy example for directed graph clustering. (c) Toy example for block modeling clustering. (d) Toy example for overlapping graph clustering.

1.1 Main Contributions

First of all, in order to obtain robust results for noisy graphs, we propose a new formulation by explicitly adding an L1-norm sparsity penalty to each factorized component. Second, we develop a novel alternating proximal method (APANMF) to efficiently solve noisy graph clustering based on our new formulation. We prove that APNNMF converges to a stationary point without any assumption about the convexity or existence of stationary points. To the best of our knowledge, it is the first convergence proof of a coordinate descent method for solving this optimization problem, whose constraint set is convex but the objective function is non-convex and non-differentiable.

1.2 Roadmap

We briefly review the related work in Section 2, followed by the derivation of our novel formulation in Section 3 and the alternating proximal method (APANMF) in Section 4. The convergence-related propositions of our APANMF (Propositions 1, 2 and 3) are also provided in Section 4. In Section 5, we demonstrate the superiority of our APANMF by comparing with other state-of-the-art methods (SymNMF_MU [6], SymNMF_NT [5], ASymNMF [7], BNMF [8]) on synthetic networks (LFR benchmarks [12] and block modeling benchmarks [13]) as well as real-world large-scale network datasets (Facebook ego network from http://snap.stanford.edu/data/ and PIPs human protein-protein interaction (PPI) network [14]). We draw the conclusion in Section 6.

2 Related Work

For a given graph $G = (V, E)$ with $|V| = N$ vertices connected by $|E| = M$ edges, The authors in [5,6] propose to decompose the corresponding adjacency

matrix $A_{N\times N}$ for a given undirected graph, with $A_{ij}=1$ denoting that vertex i connects to vertex j and $A_{ij}=0$ otherwise, into symmetric components for community detection:

$$\min_{H\geq 0}: \Gamma(H)=\left\|A-HH^T\right\|_F^2, \tag{1}$$

where H is a non-negative matrix of size $N\times K$ and K is the number of potential clusters. H can be naturally interpreted as the cluster assignment matrix for graph clustering. A multiplicative updating algorithm SymNMF_MU [6] has been proposed to solve this problem (1). However, SymNMF_MU may not converge to a stationary point, which will be further discussed in Section 4.3. SymNMF_NT [5] is a Newton-like algorithm, which solves the problem (1) by lining up the columns of H. SymNMF_NT converges to a stationary point. However, it has relatively larger memory consumption requirement [5].

In order to handle directed graphs, the authors in [7] have presented an asymmetric NMF decomposition formulation:

$$\min_{H\geq 0,\ S\geq 0}: \Pi(H,S)=\left\|A-HSH^T\right\|_F^2, \tag{2}$$

where $S_{K\times K}$ is a $K\times K$ asymmetric matrix for handling the asymmetric adjacency matrix A of a directed graph. A multiplicative updating algorithm ASymNMF [7] has been developed to solve this problem (2). The objective function values generated by ASymNMF monotonically decrease but the solution may not converge to a stationary point, which is discussed in Section 4.3.

For block modeling graph clustering, one recent algorithm—BNMF [8]—has been derived base on the following formulation:

$$\min_{H\geq 0,\ 0\leq M\leq 1}: \left\|A-HMH^T\right\|_F^2+\lambda\left\|M^{ideal}-M\right\|_F^2, \tag{3}$$

where M and M^{ideal} represent the adjacency matrices of the introduced image graph and the "ideal image matrix", respectively. M^{ideal} is the function of M, which is defined by $M_{ij}^{ideal}=\underset{u\in\{0,1\}}{argmin}\,|u-M_{ij}|$ and approximated by a sigmoid function in the proposed projected descent algorithm. However, there is no convergence proof provided for BNMF.

3 Flexible Graph Clustering with L1-norm Regularization

Adopting different NMF-based formulations can address different graph clustering problems, such as aforementioned community detection and block modeling for either undirected or directed graphs, by different formulations (1), (2), and (3). In this section, we propose a mathematical formulation, which can deal with all the above clustering tasks in just one flexible framework. Furthermore, we explicitly control the sparsity of factorized components by adding L1-norm penalty terms to yield sparse and robust solutions for noisy graphs.

3.1 A Flexible Graph Clustering Formulation

Our graph clustering formulation is based on the similar assumption that the given adjacency matrix can be factorized by the multiplications of a clustering assignment matrix and an adjacency matrix of the image graph capturing the underlying topology of the given graph $A \approx HBH^T$ [15]:

$$\begin{aligned} \text{min: } & \left\|A - HBH^T\right\|_F^2, \\ s.t.\ & H_{ij} \in \{0,1\}, \forall i,j; \\ & B_{rs} \in \{0,1\}, \forall r,s, \end{aligned} \tag{4}$$

where H is the $N \times K$ dimensional assignment matrix with $H_{ir} = 1$ revealing that vertex i belongs to cluster r and $H_{ir} = 0$ otherwise. The introduced image graph is presented by the adjacency matrix B, in which B_{rs} indicates the connectivity between the cluster r and the cluster s with $B_{rs} = 1$ meaning that cluster r densely interacts with the cluster s and $B_{rs} = 0$ otherwise. We note that our formulation is similar to (2), but with the binary constraints on both H and B. Clustering by our formulation may provide better physical interpretations for both the assignment matrix H and the image graph B.

Fig. 1 illustrates how we formulate graph clustering under different settings. With the help of an image graph B, which captures the underlying topological organization of the graph, clusters of vertices with similar topological properties can be successfully identified by using our flexible formulation (4) for all four different graph clustering tasks.

By solving the optimization problem (4), we can obtain the promising graph clustering results. However, it is challenging to find integer solutions for this nonlinear optimization problem (4) due to the inherent NP hardness of general network clustering as a quadratic assignment problem [13,16], especially with large-scale networks. Relaxing the constraints from integer to continuous variables is one typical way to achieve high quality solutions [17]. In this paper, we relax our binary constraints as follows:

$$\varphi = \{(H,B) | 0 \le H_{ij} \le 1, 0 \le B_{rs} \le 1, \forall i,j,r,s\}. \tag{5}$$

The relaxed search space φ allows the elements in H and B range from 0 to 1. After relaxation (5), our problem becomes:

$$\min_{(H,B)\in\varphi} : \Psi(H,B) = \left\|A - HBH^T\right\|_F^2. \tag{6}$$

3.2 L1-norm Regularization

For noise free networks, such as toy examples given in Fig. 1, or networks with reasonably low noise, our proposed formulation (6) can naturally produce sparse results with original clustering structures because the assumption $A \approx HBH^T$ holds. However, for real-world networks, which often contain significant amount of noise due to limitations of interaction profiling methods,

the underlying clustering structures may be destroyed and the assumption $A \approx HBH^T$ may not be satisfied. Hence, we may not be able to have meaningful sparse results by directly solving (6). In order to address this problem, we add L1-norm regularization terms for both H and B to (6) to explicitly enforce sparse structures for H and B:

$$\min_{(H,B)\in\varphi} : \Omega(H,B) = \left\|A - HBH^T\right\|_F^2 + \alpha \left\|H\right\|_{L1} + \beta \left\|B\right\|_{L1}, \tag{7}$$

in which $\|X\|_{L1} = \sum_{i,j} |X_{ij}|$. With the newly added regularization terms, we hope for the guarantee of physically meaningful sparse results, especially for noisy networks.

4 Alternating Proximal Algorithm

To solve this sparse NMF-based graph clustering problem, we now derive a new set of optimization algorithms, which are different from the existing algorithms, mostly based on multiplicative updating algorithms for the original NMF algorithm [1]. Mathematically, our optimization problem (7) is more challenging to solve with two non-differentiable terms in $\Omega(H,B)$, compared to the optimization problems (1), (2) and (3). In order to efficiently solve this optimization problem (7), we need to make use of the structure of the objective function, which takes the sum of a differentiable component and other non-differentiable components. Based on this observation, we develop an alternating proximal method that optimizes the cluster assignment matrix H and the image matrix B in an alternating way. This alternating proximal algorithm is guaranteed to converge to a stationary point of the optimization problem (7).

4.1 Updating H

Let us first consider the optimization step with respect to the assignment matrix H by fixing the image matrix at $\hat{B}$. The decomposed optimization problem aims to solve the following problem:

$$\min_{0\le H\le 1} : F(H) = \left\|A - H\hat{B}H^T\right\|_F^2 + \alpha \left\|H\right\|_{L1}, \tag{8}$$

where we define $P(H) = \left\|A - H\hat{B}H^T\right\|_F^2$ and $P(H)$ is differentiable.

Because of the structure of the problem, we apply a proximal method to iteratively solve the optimization problem. As similarly done in [18], we propose to compute $G_k(H)$ for the approximation of $F(H)$ at the kth iteration around H^{k-1}:

$$\begin{aligned} G_k(H) = P(H^{k-1}) &+ < \nabla P(H^{k-1}), (H - H^{k-1}) > \\ &+ \frac{L_k}{2} \left\|H - H^{k-1}\right\|_F^2 + \alpha \left\|H\right\|_{L1}, \end{aligned} \tag{9}$$

where L_k is a Lipschitz constant, which can be chosen to satisfy the following inequality:

$$G_k(H^k) \geq F(H^k). \tag{10}$$

Hence, instead of finding H^k based on $F(H^{k-1})$, our proximal method solves the following problem at the kth iteration:

$$H^k = \underset{0 \leq H \leq 1}{\arg\min} : G_k(H). \tag{11}$$

After some algebraic manipulations by completing the square and removing the constant terms, the problem (11) is in fact equivalent to the following problem:

$$H^k = \underset{0 \leq H \leq 1}{\arg\min} \left\{ \alpha \left\| H \right\|_{L1} + \frac{L_k}{2} \left\| H - \left(H^{k-1} - \frac{1}{L_k} \nabla P(H^{k-1}) \right) \right\|_F^2 \right\}. \tag{12}$$

Algorithm 1 Proximal Method for updating H (PMH($H, \hat{B}$))

1. **Input:** H^0, $\hat{B}$, $k = 1$, $L_0 > 1$, $\eta > 0$ and $\xi > 0$;
2. **Output:** H^*;
3. **do**
4. Find the smallest non-negative integer i_k such that inequality (10) is satisfied with $L_k = \eta^{i_k} L_{k-1}$;
5. Obtain H^k from (14);
6. $k = k + 1$;
7. **while**$\left(F(H^{k-1}) - F(H^k) > \xi\right)$
8. $H^* = H^k$.

Furthermore, we notice that this equivalent problem (12) has a closed-form solution, which is a promising property of our proximal method. With the closed-form solution, we can efficiently solve (12) without intensive computation. The closed-from solution is provided in Proposition 1, whose proof is given in the appendix.

Proposition 1 *For the following optimization problem:*

$$H^k = \underset{0 \leq H \leq 1}{\arg\min} \left\{ \phi(H) = \alpha \left\| H \right\|_{L1} + \frac{L_k}{2} \left\| H - \bar{H} \right\|_F^2 \right\}, \tag{13}$$

where $\bar{H} = H^{k-1} - \frac{1}{L_k} \nabla P(H^{k-1})$*, the element-wise closed-form solution is*

$$H_{ij}^k = \mathbb{P}(\mathbf{prox}_H(\bar{H})_{ij}), \tag{14}$$

where $\mathbb{P}(\cdot)$ *is the projection operator and it is defined by*

$$\mathbb{P}(x) = \begin{cases} 1 & x > 1 \\ x & 0 \leq x \leq 1 \\ 0 & x < 0 \end{cases}, \tag{15}$$

and

$$\begin{aligned}&\mathbf{prox}_H(\bar{H})\\&= \arg\min_H \left\{ \phi(H) = \alpha \left\| H \right\|_{L1} + \frac{L_k}{2} \left\| H - \bar{H} \right\|_F^2 \right\},\end{aligned} \tag{16}$$

whose result is the solution of $\frac{\partial \phi(H)}{\partial H} \ni \boldsymbol{0}$ *and can be computed in the following equation:*

$$\mathbf{prox}_H(\bar{H})_{ij} = \begin{cases} 0 & |\bar{H}_{ij}| \leq \frac{\alpha}{L_k} \\ \bar{H}_{ij} - \frac{\alpha}{L_k} \text{sign}(\bar{H}_{ij}) & |\bar{H}_{ij}| > \frac{\alpha}{L_k} \end{cases} \tag{17}$$

The proximal method for updating H (PMH) is described in Algorithm 1. The convergence of PMH is guaranteed by Proposition 2 with the proof given in the appendix.

Proposition 2 *The sequence* $\left\{F(H^k)\right\}_{k\geq 0}$ *generated by the algorithm in Algorithm 1 monotonically decreases and the sequence* $\left\{S_k(H^k) = G_k(H^k) - F(H^k)\right\}_{k\geq 0}$ *converges to zero. Furthermore, when* $k \mapsto +\infty$, H^k *satisfies an asymptotic stationary point condition.*

Algorithm 2 Proximal Method for updating B (PMB($\hat{H}, B$))

1. **Input:** $\hat{H}$, B^0, $k = 1$ and $\xi > 0$;
2. **Output:** B^*;
3. **do**
4. Compute $U_k(B)$ based on (19);
5. Compute B^k based on (21);
6. $k = k + 1$
7. **while**$(E(B^{k-1}) - E(B^k) > \xi)$
8. $B^* = B_k$.

4.2 Updating B

Updating B is similar as updating H because the optimization with B has the same structure as (8). Given an assignment matrix $\hat{H}$, the optimization problem we want to solve is:

$$\min_{0\leq B\leq 1} : \; E(B) = \left\| A - \hat{H}B(\hat{H})^T \right\|_F^2 + \beta \left\| B \right\|_{L1}, \tag{18}$$

where $\|B\|_{L1}$ is the non-smooth term while $\Phi(B) = \left\| A - \hat{H}B(\hat{H})^T \right\|_F^2$ is differentiable with the gradient $\nabla\Phi(B) = 2((\hat{H})^T\hat{H}B(\hat{H})^T\hat{H} - (\hat{H})^T A\hat{H})$. Here, the square of the largest eigenvalue of $(\hat{H})^T\hat{H}$ is $\Phi(B)$'s Lipschitz constant L_B, which can be proven with Lemma 1 in the appendix.

We adopt a similar proximal method by approximating $E(B)$ in (18) at B^{k-1} by an upper-bound function:

$$\begin{aligned} U_k(B) &= \Phi(B^{k-1}) + < \nabla\Phi(B^{k-1}), (B - B^{k-1}) > \\ &+ \frac{L_B}{2} \left\| B - B^{k-1} \right\|_F^2 + \beta \left\| B \right\|_{L1} \\ &= \beta \left\| B \right\|_{L1} + \frac{L_B}{2} \left\| B - \bar{B} \right\|_F^2, \end{aligned} \tag{19}$$

where $\bar{B} = B^{k-1} - \frac{1}{L_B}\nabla\Phi(B^{k-1})$. At the kth iteration, we solve the optimization problem:

$$B^k = \underset{0 \le B \le 1}{\arg\min} : U_k(B). \tag{20}$$

The corresponding closed-form optimal solution is derived similarly as Proposition 1

$$B_{ij}^k = \mathbb{P}(\mathbf{prox}_B(\bar{B})_{ij}), \tag{21}$$

where

$$\mathbf{prox}_B(\bar{B})_{ij} = \begin{cases} 0 & |\bar{B}_{ij}| \le \frac{\beta}{L_B} \\ \bar{B}_{ij} - \frac{\beta}{L_B}\text{sign}(\bar{B}_{ij}) & |\bar{B}_{ij}| > \frac{\beta}{L_B} \end{cases}, \tag{22}$$

Algorithm 2 details the procedure of the proximal method for updating B (PMB). We note that $E(B)$ is convex with respect to B and the constraint set $0 \le B \le 1$ is also convex. Therefore, the algorithm (PMB) converges to an optimal solution for a fixed $\hat{H}$ [19].

4.3 Alternating Proximal Algorithm for NMF (APANMF)

With both the algorithms PMH and PMB in hands, we summarize the alternating proximal algorithm (APANMF) in Algorithm 3. The convergence of APANMF is guaranteed by Proposition 3, whose proof is provided in the appendix.

Proposition 3 *The sequence of* $\{\Omega(H^t, B^t)\}_{t\ge 0}$ *monotonically decreases*

$$\Omega(H^{t+1}, B^{t+1}) \le \Omega(H^t, B^t). \tag{23}$$

Furthermore, the sequence $\{(H^t, B^t)\}_{t\ge 0}$ *converges to an asymptotic stationary point.*

One profound contribution of our APANMF is that APANMF has the theoretical guarantee to converge to a stationary point, which neither SymNMF_MU nor ASymNMF has provided. Additionally, to the best of our knowledge, this is the first convergence proof of a coordinate descent method for solving the NMF problem, one of whose decomposed optimization problems is non-convex and non-smooth. Therefore, our proof could provide insightful guidance for the convergence proof of the NMF problems with similar

Algorithm 3 Alternating Proximal Algorithm

1. **Input:** $A_{N\times N}$ and K;
2. **Output:** H and B;
3. Initialization: $H^0_{N\times K} > 0$, $B^0_{K\times K} > 0$ and $t = 1$;
4. **do**
5. $\quad H^{t+1}$ =PMH(H^t, B^t);
6. $\quad B^{t+1}$ =PMB(H^{t+1}, B^t);
7. $\quad t = t + 1$;
8. **while**$(\Omega(H^{t-1}, B^{t-1}) - \Omega(H^t, B^t) > \xi)$
9. Compute H by normalizing each row of H^t to have the unit length.

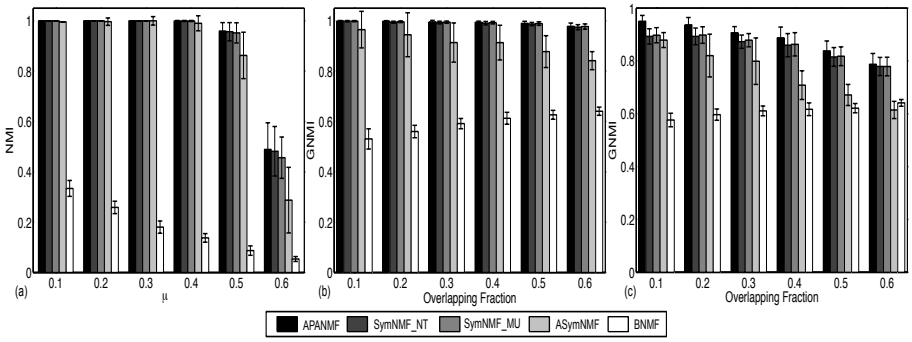

Fig. 2: Performance comparison for undirected graph clustering: (a) NMI comparison (non-overlapping) with increasing mixing parameter μ. (b) GNMI comparison (overlapping) with increasing overlapping fraction values θ when $\mu = 0.1$. (c) GNMI comparison (overlapping) with increasing overlapping fraction values θ when $\mu = 0.3$.

structures. For SymNMF in general settings, the stationary points of the optimization problem in (1) necessarily contain zero elements: $\exists i, j,\ H^*_{ij} = 0$ (Proposition 4 in the appendix). Meanwhile, the proposed multiplicative algorithm SymNMF [5] always generates iterative updates in the positive orthant (Proposition 6): $H^k_{ij} > 0, \forall i, j$. Therefore, SymNMF may not converge in general when $\exists i, j,\ H^*_{ij} = 0$. Similarly for ASymNMF [7], although the authors have shown that the sequence of objective function values during the iterative procedure of ASymNMF monotonically decreases, it is not enough to say that ASymNMF converges to a stationary point. Specifically, Proposition 7 shows that the algorithm updates in the positive orthants for both H and C $(H^k_{ij} > 0, C^k_{rs} > 0,\ \forall i, j, r, s)$ while Proposition 5 indicates that the stationary points contain zero elements in general $(\exists i, j, r, s,\ H^*_{ij} = 0$ or $C^*_{rs} = 0$ in the stationary point). Hence, no convergence properties of the sequences $\left\{H^k_{ij}\right\}$ and $\left\{C^k_{rs}\right\}$ can be established. Additionally, the denominators of the multiplicative updating equations of both SymNMF and ASymNMF are not well-defined when they approach zeros, which may cause numerical problems.

Through the procedure of APANMF, the dominant computational cost is a relatively cheap matrix multiplication involving the adjacency matrix A. Assuming that PMH and PMB respectively take k and l iterations in average to converge, the time complexity for updating H^t and B^t are $O(kN^2K)$ and

$O(lN^2K)$. Furthermore, if APANMF takes t iterations of PMH and PMB steps, then the overall time complexity of APANMF is $O(t(k+l)N^2K)$.

4.4 Initialization

Our flexible graph clustering formulation is not jointly convex with respect to H and B. Therefore, a good initial point is important to achieve high quality solutions. In this paper, we select the initialization points (H^0, B^0) as follows: First, we consider $\Psi(H, B)$ as an unconstrained optimization problem for B with randomly generated H. Setting $\nabla_B \Psi(H, B) = 0$ to obtain

$$\hat{B}^0 = (H^T H)^{-1} H^T A H (H^T H)^{-1}. \tag{24}$$

Then for $\Psi(H, B^0)$ we set $\nabla_H \Psi(H, B^0) = 0$ and get

$$\hat{H}^0 = AH\hat{B}^0(\hat{B}^0 H^T H \hat{B}^0)^{-1}. \tag{25}$$

We project $(\hat{H}^0, \hat{B}^0)$ to the non-negative orthant and choose the best (H^0, B^0) that gives the minimum objective function value as our initialization point.

4.5 Selection of α and β

We explore the stochastic nature of the proposed algorithm to determine α and β. A similar strategy has been adopted in [20]. We propose to estimate the robustness of a specific combination of α and β by measuring the differences and similarities of multiple realizations. For each realization, we compute a connectivity matrix $C = H^I B^I (H^I)^T$, where H^I and B^I are binary matrices recovered from H and B obtained from the algorithm 3. $H^I_{ij} = 1$ when $H_{ij} \geq \epsilon$ and $H^I_{ij} = 0$ when $H_{ij} < \epsilon$, where ϵ is a user-defined threshold controlling the number of memberships of the overlapping vertices [9]. Similarly, $B^I_{rs} = 1$ if $B_{rs} \geq 0.5$ (meaning the probability of cluster r interacting with cluster s is larger than 0.5), otherwise $B^I_{rs} = 0$. Then we can compute the consensus matrix $\bar{C}$ defined as the average connectivity matrix over many realizations. The entry $\bar{C}_{ij}$ of $\bar{C}$ ranges from 0 to 1 and reveals the probability that vertex i connects to vertex j.

After we obtain $\bar{C}$, we can estimate the entropy, which measures the stability of the common network structure. Assuming $\bar{C}_{ij}$ is independent of each other, we define the entropy score as

$$E_n = \frac{1}{N^2} \sum_{i,j} \left[\bar{C}_{ij} \log(\bar{C}_{ij}) + (1 - \bar{C}_{ij}) \log(1 - \bar{C}_{ij}) \right]. \tag{26}$$

For certain α and β, $E_n = 1$ means the network structure is totally unstable ($\bar{C}_{ij} = 0.5$), while $E_n = 0$ indicates that the edges in $\bar{C}$ are perfectly stable ($\bar{C}_{ij} = 1$ or $\bar{C}_{ij} = 0$). We demonstrate that the E_n score can help to select α and β in Section 5.4.

5 Experiments

In this section, in order to show the improved noise tolerance of our new graph clustering formulation and the effectiveness of our novel proximal algorithm APANMF for solving noisy graph clustering, we compare our APANMF with SymNMF_MU [6], SymNMF_NT [5], ASymNMF [7] and BNMF [8] on both synthetic benchmarks under different noise levels as well as real-world large-scale networks.

To demonstrate the robustness of our APANMF with respect to the noise, we explicitly tune the noise level of synthetic networks. Benchmarks for undirected and directed graphs are simulated by the LFR algorithm [12] with different mixing parameters μ to control the noise level. For block modeling benchmarks [16,13], the Maslov-Sneppen procedure [21] is applied to shuffle different fractions of edges to add in noise. Both the mixing parameter μ and the Maslov-Sneppen procedure have the same effect, which is to perturb the fraction of edges within the correct communities. For simplicity, we use μ to present the noise level for all synthetic networks (undirected and directed benchmarks [12] and block modeling benchmarks [16,13]). For example, $\mu = 0.1$ means that 10% of correct edges are perturbed to connect to the wrong vertices that do not follow the underlying interaction patterns. Because the perturbation of correct edges simulates the false positive and false negative edges in real-world networks, the robustness of our formulation with respect to potential noise in real-world networks can be verified by testing our APANMF on noisy benchmarks with different μ.

For the same noise level μ, we randomly generate 20 networks. For each random network, we implement each algorithm 10 times and choose the one with the best objective function value as the solution for this network. For all competing algorithms, we stop the algorithms when the objective function value does not decrease more than 0.1. The regularization parameters α and β of APANMF are determined by brute-force search in $S = \{(\alpha, \beta)|\alpha, \beta \in \{0, ..., 5\}\}$. For every network, we compute the entropy score based on (26) for every combination of α and β in S from 10 different realizations (initializations), and we choose the best α and β that yield the minimum entropy score. For λ of BNMF, we use the same procedure and set λ from 0 to 5 with an interval of 1. To quantitatively evaluate the performance of each algorithm for synthetic networks, we use the Normalized Mutual Information (NMI) [22] as the performance index for non-overlapping clustering comparison and the Generalized Normalized Mutual Information (GNMI) [23] for overlapping clustering comparison. The evaluation criteria for real-world datasets are introduced in the corresponding sections. All experiments are implemented on a MacBookPro laptop with an Intel i5 dual core processor and 8 GB memory.

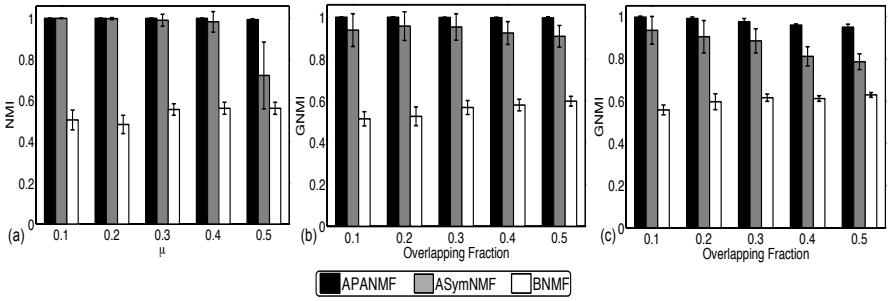

Fig. 3: Performance comparison for directed graph clustering: (a) NMI comparison (non-overlapping) with increasing mixing parameter μ. (b) GNMI comparison (overlapping) with increasing overlapping fraction values θ when $\mu = 0.1$. (c) GNMI comparison (overlapping) with increasing overlapping fraction values θ when $\mu = 0.3$.

5.1 Undirected Graph Clustering

To generate undirected graph benchmarks, we adopt the well-known LFR algorithm [12], in which the distributions of vertex degree and cluster size are both based on power laws with tunable exponents. In this paper, the benchmark networks are randomly generated based on the similar parameters adopted in [9]: The number of vertices $N = 400$; the average vertex degree is 20 and the cluster size ranges from $c_{min} = 40$ to $c_{max} = 80$. To validate the performance with different parameters, we further tune the mixing parameter (noise level) $\mu = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6\}$, which can be understood as the noise level indicating the portion of a given vertex's edges that connect to the vertices outside the community. This simulates potential noise at different levels in these randomly generated networks. When evaluating the performance for overlapping clustering, we set the overlapping fraction $\theta = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6\}$, which measures the fraction of vertices belonging to more than one clusters.

The comparison among all competing algorithms is shown in Fig. 2. The mean values and standard deviations achieved by all competing algorithms for each parameter setting are obtained from 20 randomly generated benchmarks. Fig. 2(a) illustrates the performance comparison on non-overlapping benchmarks with various mixing parameters μ. From the figure, we observe that the NMI bar from our APANMF is consistently higher than bars of all the other state-of-the-art algorithms, which indicates that APANMF identifies clusters that are closest to the ground truth. We also notice that our APANMF behaves marginally better than SymNMF_MU and SymNMF_NT, especially for large mixing parameters, which demonstrates that APANMF is more robust to noise than SymNMF_MU and SymNMF_NT since it explicitly enforces the sparsity of B and H.

Fig. 2(b) and (c) illustrate the performance for overlapping community detection under $\mu = 0.1$ and $\mu = 0.3$, respectively. With the increasing overlapping fraction values, the difficulty for graph clustering increases. We still find that the GNMI bar of our APANMF is consistently higher than bars of

the other competing algorithms with respect to different overlapping fraction values.

Furthermore, we test the statistical significance of our APANMF by comparing APANMF with SymNMF_MU and SymNMF_NT respectively as SymNMF_MU and SymNMF_NT are empirically the best-preforming algorithms in addition to our APNNMF. By two-sample t-test with unequal variances, we find that APANMF performs significantly better than SymNMF_MU and SymNMF_NT at the noise level $\mu = 0.6$ in the experiments illustrated in Fig. 2 (a) and (c) at the significant level of 0.05.

5.2 Directed Graph Clustering

We further generate directed graph benchmarks by LFR [12]. Similarly, to test the behavior of the competing algorithms, we simulate non-overlapping and overlapping directed benchmarks with different mixing parameters μ (noise levels). We set the number of vertices $N = 400$, the average vertex degree to 20 and the cluster size from $c_{min} = 40$ to $c_{max} = 80$. For non-overlapping directed graph benchmarks, we randomly generate benchmarks with the increasing mixing parameters (noise level): $\mu = \{0.1, 0.2, 0.3, 0.4, 0.5\}$. While for overlapping directed graph benchmarks, we simulate benchmarks with the increasing overlapping fraction values $\theta = \{0.1, 0.2, 0.3, 0.4, 0.5\}$. We compare our APANMF with all the other methods except SymNMF_MU and SymNMF_NT as SymNMF_MU and SymNMF_NT can not handle directed graphs.

Fig. 3 shows the comparison results for non-overlapping and overlapping clustering for directed graphs. For non-overlapping clustering comparison shown in Fig. 3(a), APANMF and ASymNMF are competitive when the mixing parameter μ is small. However, when it reaches $\mu = 0.5$, APANMF performs significantly better than ASymNMF, which further validates that with high noise level, the sparsity regularization in APANMF can help obtain better results. For overlapping clustering comparison shown in Fig. 3(b) and (c), with the increasing overlapping fraction values at fixed $\mu = 0.1$ and $\mu = 0.3$ respectively, the bars of GNMI values obtained by APANMF are consistently higher than other two competing algorithms. Additionally, the GNMI values of APANMF are the most stable one with the smallest standard deviation. Therefore, Figs. 3(b) and (c) demonstrate that APANMF is also robust to the overlapping fraction. In summary, obviously our APANMF outperforms ASymNMF and BNMF for both non-overlapping and overlapping clustering of directed graphs.

5.3 Block Modeling

Our APANMF can also solve block modeling clustering problems. We generate synthetic networks as similarly done in [16,13] with known ground truth block structures. The generated benchmark networks have block structures

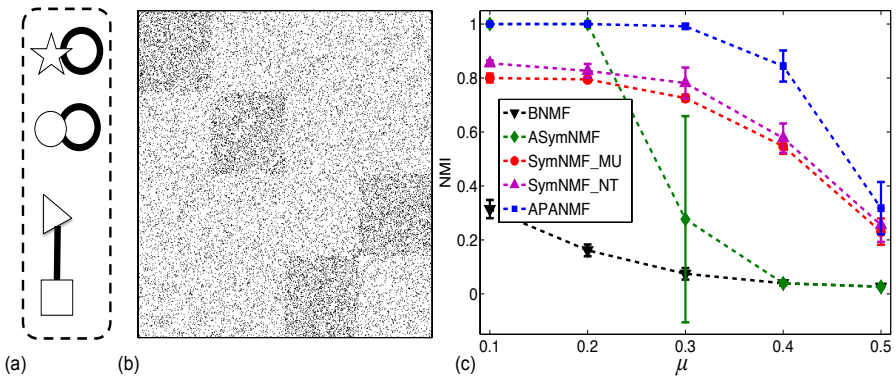

Fig. 4: (a) Underlying blockmodel structure of synthetic blockmodel benchmarks. (b) Example of a random network with $\mu = 0.4$. (c) NMI comparison with the increasing noise level for all the competing algorithms.

with two densely connected clusters and two clusters with only edges across each other as shown in Fig. 4(a). We first simulate the noise-free networks with the size of each block clusters set at 100. To vary the difficulty of the block modeling clustering problem, we instill the noise to the network topology with different levels, which can be controlled by permuting the percentage of correct edges based on the Maslov-Sneppen algorithm [21]. In the Maslov-Sneppen algorithm, two unconnected edges are randomly drawn and then mutually rewired. The noise level μ controls the percentage of correct edges to be permuted. Fig. 4(b) provides an example with 40% edges being permuted ($\mu = 0.4$).

Fig. 4(c) illustrates the comparison in terms of NMI. From the figure, we observe that the NMI curve of our APANMF is consistently on top of all the other competing algorithms at all noise levels. In addition, we discover that when the noise level is low, both APANMF and ASymNMF have competitive performance, better than the other two algorithms, since the introduction of the image graph B in both methods. However, with the increasing noise level, ASymNMF fails to detect the block structures, which consolidates that our APANMF is more robust to noise with additional sparsity regularization. For SymNMF_MU and SymNMF_NT, as they are designed to identify densely connected clusters, the bipartite-like clusters in Fig. 4(a) can not be detected even when there is no noise. Additionally, SymNMF_NT performs marginally better than SymNMF_MU because SymNMF_NT converges to a stationary point. For BNMF, the approximation of M^{ideal} may not capture the latent structure of the graph, which influences its performance.

5.4 Effect and determination of α and β

The regularization coefficients α and β control the sparsity of H and B in APANMF. The larger α and β are, the sparser H and B become. To discover

the relationships among the selection of α and β, clustering accuracy and entropy scores under a high noise level, we implement the following experiment. We randomly generate a synthetic network with the noise level of $\mu = 0.4$. The underlying block structure is the same as illustrated in Fig. 4(a). We select the (α, β) pair from $S = \{(\alpha, \beta) | \alpha \in \{0, 1, 2, 3, 4, 5\}$ and $\beta \in \{0, 1, 2, 3, 4, 5\}\}$. To demonstrate the effectiveness of the regularization terms, for each initialization we implement our algorithm through all (α, β) pairs in set S. We apply 10 different initializations and compute the average NMI value and entropy score of each (α, β) pair. Fig. 5(a) displays the surface of NMI values for every (α, β) pair and Fig. 5(b) illustrates the surface of entropy scores for every (α, β) pair. Fig. 5(a) shows that better NMI values can be achieved by appropriately selecting (α, β), which further indicates the necessity of using regularization terms to obtain robust results for noisy networks. Additionally, we discover that the best average NMI values and the minimum entropy scores are attained at the same point $(\alpha, \beta) = (5, 3)$, which demonstrates that using entropy scores can help us choose appropriate α and β.

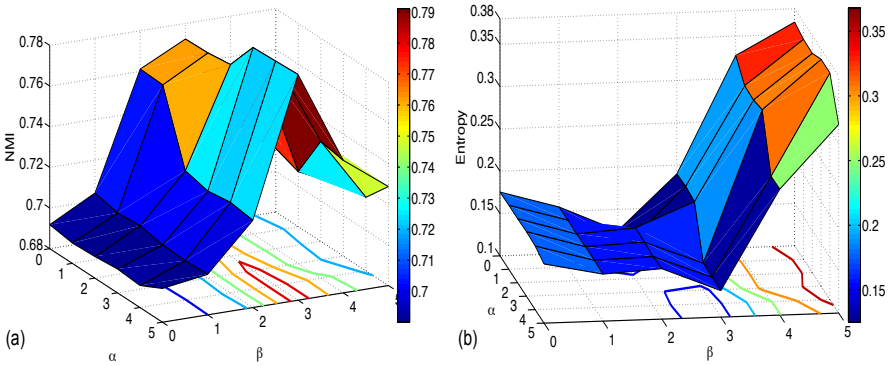

Fig. 5: (a) Surface of average NMI values for every (α, β) pair. (b) Surface of entropy scores for every (α, β) pair.

5.5 Facebook Ego Network

The Facebook ego network is obtained from the SNAP library [24]. The Facebook ego network combines 10 ego networks with 4,039 vertices as Facebook users and 88,234 edges denoting virtual friendship. This combined ego network has manually labeled ground truth from Facebook circles. Our task is to detect the overlapping communities within the ego network. Because we have the ground truth, we can evaluate the performance of all competing algorithms. The measure we applied is the geometric mean of two other measures, which are the cluster-wise sensitivity (Sn) and the cluster-wise positive predictive value (PPV) [25]. Given r predicted and s reference communities, let t_{ij} denote the number of vertices that exist in both predicted community i and

reference community j, and w_j represent the number of vertices in reference community j. Then Sn and PPV can be defined as

$$Sn = \frac{\sum_{j=1}^{s} \underset{i=1,\ldots,r}{max}\ t_{ij}}{\sum_{j=1}^{s} w_j}\ ,\ PPV = \frac{\sum_{i=1}^{r} \underset{j=1,\ldots,s}{max}\ t_{ij}}{\sum_{i=1}^{r}\sum_{j=1}^{s} t_{ij}}. \tag{27}$$

We use their geometric mean as our "accuracy" index to balance these two measures ($Acc = \sqrt{Sn \times PPV}$) [25].

We set the number of potential communities $K = 200$ for all the algorithms and choose $\alpha = 5$ and $\beta = 2$ for APANMF and $\lambda = 1$ for BNMF based on the entropy score (26). The performance comparison for this Facebook ego network is provided in Table 1, from which it is clear that our APANMF obtains the best Sn, PPV, and Acc scores. The results further demonstrate that APANMF is the best graph clustering method for this application. SymNMF_NT fails to implement due to the memory limitation. Hence, SymNMF_NT is not included in Table 1.

Table 1: Comparison on Facebook ego network.

	APANMF	SymNMF_MU	ASymNMF	BNMF
Sn	**0.4243**	0.3905	0.3978	0.2078
PPV	**0.5731**	0.5614	0.5070	0.2843
Acc	**0.4931**	0.4682	0.4491	0.2431

5.6 Human Protein-Protein Interaction Netwrok

To further illustrate the practical usage of our flexible method in computational biology, we apply all the competing algorithms and compare their performances on a human protein-protein interaction (PPI) network extracted from the PIPs dataset (HsaPIPs) [14]. This HsaPIPs network has 5,445 proteins and 74,686 edges denoting whether two corresponding proteins bind with each other. For this biological network, we do not have the clustering ground truth, which is often the case for most of the real-world network datasets. As typically done in computational biology, we evaluate the performance based on manual curations of genes and/or proteins in this network, for example, based on Gene Ontology (GO) terms [26]. GO terms annotate groups of genes representing certain gene product properties in cells. GO term enrichment analysis [27] can help interpret the corresponding cellular functions for the proteins in detected clusters by statistically detecting whether they correspond to a specific GO term. Assuming a detected cluster has n proteins with m proteins annotated to a GO term and the whole network has N proteins with M proteins annotated with the same GO term. Then the p-value of the identified cluster with respect to the enrichment of proteins within that GO term can be calculated

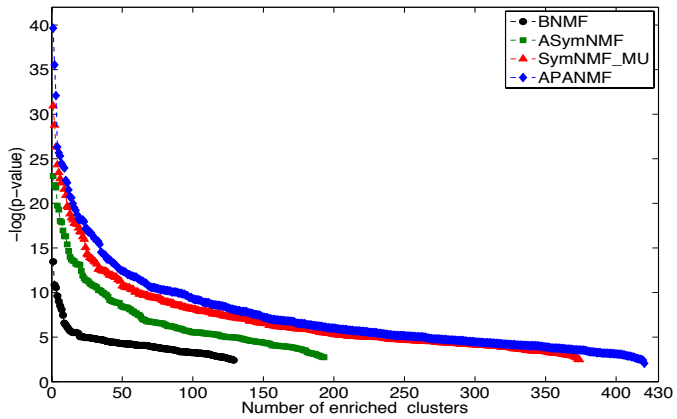

Fig. 6: GO enrichment comparison for all the competing algorithms.

as [27]

$$\text{p-value} = \sum_{i=m}^{n} \frac{\binom{m}{i}\binom{N-M}{N-i}}{\binom{N}{n}}. \tag{28}$$

In this implementation, we set $K = 600$ for all the competing algorithms and choose $\alpha = 4$ and $\beta = 2$ for APANMF and $\lambda = 1$ for BNMF based on the entropy score computation (26). SymNMF_NT again fails to run due to its memory issue. For performance comparison, a GO term is considered enriched when there is a detected cluster significantly enriched with this GO term with the corresponding p-value less than $1e{-}3$. For each cluster, we choose the lowest p-value of all its enriched GO terms as the corresponding p-value of this cluster. Fig. 6 illustrates the performance comparison in terms of the negative logarithms of the p-values for every identified clusters in the descending order. We find that the curve of APANMF is the longest one, which indicates that APANMF detects the largest number of biologically meaningful clusters (420) with the corresponding p-values lower than $1e{-}3$. Additionally, we notice that the curve of APANMF is on top of all the other curves, which implies that the significant level of the clusters identified by APANMF is higher than the others. Furthermore, we count the total number of the enriched GO terms obtained by each competing algorithm. We find that 1637 GO terms are enriched by the clusters detected by APANMF. For SymNMF_MU, ASymNMF, and BNMF, 1390, 809, and 283 GO terms are significantly enriched, respectively. Obviously, APANMF covers the largest number of enriched GO terms which indicates that APANMF unearths richer biological information. It is not surprising because many researchers [13,16] have discovered that PPI networks have block modeling structures and our APANMF is more powerful for discovering the block modeling structures of noisy graphs.

6 Conclusions

In this paper, for clustering noisy networks, we propose a flexible NMF-based formulation with the explicit sparsity regularization of all factorized components. Our new framework is noise tolerant and can solve graph clustering with different settings such as both undirected graph community detection and directed graph clustering with either overlapping or non-overlapping clustering structures, and more general block modeling clustering problems as well. Furthermore, we propose an alternating proximal method APANMF to solve our new optimization problem with the convergence guarantee. The results on synthetic benchmarks and real-world networks demonstrate that our method outperforms other NMF-based state-of-the-art graph clustering algorithms. The proposed APANMF has the potential to derive useful knowledge in diverse applications including social and biological network analysis.

7 Appendices

In this section, we provide the proofs for the lemma and propositions in the paper.

7.1 Proof for Proposition 1

Proof To solve the constrained optimization problem in (13), we can first write out the following Karush-Kuhn-Tucker (K.K.T.) conditions:

$$\begin{aligned} \frac{\partial \phi(H)}{\partial H} - \Delta + \Gamma \ni \mathbf{0} \qquad & (29.a), \\ H_{ij} \geq 0 \qquad & (29.b), \\ H_{ij} \leq 1 \qquad & (29.c), \\ \Delta \otimes H = \mathbf{0} \qquad & (29.d), \\ \Gamma \otimes (H - \mathbf{1}) = \mathbf{0} \qquad & (29.e), \end{aligned} \tag{29}$$

where $\otimes$ is an element-wise multiplication operator and the first equation is derived from the sub-gradient of the Lagrange relaxation of the original problem

$$\begin{aligned} \mathbf{0} \in & \frac{\partial L(H, \Delta, \Gamma)}{\partial H} \\ = & \frac{\partial \big[\phi(H) - trace(\Delta^T H) + trace(\Gamma^T (H - \mathbf{1}))\big]}{\partial H} \\ = & \frac{\partial \phi(H)}{\partial H} - \Delta + \Gamma. \end{aligned} \tag{30}$$

Multiplying $H \otimes (\mathbf{1} - H)$ on both sides of (29.a) and using (29.d) and (29.e), we obtain the following important equation:

$$\begin{aligned} \Big(\frac{\partial \phi(H)}{\partial H} - \Delta + \Gamma\Big) \otimes H \otimes (H - \mathbf{1}) \ni \mathbf{0} \\ \frac{\partial \phi(H)}{\partial H} \otimes H \otimes (\mathbf{1} - H) \ni \mathbf{0}. \end{aligned} \tag{31}$$

With that, we now have

$$\begin{aligned} \frac{\partial \phi(H)}{\partial H} \otimes H \otimes (\mathbf{1} - H) \ni \mathbf{0} \qquad & (32.a), \\ H_{ij} \geq 0 \qquad & (32.b), \\ H_{ij} \leq 1 \qquad & (32.c). \end{aligned} \tag{32}$$

Obviously, because $\mathbf{prox}_H(\bar{H})$ is the solution of $\frac{\partial \phi(H)}{\partial H} \ni \mathbf{0}$, the element-wise solution to the above system is

$$H_{ij}^k = \mathbb{P}(\mathbf{prox}_H(\bar{H})_{ij}), \tag{33}$$

which proves the proposition.

7.2 Proof for Proposition 2

Proof We can prove the fact that $\left\{F(H^k)\right\}_{k\geq 0}$ is non-increasing and convergent due to the following inequalities:

$$F(H^k) \leq G_k(H^k) \leq G_k(H^{k-1}) = F(H^{k-1}). \tag{34}$$

The first inequality comes from the fact that $G_k(H)$ is the upper bound of $F(H)$ (10). We have the second inequality as the proximal method solves (11). The last equality can be obtained by substituting H with H^{k-1} in (9). Because $\left\{F(H^k)\right\}_{k\geq 0}$ is bounded, we define F^* as its limit. Based on (34) and $S_k(H^k) = G_k(H^k) - F(H^k)$, we have:

$$S_k(H^k) \leq F(H^{k-1}) - F(H^k). \tag{35}$$

By adding all the terms over k, we have

$$\sum_k S_k(H^k) \leq F(H^0) - F^*, \tag{36}$$

which is also bounded. Therefore, $\left\{S_k(H^k)\right\}_{k\geq 0}$ necessarily converges to zero.

Furthermore, we notice that $S_k(H)$ is differentiable and Lipschitz continuous because

$$\begin{aligned} S_k(H) =& G_k(H) - F(H) \\ =& P(H^{k-1}) - P(H) + \frac{L_k}{2}\left\|H - H^{k-1}\right\|_F^2 \\ &+ < \nabla P(H^{k-1}), (H - H^{k-1}) > . \end{aligned} \tag{37}$$

Therefore, for any H^k and H', $S_k(H)$ satisfies the classical lemma (lemma 1.2.3 in [28]), which yields

$$S_k(H') \leq S_k(H^k) - \frac{1}{2L_k} \left\| \nabla S_k(H^k) \right\|_F^2, \tag{38}$$

where we define $H' = H^k - \frac{1}{L_k} \nabla S_k(H^k)$. From (38), we derive

$$\begin{aligned} \left\| \nabla S_k(H^k) \right\|_F^2 &\leq 2L_k (S_k(H^k) - S_k(H')) \\ &\leq 2L_k S_k(H^k) \overset{k \mapsto +\infty}{\mapsto} 0, \end{aligned} \tag{39}$$

where we take the fact that $S_k(H') \geq 0$ because (34) and $\left\{ S_k(H^k) \right\}_{k \geq 0}$ converges to zero.

Now, we compute the directional derivative $\nabla_{H-H^k} F(H^k)$ of $F(\cdot)$ at H^k in the direction $H - H^k$,

$$\begin{aligned} \nabla_{H-H^k} F(H^k) =& \nabla_{H-H^k} G_k(H^k) \\ & - < \nabla S_k(H^k), H - H^k > . \end{aligned} \tag{40}$$

Here we make a mild assumption that, for all H and H^k in the constraint set, the directional derivative $\nabla_{H-H^k} F(H^k)$ always exists. A similar assumption has been made for proving the convergence of a constrained optimization problem [18]. Note that H^k minimizes G_k on $\{H | 0 \leq H \leq 1\}$ and therefore $\nabla_{H-H^k} G_k(H^k) \geq 0$ [29]. With these,

$$\nabla_{H-H^k} F(H^k) \geq - \left\| \nabla S_k(H^k) \right\|_F \left\| H - H^k \right\|_F, \tag{41}$$

based on Cauchy-Schwarz inequality. Then,

$$\lim_{k \mapsto +\infty} \frac{\nabla_{H-H^k} F(H^k)}{\left\| H - H^k \right\|_F} \quad \geq \quad \lim_{k \mapsto +\infty} - \left\| \nabla S_k(H^k) \right\|_F = 0, \tag{42}$$

which further indicates that H^k is the stationary point of $F(H)$ when k approaches $+\infty$ based on the definition of an asymptotic stationary point proposed in [18].

7.3 Lemma 1 and Its Proof

Lemma 1 *$\Phi(B) = \left\| A - HBH^T \right\|_F^2$ is Lipschitz continuous and its Lipschitz constant Π is equal to the square of the largest eigenvalue of $H^T H$ ($L_B = \delta_{max}^2(H^T H)$).*

Proof Given two matrices X and Y, we have

$$\begin{aligned} & \left\| \nabla \Phi(X) - \nabla \Phi(Y) \right\|_F^2 \\ &= \left\| H^T H (X - Y) H^T H \right\|_{F2}^2 \\ &= trace(H^T H (X - Y)^T H^T H H^T H (X - Y) H^T H), \end{aligned} \tag{43}$$

where H^TH is a positive semi-definite symmetric matrix. Hence, we can write $H^TH = U\Sigma U^T$ by SVD (singular value decomposition) with $UU^T = I_n$ and $U^TU = I_k$. By straightforward algebraic manipulations, (43) is equivalent to

$$\begin{aligned}
&\|\nabla\Phi(X) - \nabla\Phi(Y)\|_F^2 \\
&= trace(U\Sigma U^T(X-Y)^TU\Sigma U^TU\Sigma U^T(X-Y)U\Sigma U^T) \\
&= trace(U^T(X-Y)^TU\Sigma^2U^T(X-Y)U\Sigma^2) \\
&\leq \delta_{max}^4 trace(U^T(X-Y)^TUU^T(X-Y)U) \\
&= \delta_{max}^4 \|X-Y\|_F^2,
\end{aligned} \tag{44}$$

where δ_{max} is the largest eigenvalue of H^TH. From (44), we have the following inequality:

$$\|\nabla\Phi(X) - \nabla\Phi(Y)\|_F \leq \delta_{max}^2 \|X-Y\|_F. \tag{45}$$

Therefore, $\Phi(B)$ is Lipschitz continuous and the Lipschitz constant L_B is equal to the square of the largest eigenvalue of H^TH.

7.4 Proof for Proposition 3

Proof At the tth iteration, we have $\Omega(H^t, B^t)$. Based on Proposition 2 (34) for a fixed B^t, we get

$$\Omega(H^{t+1}, B^t) \leq \Omega(H^t, B^t). \tag{46}$$

Furthermore, based on (42), we have

$$\frac{\nabla_{H-H^{t+1}}F(H^{t+1})}{\|H-H^{t+1}\|_F} \geq 0 \Leftrightarrow \frac{\nabla_{H-H^{t+1}}\Omega(H^{t+1}, B^t)}{\|H-H^{t+1}\|_F} \geq 0, \tag{47}$$

where H^{t+1} is an asymptotic stationary point and we define $Q^{t+1,t} = [H^{t+1}; B^t]$.

Similarly, the proof in [30] demonstrates that for a fixed H^{t+1} we can obtain B^{t+1} satisfying

$$\Omega(H^{t+1}, B^{t+1}) \leq \Omega(H^{t+1}, B^t) \tag{48}$$

as $\Omega(H^{t+1}, B^t)$ is convex with respect to B^t for the given H^{t+1}. Similar to (47), for the asymptotic stationary point B^{t+1}, we have

$$\frac{\nabla_{B-B^{t+1}}E(B^{t+1})}{\|B-B^{t+1}\|_F} \geq 0 \Leftrightarrow \frac{\nabla_{B-B^{t+1}}\Omega(H^{t+1}, B^{t+1})}{\|B-B^{t+1}\|_F} \geq 0. \tag{49}$$

From (46) and (48), we know

$$\Omega(H^{t+1}, B^{t+1}) \leq \Omega(H^{t+1}, B^t) \leq \Omega(H^t, B^t) \tag{50}$$

Obviously, the sequence $\{(H^t, B^t)\}$ is non-increasing. $\{(H^t, B^t)\}$ is also bounded because of $0 \leq H, B \leq 1$. Therefore, we further assume that $(\tilde{H}, \tilde{B})$ is a limit

point of the sequence. Based on (47) and (49), for any (H, B) in the constraint set, when $t \mapsto +\infty$ we obtain

$$\begin{aligned} \lim_{t\mapsto+\infty} \frac{\nabla_{H-H^{t+1}}\Omega(H^{t+1}, B^t)}{\|H - H^{t+1}\|_F} &= \frac{\nabla_{H-\tilde{H}}\Omega(\tilde{H}, \tilde{B})}{\left\|H - \tilde{H}\right\|_F} \geq 0, \\ \lim_{t\mapsto+\infty} \frac{\nabla_{B-B^{t+1}}\Omega(H^{t+1}, B^{t+1})}{\|B - B^{t+1}\|_F} &= \frac{\nabla_{B-\tilde{B}}\Omega(\tilde{H}, \tilde{B})}{\left\|B - \tilde{B}\right\|_F} \geq 0, \end{aligned} \tag{51}$$

which implies that $(\tilde{H}, \tilde{B})$ is an asymptotic stationary point of $\Omega(H, B)$ [18, 31].

7.5 Proposition 4 and Its Proof

Proposition 4 *If $A \neq HH^T$, then any stationary point of the optimization problem* (1) *is on the boundary of its constraint set $\{H|H \geq 0\}$.*

Proof By definition, a stationary point of the optimization problem (1) should satisfy the Karush-Kuhn-Tucker (KKT) optimality condition [5]:

$$\big((A - HH^T)H\big)_{ij} H_{ij} = 0. \tag{52}$$

With the assumption $A \neq HH^T$ in general, we find that the stationary points of (1) necessarily contain zero elements ($\exists i, j, H_{ij} = 0$) in H. This implies that the stationary points of (1) are on the boundary of $\{H|H \geq 0\}$.

7.6 Proposition 5 and Its Proof

Proposition 5 *If $A \neq HCH^T$, then a stationary point of the optimization* (2) *is on the boundary of its constraint set $\{(H, C)|H \geq 0, C \geq 0\}$.*

Proof Based on [7], a stationary point of the optimization problem (2) should satisfy the following Karush-Kuhn-Tucker (KKT) optimality condition:

$$\begin{cases} \big((HCH^T - A)HC^T + (HC^TH^T - A^T)HC\big)_{ij} H_{ij} = 0; \\ \big(H^T(HCH^T - A)H\big)_{rs} C_{rs} = 0. \end{cases} \tag{53}$$

Because in general, $A \neq HCH^T$ and $A^T \neq HC^TH^T$, it requires that there exists $H_{ij} = 0$ or $C_{rs} = 0$ in the stationary point to satisfy the KKT condition, which implies that the stationary points of (2) are on the boundary of $\{(H, C)|H \geq 0, C \geq 0\}$.

7.7 Proposition 6 and Its Proof

Proposition 6 *If A has neither zero column nor zero row, and the initialization point of SymNMF $H_{ij}^0 > 0, \forall i, j$, then*

$$H_{ij}^k > 0, \forall i, j, \forall k \geq 0. \tag{54}$$

Proof From [5], we know the updating rule of SymNMF is

$$H_{ij}^{k+1} \leftarrow H_{ij}^k (\frac{1}{2} + \frac{(AH^k)_{ij}}{2(H^k(H^k)^T H^k)_{ij}}). \tag{55}$$

When $k = 0$, the equation (54) holds by the assumption. By induction, if (54) is correct at k, then it is correct at $k+1$ too. The nominator and denominator in (55) are both strictly positive under the assumption that A has neither zero column nor zero row. Therefore, $H_{ij}^{k+1} > 0$.

7.8 Proposition 7 and Its Proof

Proposition 7 *If A has neither zero column nor zero row, and the initialization point of ASymNMF $H_{ij}^0 > 0$ and $C_{rs}^0 > 0$, $\forall i, j, r, s$, then*

$$H_{ij}^k > 0, C_{rs}^k > 0, \ \forall i, j, r, s, \ \forall k \geq 0. \tag{56}$$

Proof From [7], we know the updating rule of ASymNMF is

$$\begin{aligned} &H_{ij}^{k+1} \leftarrow H_{ij}^k \cdot \\ &\Big(\frac{(A^T H^k C^k + A H^k (C^k)^T)_{ij}}{\big(H^k (C^k (H^k)^T H^k (C^k)^T + (C^k)^T (H^k)^T H^k C^k)\big)_{ij}}\Big)^{\frac{1}{4}}; \\ &C_{rs}^{k+1} \leftarrow C_{rs}^k \frac{((H^k)^T A H^k)_{rs}}{((H^k)^T H^k C^k (H^k)^T H^k)_{rs}}. \end{aligned} \tag{57}$$

When $k = 0$, the equation (56) holds by the assumption. By induction, if (56) is correct at k, then it is correct at $k+1$. Both the nominator and denominator in (57) are strictly positive under the assumption that A has neither zero column nor zero row. Therefore, (56) holds at $k+1$, and the proof is complete.

References

1. D. Lee, H. Seung, Nature **401**(788-791) (1999)
2. M. Sun, H.V. hamme, IEEE Transactions on Signal Processing **60**(7), 3876 (2012)
3. M. Wang, S. Yan, Z.J. Zha, H. Zhang, T.S. Chua, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2012), pp. 2464–2471
4. D. Cai, X. He, J. Han, T.S. Huang, IEEE Transactions on Pattern Analysis and Machine Intelligence **33**(8), 1548 (2011)
5. D. Kuang, C. Ding, H. Park, in *Proceedings of the SIAM International Conference on Data Mining* (2012)

6. C. Ding, X. He, H.D. Simon, in *SIAM International Conference on Data Mining* (2005)
7. F. Wang, T. Li, X. Wang, et al, Data Min Knowl Disc **22**(3), 493 (2011)
8. J. Chan, W. Liu, A. Kan, et al, in *ACM International Conference on Information and Knowledge Management* (2013)
9. N.P. Nguyen, M.T. Thai, in *The 31st Military Communication Conference* (2012)
10. Y. Zhang, D. Yeung, in *ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (2012)
11. C.J. Lin, IEEE Trans. on Neural Networks **18**(6), 1589 (2007)
12. A. Lancichinetti, S. Fortunato, Phys Rev E **80**(016118) (2009)
13. Y. Wang, X. Qian, BMC Bioinformatics **14**(Suppl 2), S23 (2013)
14. M. McDowall, M. Scott, G. Barton, Nucleic Acids Res **37**, 651 (2009)
15. P. Holland, S. Leinhardt, Social networks **5**(2), 109 (1983)
16. Y. Wang, X. Qian, Bioinformatics **30**(1), 81 (2014)
17. N. Fan, P. Pardalos, Journal of Combinatorial Optimization **23**(2), 224 (2010)
18. J. Mairal, in *International Conference on Machine Learning (ICML)* (2013)
19. A. Beck, M. Teboulle, SIAM J. on Imaging Sciences **2**(1), 183 (2009)
20. S. Zhang, J. Zhao, X. Zhang, Phys Rev E **85**(056110) (2012)
21. S. Maslov, K. Sneppen, Science **296**, 910 (2002)
22. L. Ana, A.K. Jain, in *CVPR* (2003)
23. A. Lancichinetti, S. Fortunato, K. Jnos, New Journal of Physics **11**(3), 033015 (2009)
24. J. Leskovec, A. Krevl. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data (2014)
25. X. Li, M. Wu, C. Kwoh, S. Ng, BMC Genomics **11**(Suppl 1), S3 (2010)
26. M. Ashburner, C. Ball, J. Blake, et al, Nat Genet **25**(1), 25 (2000)
27. Y. Wang, X. Qian, BMC Systems Biology **8**(suppl 1), S9 (2014)
28. Y. Nesterov, *Introductory lectures on convex optimization* (Kluwer Academic Publishers, 2004)
29. J. Borwein, A. Lewis, *Convex analysis and nonlin- ear optimization: theory and examples* (Springer, 2006)
30. F. Bash, R. Jenatton, J.M.G. Obozinski, Foundations and Trends in Machine Learning **4**(1), 1 (2012)
31. L. Grippo, M. Sciandrone, Operations Research Letters **26**(3), 127 (2000)