Autonomous Semantic Community Detection via Adaptively Weighted Low-rank Approximation

LIANG YANG, YUEXUE WANG, and JUNHUA GU, Hebei University of Technology, China
XIAOCHUN CAO, Institute of Information Engineering, Chinese Academy of Sciences, China
XIAO WANG, Beijing University of Posts and Telecommunications, China
DI JIN, Tianjin University, China
GUIGUANG DING, Tsinghua University, China
JUNGONG HAN, University of Warwick, UK
WEIXIONG ZHANG, Washington University, USA

Identification of semantic community structures is important for understanding the interactions and sentiments of different groups of people and predicting the social emotion. A robust community detection method needs to autonomously determine the number of communities and community structure for a given network. Nonnegative matrix factorization (NMF), a component decomposition approach for latent sentiment discovery, has been extensively used for community detection. However, the existing NMF-based methods require the number of communities to be determined *a priori*, limiting their applicability in practice of affective computing. Here, we develop a novel NMF-based method to autonomously determine the number of semantic communities and community structure simultaneously. In our method, we use an initial number of semantic communities, larger than the actual number, in the NMF formulation, and then suppress some of the communities by introducing an adaptively weighted group-sparse low-rank regularization to derive the target number of communities and at the same time the corresponding community structure. Our method not only maintains the efficiency without increasing the complexity compared to the original NMF method but also can be straightforwardly extended to handle the non-network data. We thoroughly examine the new method, showing its superior performance over several competing methods on synthetic and large real-world social networks.

CCS Concepts: • **Information systems → Social networks**;

This work was supported in part by the National Key R&D Program of China (Grants No. 2017YFC0820106 and No. 2016YFB0800403), National Natural Science Foundation of China (Grants No. 61972442, No. U1636214, and No. 61772361) and Beijing Natural Science Foundation (Grant No. 4172068).

Authors' addresses: L. Yang, Y. Wang, and J. Gu (corresponding author), School of Artificial Intelligence, Hebei Province Key Laboratory of Big Data Calculation, Hebei University of Technology, 5340 Xiping Road, Tianjin, 300130, China; emails: yangliang@vip.qq.com, yuexuewang@yeah.net, jhgu@hebut.com; X. Cao, State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, 89A Minzhuang Road, Beijing, 100093, China; email: caoxiaochun@iie.ac.cn; X. Wang (corresponding author), School of Computer Science, Beijing University of Posts and Telecommunications, 10 Xitucheng Road, Beijing, 100876, China; email: xiaowang@bupt.edu.cn; D. Jin, College of Intelligence and Computing, Tianjin University, 135 Yaguan Road, Tianjin, 300350, China; email: jindi@tju.edu.cn; G. Ding, School of Software, Tsinghua University, Beijing, China; email: dinggg@tsinghua.edu.cn; J. Han, University of Warwick, Coventry, UK; email: jungong.han@warwick.ac.uk; W. Zhang, Department of Computer Science and Engineering, Washington University, St. Louis, MO 63130, USA; email: weixiong.zhang@wustl.edu.

1551-6857/2019/11-ART98 $15.00
https://doi.org/10.1145/3355393

Additional Key Words and Phrases: Community detection, nonnegative matrix factorization, low-rank approximation, rank determination

ACM Reference format:
Liang Yang, Yuexue Wang, Junhua Gu, Xiaochun Cao, Xiao Wang, Di Jin, Guiguang Ding, Jungong Han, and Weixiong Zhang. 2019. Autonomous Semantic Community Detection via Adaptively Weighted Low-rank Approximation. *ACM Trans. Multimedia Comput. Commun. Appl.* 15, 3s, Article 98 (November 2019), 22 pages.
https://doi.org/10.1145/3355393

1 INTRODUCTION

The network has been evolving into a major means of communication in our society, and as it is becoming more and more social, a massive amount of content is produced and widely shared online. This also creates new opportunities for understanding the interactions and sentiments of different groups of people and predicting the social emotion, at the core of which of is the analysis of complex networks [20, 31, 35, 36]. As a key feature of networks, semantic communities or modules are subgraphs where vertices within a subgraph are more densely connected than vertices across subgraphs. Many semantic community detection algorithms [11, 12, 22] and their semi-supervised versions [46, 48, 49] have been proposed. They can be grouped into two categories, i.e., the ones that need to predetermine the number of semantic communities [15, 29, 43, 47, 50] and the methods that are able to determine the number of semantic communities and detect the semantic community structure simultaneously [5, 10, 13]. Although some of these methods can produce good results on determining the number of semantic communities, most of them cannot produce accurate community structures at the same time, and it does not seem to be straightforward to seamlessly integrate them with other accurate semantic community detection approaches when given the number of communities. This seriously hinders their applicability in practice of affective computing.

Nonnegative matrix factorization (NMF) [17, 44], which aims to decompose a system into (hidden) components or patterns embedded in the data, has been widely used in affective computing, e.g., predicting speech, image and video emotion [32, 38], image processing [17], audio processing [37], and text mining [42]. NMF has also been successfully applied to semantic community detection in complex networks [30, 43]. In semantic community detection, the number of columns of the final factorized matrix corresponds to the number of communities, and hence this number is also critical for parsing the result. Furthermore, to be effective, the NMF method often needs to know the number of communities in advance. However, this number is generally unknown or difficult to determine in practice. A related idea is to apply a low-rank approximation method to directly derive the number of communities. In essence, NMF seeks a low-rank approximation to the original data with *a known rank*, i.e., the number of components or communities. This idea has been pursued lately to directly derive the underlying low-rank representation for given data without knowing the rank, with Robust PCA (RPCA) being one of the most well known [6]. Unfortunately, the results from RPCA are often not low rank. It imposes some restrictions on given data, i.e., the noise is sparse and the rank of underlying data is sufficiently low, which cannot be met in reality, defeating its primary design objective and applicability to finding the number of components autonomously.

In this article, we consider the problem of determining the number of semantic communities and identifying the corresponding semantic communities at the same time. We develop such a method under the NMF paradigm. Our method, namely, NMF-AWL, stems from two ideas. We first introduce a weighted group-sparse low-rank regularization to help decompose a given network into components. The second is an innovative adaptive optimization scheme to learn the right rank or

number of components of the given data. To the best of our knowledge, this is the first NMF-based approach to directly determining the underlying low-rank representation for networks without knowing the rank. Compared with previous low-rank approximation methods that constrain the underlying data with uniformly weighted low-rank regularization, this is the first time to introduce an adaptively weighted low-rank regularization to NMF for the detection of communities. It is worth noting that NMF-AWL is readily extensible to other NMF-based methods and applications on non-network data.

The article is organized as follows. We briefly review the previous work on community detection with NMF and low-rank matrix approximation in Section 2. We then propose our method of Nonnegative Matrix Faction with Adaptively Weighted Low-rank approximation (NMF-AWL) in Section 3. We consider optimization of the method, provide insights to its components and analyze complexity in Section 4. Extensive experiments on synthetic and real datasets are presented in Section 5. We conclude in Section 6 by highlighting the main contributions and discussing future directions.

2 PRIOR WORK

Here, we review the previous work on low-rank matrix approximation, which forms the basis of the current work. We then discuss how NMF is used to solve the community detection and why original NMF-based methods need pre-determine the rank of the factorized matrices.

2.1 Low-rank Matrix Approximation

Since most real data are corrupted with noise, how to remove noise and reveal the structure of the data is a critical problem in many areas, such as signal processing, computer vision and pattern recognition. Low-rank approximation is a class of widely used methods, which can be divided into two groups to be discussed next, to find the underlying structure of a given dataset. They operate under the assumptions that the underlying structure of the data lies on a low dimensional subspace and the high dimension of the observed data is often due to noises.

2.1.1 Approximation with Known Rank. A given dataset over n entities (vertices), which are characterized by m features, can be represented by a matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$, where a column specifies the features of an entity. We may approximate $\mathbf{X}$ by decomposing or factorizing $\mathbf{X}$ into two low-dimensional matrices $\mathbf{U} \in \mathbb{R}^{m\times k}$ and $\mathbf{V} \in \mathbb{R}^{n\times k}$ by minimizing

$$\underset{\mathbf{U},\mathbf{V}}{\operatorname{argmin}} \operatorname{Dis}(\mathbf{X}, \mathbf{U}\mathbf{V}'), \tag{1}$$

where $k \ll \min(m, n)$ is the dimension of the latent space or the rank of the underlying data and $\operatorname{Dis}(\mathbf{X}, \mathbf{U}\mathbf{V}')$ denotes the error between the original data $\mathbf{X}$ and the reconstructed data $\mathbf{U}\mathbf{V}'$ under some specific distance metric, such as KL-divergence, ℓ_1 norm and Frobenius norm. In nonnegative matrix factorization (NMF), a well-known low-rank approximation with known rank, $\mathbf{U}$ and $\mathbf{V}$ are required to be nonnegative. Besides, $\mathbf{U}$ and $\mathbf{V}$ have their own specific meanings in many application. Taking face clustering as an example [17], each column of $\mathbf{U}$ is considered as a basis image (latent space dictionary atom), while each column of $\mathbf{V}$ is called an encoding (new representation). This new representation is more effective and robust for clustering. The most serious limitation of this kind of methods is that the dimension of the latent space, i.e., k, must be pre-determined. In reality, nevertheless, it is often difficult to determine k in advance.

2.1.2 Approximation with unknown Rank. When the rank of the data is not given, we may directly approximate X with a low-rank matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$, which is the idea of low-rank approximation [6], by minimizing

$$\underset{\mathbf{A}}{\operatorname{argmin}} \operatorname{Dis}(\mathbf{X}, \mathbf{A}) + \lambda \operatorname{rank}(\mathbf{A}),$$

where rank($\mathbf{A}$) is the rank of the matrix $\mathbf{A}$ and λ is a parameter for tradeoff between the two terms. Since the rank function is nonconvex, we can alternatively minimize its convex surrogate as

$$\underset{\mathbf{A}}{\operatorname{argmin}} \operatorname{Dis}(\mathbf{X}, \mathbf{A}) + \lambda ||\mathbf{A}||_*, \tag{2}$$

where $||\mathbf{A}||_*$ is the trace norm of $\mathbf{A}$, i.e., the sum of the singular values of $\mathbf{A}$. The most well-known approach for low-rank approximation with unknown rank is Robust PCA (RPCA) [6], which has been used for background subtraction, texture repair and subspace segmentation. RPCA uses the ℓ_0 norm to measure the difference between the original data $\mathbf{X}$ and the low-rank approximation $\mathbf{A}$, i.e., $\operatorname{Dis}(\mathbf{X}, \mathbf{A}) = ||\mathbf{X} - \mathbf{A}||_0$, where $||\mathbf{X}||_0$ is the number of nonzeros in $\mathbf{X}$. As before, RPCA directly optimizes the $||\mathbf{X}||_1$ for its convexity instead of $||\mathbf{X}||_0$ and for the equivalence of the following two problems under rather broad conditions (the error matrix $\mathbf{X} - \mathbf{A}$ is sufficiently sparse relative to the rank of $\mathbf{A}$) [6]:

$$\underset{\mathbf{A}}{\operatorname{argmin}} ||\mathbf{X} - \mathbf{A}||_0 + \lambda \operatorname{rank}(\mathbf{A}),$$
$$\underset{\mathbf{A}}{\operatorname{argmin}} ||\mathbf{X} - \mathbf{A}||_1 + \lambda ||\mathbf{A}||_*.$$

However, this method has drawbacks. It imposes some restrictions on the data, particularly, the noise is assumed to be sparse and the rank of data is sufficiently low [6]. In practice, we often cannot derive the actual rank by directly computing the rank of the resultant matrix $\mathbf{A}$, since minimizing the rank function and its convex surrogate is not equivalent for many applications when the sparsity of the error is not sufficient relative to the actual rank of the data. Second, it cannot directly obtain the latent space dictionary and representation based on this dictionary as in NMF, which is very important for many clustering problems, including community detection. Third, although it has been argued that balancing parameter λ can be done in theory under rather broad condition, the parameter still needs to be tuned in practice when the sparsity of the error is not sufficient relative to the actual rank of the data.

2.1.3 Rank Determination. While determining the rank of a dataset is critical, the available approaches are limited, which can be divided into two categories. The methods of first category do not directly determinate the rank of the data, but rather evaluate each candidate using the Markov chain Monte Carlo [7] or sampling the rank along with other parameter using computationally intensive reversible jump Markov chain Monte Carlo [60]. This type of sampling methods is computationally expensive. The methods in the second category compute the rank through a Bayesian approach. Some of them assume the elements of factorized matrix follow exponential priors [24, 25], while others assume them follow Gamma priors [4]. BNMF assumes the reconstruction error, the factorized matrix and parameters are Poisson, half-normal and Gaussian distributions, respectively [40]. There are two main drawbacks of this kind of methods. First, there are many hyperparameters, which significantly affect the performance, need to be tuned. Although the authors provide some suggestions to tune them, it often can not obtain the best performance. Second, it lacks of interpretability as many other Bayesian learning methods, which makes it hardly to tune the parameters and analyze the results.

2.2 Community Detection with NMF

The community detection problem using NMF can be modeled by a generative process of a network [30]. In particular, let x_{ij} be a variable indicating whether vertices i and j are connected. We then define $\mathbf{U} = [u_{it}] \in \mathbb{R}_+^{n \times k}$ and $\mathbf{V} = [v_{jt}] \in \mathbb{R}_+^{n \times k}$ as the membership matrices where elements u_{it} and v_{jt} represent the probabilities that vertex i generates an in-edge and an out-edge in community t, respectively. They also imply the probability that node i belongs to the in- or out-community t.

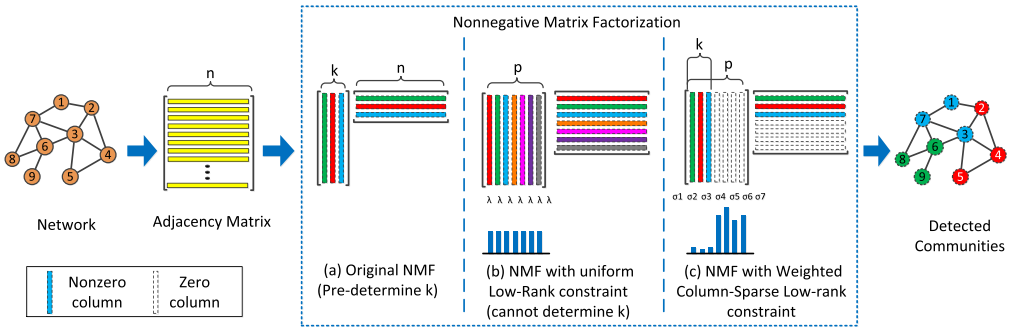


Fig. 1. Illustration and comparison of the three NMF-based methods. Community detection methods via NMF take the network adjacency matrix as input, and classify the node according to the factorized matrix. (a) Original NMF needs to pre-determine the number of communities, i.e., the number of columns in factorized matrix. (b) NMF with low-rank constraint, which is equivalent to suppressing all columns uniformly, can not make the factorized matrix column-sparse, thus it can not directly obtain the number of communities. (c) NMF with weighted column-sparse low-rank constraint can suppress most of the columns to zero, and the number of nonzero columns is that of the communities.

Since the networks that we considered here are undirected, either $\mathbf{U}$ or $\mathbf{V}$ can be used to partition the network and separate the vertices.

Whether vertices i and j are connected depends on probability that they belong to the same community. The probability of vertices i and j belonging to community t is $u_{it}v_{jt}$, and the probability that vertices i and j belong to the same community is then

$$\hat{x}_{ij} = \sum_{t=1}^{k} u_{it}v_{jt}. \tag{3}$$

Therefore, the community detection problem can be modeled as a NMF problem $\mathbf{X} \approx \hat{\mathbf{X}} = \mathbf{UV}'$. This process is illustrated in Figure 1. Viewed as low-rank matrix approximation, community detection is to find a low-dimensional (low-rank) representation of the original network. In general, the index of the largest element in each row of $\mathbf{U}$ and $\mathbf{V}$ is the community that the corresponding vertex belongs to.

We should notice that since the ith row of $\mathbf{U}$ and $\mathbf{V}$, denoted as $\mathbf{u_i}$ and $\mathbf{v_i}$, respectively, can be regarded as the membership distribution of vertex i. If the number of the communities, i.e., the rank of $\mathbf{U}$ and $\mathbf{V}$, is not correctly set, then the result may be meaningless.

2.3 Model Selection in Community Detection

The community detection methods can be divided into two categories, discrimination model and generative model. According to the detection method, a number of model selection algorithms have been proposed, such as greedy algorithm, spectral algorithm and statistical algorithms (expectation-maximization algorithms and sampling methods). Louvain [5] and Infomap [33] are two commonly used greedy algorithm. Louvain, which adopts agglomerative hierarchical method, initially assigns each node to its own community, and then moved in the community associated to the largest modularity gain. This process is repeated until no further improvement can be achieved. Infomap optimizes the map equation, which quantifies the information needed to represent some random walk in the network, using simulated annealing. In Spectral algorithm [27, 28], the leading eigenvector of modularity matrix is regarded to be correlated to the optimum assignments and the eigenvalues is used to determine the number of communities. The index of eigenvalue that gets the largest drop from previous one is considered as the number of communities. Akaike

Information Criterion (AIC) [3] and Bayesian Information Criterion (BIC) [34] are well-known statistical model selection methods adopted by Bayesian community detection including stochastic block model (SBM).

3 MODELS

Our new method NMF-AWL hinges upon the idea of introducing an adaptively weighted low-rank regularization to NMF. We first present a straightforward method, i.e., NMF with a low-rank constraint. We then discuss the rationale that it cannot yield a satisfactory result and develop a novel NMF with a weighted column-sparse low-rank constraint. Finally, we present a strategy to adaptively determine the weights of the columns.

3.1 NMF with Low-rank Constraint

A straightforward method to combine the merits of these two kinds of low-rank approximation methods discussed in the previous section is to factorize the matrix $\mathbf{A}$ in Equation (2) into $\mathbf{U} \in \mathbb{R}^{m \times p}$ and $\mathbf{V} \in \mathbb{R}^{n \times p}$,

$$\underset{\mathbf{U},\mathbf{V}}{\operatorname{argmin}} \operatorname{Dis}(\mathbf{X}, \mathbf{U}\mathbf{V}') + \lambda ||\mathbf{U}\mathbf{V}'||_*, \tag{4}$$

where $p \gg k$ and $p \leq \min(m, n)$ is a pre-defined number of columns for $\mathbf{U}$. Equation (4) can be regarded as adding a low-rank constraint $\lambda||\mathbf{U}\mathbf{V}'||_*$ to Equation (1). By doing so, we hope the rank of the resulting matrix $\mathbf{U}$ to be automatically determined as in RPCA.

3.2 Weighted Column-Sparse Low-rank Constraint

However, the resulting matrix $\mathbf{U}$ may not have a low rank in practice. Even if it is low rank, it may also be difficult to determine which columns should be selected for data representation (membership in community detection). We thus like to force the resulting matrix to be column sparse to help choose the non-zero columns as the final membership matrix. However, the original low-rank constraint cannot yield column-sparse matrix. To see this, we rewrite Equation (4) into

$$\underset{\mathbf{U},\mathbf{V}}{\operatorname{argmin}} \operatorname{Dis}(\mathbf{X}, \mathbf{U}\mathbf{V}') + ||(\mathbf{U}(\sqrt{\lambda}\mathbf{I}))(\mathbf{V}(\sqrt{\lambda}\mathbf{I})')||_*, \tag{5}$$

where $\mathbf{I}$ is the p-dimensional identity matrix. Since $\mathbf{U} = [\mathbf{u_1}, \mathbf{u_2}, \ldots, \mathbf{u_p}]$, where $\mathbf{u_i}$ is the i^{th} column of $\mathbf{U}$, $\mathbf{U}(\sqrt{\lambda}\mathbf{I}) = [\sqrt{\lambda}\mathbf{u_1}, \sqrt{\lambda}\mathbf{u_2}, \ldots, \sqrt{\lambda}\mathbf{u_p}]$. This means that the introduction of a low-rank constraint to Equation (4) equally suppresses each column of $\mathbf{U}$, which can be formally proven following a lemma by Reference [23].

Lemma 3.1. *For any matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, the following holds:*

$$||\mathbf{A}||_* = \min_{\mathbf{U},\mathbf{V},\mathbf{A}=\mathbf{U}\mathbf{V}'} \frac{1}{2}\left(||\mathbf{U}||_F^2 + ||\mathbf{V}||_F^2\right).$$

When $\operatorname{rank}(\mathbf{A}) = k \leq \min(m, n)$, the minimum is attained with a factorization $\mathbf{A} = \mathbf{U}\mathbf{V}'$ where $\mathbf{U} \in \mathbb{R}^{m \times k}$ and $\mathbf{V} \in \mathbb{R}^{n \times k}$.

Based on Lemma 1, we have the following result.

Theorem 3.2. *If $\mathbf{A}$ is the solution to Equation (2), and $(\mathbf{U}, \mathbf{V})$ is the solution to the following minimization problem:*

$$\underset{\mathbf{U},\mathbf{V}}{argmin} \operatorname{Dis}(\mathbf{X}, \mathbf{U}\mathbf{V}') + \frac{\lambda}{2}\left(||\mathbf{U}||_F^2 + ||\mathbf{V}||_F^2\right),$$

then $\mathbf{A} = \mathbf{U}\mathbf{V}'$.

Proof. According to Lemma 3.1, we can write

$$\begin{aligned}
&\operatorname*{argmin}_{\mathbf{A}} \mathrm{Dis}(\mathbf{X},\mathbf{A}) + \lambda||\mathbf{A}||_* \\
\Leftrightarrow &\operatorname*{argmin}_{\mathbf{A}} \left(\mathrm{Dis}(\mathbf{X},\mathbf{A}) + \min_{\substack{\mathbf{U},\mathbf{V}\\ \mathbf{A}=\mathbf{U}\mathbf{V}'}} \frac{\lambda}{2}\left(||\mathbf{U}||_F^2 + ||\mathbf{V}||_F^2\right)\right) \\
\Leftrightarrow &\operatorname*{argmin}_{\mathbf{A}} \min_{\substack{\mathbf{U},\mathbf{V}\\ \mathbf{A}=\mathbf{U}\mathbf{V}'}} \left(\mathrm{Dis}(\mathbf{X},\mathbf{U}\mathbf{V}') + \frac{\lambda}{2}\left(||\mathbf{U}||_F^2 + ||\mathbf{V}||_F^2\right)\right) \\
\Leftrightarrow &\operatorname*{argmin}_{\mathbf{U},\mathbf{V}} \left(\mathrm{Dis}(\mathbf{X},\mathbf{U}\mathbf{V}') + \frac{\lambda}{2}\left(||\mathbf{U}||_F^2 + ||\mathbf{V}||_F^2\right)\right).
\end{aligned}$$

So, we have $\mathbf{A} = \mathbf{U}\mathbf{V}'$. □

Therefore, minimizing $\lambda||\mathbf{U}\mathbf{V}'||_*$ is equivalent to minimizing $\frac{\lambda}{2}(||\mathbf{U}||_F^2) + ||\mathbf{V}||_F^2)$, and

$$\lambda||\mathbf{U}||_F^2 = ||\mathbf{U}(\sqrt{\lambda}\mathbf{I})||_F^2 = \lambda\sum_{t=1}^{p}||\mathbf{u_t}||_2^2 = \sum_{t=1}^{p}||\sqrt{\lambda}\mathbf{u_t}||_2^2.$$

This result is intuitive, i.e., introducing a low-rank constraint equally suppresses each column of matrices $\mathbf{U}$ and $\mathbf{V}$. Consequently, no column-sparse matrix $\mathbf{U}$ can be obtained by directly optimizing NMF with a low-rank constraint through Equation (4).

Our first idea for deriving a column-sparse matrix $\mathbf{U}$ is to introduce a low-rank regularization that is able to selectively suppress some columns of $\mathbf{U}$. To do so, we use distinct weights for different columns, i.e., use the following regularization in place of $\lambda||\mathbf{U}||_F^2$:

$$||\mathbf{U}\Sigma^{\frac{1}{2}}||_F^2 = \sum_{t=1}^{p}\sigma_t||\mathbf{u_t}||_2^2 = \sum_{t=1}^{p}||\sqrt{\sigma_t}\mathbf{u_t}||_2^2, \tag{6}$$

where $\Sigma = \mathrm{diag}([\sigma_1, \sigma_2, \ldots, \sigma_p])$, a p-dimensional diagonal matrix whose ith diagonal element is σ_i and non-diagonal elements are zero, is used to specify the degree of suppression on different columns. If $\sigma_i > \sigma_j$, then the ith column will be suppressed more than the jth column when minimizing Equation (6). Thus, Equation (5) can be generalized to matrix factorization with weighted low-rank constraints as

$$\operatorname*{argmin}_{\mathbf{U},\mathbf{V}} \mathrm{Dis}(\mathbf{X},\mathbf{U}\mathbf{V}') + ||(\mathbf{U}\Sigma^{\frac{1}{2}})(\mathbf{V}\Sigma^{\frac{1}{2}})'||_*. \tag{7}$$

The remaining problem is how to determine the weights in Σ.

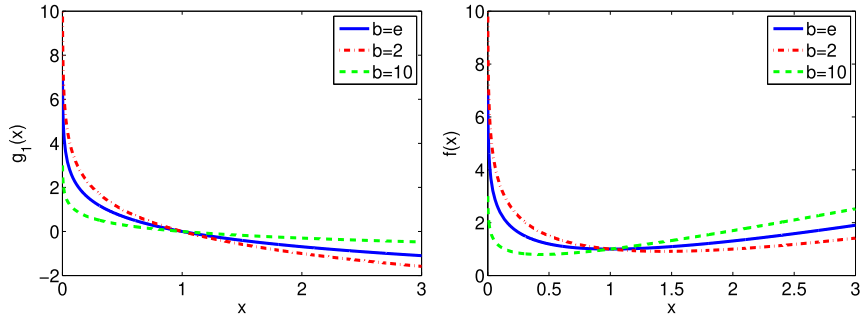
3.3 Adaptive Weights

Since there are p parameters in Σ, it is impractical to consider them once at a time. A strategy is needed to learn the parameters from the data. We first make the following two observations. First, each σ_i should not be too small; otherwise, the regularization will have little effect and the overall objective function degrades to the original low-rank matrix factorization problem that requires to have the rank of the data be known. Second, some σ_i's should be larger than the others to suppress the corresponding columns. However, not all the σ_i's should be large, otherwise, all the columns will be suppressed to zero.

To make Σ satisfy these criteria, we introduce a penalty function,

$$F(\Sigma) = G(\Sigma) + H(\Sigma), \tag{8}$$

where $G(\Sigma)$ is used to prevent all σ_i's from being too small, and $H(\Sigma)$ is used to prevent all the σ_i's from being too large.



(a) Curves of $g_1(x)$ (b) Curves of $f(x)$

Fig. 2. Curves of the penalty functions with different parameter b. (a) The curves of $g_1(x)$. When b is large, the curve decreases rapidly approaching to 0 and has a flat tail, while it decreases slowly near 0 and has a steep tail when b is set small. (b) The curves of $f(x)$. A larger b makes $f(x)$ reach its minimum at a small x, and vice versa. Thus, to make it have the minimum at $x = 1$, we choose $b = e$.

To obey the first criterion, $G(\cdot)$ should be a monotonically non-increasing function. For simplicity, we also choose a differentiable function for $G(\cdot)$. Furthermore, we suppose that we can individually penalize each σ_i and the penalty is zero if $\sigma_i = 1$. Two simple functions that meet these requirements are

$$G_1(\Sigma) = -\beta \log_b(|\Sigma|) = \beta \sum_{t=1}^{p} -\log_b(\sigma_t) = \beta \sum_{t=1}^{p} g_1(\sigma_t),$$
$$G_2(\Sigma) = \beta \sum_{t=1}^{p} \left(\frac{1}{\sigma_t} - 1\right) = \beta \sum_{t=1}^{p} g_2(\sigma_t), \tag{9}$$

where β is a positive value to determine the strength of $G(\cdot)$ and $|\Sigma|$ denotes the determinant of Σ. Figure 2(a) shows $g_1(x)$. Since Σ is a diagonal matrix, $|\Sigma| = \prod_{t=1}^{p} \sigma_t$. Herein, we will fix parameter 2β to the total number of rows of the targeting matrices $\mathbf{U}$ and $\mathbf{V}$, i.e., $m + n$; the rationale for doing so will be discussed in Section 4.2. Furthermore, $g_1(\cdot)$ and $g_2(\cdot)$ are collectively referred to $g(\cdot)$, and we will verify that natural logarithm, i.e., $b = e$, often yields good results in reality.

However, to obey the second criterion, $H(\cdot)$ should be a monotonically non-decreasing function. For simplicity, we use the trace of Σ as $H(\cdot)$, i.e.,

$$H(\Sigma) = \alpha \operatorname{tr}(\Sigma) = \alpha \sum_{t=1}^{p} \sigma_t = \alpha \sum_{t=1}^{p} h(\sigma_t), \tag{10}$$

where $\operatorname{tr}(\cdot)$ is the trace of a matrix and α is a positive value to determine the contribution of $H(\cdot)$. Since $H(\cdot)$ is not as important as $G(\cdot)$, we should choose a small α. Unless noted, we set $\alpha = 1$. The role of α will also be discussed in detail in Section 4.2. Thus, Equation (8) can be expressed as

$$F(\Sigma) = \beta \sum_{t=1}^{p} g(\sigma_t) + \alpha \sum_{t=1}^{p} h(\sigma_t)$$
$$= \sum_{t=1}^{p} \Big(\beta g(\sigma_t) + \alpha h(\sigma_t)\Big). = \sum_{t=1}^{p} f(\sigma_t). \tag{11}$$

Herein, $f(\sigma)$ is the penalty function to make σ's meet the above mentioned criteria, and it is shown in Figure 2(b) with $\alpha = \beta = 1$. Furthermore, to make $f(\sigma)$ reach its minimum at $\sigma = 1$, $b = e$ is a natural choose.

3.4 Reconstruction Error

Traditional nonnegative matrix factorization (NMF) seeks the latent representation by minimizing the reconstruction error from the factorized nonnegative matrices as in Equation (1). The proposed NMF-AWL, however, balances between the reconstruction error and the regularization terms of low-rank constraint and adaptive weights. Therefore, the reconstruction error of NMF-AWL should theoretically be larger than that of traditional NMF with the same fixed rank. Reconstruction error is also a commonly used criterion for model selection such as minimum description length (MDL)-based model selection in Reference [14]. The main drawback of reconstruction error-based model selection is its complexity. It needs to traverse all possible rank values and choose the one with the smallest reconstruction error. Different from reconstruction error-based model selection, our proposed NMF-AWL does not need this traversal, but simultaneously detects community structure and the number of communities. Therefore, advantage in complexity is very obvious.

4 OPTIMIZATION AND INSIGHT

We now consider the optimization algorithm, provide some insights to the final model, discuss how to set the model parameters, present the algorithm for autonomously detecting community and analyze the complexity of the algorithm.

4.1 Objective Function and Optimization

The overall objective function consists of the weighted group-sparse low-rank constraint in Equation (7) and the weights for penalty in Equation (11). The KL-divergence is adopted as the distance measure between the original data and reconstructed data, since it is more robust to measure the distance between two probability distributions than L2 norm. Specifically, each element x_{ij} in original data $\mathbf{X} \in \{0, 1\}^{N \times N}$ is either 0 or 1 to represent whether an edge between corresponding nodes i and j is observed. Each element y_{ij} in the reconstructed data $\mathbf{Y} = \mathbf{U}\mathbf{V}'$ is the probability of edge existence according to community assignments $\mathbf{U}$ and $\mathbf{V}$. Therefore, KL-divergence is used to measure the difference between these two probability distributions. Besides, the three matrices are forced to be nonnegative for interpretability of the expected result. The final objective function is

$$\underset{\mathbf{U}>\mathbf{0},\mathbf{V}>\mathbf{0},\Sigma>\mathbf{0}}{\operatorname{argmin}} \quad \mathcal{L}_{KL}(\mathbf{X}||\mathbf{U}\mathbf{V}') + ||\mathbf{U}\Sigma\mathbf{V}'||_* + f(\Sigma). \tag{12}$$

Herein,

$$\mathcal{L}_{KL}(\mathbf{X}||\mathbf{U}\mathbf{V}') = \sum_{i,j}\left(x_{ij}\log\left(\frac{x_{ij}}{y_{ij}}\right) - x_{ij} + y_{ij}\right), \tag{13}$$

where $\mathbf{X} = [x_{ij}]$ and $\mathbf{Y} = [y_{ij}] = \mathbf{U}\mathbf{V}'$. Following Theorem 3.1 and Equations (13), (9), and (10), the overall objective function Equation (12) can be written as

$$\begin{aligned}\mathcal{L}(\mathbf{U},\mathbf{V},\Sigma) \quad = \quad & \mathcal{L}_{KL}(\mathbf{X}||\mathbf{U}\mathbf{V}') + \alpha\sum_{t=1}^{p}\sigma_t + \beta\sum_{t=1}^{p}g(\sigma_t) \\ & + \frac{1}{2}\sum_{t=1}^{p}\sigma_t\left(||\mathbf{u_t}||_2^2 + ||\mathbf{v_t}||_2^2\right).\end{aligned}$$

Since the loss function $\mathcal{L}(\mathbf{U}, \mathbf{V}, \Sigma)$ is not convex for all $\mathbf{U}$, $\mathbf{V}$, and Σ together as in the original KL-divergence-based NMF, it is difficult to find the global minima. We develop an iterative algorithm similar to the multiplicative updating rules in Reference [18], which can reach local minima. The procedure has two major steps. First, we compute the gradients of the object function $\mathcal{L}(\mathbf{U}, \mathbf{V}, \Sigma)$ with respect to $\mathbf{U}$, $\mathbf{V}$ and Σ. Second, we update $\mathbf{U}$ and $\mathbf{V}$ by multiplying their current values with the ratio between the positive to the negative parts of the gradients. The updating rules for $\mathbf{U}$ and $\mathbf{V}$ are as follows:

$$u_{it} \leftarrow u_{it} \frac{\sum_j (x_{ij} v_{jt} / \sum_t u_{it} v_{jt})}{\sigma_t u_{it} + \sum_j v_{jt}}, \tag{14}$$

$$v_{jt} \leftarrow v_{jt} \frac{\sum_i (x_{ij} u_{it} / \sum_t u_{it} v_{jt})}{\sigma_t v_{jt} + \sum_i u_{it}}. \tag{15}$$

We update Σ by setting its derivative equal to zero as it is analytically calculated given $\mathbf{U}$ and $\mathbf{V}$. The updating rules for Σ with $G_1(\cdot)$ and $G_2(\cdot)$ are as follows:

$$\sigma_t \leftarrow \frac{\beta}{\left(\frac{1}{2}(||\mathbf{u_t}||_2^2 + ||\mathbf{v_t}||_2^2) + \alpha\right) \ln b}, \tag{16}$$

$$\sigma_t \leftarrow \sqrt{\frac{\beta}{(||\mathbf{u_t}||_2^2 + ||\mathbf{v_t}||_2^2)/2 + \alpha}},$$

where b is the base of logarithm, and α and β are parameters to be set, as discussed next. For network data, each row of $\mathbf{U}$ or $\mathbf{V}$ represents one node. For non-network data (such as face image collection in Reference [17]), rows of $\mathbf{U}$ and $\mathbf{V}$ possess different meanings (such as one image and pixel, respectively). However, no matter network or non-network data, $\mathbf{u_t}$ and $\mathbf{v_t}$, which are the tth columns of factorized matrix $\mathbf{U}$ and $\mathbf{V}$, have the same role, i.e., one latent component. Therefore, Equation (16) makes sense for both network and non-network data.

4.2 Parameter Setting

Based on our experiments (Section 5.2), $b = e$, i.e., taking the natural logarithm, often gives rise to good results, which is the value we will use. With $b = e$ and $\alpha = 0$, Equation (16) can be rewritten as

$$\sigma_t \leftarrow \frac{2\beta}{||\mathbf{u_t}||_2^2 + ||\mathbf{v_t}||_2^2}. \tag{17}$$

This means that if the ℓ_2 norm of the kth columns of $\mathbf{U}$ and $\mathbf{V}$ is small, the corresponding weight σ_t will be large, so that these two columns will be heavily suppressed in the next iteration. If we set $\beta = (m + n)/2$, then σ_k is the reciprocal of the average of the square of all the $m + n$ elements in $\mathbf{u_t}$ and $\mathbf{v_t}$, i.e.,

$$\sigma_t \leftarrow \frac{1}{\frac{1}{m+n}(\sum_{i=1}^m u_{it}^2 + \sum_{j=1}^n v_{jt}^2)}.$$

It implies that whether $\sigma_t > 1$ should be determined by whether the average of the square of elements in $\mathbf{u_t}$ and $\mathbf{v_t}$ is smaller than 1.

Second, if $\alpha = 0$, then σ_t will approach infinity if the tth column is suppressed to zero. To avoid this situation, we set α to a smaller number than β. This is consistent with the rule that the first criterion of forcing some columns to zero predominates the second of not forcing all columns to zero. As our experimental results suggested, α can be simply set to 1 to obtain satisfactory results. From Equation (16), it is evident that a smaller α makes σ_t's larger, so that more columns will be

ALGORITHM 1: NMF with adaptively weighted low-rank constraint (NMF-AWL) for community detection

Input: Adjacency matrix $\mathbf{X} \in \mathbb{R}^{m \times m}$, the number of columns $p = m/2$, $\beta = m$ and $\alpha = 1$ for normal circumstances or $\alpha \in \{1, 2, 3\}$ for hierarchical community detection.
Output: The number of communities k and the nodes' affiliation.
Initialization: $\Sigma = \mathbf{I}$, random matrix $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{m \times p}$
for $t = 1 : n_{iter}$ **do**
  Update $\mathbf{U}$ via Equation (14);
  Update $\mathbf{V}$ via Equation (15);
  Update Σ via Equation (16);
end
The number of communities k is that of columns of $\mathbf{U}$ whose ℓ_2 norms are less than 10^{-200}.
By removing the zero columns of $\mathbf{U}$, we obtain the final affiliation matrix $\mathbf{U}^* = [u^*_{ij}] \in \mathbb{R}^{m \times k}$.
We assign the i^{th} node to the t^{th} community if u^*_{ik} is the largest element in the i^{th} row of $\mathbf{U}^*$

suppressed to zero, resulting in a smaller number of communities. Thus, if we want to divide the network into smaller communities, then we can properly increase α.

Finally, based on Equations (14) and (15) the only difference between the updating rules of the conventional KL-divergence-based NMF and NMF-AWL is the added term $\sigma_t u_{it}$ or $\sigma_t v_{jt}$ in the denominator. Since σ_t is the reciprocal of the average of the square of all the elements in $\mathbf{u_t}$ and $\mathbf{v_t}$, if u_{it} or v_{jt} is larger than this average, $\sigma_t u_{it}$ or $\sigma_t v_{jt}$ is larger than 1 and Equation (14) or Equation (15) makes it (u_{it} and v_{jt}) smaller, i.e., suppresses it to zero.

4.3 Autonomous Community Detection

The adjacency matrix of an undirected and unweighed graph $G = (V, E)$ over m vertices is a symmetric nonnegative binary matrix $\mathbf{X} = [x_{ij}] \in \mathbb{R}^{m \times m}_+$. Assume the actual rank k of $\mathbf{X}$ is much smaller than m, i.e., $k \ll m$, we may set the initial number of columns of factorized matrices to $p = m/2 \gg k$. and α to 1. In general, if we can estimate an upper bound of k, then we may set the initial value of p to the upper bound. We then factorize the adjacency matrix $\mathbf{X}$ into a component (community) matrix $\mathbf{U}$ by minimizing $\mathcal{L}(\mathbf{U}, \mathbf{V}, \Sigma)$. This community detection method is specified in Algorithm 1. The iteration is terminated if the relative change of the maximum value of σ_t's is little than 1e-5. The number of communities is the number of nonzero columns of $\mathbf{U}$, i.e., k, and the resulting communities correspond to the non-zero columns. By removing zero columns of $\mathbf{U}$, we have the final community membership matrix $\mathbf{U}^* = [u^*_{it}] \in \mathbb{R}^{m \times k}$. Following convention, we may assign a vertex to the community to which it has the highest membership.

4.4 Computational Complexity

For convenience, we only analyze the complexity of Equations (14) and (16). We reformulate Equation (14) as

$$u_{it} \leftarrow u_{it} \frac{\sum_j (\frac{x_{ij}}{\sum_t u_{it} v_{jt}} v_{jt})}{\sigma_t u_{it} + \sum_j v_{jt}}. \tag{18}$$

Since the size of $\mathbf{U}$ is np, the multiplication between u_{it} and the fraction will be repeated np times. The denominator consists of two parts. The first part $\sigma_t u_{it}$ will be repeated for each i and k, thus needs np floating-point multiplications. The second part $\sum_j v_{jt}$ will be repeated for each t, thus needs np floating-point additions. The numerator is made up of a summation of n elements, which will be repeated np times, thus it requires $n^2 p$ floating-point additions. Similarly, the multiplication inside the brackets needs $n^2 p$ floating-point multiplications. Since $\frac{x_{ij}}{\sum_t u_{it} v_{jt}}$ is independent of

Table 1. Real-world Networks that Were Experimentally Analyzed

Datasets	m	n	k	Description
Karate [52]	34	78	2	Zachary's karate club
Dolphins [21]	62	159	2	Dolphin social network
Friendship6 [45]	68	220	6	High school friendship
Friendship7 [45]	68	220	7	High school friendship
Football [13]	115	613	12	American College football
Polbooks [28]	105	441	3	Books about US politics
Polblogs [1]	1,490	16,718	2	Blogs about US politics
Cora [51]	2,708	5,429	7	Publication citation dataset from ML
Citeseer [51]	3,312	4,732	6	Publication citation dataset from Citerseer site
Syracuse [41]	13,653	543,982	7	Facebook networks at Syracuse University
NYU [41]	21,679	715,715	7	Facebook networks at New York University
Word Association [26]	5,017	29,148	-	Words that people always associate

k, it only needs n^2 floating-point divisions. The summation $\sum_t u_{it}v_{jt}$ is repeated for each i and j, resulting in n^2p floating-point additions. Overall, Equation (14) needs $2n^p + np$ floating-point additions, $2n^p + np$ floating point multiplications and $2n^2 + np$ floating-point divisions. The overall complexity is $O(n^2p)$. Thus, according to Equations (14), (15), and (16), NMF-AWL does not incur additional complexity with respect to the conventional KL-divergence-based NMF.

Due to the sparsity of the adjacency matrix $\mathbf{X}$, the n^2p and n^2 operations can be reduced to mp and m, respectively. For example, since there are only m nonzero elements x_{ij}, we do not have to compute all the $\sum_t u_{it}v_{jt}$, but only need to perform the multiplication for i and j when $x_{ij} \neq 0$. Thus its complexity is reduced to mp. Thus, the complexity of Equation (14) is $O((m + n)p)$. Although we iteratively update σ_t's in Equation (16), its overall complexity is only $O(np)$. Therefore, the complexity of the NMF-AWL is $O((m + n)p)$. If we can properly choose $p = O(k)$, i.e., p and k are in the same order of magnitude, then the complexity of NMF-AWL is nearly linear in m and n.

Furthermore, from the experiments in Section 5.3, we observed that most of the columns of the model in Section 3 were suppressed to zero over iterations. Thus, by considering the column-sparsity of $\mathbf{U}$ and $\mathbf{V}$, the complexity of each iteration is $O((m + n)p)$ over the initial few iterations, while it reduces to $O((m + n)k)$, which is independent of the initial number of columns p, over most of the iterations. Therefore, NMF-AWL is nearly linear in network size, and thus efficient on large networks.

Note that, since the only difference between our NMF-AWL and conventional KL-divergence-based NMF is the multiplication factors of $\sigma_t u_{it}$ and $\sigma_t v_{jt}$ in the denominators of the updating rules for $\mathbf{U}$ and $\mathbf{V}$, the parallel [8] and distributed [19] computing technologies designed for NMF can be applied to NMF-AWL to make it applicable to larger networks.

5 EXPERIMENTS

To evaluate NMF-AWL, we carried out experiments on two types of synthetic networks and several widely used real-world networks, listed in Table 1. We compared NMF-AWL with four state-of-the-art approaches for autonomous community detection methods, i.e., the louvain algorithm [5], the spectral (SP) algorithm due to Reference [28], the external optimization (EO) algorithm by Reference [10], and the Bayesian NMF (BNMF) algorithm from Reference [30]. We adopted two metrics for performance evaluation, the normalized mutual information (NMI) [39] and the difference between the ground-truth communities and the detected communities. The first metric

directly evaluates the accuracy of detected community structures, while the second metric only assess the number of communities detected; the first metric is more informative than the second as the former measures community structures. To compare the accuracy of the number of detected communities, we use the following measurement:

$$\text{diff}(k_{gt}, k_{detected}) = |k_{gt} - k_{detected}| + 1, \tag{19}$$

where k_{gt} is the number of ground-truth communities and $k_{detected}$ is the number of detected communities. It is evident that this function is equal to 1 if and only if $k_{gt} = k_{detected}$ and increases with the difference between k_{gt} and $k_{detected}$ increases. Suppose φ^a is the ground-truth of community structure and φ^b is the result from the algorithm, then the NMI of this algorithm is defined as

$$\text{NMI}(\varphi^a, \varphi^b) = \frac{\sum_{i=1}^{k_{gt}} \sum_{j=1}^{k_{detected}} n_{ij} \log\left(\frac{n \cdot n_{ij}}{n_i^a \cdot n_j^b}\right)}{\sqrt{\left(\sum_{i=1}^{k_{gt}} n_i^a \log \frac{n_i^a}{n}\right)\left(\sum_{j=1}^{k_{detected}} n_j^b \log \frac{n_j^b}{n}\right)}}, \tag{20}$$

in which n is the number of nodes, n_{ij} is the number of nodes both in ground-truth community i and in result community j, n_i^a is the number of nodes in ground-truth community i, and n_j^b is the number of nodes in the result community j. NMI is more informative than just simply counting the number of misclassified nodes or computing the accuracy [39]. It is especially suitable for imbalanced datasets such as Lancichinetti-Fortunato-Radicchi networks (LFR networks) benchmark and some real-world networks, which will be discussed in the following sections.

All experiments were conducted on a single PC (Intel(R) Core(TM) i7-2600 CPU @ 3.40 GHz. processor with 4 G memory). Since algorithms only converge to local minima, we repeated each algorithm many times with random initialization and chose the result giving the smallest objective function value. The source code of all the algorithms used in this article can be downloaded from the authors' websites.

5.1 Synthetic Networks

We considered two types of synthetic networks in our experiments, i.e., Girvan-Newman networks (GN networks) [13] and Lancichinetti-Fortunato-Radicchi networks (LFR networks) [16].

Each GN network consists of 128 nodes that are divided into 4 communities of 32 nodes each. Each node has on average 16 edges, among which Z_{out} edges are inter-community edges. As Z_{out} increases, community detection is more difficult as the community structure becomes weaker. In our experiments, we set the parameter $\alpha = 2$ (similar results with $\alpha = 1$). The results are shown in Figure 3. As shown, NMF-AWL, the louvain, and the external optimization can correctly detect the number of communities (right panel of Figure 3) and our method outperforms the other methods, especially on networks with large Z_{out} (left panel of Figure 3).

The LFR networks are more complicated than the GN networks. Its generator allows to specify the number of node, average degree, community size distribution, degree distribution, minimum and maximum of the community sizes and the fraction of the inter-community edge (mixing parameter μ). Following the experiment setting suggested by Reference [16], we set the number of nodes to 1,000, the minimum community size to 20, the maximum community size to 100, the average degree to 20, the exponent of a vertex degree and the community size to -2 and -1, respectively, and vary the mixing parameter μ from 0.1 to 0.6. We set $\alpha = 1$ and the diagonal elements of adjacency matrix as the degree of corresponding nodes. The results are shown in Figure 4. The right subfigure indicates that NMF-AWL are more accurate than many other algorithms on determining the number of communities (except for louvain on networks with $\mu = 0.5$). From the left

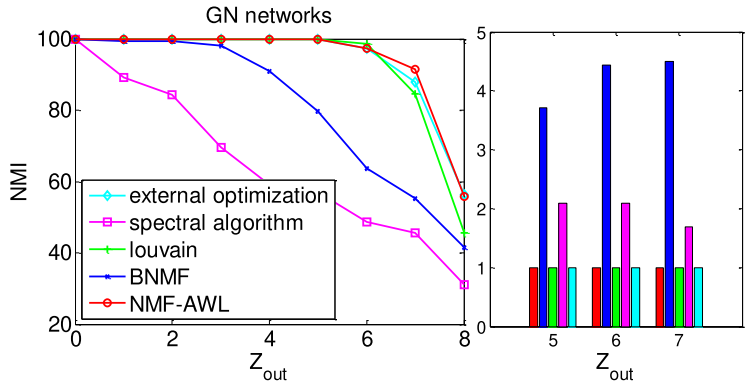


Fig. 3. Comparison of NMF-AWL and four existing methods on GN networks. The cyan, magenta, green, blue, and red curves (bars) in the left (right) panel are the results of external optimization, spectral clustering, leaven, Bayesian NMF, and NMF-AWL, respectively. The left panel shows the result on NMI; the larger the value shown, the better the result. The right panel shows the results on the accuracy of the number of detected communities according to Equation (19); the smaller the value shown, the better the result.

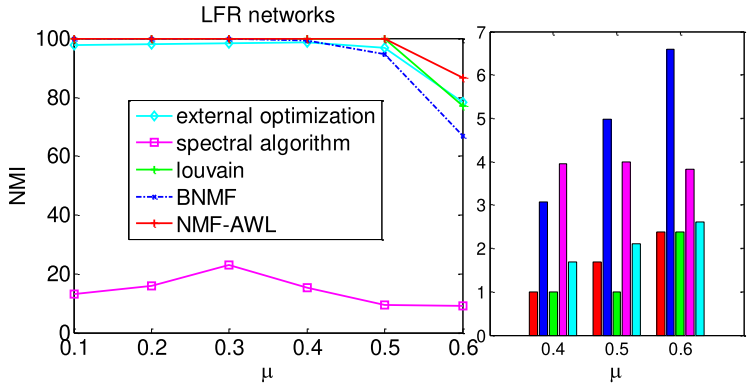


Fig. 4. Comparison of NMF-AWL and four existing methods on LFR networks. Comparison criteria are the same as that for Figure 3.

subfigure, we can find out that NMF-AWL outperforms all the other methods, especially on networks with large μ. These two experiments illustrate that NMF-AWL outperforms other methods on synthetic networks.

5.2 Effect of Parameters

To assess the effect of the two parameters, the base b of the logarithm function and α in Equation (9), on result quality, we experimented on the LFR networks as they capture many features of real networks. In addition, we tested whether it is helpful to set the diagonal elements of the adjacency matrix as the degree of corresponding nodes. The results (Figure 5) show that most combinations of α and b can yield satisfactory results when μ is small; the best performance is achieved when b is near e, the base for natural logarithm; and if we set the diagonal elements to be the degree of corresponding nodes, then the best performance is achieved with $\alpha = 1$. Thus, we follows these results by using $\alpha = 1$ on real-world networks. The effect of α on dividing network into different scale communities in real-world networks is shown in Section 5.4 and Figure 9.

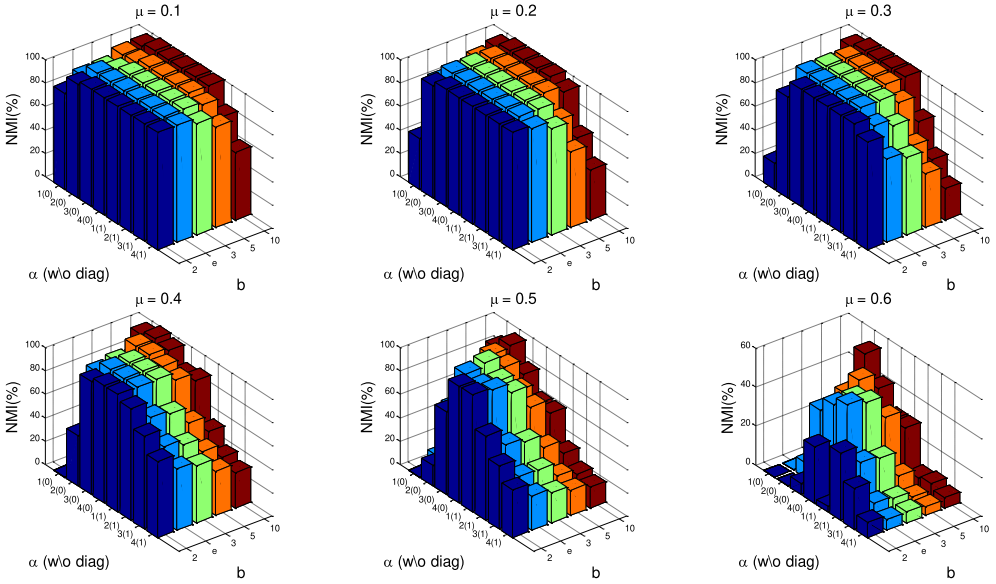


Fig. 5. The effects of parameters α and b on LFR networks. We show the results on NMI (the z axes) of different combinations of α and b. The results show that most combinations of α and b can yield satisfactory results when μ is small; the best performance is achieved when b is near e, the base for natural logarithm.

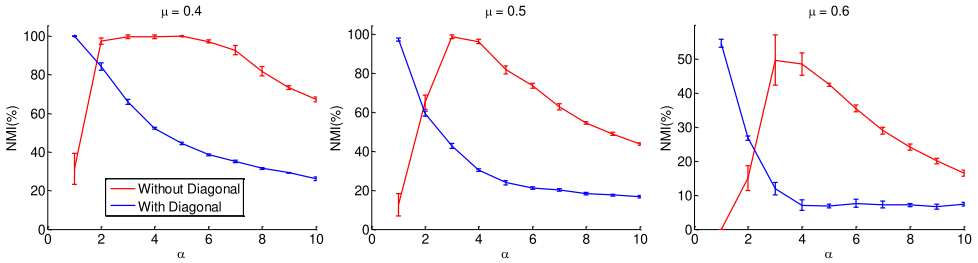


Fig. 6. The slices of the 3D histogram in Figure 5 with μ from 0.4 to 0.6 by setting $b = e$. If we set the diagonal elements to be the degree of corresponding nodes, then the best performance is achieved with $\alpha = 1$.

In Algorithm 1, we set the initial number of columns as half of the number of nodes, i.e., $p = m/2$. However, this initial number does not significantly affect the performance and the number of final detected communities, which fully illustrates the robustness of our framework. To verify this independence, we adopted different proportions of initial columns (from m, $m/2$, to $m/10$) on six real-world networks. The results are shown in Figure 8. With the changes of the proportion, the performances (NMI) on all of the networks change slightly, and the number of detected communities on most of the networks does not change except the Football network. This is because the Football network consists of 115 nodes that form 12 communities. With the decrease in the proportion, the initial number of columns is smaller than the actual number of communities, and it is the target number of communities to be detected.

5.3 Framework Verification

To obtain the number of communities, the model is designed to assign different penalty parameters σ_t to different columns of the factorized matrices and suppress some of them to zero. To verify

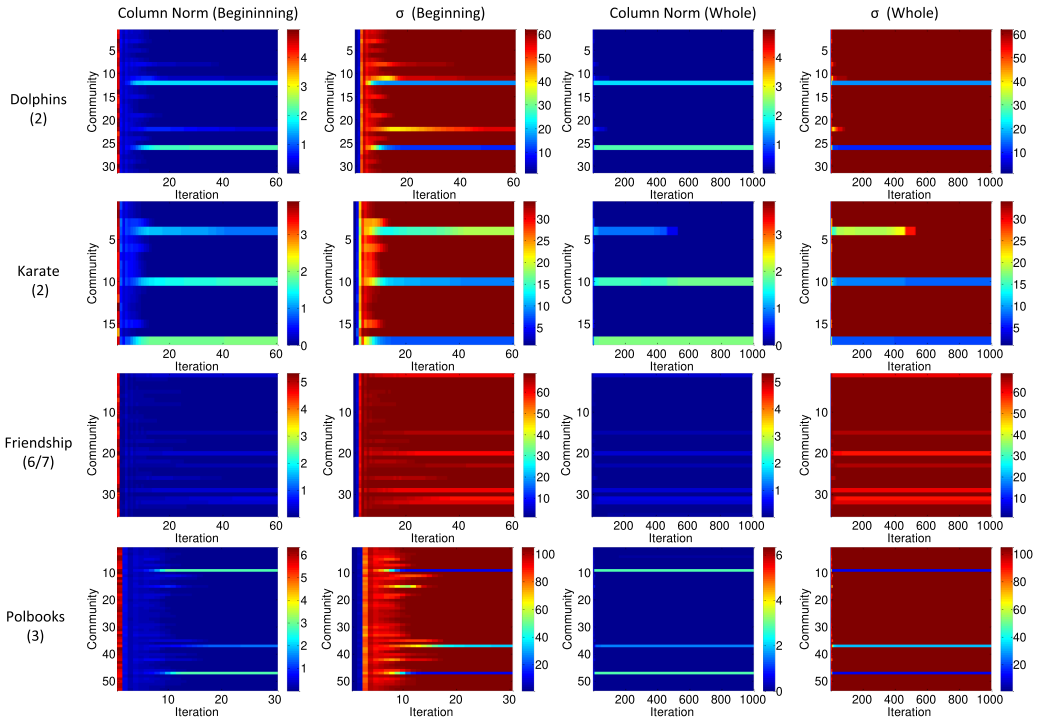


Fig. 7. The changes of column norms and σ's over iterations. The four rows are the results on Dolphins, Karate, Friendship and Polbooks networks, respectively. The first and second columns are the changes over the beginning few iterations, while the third and fourth ones are those over the whole iterations. Herein, each row of the subfigures in the first and third columns represents one column of factorized matrices $\mathbf{U}$, while that in the second and fourth columns represents the σ corresponding to the column. The color of each element of the subfigures in the first and third columns denotes the value of column norm, while that in the second and fourth columns denotes the value of σ. We can find out that the results are consistent with our expectations.

this procedure, we conducted experiments on the real-world networks, and the results on four networks (one row for one network) are shown in Figure 7. The figure shows the changes of column norms and σ's over iterations. The third and fourth columns show the changes over the whole iterations, while the first and second columns provide some details over the initial few iterations. The color of each element in the sub-figure represents the value of the factorized matrix column norm (the first and third columns) and σ (the second and fourth columns).

It is evident that the results are consistent with our model. Most of the σ's become large (the dark red in the second and fourth columns), and only a few columns become relatively small (the blue or bright red in the second and fourth columns) at the end of the iteration. Meanwhile, the norms of the columns corresponding to large σ's are suppressed to zero (the dark blue in the first and third columns). The remaining non-zero columns are then the detected communities. Note that this process doesn't correspond to the hierarchical community detection, since neither σ's nor column norms are monotonous as shown in Figure 7 (colors are not gradually varied). This means that communities don't hierarchical merge but simultaneously merge and split during the iterations. Taking the Polbooks network as an example, there are 105 nodes in the networks, and we set the initial number of columns to 53 ($\lceil 105/2 \rceil$). In the first iteration, we set all the σ's and

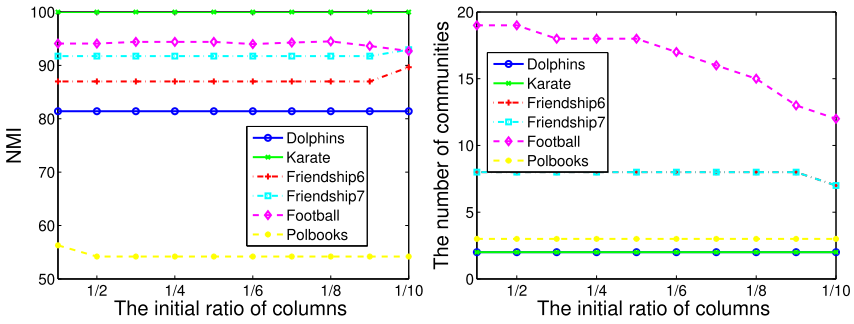


Fig. 8. The impact of the initial number of columns p on performance (NMI) (a) and the number of detected communities (b). The x-axis indicates the initial ratio of columns. With the decrease in the proportion, the changes in performance and the number of detected communities are very slight except the Football network.

Table 2. NMI(%) and the Number of Detected Communities on Real-world Networks

Datasets	Bayesian NMF	Louvain	Spectral Algorithm	EO	NMF-AWL
Dolphins	37.41 (16)	51.62 (5)	75.32 (2)	57.92 (4)	**81.41 (2)**
Karate	47.48 (9)	58.66 (4)	**100.00 (2)**	58.66 (4)	**100.00 (2)**
Friendship6	79.16 (12)	85.18 (7)	41.76 (2)	**95.21 (6)**	86.99 (8)
Friendship7	83.80 (12)	87.84 (7)	47.73 (2)	90.99 (6)	**91.79 (8)**
Polbooks	39.84 (15)	57.44 (3)	**59.79 (2)**	55.60 (5)	54.20 (3)
Football	93.63 (13)	89.03 (10)	33.35 (2)	88.48 (10)	**93.83 (14)**
Polblogs	23.30 (383)	**37.52 (276)**	18.76 (13)	19.00 (14)	36.29 (298)
Cora	42.01 (227)	25.96 (104)	29.49 (141)	44.07 (134)	**46.72 (163)**
Citeseer	33.19 (538)	24.38 (468)	27.70 (518)	32.75 (486)	**34.24 (654)**
Syracuse	22.92(36)	19.32(45)	20.28(69)	21.83(51)	**29.22(21)**
NYU	40.51(53)	23.99(99)	36.17(133)	41.62(109)	**46.82(9)**

the norms of all the columns to 1. After the third iteration, most of the σ's begin to increase, and the corresponding columns are suppressed to zero. After the twentieth iteration, only three σ's less than 50, while others reach 100. At the same time, the columns corresponding to large σ's are zero. Thus, the number of final detected communities is three, and the non-zero columns are used to divide the network into three communities. This result adequately verifies our model.

5.4 Real-world Networks

We considered nine widely used real-world networks, listed in the first nine rows of Table 1. Herein, m, n, and k are the number of vertices, edges and communities, respectively. In our experiments, we used $\alpha = 1$ on most of the networks except Football. We set $\alpha = 2$ on Football network, since its community size is very small (about 10 nodes on average).

Among the 11 networks tested, NMF-AWL achieves the best performance (in bold) on 8 networks (Dolphins, Karate, Friendship7, Football, Cora, Citeseer, Syracuse, and NYU) and the second best performance (with underlined) on 2 networks (Friendship6 and Polblogs) (Table 2). In comparison, BNMF, the other method that used NMF, performed poorly on all the networks except Football. Although the external optimization algorithm also obtained some satisfactory results on 6 networks (Friendship, Polbooks, Football, Cora, Citesseer, and NYU), its performance was slightly lower than NMF-AWL on these networks, but performed poorly on Dolphins, Karate and Polblogs networks.

The other methods were not consistent on the 11 networks. Louvain only obtained good results on Friendship, Polbooks, Football and Polblogs networks, and the spectral algorithm only achieved reasonable results on three networks (Dolphins, Karate, and Polbooks). Among the 11 real networks, the results on the two largest networks, Syracuse and NYU, are the most illustrative and convincing ones. Specifically, our proposed NMF-AWL has significantly outperformed the other state-of-the-art methods for the community structure detection task by 27.5% ((29.22–22.92)/22.92) and 12.5% ((46.82–41.62)/41.62) on Syracuse and NYU, respectively, due to the accurate determination of the number of communities by NMF-AWL as shown in the brackets of Table 2. Some of the illustrative examples of the detected communities are shown in Figures 9(a), 9(d), and 9(g).

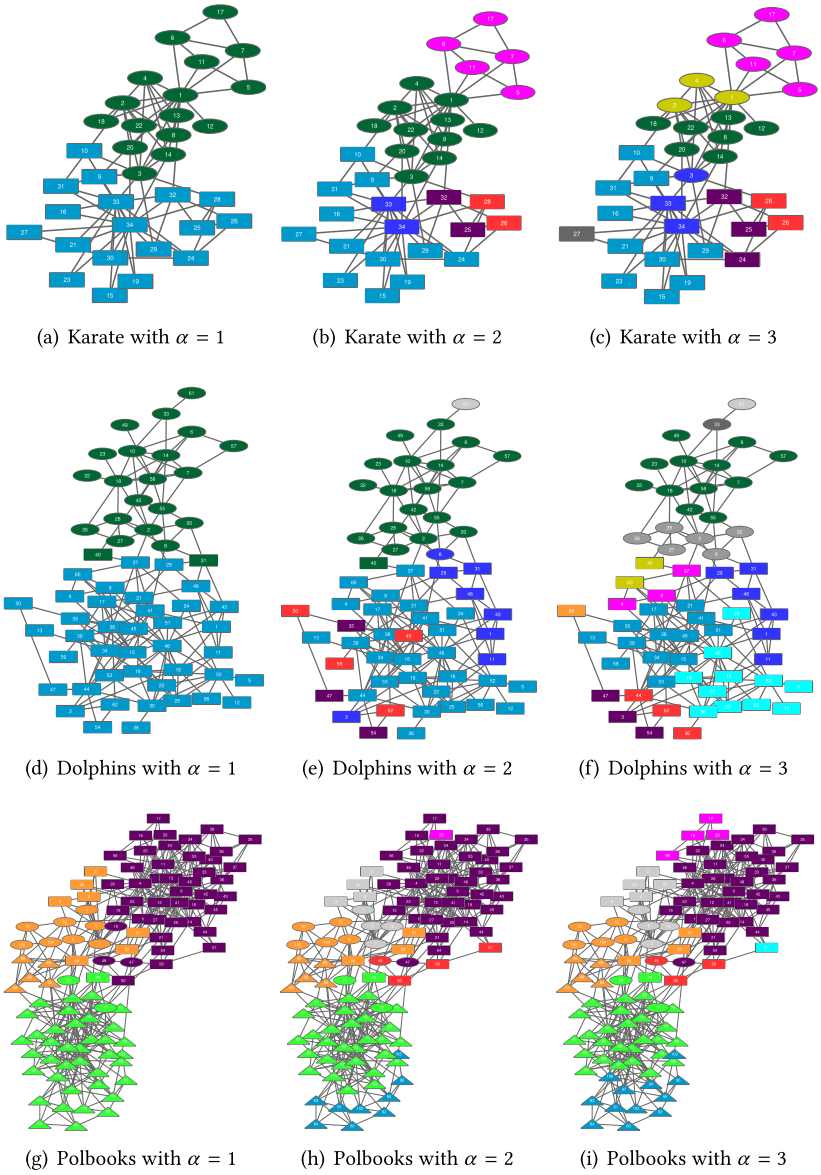
As illustrated in Section 4.2, a smaller α makes σ's larger, so that more columns will be suppressed to zero, resulting in a smaller number of communities, and vice versa. To appreciate the effect of α on dividing network into different scale communities, we conducted experiments by, respectively, setting $\alpha = 1$, 2, and 3 on Karate, Dolphins, and Polbooks networks. The results are shown in Figure 9. As shown, with increase of α, some communities are divided into smaller sub-communities. For example, on Karate network, the green community in Figure 9(a), is divided into two sub-communities, i.e., the new green sub-community and purple sub-community in Figure 9(b) by increasing α from 1 to 2, and the new green sub-community is further partitioned into two much smaller sub-communities, i.e., the yellow community and the new green community in Figure 9(c) by increasing α to 3. The same observation can be made on the other two networks. Therefore, we consider α as a parameter to control the scale of the communities to be detected.

Moreover, we compared these methods on the word association network [26], where a node represents a word and a link an association between two words. We use the enrichment of vertex pair similarity [2]—the average metadata similarity between all pairs of vertices that share a community divided by the average metadata similarity between all pairs of vertices—to evaluate the quality of detected communities. The higher an enrichment, the better the result. The enrichment of the result from NMF-AWL is 2.53, which is larger than that of BNMF (2.47) and the spectral algorithm (1.22). The results showed that NMF-AWL outperformed the other methods on this large real network.

6 CONCLUSION AND FUTURE DISCUSSION

We proposed a novel nonnegative matrix factorization approach for autonomous semantic community detection, namely, NMF-AWL. Two key ideas made the new method effective and efficient. The first is a weighted group-sparse low-rank regularization to help decompose a given network into components. The second is an adaptive, optimization scheme to learn the correct number of components from the data by removing some of the initial components. To the best of our knowledge, this is the first NMF-based approach to directly derive the underlying low-rank representation for networks without knowing the rank. Compared with previous low-rank approximation methods that directly constrain the underlying data with uniformly weighted low-rank regularization, this is the first time to introduce an adaptively weighted low-rank regularization to NMF to community detection. Extensive experiments on both synthetic and real-world networks have shown the superior performance of NMF-AWL over four state-of-the-art approaches, showing NMF-AWL's superior performance on real networks. More importantly, NMF-AWL is readily extensible to other NMF-based methods and applications on asymmetric data representation, e.g., NMF-based classification and clustering on non-network data.

We will pursue several lines of future research. One area is application of NMF-AWL to study various complex network community structures, including hierarchical, overlapping, and dynamic community structures. The second area is to extend the NMF-AWL method to many variants of NMF, such as constrained NMF, structured NMF, semi-NMF, and nonnegative matrix



(a) Karate with $\alpha = 1$ (b) Karate with $\alpha = 2$ (c) Karate with $\alpha = 3$

(d) Dolphins with $\alpha = 1$ (e) Dolphins with $\alpha = 2$ (f) Dolphins with $\alpha = 3$

(g) Polbooks with $\alpha = 1$ (h) Polbooks with $\alpha = 2$ (i) Polbooks with $\alpha = 3$

Fig. 9. The effects of parameter α on real-world networks. The first, second, and third rows are the results on Karate, Dolphins and Polbooks networks, respectively. And the first, second, and third columns are the result with $\alpha = 1,\ 2$, and 3, respectively. In each plot, the shapes represent the ground-truth communities, while the colors represent the estimated communities. We can find out that as α grows, the number of communities increases and many communities are divided into smaller sub-communities. In the first column, the inconsistency between color and shape indicates the wrong prediction. For example, all nodes are correctly classified in subfigure (a), while the green rectangle (nodes 31 and 40) indicates the wrong prediction in subfigure (d) .

tri-factorization (NMTF), to develop a unified approach for rank determination under the NMF framework. The third area is to adapt the adaptively weighted low-rank regularization for feature selection as it is able to selectively boost and/or suppressed some columns (features) in NMF. These attempts will benefit many problems in affective computing, including emotion recognition and prediction [9, 53–59].

REFERENCES

[1] Lada A. Adamic and Natalie Glance. 2005. The political blogosphere and the 2004 U.S. election. In *Proceedings of the 3rd International Workshop on Link Discovery*. ACM, 36–43.

[2] Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann. 2010. Link communities reveal multiscale complexity in networks. *Nature* 466 (2010), 761–764.

[3] H. Akaike. 1974. A new look at the statistical model identification. *IEEE Trans. Automat. Control* 19, 6 (Dec. 1974), 716–723. DOI: https://doi.org/10.1109/TAC.1974.1100705

[4] David M. Blei, Perry R. Cook, and Matthew Hoffman. 2010. Bayesian nonparametric matrix factorization for recorded music. In *Proceedings of the 27th International Conference on Machine Learning (ICML'10)*. Omnipress, 439–446. Retrieved from http://www.icml2010.org/papers/523.pdf.

[5] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *J. Stat. Mech.: Theory Exp.* 2008, 10 (2008), P10008. http://stacks.iop.org/1742-5468/2008/i=10/a=P10008.

[6] Emmanuel J. Candès, Xiaodong Li, Yi Ma, and John Wright. 2011. Robust principal component analysis? *J. ACM* 58, 3, Article 11 (June 2011), 37 pages. DOI: https://doi.org/10.1145/1970392.1970395

[7] Ali Taylan Cemgil. 2009. Bayesian inference for nonnegative matrix factorisation models. *Intell. Neurosci.* 2009, Article 4 (Jan. 2009), 17 pages. DOI: https://doi.org/10.1155/2009/785152

[8] James W. Demmel, Michael T. Heath, and Henk A. Van Der Vorst. 1993. Parallel numerical linear algebra. *Acta Numerica* 2 (1993), 111–197.

[9] Guiguang Ding, Wenshuo Chen, Sicheng Zhao, Jungong Han, and Qiaoyan Liu. 2018. Real-time scalable visual tracking via quadrangle kernelized correlation filters. *IEEE Trans. Intell. Transport. Syst.* 19, 1 (2018), 140–150. DOI: https://doi.org/10.1109/TITS.2017.2774778

[10] Jordi Duch and Alex Arenas. 2005. Community detection in complex networks using extremal optimization. *Phys. Rev. E* 72 (Aug. 2005), 027104. Issue 2. DOI: https://doi.org/10.1103/PhysRevE.72.027104

[11] Santo Fortunato. 2010. Community detection in graphs. *Phys. Rep.* 486, 3 (2010), 75–174.

[12] Santo Fortunato and Darko Hric. 2016. Community detection in networks: A user guide. *Phys. Rep.* 659 (2016), 1–44. DOI: https://doi.org/10.1016/j.physrep.2016.09.002 Community detection in networks: A user guide.

[13] Michelle Girvan and Mark E. J. Newman. 2002. Community structure in social and biological networks. *Proc. Natl. Acad. Sci. U.S.A.* 99, 12 (2002), 7821–7826.

[14] Keith Henderson, Brian Gallagher, Tina Eliassi-Rad, Hanghang Tong, Sugato Basu, Leman Akoglu, Danai Koutra, Christos Faloutsos, and Lei Li. 2012. RolX: Structural role extraction & mining in large graphs. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'12)*. 1231–1239. DOI: https://doi.org/10.1145/2339530.2339723

[15] Brian Karrer and M. E. J. Newman. 2011. Stochastic blockmodels and community structure in networks. *Phys. Rev. E* 83 (Jan 2011), 016107. Issue 1. DOI: https://doi.org/10.1103/PhysRevE.83.016107

[16] Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. 2008. Benchmark graphs for testing community detection algorithms. *Phys. Rev. E* 78, 4 (2008), 046110.

[17] Daniel D. Lee and H. Sebastian Seung. 1999. Learning the parts of objects by non-negative matrix factorization. *Nature* 401, 6755 (1999), 788–791.

[18] Daniel D. Lee and H. Sebastian Seung. 2000. Algorithms for non-negative matrix factorization. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS'00)*. 556–562.

[19] Chao Liu, Hung-chih Yang, Jinliang Fan, Li-Wei He, and Yi-Min Wang. 2010. Distributed nonnegative matrix factorization for web-scale dyadic data analysis on mapreduce. In *Proceedings of the 19th International Conference on World Wide Web*. ACM, New York, NY, 681–690. DOI: https://doi.org/10.1145/1772690.1772760

[20] Renquan Lu, Wenwu Yu, Jinhu Lu, and Anke Xue. 2014. Synchronization on complex networks of networks. *IEEE Trans. Neural Netw. Learn. Syst.* 25, 11 (Nov. 2014), 2110–2118. DOI: https://doi.org/10.1109/TNNLS.2014.2305443

[21] David Lusseau and Mark E. J. Newman. 2004. Identifying the role that animals play in their social networks. *Proc. Roy. Soc. London. Series B: Biol. Sci.* 271, S 6 (2004), 477–481.

[22] Fragkiskos D. Malliaros and Michalis Vazirgiannis. 2013. Clustering and community detection in directed networks: A survey. *Phys. Rep.* 533, 4 (2013), 95–142.

[23] Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. 2010. Spectral regularization algorithms for learning large incomplete matrices. *J. Mach. Learn. Res.* 11 (Aug. 2010), 2287–2322. Retrieved from http://dl.acm.org/citation.cfm?id=1756006.1859931.

[24] M. Morup and L. K. Hansen. 2009. Tuning pruning in sparse non-negative matrix factorization. In *Proceedings of the 17th European Signal Processing Conference*. 1923–1927.

[25] Morten Morup and Lars Kai Hansen. 2009. Automatic relevance determination for multi-way models. *J. Chemometr.* 23, 7–8 (2009), 352–363. DOI: https://doi.org/10.1002/cem.1223

[26] D. L. Nelson, McEvoy, C. L., and T. A. Schreiber. 1998. The University of South Florida word association, rhyme, and word fragment norms. Retrieved from http://w3.usf.edu/FreeAssociation/.

[27] Mark E. J. Newman. 2006. Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev. E* 74, 3 (2006), 036104.

[28] Mark E. J. Newman. 2006. Modularity and community structure in networks. *Proc. Natl. Acad. Sci. U.S.A.* 103, 23 (2006), 8577–8582.

[29] M. E. J. Newman. 2013. Spectral methods for community detection and graph partitioning. *Phys. Rev. E* 88 (Oct 2013), 042822. Issue 4. DOI: https://doi.org/10.1103/PhysRevE.88.042822

[30] Ioannis Psorakis, Stephen Roberts, Mark Ebden, and Ben Sheldon. 2011. Overlapping community detection using bayesian non-negative matrix factorization. *Phys. Rev. E* 83, 6 (2011), 066114.

[31] Jiahu Qin, Huijun Gao, and Wei Xing Zheng. 2015. Exponential synchronization of complex networks of linear systems and nonlinear oscillators: A unified analysis. *IEEE Trans. Neural Netw. Learn. Syst.* 26, 3 (Mar. 2015), 510–521. DOI: https://doi.org/10.1109/TNNLS.2014.2316245

[32] Nurlaila Rosli, Nordiana Rajaee, and David Bong. 2016. Non negative matrix factorization for music emotion classification. In *Advances in Machine Learning and Signal Processing*, Ping Jack Soh, Wai Lok Woo, Hamzah Asyrani Sulaiman, Mohd Azlishah Othman, and Mohd Shakir Saat (Eds.). Springer International Publishing, Cham, 175–185.

[33] Martin Rosvall and Carl T. Bergstrom. 2008. Maps of random walks on complex networks reveal community structure. *Proc. Natl. Acad. Sci. U.S.A.* 105, 4 (2008), 1118–1123. DOI: https://doi.org/10.1073/pnas.0706851105 arXiv: https://www.pnas.org/content/105/4/1118.full.pdf.

[34] Gideon Schwarz. 1978. Estimating the dimension of a model. *Ann. Stat.* 6, 2 (3 1978), 461–464. DOI: https://doi.org/10.1214/aos/1176344136

[35] Bo Shen, Zidong Wang, Derui Ding, and Huisheng Shu. 2013. H_∞ state estimation for complex networks with uncertain inner coupling and incomplete measurements. *IEEE Trans. Neural Netw. Learn. Syst.* 24, 12 (Dec. 2013), 2027–2037. DOI: https://doi.org/10.1109/TNNLS.2013.2271357

[36] T. C. Silva and Liang Zhao. 2012. Stochastic competitive learning in complex networks. *IEEE Trans. Neural Netw. Learn. Syst.* 23, 3 (Mar. 2012), 385–398. DOI: https://doi.org/10.1109/TNNLS.2011.2181866

[37] Paris Smaragdis and Judith C Brown. 2003. Non-negative matrix factorization for polyphonic music transcription. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. IEEE, 177–180.

[38] P. Song, S. Ou, W. Zheng, Y. Jin, and L. Zhao. 2016. Speech emotion recognition using transfer non-negative matrix factorization. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'16)*. 5180–5184. DOI: https://doi.org/10.1109/ICASSP.2016.7472665

[39] Alexander Strehl and Joydeep Ghosh. 2003. Cluster ensembles—A knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.* 3 (2003), 583–617.

[40] V. Y. F. Tan and C. Fevotte. 2013. Automatic relevance determination in nonnegative matrix factorization with the /spl beta/-Divergence. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 7 (2013), 1592–1605. DOI: https://doi.org/10.1109/TPAMI.2012.240

[41] Amanda L. Traud, Peter J. Mucha, and Mason A. Porter. 2012. Social structure of facebook networks. *Phys. A: Stat. Mech.anics and its Appl.* 391, 16 (2012), 4165–4180. DOI: https://doi.org/10.1016/j.physa.2011.12.021

[42] Dingding Wang, Tao Li, Shenghuo Zhu, and Chris Ding. 2008. Multi-document summarization via sentence-level semantic analysis and symmetric matrix factorization. In *Proceedings of the 31st Annual International ACM Special Interest Group on Information Retrieval (SIGIR'08)*. ACM, 307–314.

[43] Fei Wang, Tao Li, Xin Wang, Shenghuo Zhu, and Chris Ding. 2011. Community discovery using nonnegative matrix factorization. *Data Min. Knowl. Discov.* 22, 3 (May 2011), 493–521. DOI: https://doi.org/10.1007/s10618-010-0181-y

[44] Yu-Xiong Wang and Yu-Jin Zhang. 2013. Nonnegative matrix factorization: A comprehensive review. *IEEE Trans. Knowl. Data Eng.* 25, 6 (June 2013), 1336–1353. DOI: https://doi.org/10.1109/TKDE.2012.51

[45] Jierui Xie, Stephen Kelley, and Boleslaw K. Szymanski. 2013. Overlapping community detection in networks: The state-of-the-art and comparative study. *ACM Comput. Surveys* 45, 4 (2013), 43.

[46] Liang Yang, Xiaochun Cao, Dongxiao He, Chuan Wang, Xiao Wang, and Weixiong Zhang. 2016. Modularity-based community detection with deep learning. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI'16)*. 2252–2258. http://www.ijcai.org/Abstract/16/321

[47] Liang Yang, Xiaochun Cao, Di Jin, Xiao Wang, and Dan Meng. 2015. A unified semi-supervised community detection framework using latent space graph regularization. *IEEE Trans. Cybernet.* 45, 11 (2015), 2585–2598. DOI: https://doi.org/10.1109/TCYB.2014.2377154

[48] Liang Yang, Meng Ge, Di Jin, Dongxiao He, Huazhu Fu, Jing Wang, and Xiaochun Cao. 2017. Exploring the roles of cannot-link constraint in community detection via multi-variance mixed gaussian generative model. *PloS One* 12, 7 (2017), e0178029.

[49] Liang Yang, Di Jin, Dongxiao He, Huazhu Fu, Xiaochun Cao, and Francoise Fogelman-Soulie. 2017. Improving the efficiency and effectiveness of community detection via prior-induced equivalent super-network. *Sci. Rep.* 7, 1 (2017), 634.

[50] Liang Yang, Di Jin, Xiao Wang, and Xiaochun Cao. 2015. Active link selection for efficient semi-supervised community detection. *Sci. Rep.* 5, 1 (2015), 9039.

[51] Tianbao Yang, Rong Jin, Yun Chi, and Shenghuo Zhu. 2009. Combining link and content for community detection: A discriminative approach. In *Proceedings of the 15th ACM Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD'09)*. 927–936. DOI: https://doi.org/10.1145/1557019.1557120

[52] W. Zachary. 1977. An information flow modelfor conflict and fission in small groups1. *J. Anthropol. Res.* 33, 4 (1977), 452–473.

[53] Sicheng Zhao, Guiguang Ding, Yue Gao, and Jungong Han. 2017. Approximating discrete probability distribution of image emotions by multi-modal features fusion. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI'17)*. 4669–4675. DOI: https://doi.org/10.24963/ijcai.2017/651

[54] Sicheng Zhao, Guiguang Ding, Yue Gao, Xin Zhao, Youbao Tang, Jungong Han, Hongxun Yao, and Qingming Huang. 2018. Discrete probability distribution prediction of image emotions with shared sparse learning. *IEEE Trans. Affect. Comput.* (2018), 1–1. DOI: https://doi.org/10.1109/TAFFC.2018.2818685

[55] Sicheng Zhao, Yue Gao, Guiguang Ding, and Tat-Seng Chua. 2018. Real-time multimedia social event detection in microblog. *IEEE Trans. Cybernet.* 48, 11 (2018), 3218–3231. DOI: https://doi.org/10.1109/TCYB.2017.2762344

[56] Sicheng Zhao, Amir Gholaminejad, Guiguang Ding, Yue Gao, Jungong Han, and Kurt Keutzer. 2019. Personalized emotion recognition by personality-aware high-order learning of physiological signals. *ACM Trans. Multimedia Comput. Commun. Appl.* 15, 1s (2019), 14:1–14:18. DOI: https://doi.org/10.1145/3233184

[57] Sicheng Zhao, Hongxun Yao, Yue Gao, Guiguang Ding, and Tat-Seng Chua. 2018. Predicting personalized image emotion perceptions in social networks. *IEEE Trans. Affect. Comput.* 9, 4 (2018), 526–540. DOI: https://doi.org/10.1109/TAFFC.2016.2628787

[58] Sicheng Zhao, Hongxun Yao, Yue Gao, Rongrong Ji, and Guiguang Ding. 2017. Continuous probability distribution prediction of image emotions via multitask shared sparse regression. *IEEE Trans. Multimedia* 19, 3 (2017), 632–645. DOI: https://doi.org/10.1109/TMM.2016.2617741

[59] Sicheng Zhao, Xin Zhao, Guiguang Ding, and Kurt Keutzer. 2018. EmotionGAN: Unsupervised domain adaptation for learning discrete probability distributions of image emotions. In *Proceedings of the ACM Multimedia Conference on Multimedia Conference (MM'18)*. 1319–1327. DOI: https://doi.org/10.1145/3240508.3240591

[60] Mingjun Zhong and Mark Girolami. 2009. Reversible jump MCMC for non-negative matrix factorization. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*. 663–670.

Received January 2019; revised July 2019; accepted August 2019