Neural Computing and Applications (2024) 36:897–912
https://doi.org/10.1007/s00521-023-09066-y

ORIGINAL ARTICLE

Community detection in attributed networks via adaptive deep nonnegative matrix factorization

Junwei Cheng[1,2] · Yong Tang[1,2] · Chaobo He[1,2] · Kunlin Han[3] · Ying Li[1] · Jinhui Wei[1]

Received: 8 July 2022 / Accepted: 14 September 2023 / Published online: 5 October 2023

Abstract
Community detection plays an important role in analyzing attributed networks. It attempts to find the optimal cluster structures to identify valuable information. Although deep nonnegative matrix factorization (DNMF) is widely used in community detection, it cannot be used to analyze attributed networks since only topology information is considered. Recent researches have taken attribute information into account, but we still need to face the following challenges. First, it is difficult to deal with topology noise and attribute noise in attributed networks at one stroke. Second, we need to balance the coupling between topology and node attributes with hyperparameters in most methods. However, with inappropriate hyperparameters, it is easy to cause interference and compromise between them. For the above challenges, in this paper, we propose a novel method, namely adaptive deep nonnegative matrix factorization. Specifically, we handle the inherent noise of attributed networks via dual-DNMF with autoencoder. And then, we use the attention mechanism to adaptively integrate topology information and attribute information without adjusting hyperparameters manually. Overall, our method not only handles the inherent noise in attributed networks, but also resolves the interference and compromise between topology and attributes in a generalized way. The results of comprehensive experiments support our conclusions and demonstrate that our method outperforms the state-of-the-art methods in most datasets.

Keywords Deep nonnegative matrix factorization · Community detection · Attention mechanism · Attributed networks · Autoencoder

1 Introduction

Attributed networks are ubiquitous and widely used to represent the real world, such as social networks [1], biological networks [2] and citation networks [3]. Apparently, the research field of attributed networks contains many popular topics [4–6] and we focus on community detection among those topics in this paper.

Community detection plays a major role in mining attributed networks for valuable information. It usually assigns each node to only one cluster, making the correlations within clusters dense and the correlations between clusters sparse [7–9]. To this end, how to find the optimal cluster structures, that satisfy the above conditions, is a key issue. Notably, nonnegative matrix factorization (NMF)-based methods [10–12], graph neural networks (GNNs)-based methods [13–15] and evolutionary clustering-based methods [16, 17] have contributed to the development of community detection.

In recent years, NMF-based methods have attracted great attention for their high interpretability and extensibility [18]. Sun et al. [19] proposed a NMF-based method with nonnegative symmetric autoencoder. However, this method is limited by topology noise, making it difficult to play its advantage in community detection on attributed networks. Inspired by community characteristics, Wang et al. [20] proposed SCI method, which takes both topology and node attributes into consideration. Their experimental results demonstrate that the complementarity between

✉ Chaobo He
hechaobo@m.scnu.edu.cn

1 School of Computer Science, South China Normal University, Guangzhou 510631, China

2 Pazhou Lab, Guangzhou 510335, China

3 Computer Science Department, University of Southern California, Los Angeles 90089, USA

topology information and attribute information helps to improve the ability to detect optimal communities and compensate for the trouble caused by the linear operations of NMF [21].

Unfortunately, recent researches have shown that it is difficult to exploit topology and attribute information at one stroke. On the one hand, most methods cannot solve the noise problem of attributed networks. For example, Jin et al. [22] proposed a new strategy to address mismatches between topology-based clusters and attribute-based clusters. However, it still fails to address the inherent noise of topology and attribute information. On the other hand, the interference and compromise between topology and attributes make them difficult to be integrated. Specifically, topology tends to interfere with the node attributes [23]. This perturbation of topology leads to the loss of ability to express attributes, which eventually destroys the characteristics of clusters, i.e., clusters are no longer densely correlated internally. Moreover, attributes tend to distort the topology structures. For example, WSCDSM [24], ANEM_B and ANEM_M [25] need to balance the coupling between topology and attributes through hyperparameters. This is a tricky problem because the choice of hyperparameters is a difficult and irregular exploration process. So it is difficult for us to find suitable and consistent hyperparameters to detect communities of different attributed networks.

In view of the above, two questions naturally arise: *How to deal with the inherent noise of topology and attributes?* and *How to resolve the interference and compromise between topology and attributes in a generalized way?* In this paper, we propose a novel method, namely adaptive deep nonnegative matrix factorization (ADNMF), to answer the aforementioned problems. For the first question, inspired by characteristics of DNMF and hypothesis of homogeneity [26], we construct a K-nearest neighbor (KNN) graph and employ multilayer subspace of dual-DNMF with autoencoder to deal with the inherent noise of attributed networks. *Refineissuitable*? Especially, the encoder refines the hierarchical mapping and filters the inherent noise of attributed networks through dimension reduction. And the core information of networks will be preserved by decoder. For the other question, we focus on resolving the interference and compromise between topology and attributes. To this end, we propose an attention mechanism that integrates topology and attribute information in a generalized way without manual tuning of hyperparameters. In summary, the work of this paper can be summarized as:

- We propose a novel method, namely ADNMF, which not only deals with the inherent noise of topology and attributes, but also resolves the interference and compromise between them.
- We decouple the relation between topology and attributes via dual DNMF with autoencoder, which can mitigate their inherent noise. And we propose an attention mechanism to adaptively integrate topology and attribute information so that we no longer need to manually tune hyperparameters to balance their coupling relationships. Obviously, in this way, we can effectively resolve the interference and compromise between topology and attributes.
- We conduct sufficient experiments on a series of benchmark datasets, and results demonstrate that ADNMF is superior over the state-of-the-art community detection methods.

The rest of the paper is organized as follows. Section 2 introduces the related work of NMF-based and DNMF-based methods. Section 3 provides a detailed description of ADNMF. Section 4 presents related experiments and results on synthetic datasets and real attribute datasets. Finally, we draw conclusions in Sect. 5.

2 Related work

NMF plays an important role in community detection because it is more interpretable, simpler and more general than other types of methods [27, 28]. The traditional NMF was proposed by Lee and Seung [29]. It uses the product of two nonnegative matrices, i.e., the bias matrix and community membership matrix, to approximate the original nonnegative data. On the basis of their work, Sun et al. [19] proposed a NMF-based method with autoencoder. This is a groundbreaking design because the method successfully applies autoencoder to NMF and improves performance. Furthermore, Yuan et al. [30] believed that the community membership matrix should be sparse, since each node only belongs to one community. Thus, they proposed PNMF method, which maintains the sparsity of community membership matrix when updating the matrix. Wang et al. [31] observed networks from a mesoscopic perspective and proposed MNMF that preserves the node microstructure and cluster structures. Although various NMF-based community detection methods have been proposed and achieve effectiveness, they cannot overcome their limitations.

They are limited because NMF-based methods rely on the product of two nonnegative matrices to approximate the original nonnegative data. Obviously, methods that only employ simple linear operations and topology information have difficulty in dealing with complex networks. To this end, an effective way is to additionally consider attribute information and integrate it with topology information to

improve the performance. SCI considers the coupling of topology space and attribute space and balances the coupling relationship between them through hyperparameters. Unfortunately, the choice of hyperparameters is a tricky problem, leading us to explore only local optimum rather than global optimum. In addition, inappropriate hyperparameters can easily lead to the following problems. Firstly, each cluster has its own unique semantics, but the semantics of clusters become confusing when topology interferes with attributes. Secondly, attributes tend to distort the topology structures, i.e., clusters lose their community characteristics and become sparse and disorganized. The main reason is that inappropriate hyperparameters destroy the balance between topology structure and node attributes, forcing topology to compromise toward attributes. Huang et al. [32] used a shared bias matrix to establish coupling relationship between topology and attributes. But the coupling relationship between them still needs to be manually balanced by hyperparameters. In addition, Wang et al. [24], Li et al. [25] and Jin et al. [33] all used prior information to improve the performance of their methods. Unfortunately, they still encounter the same problem, i.e., how to choose appropriate hyperparameters. In recent years, Trigeorgis et al. were [34] inspired by deep learning and proposed deep nonnegative factorization matrix (DNMF), which effectively deals with topology noise through multilayer matrix mapping. After this, Ye et al. [35] proposed DNMF-based method with autoencoder and graph regularization, namely DANMF. Also, Huang et al. [36] proposed DNMF-based method with modularity. Although these methods promote the development of DNMF, they can only be used on topology networks but not on attributed networks because of their inherent limitation. In all, we can observe from previous researches that integrating topology and attribute information is the key to detect communities of attributed networks. But there are two problems to be solved: (1) How to deal with the inherent noise of topology and attributes. (2) How to resolve the interference and compromise between topology and node attributes in a generalized way. To this end, in this paper, we propose ADNMF to resolve these two problems.

3 ADNMF: the proposed method

In this section, we first introduce some notations that are used in this paper and outline the design ideas of ADNMF. Then, the proposed method ADNMF is described. Finally, we prove that the update rules of ADNMF are converging.

3.1 Notations

An attributed network can be defined as a non-directed graph $\mathcal{G} = (\mathbf{A}, \mathbf{X})$, where $\mathbf{A} \in \mathbb{R}^{n\times n}$ denotes the adjacency matrix with n nodes, and $\mathbf{A}_{ij} = 1$ represents the relationship between i and j, otherwise, $\mathbf{A}_{ij} = 0$. $\mathbf{X} \in \mathbb{R}^{n\times d}$ indicates the attribute matrix with d features. Assume that graph $\mathcal{G}$ can be divided into k communities, i.e., $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$, where $\mathcal{C} = \{C_i | C_i \neq \emptyset, C_i \neq C_j, i \neq j\}$. Suppose that non-negative matrices $\mathbf{U}_i \in \mathbb{R}_+^{n\times d_i}$ and $\mathbf{V}_i \in \mathbb{R}_+^{d_i\times n}$ correspond to bias matrix and community membership matrix of i-th layer, where d_i is dimension of i-th layer matrix.

3.2 Overview

The overall framework of ADNMF is shown in Fig. 1. In order to deal with the inherent noise of topology and attributes, an ingenious design is to split the attributed networks into two graphs, i.e., topology graph and KNN graph. Notably, in pairs of nodes, the similarity of attributes is positively correlated with the probability that they have edges. Therefore, we construct a KNN graph based on node attributes to initially filter some irrelevant node relations. Next, we use dual-DNMF with autoencoder to deal with the inherent noise of topology graph and KNN graph. In this way, the inherent noise of these two graphs will be filtered out during multilayer encoding. Obviously, each DNMF processes only one kind of information, which reduces the difficulty of dealing with noise. Finally, we can obtain two refined community membership matrices, i.e., $\mathbf{V}_t$ and $\mathbf{V}_k$, which contain topology information and attribute information, respectively. In general, most methods use hyperparameters to balance the coupling between these two community membership matrices, but in this way we will have a problem, i.e., topology interferes with the node attributes, and the attributes distort the topology. To this end, we employ a generalized way to resolve the interference and compromise between topology and attributes, i.e., attention mechanism, which means that our method no longer needs to manually balance coupling relationships. Finally, we find the optimal clusters in an end-to-end manner through the integrated matrix $\mathbf{Y}_f$.

3.3 DNMF module

3.3.1 K-nearest neighbor graph

To filter the inherent noise of node attributes, based on the homogeneity assumption, we construct a KNN graph $\mathcal{G}_k$. We use $\mathbf{A}_k \in \mathbb{R}^{n\times n}$ to represent the adjacency matrix of KNN graph, where $\mathbf{A}_{k,ij} = 1$ represents there is an edge between nodes i and j, otherwise, 0. Especially, in order to

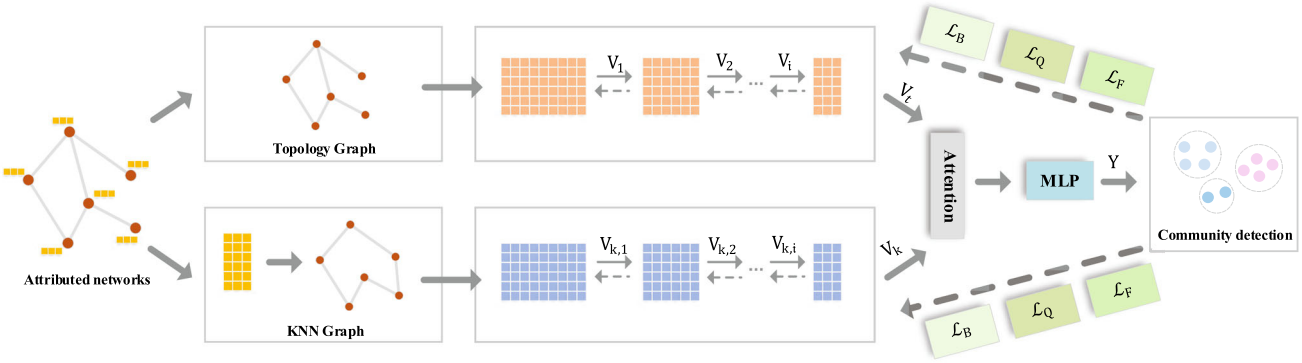


Fig. 1 The framework of ADNMF

construct $\mathbf{A}_k$, we first calculate the similarity matrix $\mathbf{S} \in \mathbb{R}^{n\times n}$ between n nodes via Cosine Similarity.

And then, we select top-k node pairs with the most similar attributes to compose $\mathbf{A}_k$. The cosine similarity formula is as follows:

$$S_{ij} = \frac{x_i \cdot x_j}{|x_i||x_j|} \tag{1}$$

where x_i and x_j are attribute vectors of nodes i and j, respectively.

3.3.2 DNMF encoder

Given a non-directed $\mathcal{G}$, traditional NMF uses the product of two nonnegative matrices, i.e., $\mathbf{U}$ and $\mathbf{V}$, to approximate the original adjacency matrix $\mathcal{A}$ (For convenience, we use $\mathcal{A}$ to refer to both $\mathbf{A}$ and $\mathbf{A}_k$). However, real attributed networks contain complexity, uncertainty and noise, which makes it difficult for many variants of NMF-based methods to detect optimal communities. As well known, deep learning is able to narrow the gap between the original core data and low-dimensional mapping data [37]. To this end, the DNMF encoder performs multilayer decomposition of $\mathcal{A}$, which not only extracts core hierarchy information, but also filters inherent noise. The adjacency matrix $\mathcal{A}$ is decomposed into $p+1$ nonnegative factor matrices as follows:

$$\mathcal{A} \approx \mathbf{U}_1\mathbf{U}_2\mathbf{U}_3 ... \mathbf{U}_p\mathbf{V}_p \tag{2}$$

where we use $\mathbf{V}_p$ to refer to both $\mathbf{V}_t$ and $\mathbf{V}_k$. The Eq. (2) is limited by the following conditions:

$$\begin{aligned} &\mathcal{A} \approx \mathbf{U}_1\mathbf{V}_1 \\ &\mathbf{V}_1 \approx \mathbf{U}_2\mathbf{V}_2 \\ &\cdots \\ &\mathbf{V}_{p-1} \approx \mathbf{U}_p\mathbf{V}_p \end{aligned} \tag{3}$$

To this end, this multilayer structure can extract the core information from each layer and filter out the inherent noise. Finally, the objective function can be written as:

$$\begin{aligned} &\min_{\mathbf{U}_i,\mathbf{V}_i} \mathcal{L}_e = \|\mathcal{A} - \mathbf{U}_1\mathbf{U}_2 \cdots \mathbf{U}_p\mathbf{V}_p\|_F^2 \\ &s.t.\ \mathbf{U}_i \geq 0,\ \mathbf{V}_i \geq 0,\ i = 1, 2, \ldots, p \end{aligned} \tag{4}$$

3.3.3 DNMF decoder

From the literature [19], we can know that the community membership matrix $\mathbf{V}_p$ needs to satisfy two conditions, i.e., nonnegativity and sparsity. Therefore, a simple yet efficient decoder is to use the product of two matrices, $\mathbf{U}$ and $\mathcal{A}$, to approximate the community membership matrix $\mathbf{V}_p$. To this end, we extend original decoder to a multilayer decomposition such that the community membership matrix $\mathbf{V}_p$ is decomposed into $p+1$ nonnegative factor matrices, as follows:

$$\mathbf{V}_p \approx \mathbf{U}_p^T\mathbf{U}_{p-1}^T\mathbf{U}_{P-2}^T ... \mathbf{U}_1^T\mathcal{A} \tag{5}$$

where Eq. (5) is limited by the following conditions:

$$\begin{aligned} &\mathbf{V}_p \approx \mathbf{U}_p^T\mathbf{V}_{p-1} \\ &\mathbf{V}_{p-1} \approx \mathbf{U}_{p-1}^T\mathbf{V}_{p-2} \\ &\cdots \\ &\mathbf{V}_1 \approx \mathbf{U}_1^T\mathcal{A} \end{aligned} \tag{6}$$

To this end, the community membership matrix not only satisfies nonnegativity and sparsity, but also retains core information of graph. Finally, the objective function can be written as:

$$\begin{aligned} &\min_{\mathbf{U}_i,\mathbf{V}_i} \mathcal{L}_d = \|\mathbf{V}_p - \mathbf{U}_p^T\mathbf{U}_{p-1}^T \cdots \mathbf{U}_1^T\mathcal{A}\|_F^2 \\ &s.t.\ \mathbf{U}_i \geq 0,\ \mathbf{V}_i \geq 0,\ i = 1, 2, \ldots, p \end{aligned} \tag{7}$$

By combining Eqs. (4) and (7), we can obtain the unified objective function suitable for topology graph and KNN graph, as follows:

$$\begin{aligned}\min_{\mathbf{U}_i,\mathbf{V}_i}\mathcal{L}&=\mathcal{L}_e+\mathcal{L}_d=\|\mathcal{A}-\mathbf{U}_1\mathbf{U}_2\cdots\mathbf{U}_p\mathbf{V}_p\|_F^2\\&+\|\mathbf{V}_p-\mathbf{U}_p^T\mathbf{U}_{p-1}^T\cdots\mathbf{U}_1^T\mathcal{A}\|_F^2\\ s.t.\ &\mathbf{U}_i\geq 0,\ \mathbf{V}_i\geq 0,\ i=1,2,\ldots,p\end{aligned}\tag{8}$$

3.3.4 Optimization

We rewrite Eq. (8) in the form of a trace so that we can update various variables and optimize the objective function, as follows:

$$\begin{aligned}\mathcal{L}=&tr(\mathcal{A}^T\mathcal{A}-2\mathcal{A}^T\Phi_{i-1}\mathbf{U}_i\Psi_{i+1}\mathbf{V}_p\\&+\mathbf{V}_p^T\Psi_{i+1}^T\mathbf{U}_i^T\Phi_{i-1}^T\Phi_{i-1}\mathbf{U}_i\Psi_{i+1}\mathbf{V}_p)\\&+tr(\mathbf{V}_p^T\mathbf{V}_p-2\mathbf{V}_p^T\Psi_{i+1}^T\mathbf{U}_i^T\Phi_{i-1}^T\mathcal{A}\\&+\mathcal{A}^T\Phi_{i-1}\mathbf{U}_i\Psi_{i+1}\Psi_{i+1}^T\mathbf{U}_i^T\Phi_{i-1}^T\mathcal{A})\end{aligned}\tag{9}$$

where $\Phi_{i-1}=\mathbf{U}_1\mathbf{U}_2\cdots\mathbf{U}_{i-1}$ and $\Psi_{i+1}=\mathbf{U}_{i+1}\mathbf{U}_{i+2}\cdots\mathbf{U}_p$.

By fixing all the variables except $\mathbf{U}_i$, the partial derivative of Eq. (9) is:

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\mathbf{U}_i}=&-4\Phi_{i-1}^T\mathcal{A}\mathbf{V}_p^T\Psi_{i+1}^T+2\Phi_{i-1}^T\Phi_{i-1}\mathbf{U}_i\Psi_{i+1}\mathbf{V}_p\mathbf{V}_p^T\Psi_{i+1}^T\\&+2\Phi_{i-1}^T\mathcal{A}\mathcal{A}^T\Phi_{i-1}\mathbf{U}_i\Psi_{i+1}\Psi_{i+1}^T\end{aligned}\tag{10}$$

By fixing all the variables except $\mathbf{V}_i$, the partial derivative of Eq. (9) is:

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\mathbf{V}_i}=&-4\Psi_{i+1}^T\mathbf{U}_i^T\Phi_{i-1}^T\mathcal{A}+2\mathbf{V}_i\\&+2\Psi_{i+1}^T\mathbf{U}_i^T\Phi_{i-1}^T\Phi_{i-1}\mathbf{U}_i\Psi_{i+1}\mathbf{V}_i\end{aligned}\tag{11}$$

To satisfy the Karush–Kuhn–Tucker optimization condition, we let Eq. (10) = 0 and Eq. (11) = 0 to get the following update rules:

$$\mathbf{U}_i=\mathbf{U}_i\frac{2\Phi_{i-1}^T\mathcal{A}\mathbf{V}_p^T\Psi_{i+1}^T}{\Phi_{i-1}^T\Phi_{i-1}\mathbf{U}_i\Psi_{i+1}\mathbf{V}_p\mathbf{V}_p^T\Psi_{i+1}^T+\Phi_{i-1}^T\mathcal{A}\mathcal{A}^T\Phi_{i-1}\mathbf{U}_i\Psi_{i+1}\Psi_{i+1}^T}\tag{12}$$

$$\mathbf{V}_i=\mathbf{V}_i\frac{2\Psi_{i+1}^T\mathbf{U}_i^T\Phi_{i-1}^T\mathcal{A}}{\mathbf{V}_i+\Psi_{i+1}^T\mathbf{U}_i^T\Phi_{i-1}^T\Phi_{i-1}\mathbf{U}_i\Psi_{i+1}\mathbf{V}_i}\tag{13}$$

3.3.5 Convergence analysis

Theorem 1 *The Eqs. (12) and (13) guarantee that the objective function $\mathcal{L}$ is monotonically decreasing.*

We note that the convergence proofs for $\mathbf{U}_i$ and $\mathbf{V}_i$ are similar. For convenience, we take the update rule of $\mathbf{U}_i$ as an example to prove that $\mathcal{L}$ is monotonically decreasing under the update rule of $\mathbf{U}_i$. To this end, we can prove this theorem is following the idea of the auxiliary function proposed in [38]. Firstly, we briefly introduce the definition of auxiliary function and a related lemma, as follows:

Definition 1 If a function satisfies $G(\mathbf{U},\mathbf{U}')\geq\mathcal{L}(\mathbf{U})$, then we regard this function $G(\mathbf{U},\mathbf{U}')$ as an auxiliary function of $\mathcal{L}(\mathbf{U})$.

Lemma 1 *If $G(\mathbf{U},\mathbf{U}')$ is an auxiliary function, then $\mathcal{L}(\mathbf{U})$ is monotonically decreasing under the update rule of $\mathbf{U}^{t+1}=\arg\min G(\mathbf{U},\mathbf{U}^t)$, where t and $t+1$ represent the current and next numerical calculations, respectively.*

Proof $\mathcal{L}(\mathbf{U}^{t+1})\leq G(\mathbf{U}^{t+1},\mathbf{U}^t)\leq G(\mathbf{U}^t,\mathbf{U}^t)=\mathcal{L}(\mathbf{U}^t)$ □

Note that $\mathcal{L}(\mathbf{U}^{t+1})=\mathcal{L}(\mathbf{U}^t)$ holds only if $\mathbf{U}^t$ is a local minimum of $G(\mathbf{U},\mathbf{U}^t)$. Thus, by iteratively updating $\mathbf{U}^t$, $\mathcal{L}(\mathbf{U})$ can converge to the local minimum $\mathbf{U}_{min}=\arg\min_{\mathbf{U}}\mathcal{L}(\mathbf{U})$. Thereby, we can obtain a sequence of estimates: $\mathcal{L}(\mathbf{U}_{min})\leq\ldots\leq\mathcal{L}(\mathbf{U}^{t+1})\leq\mathcal{L}(\mathbf{U}^t)\leq\ldots\leq\mathcal{L}(\mathbf{U}^1)\leq\mathcal{L}(\mathbf{U}^0)$

Now, we proof that the $\mathcal{L}$ is convergent under update rule of Eq. (12). For convenience of description, we hide the subscripts of some variables, which makes our proof idea looks clearer. Firstly, the function $\mathcal{L}_{u_{ab}}(u)$ denotes the part of unified objective function $\mathcal{L}$ related to u_{ab}, where $\forall u_{ab}\in\mathbf{U}$. We can get the first-order partial derivative of $\mathcal{L}$ with respect to u_{ab} as described in Eq. (10):

$$\begin{aligned}\mathcal{L}'_{u_{ab}}(u_{ab})=\frac{\partial\mathcal{L}}{\partial u_{ab}}=&-4(\Phi^T\mathcal{A}\mathbf{V}_p^T\Psi^T)_{ab}\\&+2(\Phi^T\Phi\mathbf{U}\Psi\mathbf{V}_p\mathbf{V}_p^T\Psi^T)_{ab}+2(\Phi^T\mathcal{A}\mathcal{A}^T\Phi\mathbf{U}\Psi\Psi^T)_{ab}\end{aligned}\tag{14}$$

Based on Eq. (14), we can calculate the second- and third-order partial derivatives of $\mathcal{L}$ with respect to u_{ab}, as follows:

$$\begin{aligned}\mathcal{L}''_{u_{ab}}(u_{ab})=&2(\Phi^T\Phi)_{aa}\mathcal{J}_{ab}(\Psi\mathbf{V}_p\mathbf{V}_p^T\Psi^T)_{bb}\\&+2(\Phi^T\mathcal{A}\mathcal{A}^T\Phi)_{aa}\mathcal{J}_{ab}(\Psi\Psi^T)_{bb}\end{aligned}\tag{15}$$

$$\mathcal{L}'''_{u_{ab}}(u_{ab})=0\tag{16}$$

where $\mathcal{J}_{ab}$ is single-entry matrix with 1 in coordinate (a,b) and 0 elsewhere. To this end, the Taylor expansion of $\mathcal{L}_{u_{ab}}$ at u_{ab} can be expressed as:

$$\begin{aligned}\mathcal{L}_{u_{ab}}(u) &= \mathcal{L}_{u_{ab}}(u_{ab}) + \mathcal{L}'_{u_{ab}}(u_{ab})(u-u_{ab}) \\ &\quad + \frac{1}{2}\mathcal{L}''_{u_{ab}}(u_{ab})(u-u_{ab})^2 \\ &= \mathcal{L}_{u_{ab}}(u^t_{ab}) + \mathcal{L}'_{u_{ab}}(u^t_{ab})(u-u^t_{ab}) \\ &\quad + [(\Phi^T\Phi)_{aa}\mathcal{J}_{ab}(\Psi\mathbf{V}_p\mathbf{V}_p^T\Psi^T)_{bb} \\ &\quad + (\Phi^T\mathcal{A}\mathcal{A}^T\Phi)_{aa}\mathcal{J}_{ab}(\Psi\Psi^T)_{bb}](u-u^t_{ab})^2\end{aligned} \tag{17}$$

Meanwhile, we can define a new function $G_{u_{ab}}(\cdot)$, as follows:

$$\begin{aligned}G_{u_{ab}}(u,u^t_{ab}) &= \mathcal{L}_{u_{ab}}(u^t_{ab}) + \mathcal{L}'_{u_{ab}}(u^t_{ab})(u-u^t_{ab}) \\ &\quad + \frac{(\Phi^T\Phi\mathbf{U}\Psi\mathbf{V}_p\mathbf{V}_p^T\Psi^T + \Phi^T\mathcal{A}\mathcal{A}^T\Phi\mathbf{U}\Psi\Psi^T)_{ab}}{u^t_{ab}}(u-u^t_{ab})^2\end{aligned} \tag{18}$$

Theorem 2 *$G_{u_{ab}}(u,u^t_{ab})$ is an auxiliary function of $\mathcal{L}_{u_{ab}}(u)$.*

Proof We can find that $G_{u_{ab}}(u,u^t_{ab}) = \mathcal{L}_{u_{ab}}(u)$ holds when $u^t_{ab} = u$. Therefore, we need to prove that $G_{u_{ab}}(u,u^t_{ab}) \geq \mathcal{L}_{u_{ab}}(u)$ holds when $u^t_{ab} \neq u$ afterward.

$$\begin{aligned}&\because \quad (\Phi^T\Phi\mathbf{U}\Psi\mathbf{V}_p\mathbf{V}_p^T\Psi^T + \Phi^T\mathcal{A}\mathcal{A}^T\Phi\mathbf{U}\Psi\Psi^T)_{ab} \\ &= \sum_k\sum_j(\Phi^T\Phi)_{ak}\mathbf{U}^t_{kj}(\Psi\mathbf{V}_p\mathbf{V}_p^T\Psi^T)_{jb} \\ &\quad + \sum_k\sum_j(\Phi^T\mathcal{A}\mathcal{A}^T\Phi)_{ak}\mathbf{U}^t_{kj}(\Psi\Psi^T)_{jb} \\ &\geq (\Phi^T\Phi)_{aa}\mathbf{U}^t_{ab}(\Psi\mathbf{V}_p\mathbf{V}_p^T\Psi^T)_{bb} \\ &\quad + (\Phi^T\mathcal{A}\mathcal{A}^T\Phi)_{aa}\mathbf{U}^t_{ab}(\Psi\Psi^T)_{bb}\end{aligned} \tag{19}$$

$$\begin{aligned}&\therefore \quad \frac{(\Phi^T\Phi\mathbf{U}\Psi\mathbf{V}_p\mathbf{V}_p^T\Psi^T + \Phi^T\mathcal{A}\mathcal{A}^T\Phi\mathbf{U}\Psi\Psi^T)_{ab}}{u^t_{ab}} \\ &\geq (\Phi^T\Phi)_{aa}\mathcal{J}_{ab}(\Psi\mathbf{V}_p\mathbf{V}_p^T\Psi^T)_{bb} \\ &\quad + (\Phi^T\mathcal{A}\mathcal{A}^T\Phi)_{aa}\mathcal{J}_{ab}(\Psi\Psi^T)_{bb}\end{aligned} \tag{20}$$

□

In conclusion, we have proved that $G_{u_{ab}}(u,u^t_{ab})$ is an auxiliary function of $\mathcal{L}_{u_{ab}}(u)$. Notably, we can easily get the local minimum of $\mathcal{L}_{u_{ab}}(u)$ when $G_{u_{ab}}(u,u^t_{ab})$ reaches the local minimum. To this end, we calculate the first-order partial derivative of $G_{u_{ab}}(u,u^t_{ab})$ with respect to u, as follows:

$$\begin{aligned}G'_{u_{ab}}(u,u^t_{ab}) &= \mathcal{L}'_{u_{ab}}(u^t_{ab}) \\ &\quad + 2\frac{(\Phi^T\Phi\mathbf{U}\Psi\mathbf{V}_p\mathbf{V}_p^T\Psi^T + \Phi^T\mathcal{A}\mathcal{A}^T\Phi\mathbf{U}\Psi\Psi^T)_{ab}}{u^t_{ab}}(u-u^t_{ab}) \\ &= -4(\Phi^T\mathcal{A}\mathbf{V}_p^T\Psi^T)_{ab} + 2(\Phi^T\Phi\mathbf{U}\Psi\mathbf{V}_p\mathbf{V}_p^T\Psi^T)_{ab} \\ &\quad + 2(\Phi^T\mathcal{A}\mathcal{A}^T\Phi\mathbf{U}\Psi\Psi^T)_{ab} \\ &\quad + 2\frac{(\Phi^T\Phi\mathbf{U}\Psi\mathbf{V}_p\mathbf{V}_p^T\Psi^T + \Phi^T\mathcal{A}\mathcal{A}^T\Phi\mathbf{U}\Psi\Psi^T)_{ab}}{u^t_{ab}}(u-u^t_{ab})\end{aligned} \tag{21}$$

Let Eq. (21) = 0, we can get the update rule about $\mathbf{U}_i$, as follows:

$$u = u^t_{ab}\frac{2(\Phi^T\mathcal{A}\mathbf{V}_p^T\Psi^T)_{ab}}{(\Phi^T\Phi\mathbf{U}\Psi\mathbf{V}_p\mathbf{V}_p^T\Psi^T + \Phi^T\mathcal{A}\mathcal{A}^T\Phi\mathbf{U}\Psi\Psi^T)_{ab}} \tag{22}$$

We can get the local minimum value of auxiliary function $G_{u_{ab}}(u,u^t_{ab})$ under above iterative update rule because the Newton iterative update rule Eq. (22) is convergent quadratically. Meanwhile, we can also get the local minimum of $\mathcal{L}_{u_{ab}}(u)$. Obviously, the Eqs. (22) and (12) are equivalent, thus the Eq. (12) guarantees that the objective function $\mathcal{L}$ is monotonically decreasing, and the update rules of other variables can also guarantee this property.

3.4 Adaptive mechanism

The attributed network is extracted into two community membership matrices, i.e., $\mathbf{V}_t$ and $\mathbf{V}_k \in \mathbb{R}^{n\times h}$, which contain topology information and attribute information, respectively. Notably, these two matrices have been filtered their respective noises out in the DNMF module. Therefore, the most important task at present is to integrate topology information and attribute information. However, the traditional way, i.e., manually tune coupling via hyperparameters, not only easily destroys the cluster structures, but also loses attributes expressiveness. Specifically, topology tends to interfere with the node attributes and attributes tend to distort the topology structure and even make topology to compromise to attributes. In view of this, we use the attention mechanism, i.e., $att(\mathbf{V}_t,\mathbf{V}_k)$, to adaptively adjust the coupling relationship between topology and attributes, as follows:

$$(\alpha_t,\alpha_k) = att(\mathbf{V}_t,\mathbf{V}_k) \tag{23}$$

where α_t and $\alpha_k \in \mathbb{R}^{n\times 1}$ are the coupling weights of $\mathbf{V}_t,\mathbf{V}_k$, respectively.

For convenience, we take $\mathbf{V}_t$ as an example to illustrate how to obtain coupling weight. A simple yet effective way is to perform nonlinear transformations on this matrix to obtain low-dimensional attention weight, as follows:

$$\omega_t = \tanh(\mathbf{V}_t\mathbf{W}^T + \mathbf{B})\mathbf{Q}^T \tag{24}$$

where $\mathbf{W} \in \mathbb{R}^{h' \times h}$ and $\mathbf{Q} \in \mathbb{R}^{1 \times h'}$ are shared weight matrices, and $\mathbf{B} \in \mathbb{R}^{n \times h'}$ is shared bias matrix.

Meanwhile, we can get the attention weight ω_k about $\mathbf{V}_k$ in the same way. Finally, we obtain coupling weights by normalizing these two attention weights, as follows:

$$\alpha_t = \mathrm{softmax}(\omega_t) = \frac{\exp(\omega_t)}{\exp(\omega_t) + \exp(\omega_k)} \tag{25}$$

$$\alpha_k = softmax(\omega_k) = \frac{exp(\omega_k)}{exp(\omega_t) + exp(\omega_k)} \tag{26}$$

where the larger value of α_t and α_k implies that the corresponding community membership matrix is more important.

So far, we have obtained the integrated community membership matrix $\mathbf{V}_f$ under the combination of community membership matrices $\mathbf{V}_t$ and $\mathbf{V}_k$. The matrix $\mathbf{V}_f$ not only contains the core information of topology and attribute, but also fully characterizes the coupling relationship between them, as follows:

$$\mathbf{V}_f = \alpha_T\mathbf{V}_t + \alpha_K\mathbf{V}_k \tag{27}$$

where $\alpha_T = \mathrm{diag}(\alpha_t)$ and $\alpha_K = \mathrm{diag}(\alpha_k) \in \mathbb{R}^{n \times n}$.

3.5 The unified loss function

We introduce three loss functions as objective functions to train ADNMF, ensuring that the community detection results satisfy the following characteristics (1) dense intra-edge and sparse inter-edge and (2) similar intra-attributes.

Modularity maximization was proposed by Newman [17] and has been widely used in community detection. Its main idea is based on the characteristics of clusters, i.e., dense edges in the inner clusters and sparse edges outside the clusters. The loss function of modularity is defined as follows:

$$\mathcal{L}_Q = tr\left(\mathbf{Y}^T\left(\mathbf{A} - \frac{\mathbf{D}\mathbf{D}^T}{\sum\sum\mathbf{D}} + \mathbf{A}_k - \frac{\mathbf{D}_k\mathbf{D}_k^T}{\sum\sum\mathbf{D}_k}\right)\mathbf{Y}\right) \tag{28}$$

where $\mathbf{D}$ and $\mathbf{D}_k$ are the degree matrices of the topology graph and KNN graph, respectively.

In order to enhance the ability of the attention module to capture intrinsic relationship between topology information and attribute information, we impose additional constraints on this module, i.e., cross-entropy function and Frobenius-normal form, as follows:

$$\begin{aligned}\mathcal{L}_B = -\frac{1}{N^2}\sum\sum(\mathbf{A} + \mathbf{A}_k)\log p(\tilde{\mathbf{A}} = 1) \\ + (1 - \mathbf{A} + \mathbf{A}_k)\log p(\tilde{\mathbf{A}} = 0)\end{aligned} \tag{29}$$

$$\mathcal{L}_F = \|\mathbf{A} + \mathbf{A}_k - \mathbf{Y}\mathbf{Y}^T\|_F^2 \tag{30}$$

where N is the number of nodes, $p(\tilde{\mathbf{A}} = 1) = sigmoid(\mathbf{Y}\mathbf{Y}^T)$, and $p(\tilde{\mathbf{A}} = 0) = 1 - sigmoid(\mathbf{Y}\mathbf{Y}^T)$.

In summary, we combine $\mathcal{L}_Q$, $\mathcal{L}_B$ and $\mathcal{L}_F$ as a unified loss to constrain our method as follows:

$$\mathcal{L} = \alpha\mathcal{L}_F - \beta\mathcal{L}_Q + \mathcal{L}_B \tag{31}$$

where α and β are coefficients that control the contribution of the corresponding loss function. In our method, we always set these two coefficients to 0.1.

With the cooperation of all objective functions and DNMF module, we can optimize the ADNMF and detect communities in an end-to-end manner, as follows:

$$\mathbf{Y} = \arg\max \sigma(\mathbf{V}_f\mathbf{W}_f^T + \mathbf{B}_f) \tag{32}$$

where $\sigma(\cdot)$ is *softmax* activation function, $\mathbf{W}_f$ is weight matrix and $\mathbf{B}_f$ is bias matrix.

3.6 Time complexity

The complete procedure of ADNMF is outlined in Algorithm 1, where main computational cost is on the encoder module, the decoder module, the attention mechanism and the detect communities. The computational cost for the encoder module is of order $O(T_1 p(n^2 h + h^2 n))$ for T_1 iterations to converge. The computational cost for the decoder module is of order $O(T_{\max} p(n^2 h + h^2 n + h^3))$ for $T_{\max}$ iterations to converge. For attention mechanism module, the computational cost is of order $O(T_{\max} n h h') + O(T_{\max} n h')$. For the final community detection step, the computational cost is of order $O(T_{\max} n h k)$. In general, $h \leq n$ and $h \approx h' \approx k$, thus theoretically the total time complexity is $O((T_1 + T_{\max}) p n^2 h)$. In fact, because many matrices are sparse, the total computational cost is probably lower than the theoretical value. Thus, similar to most of other methods, ADNMF can converge in a short time.

Algorithm 1 ADNMF

Require: Attributed graph $\mathcal{G}$, Number of layers p, Number of communities k, Maximum iterations T_{max}, Parameter k in KNN graph

► **Construct KNN graph:**
Calculate $\mathbf{A}_k$ via Eq. (1)
► **Encoder module of DNMF:**
for $i \in p$ **do**
  $\mathbf{U}_i, \mathbf{U}_{k,i}, \mathbf{V}_i, \mathbf{V}_{k,i} \leftarrow$ Run NMF with hierarchical restrictions, i.e., Eq. (2) and Eq. (3), until convergence
end for
while t $\in T_{max}$ **do**
  ► **Decoder module of DNMF:**
  for $i \in p$ **do**
    Update $\mathbf{U}_i$ and $\mathbf{U}_{k,i}$ via Eq. (12)
    Update $\mathbf{V}_i$ and $\mathbf{V}_{k,i}$ via Eq. (13)
  end for
  ► **Attention mechanism:**
  Calculate $\mathbf{V}_f$ via Eq. (24), Eq. (25) and Eq. (27)
  ► **Unified loss function:**
  Calculate $\mathcal{L}_F$, $\mathcal{L}_Q$ and $\mathcal{L}_B$ via Eq. (30), Eq. (28) and Eq. (29) ,respectively
  Calculate the unified loss function $\mathcal{L}$ via Eq. (31)
  ► **Detect communities:**
  Calculate $\mathbf{Y}$ via Eq. (32)
end while
return Community detection results $\mathbf{Y}$

4 Experiments

To verify the effectiveness of ADNMF, in this section, we first conduct performance experiments on synthetic networks and real attributed networks, and then deeply analyze the results and demonstrate that ADNMF can effectively solve the two problems raised above. We implement all methods using Python 3.6. All experiments are conducted on a PC with 64-bit Windows 10 system, 3.40 GHz Intel Core i7-6700 CPU and 32 GB RAM.

4.1 Benchmark datasets

4.1.1 Synthetic benchmark datasets

We generate synthetic networks with ground-truth communities via the well-known LFR method [39]. By fixing the number of nodes to 3000 and fine-tuning the mixing coefficient from 0 to 0.4, we generate 5 sets of synthetic networks, as shown in Table 1. Next, based on the ground-truth communities label, we assign attributes for each node under a certain probability to satisfy the high attribute similarity within the communities. The details of generating the attributes are given in the literature [40].

4.1.2 Real benchmark datasets

All methods are conducted performance experiments on eight real attributed networks datasets, which are summarized in Table 2. All datasets can be found at https://github.com/GDM-SCNU/Datasets.

4.2 Baselines

We compare ADNMF with three types of methods, including two DNMF-based methods, four NMF-based methods and one network embedding method.

- *DANMF* [35]: DANMF expands the original DNMF through autoencoder and graph regularization to learn a hierarchical mapping between original nonnegative data and the final data.
- *MDNMF* [36]: Inspired by characteristics of clusters, Huang et al. proposed DNMF with modularity.
- Sun et al. [19]: Sun et al. proposed a nonnegative symmetric structure to detect communities. The method aims to minimize $\|\mathbf{A} - \mathbf{UV}\|_F^2 + \|\mathbf{V} - \mathbf{U}^T\mathbf{A}\|_F^2$.
- *PNMF* [30]: Since nodes usually belong to only one community, the community membership matrix is

Table 1 The statistics of the synthetic network datasets

Datasets	# Node	# Attributes	# Mixing coefficient	# Communities
SN1	3000	100	0	20
SN2	3000	100	0.1	20
SN3	3000	100	0.2	20
SN4	3000	100	0.3	20
SN5	3000	100	0.4	20

Table 2 The statistics of the real attributed network datasets

Datasets	# Nodes	# Edges	# Attributes	# Communities
Washington	230	446	1703	5
Wisconsin	265	530	1703	5
Cornell	195	304	1703	5
Texas	187	328	1703	5
Cora	2708	5429	1433	7
Citeseer	3312	4715	3703	6
BlogCatalog	5196	1,71,743	8189	6
UAI2010	3067	28,311	4973	19

sparse. Based on this assumption, PNMF aims to minimize $\|\mathbf{A} - \mathbf{V}\mathbf{V}^T\mathbf{A}\|_F^2$.

- *SCI* [20]: SCI aims to integrate topology and node attributes into a unified function, and it considers the sparsity of the community membership matrix.
- *NMFjGO* [32]: NMFjGO is used to solve the heterogeneity of structure and attributes. It achieves the integration of topology and latent feature of node attributes.
- *MNMF* [31]: MNMF is a variant of NMF, which takes modularity into consideration additionally. It incorporates the cluster structures into network embeddings.

4.3 Parameter settings and evaluation metrics

For ADNMF, we use the Adam optimizer and set the learning rate as 0.001. For each synthetic datasets, we set K to 13 and hidden-layers of encoder to 128, 86 and 64, respectively. For each real attributed dataset, the detail configuration is provided in Table 2. For each benchmark dataset, our method is trained for 500 iterations and we report the optimal result. Other baseline methods employ the settings described in their paper. To comprehensively evaluate the performance of methods, we employ modularity (Q) and normalized mutual information (NMI) to evaluate performance of methods on synthetic datasets. Furthermore, we use NMI, adjusted rand index (ARI), accuracy (ACC) and F1-macro (F1) to evaluate the performance of methods on real attributed datasets. For all metrics, larger value indicates better performance.

4.4 Performance comparison

4.4.1 Experiments on synthetic datasets

The experiment results on synthetic datasets are reported in Table 4. We have the following observations:

- ADNMF achieves the best performance on all synthetic datasets. Notably, for NMI and Q, ADNMF respectively achieves maximum relative improvements of 85.3% and 90.6% on SN1. The result implies the effectiveness of ADNMF to detect communities.
- Although all methods have degraded performance with the increase of the mixing coefficient, ADNMF still outperforms other methods, which indicates that ADNMF can effectively deal with network noise. We conclude the probable reasons as follows: on the one hand, the multilayer characteristic of DNMF and its autoencoder can filter the noise in the network to a large extent. On the other hand, each DNMF individually handles topology and attribute noise. To this end, even with a lot of topology noise interference, the extraction of attribute information is still unaffected and maintains good performance generally through the attention mechanism module.

Table 3 Layers configuration on real datasets

Datasets	Washington	Wisconsin	Cornell	Texas	Cora	Citeseer	BlogCatalog	UAI2010
Layers	32-20-16	32-22-16	64-32-20	32-20-16	128-86-64	150-86-64	64-32-16	100-64-50
k	19	16	11	13	13	15	14	18

Table 4 Performance evaluation on synthetic benchmark datasets

Datasets	Metrics	ADNMF	DANMF	MDNMF	Sun et al	PNMF	NMFjGO	SCI	MNMF
SN1	NMI	**1**	0.960	0.094	0.742	0.991	0.326	0.984	0.963
	Q	**0.941**	0.898	0.088	0.677	0.928	0.242	0.924	0.908
SN2	NMI	**0.999**	0.994	0.116	0.54	0.965	0.326	0.98	0.951
	Q	**0.85**	0.847	0.09	0.415	0.821	0.225	0.827	0.807
SN3	NMI	**0.979**	0.978	0.091	0.46	0.954	0.326	0.969	0.927
	Q	**0.745**	0.744	0.085	0.344	0.744	0.211	0.742	0.703
SN4	NMI	**0.967**	0.954	0.061	0.296	0.94	0.326	0.962	0.948
	Q	**0.644**	**0.644**	0.09	0.217	0.625	0.195	0.639	0.633
SN5	NMI	**0.933**	0.887	0.034	0.195	0.824	0.326	0.913	0.828
	Q	**0.546**	0.522	0.085	0.168	0.501	0.174	0.544	0.495

Bold numbers denote the best results

Table 5 Performance evaluation based on NMI, F1, ACC and ARI

Datasets	Metrics	ADNMF	DANMF	MDNMF	Sun et al	PNMF	NMFjGO	SCI	MNMF
Washington	NMI	**0.418**	0.041	0.112	0.092	0.098	0.405	0.125	0.062
	F1	**0.41**	0.386	0.137	0.197	0.12	0.164	0.261	0.192
	ACC	**0.543**	0.201	0.174	0.235	0.117	0.226	0.43	0.226
	ARI	**0.404**	0.074	0.132	0.096	0.169	**0.404**	0.154	0.042
Wisconsin	NMI	0.451*	0.05	0.037	0.04	0.062	**0.465**	0.073	0.097
	F1	**0.418**	0.162	0.178	0.174	0.131	0.176	0.173	0.161
	ACC	**0.517**	0.16	0.317	0.2	0.155	0.204	0.185	0.189
	ARI	0.351*	0.041	0.028	0.015	0.095	**0.392**	0.051	0.052
Cornell	NMI	**0.422**	0.063	0.058	0.142	0.081	0.32	0.074	0.056
	F1	**0.328**	0.256	0.163	0.222	0.217	0.164	0.194	0.186
	ACC	**0.451**	0.234	0.426	0.426	0.303	0.174	0.2	0.221
	ARI	**0.324**	0.037	0.013	0.098	0.019	0.241	0.036	0.035
Texas	NMI	**0.385**	0.059	0.07	0.178	0.192	0.378	0.087	0.062
	F1	**0.284**	0.16	0.157	0.253	0.143	0.16	0.144	0.185
	ACC	**0.401**	0.13	0.23	0.289	0.139	0.166	0.193	0.219
	ARI	0.243	0.14	0.074	0.186	**0.299**	0.28	0.123	0.109
Cora	NMI	0.289	**0.374**	0.057	0.144	0.224	0.167	0.326	0.231
	F1	**0.386**	0.24	0.137	0.172	0.116	0.147	0.251	0.163
	ACC	**0.407**	0.401	0.302	0.318	0.267	0.152	0.357	0.322
	ARI	0.232	**0.263**	0.027	0.055	0.118	0.121	0.238	0.083
Citeseer	NMI	**0.265**	0.173	0.046	0.085	0.105	0.219	0.19	0.11
	F1	**0.409**	0.218	0.172	0.185	0.078	0.21	0.179	0.167
	ACC	**0.438**	0.296	0.187	0.192	0.191	0.21	0.185	0.174
	ARI	**0.24**	0.171	0.001	0.008	0.013	0.211	0.17	0.012
BlogCatalog	NMI	**0.264**	0.221	0.008	0.005	0.155	0.144	0.224	0.202
	F1	**0.38**	0.062	0.047	0.173	0.143	0.163	0.218	0.163
	ACC	**0.405**	0.166	0.166	0.173	0.153	0.163	0.227	0.163
	ARI	**0.194**	0.157	0.003	0.002	0.086	0.101	0.153	0.139
UAI2010	NMI	**0.455**	0.318	0.054	0.044	0.2	0.417	0.231	0.153
	F1	**0.139**	0.009	0.009	0.042	0.023	0.053	0.055	0.055
	ACC	**0.144**	0.03	0.072	0.051	0.03	0.062	0.069	0.053
	ARI	**0.277**	0.124	0.003	0.08	0.067	0.25	0.09	0.039

Bold numbers denote the best results, and asterisk numbers denote the runner-up results

- Comparing with the three methods, we can learn that ADNMF can achieve good performance without manually tuning the coupling relationship between topology and attribute information on all datasets. The main reason is that the attention mechanism module adaptively integrates topology and attribute information. Therefore, we do not need to focus on which one is more important in each graph, which makes our method more general. Meanwhile, it further confirms that our method effectively avoids the interference and compromise between topology and attributes.

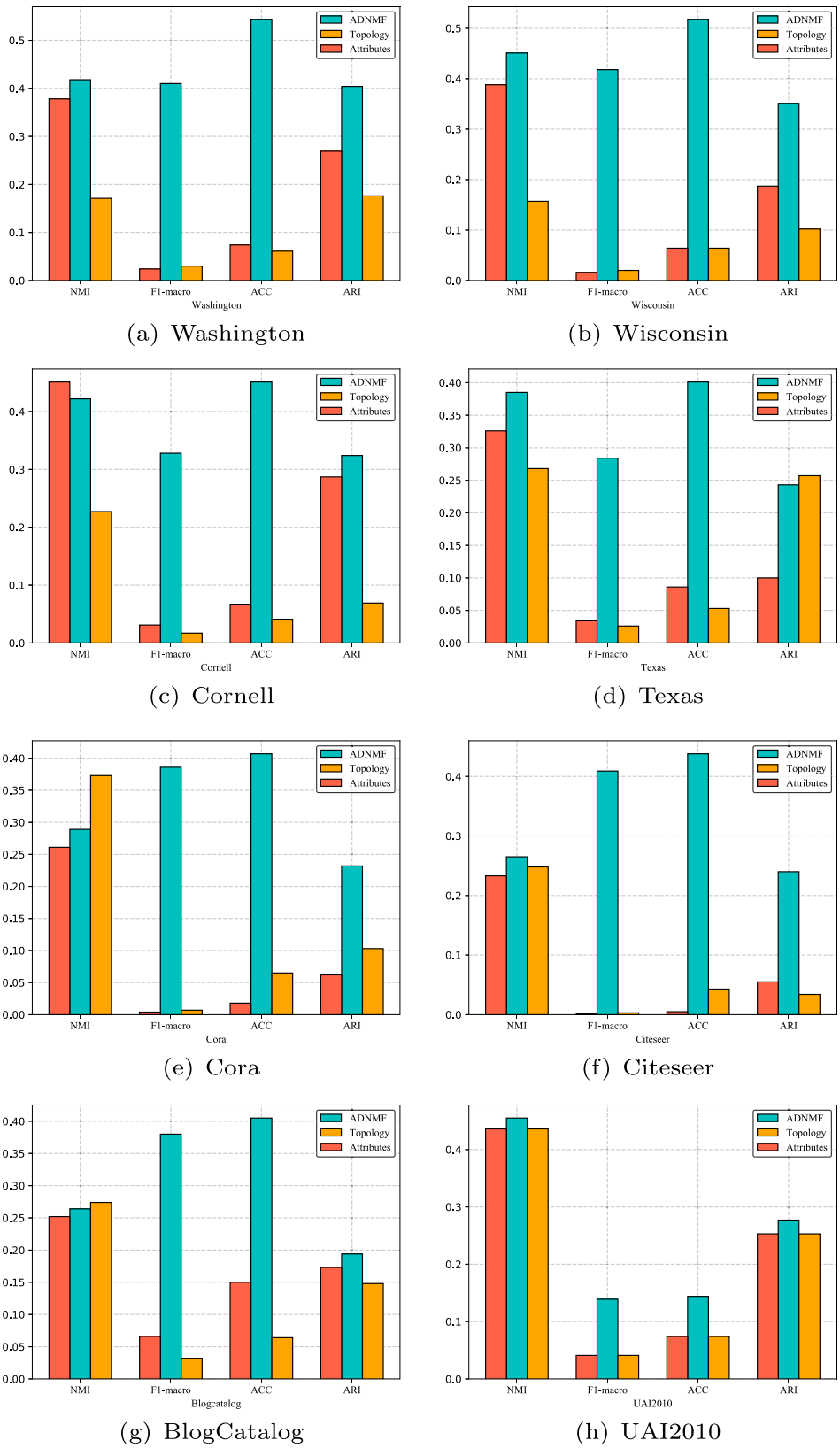
4.4.2 Experiments on real datasets

The community detection results are reported in Table 5. We have the following observations.

- *ANMF achieves better results than most methods on eight real attributed networks datasets.* For datasets from Washington, Wisconsin, Cornell and Texas, there is strong topology noise due to their hub nodes, meaning that they are basically dense networks. But as soon as the hub nodes are removed, the networks immediately become sparse. In this case, it is difficult to find optimal community structures in most methods. Fortunately, ADNMF performs better in these four datasets. Specifically, we get 24.2% (Texas), 15.4% (Cornell and Texas), 32.7% (Texas) and 18.6% (Texas) performance boosts on NMI, F1, ACC and ARI, respectively. And we achieve the largest performance gains of 32% (Wisconsin), 29.3% (Washington), 50.2% (Wisconsin) and 34.8% (Washington) on NMI, F1, ACC and ARI, respectively. From these data, we find that ADNMF significantly resolves the inherent noise of topology structures, as demonstrated by ACC and NMI. Focus on four other datasets, although these datasets do not have prominent topology noise, dealing with both topology noise and attribute noise is still a challenge. It is worth noting that ADNMF still achieves a significant improvement in performance. Generally, we get 21.9% (Citeseer), 13% (UAI2010), 11.4% (UAI2010) and 19.2% (BlogCatalog) performance boosts on NMI, F1, ACC and ARI, respectively. Among them, we achieve the largest performance gains of 41.1% (UAI2010), 33.3% (BlogCatalog), 26.4% (Citeseer) and 27.4% (UAI2010) on NMI, F1, ACC and ARI, respectively. We can find that ADNMF can still achieve good results in facing the tricky problem of handling topology and attribute noise at the same time. In all, these results support the conclusion that ADNMF can resolve attributed network inherent noise.
- *ADNMF has better results than most methods for integrating topology and node attributes.* For both NMFjGO and SCI, we find that these methods have difficulty to resolve the interference and compromise between topology and attributes. In Wisconsin, for example, their ACC is even smaller than MDNMF, which only considers topology information. Similarly, in Citeseer, their F1 is smaller than ADNMF. Fortunately, ADNMF can solve the problem through an attention mechanism. To be specific, we get 0.7% (Texas), 8.4% (UAI2010) and 5% (Cora) performance boosts on NMI, F1 and ACC, respectively. And we achieve the largest performance gains in Wisconsin and Washington, i.e., 37.8% (Wisconsin), 24.6% (Washington), 33.2% (Wisconsin) and 30% (Wisconsin) on NMI, F1, ACC and ARI, respectively. Although dealing with these datasets is a thorny challenge, these experimental results show that the attention mechanism in our method can effectively resolve the interference and compromise between topology and attributes. We no longer need hyperparameters to balance their coupling relationship. However, we can find that the NMI and ARI in Wisconsin are only runner-up. The main reason is the hypothesis of homogeneity that the existence of edges between nodes with similar attributes. In real networks, however, there are node pairs with similar attributes but no relationship between them in most cases. Therefore, some unavoidable noise will appear when we construct KNN graph.
- *ADNMF achieves better results than most methods of different types.* In comparison to the DNMF-based methods, our method achieves the largest performance gains of 41.4% (Wisconsin), 33.3% (BlogCatalog), 36.9% (Washington) and 33% (Washington) on NMI, F1, ACC and ARI, respectively. In comparison to the NMF-based methods, our method achieves the largest performance gains of 41.4% (Wisconsin and UAI2010), 33.1% (Citeseer), 42.6% (Washington) and 33.6% (Wisconsin) on NMI, F1, ACC and ARI, respectively. In comparison to the network embedding method, our method achieves the largest performance gains of 36.6% (Cornell), 25.7% (Wisconsin), 32.8% (Wisconsin) and 36.2% (Washington) on NMI, F1, ACC and ARI, respectively. The experiment results show that ADNMF outperforms different types of methods generally.

4.4.3 Method analysis

We conduct two types of experiments and learn that ADNMF generally outperforms other methods on both synthetic and real attributed networks. To this end, we now conduct a comprehensive analysis and summary of the possible reasons behind the experimental results. Overall,



(a) Washington (b) Wisconsin

(c) Cornell (d) Texas

(e) Cora (f) Citeseer

(g) BlogCatalog (h) UAI2010

Fig. 2 Ablation experiment. The bars of orange, tomato and cyan are ADNMF with topology only, with attributes only and our proposed method, respectively

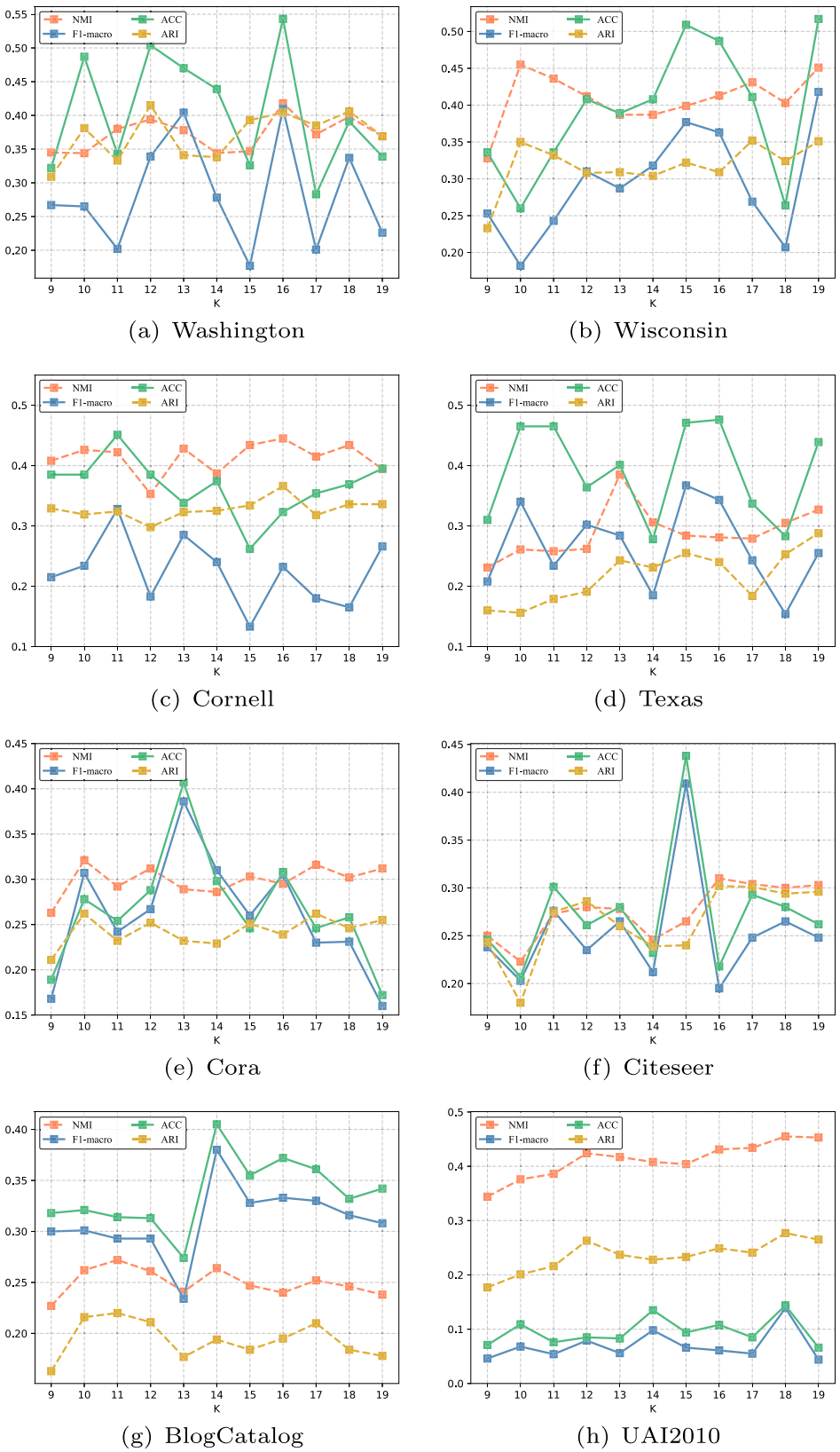


Fig. 3 *K*-sensitivity analysis. The lines of coral, steel blue, medium seagreen and golden rod are respectively corresponding NMI, F1, ACC and ARI

ADNMF has significant advantages in community detection on attributed networks because we focus on solving two major problems, i.e., network noise and coupling between topology and attributes. Obviously, network noise brings great challenges to community detection on attributed networks. Therefore, we emphasize the significance of DNMF with autoencoder. On the one hand, with the increase of the mapping layer (i.e., encoding), community membership matrix will be gradually purified, i.e., noise is filtered. On the other hand, with the recovery of the mapping layer (i.e., decoding), the core information will be preserved. Another problem that needs to be solved is how to avoid manual tuning of the coupling relationship between topology and attributes. For example, in experiments on synthetic datasets, NMF-based methods often need to adjust coefficients to balance the coupling between topology and attributes. The main reason is that they do not know which one is more important in this dataset, causing interference and compromise between topology and attributes. In our method, the attention mechanism module can adaptively integrate topology and attribute information. Guided by the objective function, our community detection result can achieve denser intra-edges and higher similar intra-attributes and avoid the aforementioned problems. All experiment results effectively demonstrate our conclusions.

4.5 Ablation experiment

We conduct an ablation experiment to intuitively feel the importance of each component in ADNMF. To this end, we design two control combinations: ADNMF with topology information only and with attribute information only, for performance comparison with ADNMF. The experimental results are shown in Fig. 2.

We can observe that ADNMF has good performance in most datasets, where F1 and ACC have significant advantages. Especially, ADNMF gets 6.3% (Wisconsin), 40.6% (Citeseer), 46.9% (Washington) and 18.5% (Citeseer) performance boosts on NMI, F1, ACC and ARI, respectively. And it achieves the largest performance gains of 29.4% (Wisconsin), 40.8% (Citeseer), 48.2% (Washington) and 25.5% (Cornell) on NMI, F1, ACC and ARI, respectively. Ablation experiment, firstly, shows that integrating topology and attributes is an effective way to improve performance. On the one hand, when we consider only one type of information, such as topology information or attribute information, it is difficult to further improve the performance of the method, as shown in Fig. 2e, f. On the other hand, the performance instability problem is very prominent. For example, Fig. 2c shows that ADNMF with only attributes has better performance in NMI and ARI. But it shows a significant gap in F1 and ACC. Figure 2e shows that ADNMF with only topology has a similar situation. Obviously, the results of the ablation experiments show that considering both topology and attribute information is helpful to solve the above problem.

Secondly, from a macroscopic perspective, ablation experiment shows that ADNMF can handle the inherent noise of attributed networks and resolve the interference and compromise between topology and attributes. Focusing on orange and cyan components, we find that ADNMF achieves the largest performance gains of 29.4% (Wisconsin), 40.6% (Citeseer), 45.3% (Wisconsin) and 25.5% (Cornell) on NMI, F1, ACC and ARI, respectively. Focusing on tomato and cyan, we find that ADNMF achieves the largest performance gains of 6.3% (Wisconsin), 40.8% (Citeseer), 46.9% (Washington) and 18.5% (Citeseer) on NMI, F1, ACC and ARI, respectively. Obviously, the experiment results confirm our conclusion.

4.6 *K*-sensitivity analysis

Since the number of the k is an important hyperparameter in construction of the KNN graph, thus we design a k-sensitivity experiment to explore the effect of different k on ADNMF. The experiment results can help us to unearth potential rules of performance variation. Figure 3 provides an overall performance ups and downs, where k $\in \{9, 10, \ldots, 19\}$. From a holistic perspective, NMI and ARI have similar increasing and decreasing trends, and ACC and F1 have same trends. Therefore, we start from these two sets of metrics to explore the sensitivity of k. Figure 3 shows that when the value of any group changes, there is a tendency for the value of the other group to change as well. However, the changes in NMI and ARI are negligible, but the changes in ACC and F1 are evident. This result implies that ACC and F1 are sensitive to k, but NMI and ARI are not sensitive to k. Fortunately, we can easily find the optimal k because the four metrics have similar increasing and decreasing trends, and the NMI and ARI have small fluctuations.

5 Conclusions

In this paper, we propose a novel method, namely ADNMF, which consists of a dual DNMF with autoencoder and an attention mechanism component. Especially, we attempt to process topology and attribute information separately by DNMF with autoencoder to make it easier for our method to deal with inherent noise in attributed networks. Then, the attention mechanism integrates community membership matrices in a generalized way, which resolves the problem of interference and compromise between topology and attribute information. Experiment

results support our conclusions and show that our method outperforms most state-of-the-art methods.

Acknowledgements This work was supported in part by the National Natural Science Foundation of China under Grant 62077045, and Grant U1811263, in part by the Humanity and Social Science Youth Foundation of Ministry of Education of China under Grant 19YJCZH049.

Data availability The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The authors declared that there is no competing interests.

References

1. Teng XY, Liu J, Li MM (2021) Overlapping community detection in directed and undirected attributed networks using a multiobjective evolutionary algorithm. IEEE Trans Cybern 51(1):138–150
2. Menche J, Sharma A, Kitsak M, Ghiassian SD, Vidal M, Loscalzo J, Barab AL (2015) Uncovering disease-disease relationships through the incomplete interactome. Am Assoc Adv Sci 347(6224):1257601
3. Kipf TN, Welling M (2016) Semi-supervised classification with graph convolutional networks. In: International conference on learning representations (ICLR)
4. Liu FZ, Li Z, Wang BK, Wu J, Yang J, Huang JM, Zhang YQ, Wang WQ, Xue S, Nepal S, Sheng QZ (2022) eRiskCom: an e-commerce risky community detection platform. VLDB J 31:1085–1101
5. Xu SY, Yang C, Shi C, Fang Y, Guo YX, Yang TC, Zhang LH, Hu MD (2021) Topic-aware heterogeneous graph neural network for link prediction. In: Proceedings of the 30th ACM international conference on information & knowledge management, pp 2261–2270
6. Cen YK, Zou X, Zhang JW, Yang HX, Zhou JR, Tang J (2019) Representation learning for attributed multiplex heterogeneous network. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, pp 1358–1368
7. Su X, Xue S, Liu FZ, Wu J, Yang J, Zhou C, Hu WB, Paris C, Nepal S, Jin D, Sheng QZ, Yu PS (2021) A comprehensive survey on community detection with deep learning. IEEE Trans Neural Netw Learn Syst abs/2105.12584
8. Jin D, Yu ZZ, Jiao PF, Pan SR, Yu PS, Zhang WX (2021) A survey of community detection approaches: From statistical modeling to deep learning. IEEE Trans Knowl Data Eng abs/2101.01669
9. Liu FZ, Xue S, Wu J, Zhou C, Hu WB, Paris C, Nepal S, Yang J, Yu PS (2020) Deep learning for community detection: progress, challenges and opportunities. In: Proceedings of the twenty-ninth international joint conference on artificial intelligence, pp 4981–4987
10. Cheng JW, Li WS, Han KL, Tang Y, He CB, Zhang NN (2022) SARNMF: a community detection method for attributed networks. In: 2022 IEEE 25th international conference on computer supported cooperative work in design(IEEE CSCWD 2022), pp 879–884
11. Ma XK, Dong D, Wang Q (2019) Community detection in multilayer networks using joint nonnegative matrix factorization. IEEE Trans Knowl Data Eng 31(2):273–286
12. Jin D, He J, Chai BF, He DX (2021) Semi-supervised community detection on attributed networks using non-negative matrix trifactorization with node popularity. Front Comp Sci 15(4):1–11
13. He DX, Song Y, Feng ZY, Zhang BB, Yu ZZ, Zhang WX (2020) Community-centric graph convolutional network for unsupervised community detection. In: Proceedings of the twenty-ninth international conference on international joint conferences on artificial intelligence, pp 3515–3521
14. Oleksandr S, Günnemann S (2019) Overlapping community detection with graph neural networks. preprint arXiv
15. Yang L, Zhou WM, Peng WH (2022) Graph neural networks beyond compromise between attribute and topology. In: Proceedings of the ACM web conference, pp 1127–1135
16. Liu FZ, Wu J, Xue S, Zhou C, Yang J, Sheng QZ (2020) Detecting the evolving community structure in dynamic social networks. World Wide Web 23(2):715–733
17. Liu FZ, Wu J, Zhou C, Yang J (2019) Evolutionary community detection in dynamic social networks. 2019 international joint conference on neural networks, pp 1–7
18. He CB, Fei X, Cheng QW, Li HC, Hu Z, Tang Y (2022) A survey of community detection in complex networks using nonnegative matrix factorization. IEEE Trans Comput Soc Syst 9(2):440–457
19. Sun BJ, Shen HW, Gao JH, O WT, Cheng XQ (2017) A nonnegative symmetric encoder–decoder approach for community detection. In: Proceedings of the 2017 ACM on conference on information and knowledge management, pp 597–606
20. Wang X, Jin D, Cao XC, Yang L, Z WX (2016) Semantic community identification in large attribute networks. In: Proceedings of the AAAI conference on artificial intelligence 30(1)
21. He CB, Zheng YL, Fei X, Li HC, Hu Z, Tang Y (2021) Boosting nonnegative matrix factorization based community detection with graph attention auto-encoder. IEEE Trans Big Data 8:968–981
22. Ji D, Liu Z, He RF, Wang X, He DX (2018) A robust and strong explanation community detection mehtod for attributed networks. Chin J Comput 41(7):1476–1489
23. Yang L, Chen ZY, Gu JH, Guo YF (2019) Dual self-paced graph convolutional network: towards reducing attribute distortions induced by topology. In: Proceedings of the twenty-eighth international joint conference on artificial intelligence, pp 4062–4069
24. Wang WJ, Liu X, Jiao PF, Chen X, Jin D (2018) A unified weakly supervised framework for community detection and semantic matching. Adv Knowl Discov Data Min 10939:218–230
25. Li HJ, Huang L, Wang CD, Huang D, Lai HJ, Chen P (2021) Attributed network embedding with micro-meso structure. ACM Trans Knowl Discov Data 15(4):1–26
26. McPherson M, Smith-Lovin L, Cook JM (2001) Birds of a feather: homophily in social networks. Ann Rev Sociol 27:415–444
27. Garza SE, Schaeffer SE (2019) Community detection with the label propagation algorithm: a survey. Phys A 534:122058
28. Perozzi B, Al-Rfou R, Skiena S (2014) DeepWalk: online learning of social representations. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pp 701–710
29. Lee DD, Seung HS (1999) Learning the parts of objects by nonnegative matrix factorization. Nature 401(6755):788–791
30. Yuan ZJ, Oja E (2005) Projective nonnegative matrix factorization for image compression and feature extraction. Image Analysis, 14th Scandinavian Conference 3540:333–342
31. Wang X, Cui P, Wang J, Pei J, Yang SQ (2017) Community preserving network embedding. In: Proceedings of the thirty-first AAAI conference on artificial intelligence, pp 203–209

32. Huang ZH, Zhong XX, Wang Q, Gong MG, Ma XK (2020) Detecting community in attributed networks by dynamically exploring node attributes and topological structure. Knowl-Based Syst 196:105760
33. Jin D, He J, Chai BF, He DX (2021) Semi-supervised community detection on attributed networks using non-negative matrix tri-factorization with node popularity. Front Comp Sci 15(4):154324
34. Trigeorgis G, Bousmalis K, Zafeiriou S, Schuller BW (2017) A deep matrix factorization method for learning attribute representations. IEEE Trans Pattern Anal Mach Intell 39(3):417–429
35. Ye FH, Chen C, Zheng ZB (2018) Deep autoencoder-like nonnegative matrix factorization for community detection. In: Proceedings of the 27th ACM international conference on information and knowledge management, pp 1393–1402
36. Huang J, Zhang TH, Yu WH, Zhu J, Cai EC (2020) Community detection based on modularized deep nonnegative matrix factorization. Int J Pattern Recognit Artif Intell 32(5):2159006:1-2159006:17
37. Bengio Y, Courville A, Vincent P (2013) Representation learning: a review and new perspectives. IEEE Trans Pattern Anal Mach Intell 35(8):1798–1828
38. Lee DD, Seung HS (2000) Algorithms for non-negative matrix factorization. Adv Neural Inf Proc Syst 13:556–562
39. Lancichinetti A, Fortunato S, Radicchi F (2008) Benchmark graphs for testing community detection algorithms. Phys Rev E 78(2):046110
40. He CB, Zheng YL, Cheng JW, Tang Y, Chen GH, Liu H (2022) Semi-supervised overlapping community detection in attributed graph with graph convolutional autoencoder. Inf Sci 608:1464–1479