Multi-Source Propagation Aware Network Clustering[⋆]

Tiantian He[a,b,*], Yew-Soon Ong[a,b] and Pengwei Hu[c]

[a] *School of Computer Science and Engineering, Nanyang Technological University, Singapore*
[b] *Data Science & Artificial Intelligence Research Center, Nanyang Technological University, Singapore*
[c] *Kriston AI, China PR*

ARTICLE INFO

Keywords:
Network cluster analysis, matrix factorization, non-negative matrix factorization, latent factor model

ABSTRACT

Network cluster analysis is of great importance as it is closely related to diverse applications, such as social community detection, biological module identification, and document segmentation. Aiming to effectively uncover clusters in the network data, a number of computational approaches, which utilize network topology, single vector of vertex features, or both the aforementioned, have been proposed. However, most prevalent approaches are incapable of dealing with those contemporary network data whose vertices are characterized by features collected from multiple sources. To address this challenge, in this paper, we propose a novel framework, dubbed Multi-Source Propagation Aware Network Clustering (MSPANC) for uncovering clusters in network data possessing multiple sources of vertex features. Different from most previous approaches, MSPANC is able to infer the cluster preference for each vertex utilizing both network topology and multi-source vertex features. To improve the practical significance of the discovered clusters, the learning of cluster membership is also involved into the modeling of the maximization of intra-cluster propagation regarding multi-source features. We propose a unified objective function for MSPANC to perform the clustering task and derive an alternative manner of learning algorithm for model optimization. Besides, we theoretically prove the convergence of the algorithm for optimizing MSPANC. The proposed model has been tested on five real-world datasets, including social, biological and document networks, and has been compared with several competitive baselines. The remarkable experimental results validate the effectiveness of MSPANC.

1. Introduction

Networks are omnipresent in this hyper-connected world as many real-world data samples and observed sample-sample relationship can be modeled using the terminologies of vertices and edges respectively. Representing different samples coming from various activities, e.g., people surfing on the social networking sites, documents collected for segmentation, and biological units jointly participating vital activities, networks may well preserve the complex inter-relationship between data samples for future analytical tasks. Compared with the synthetic, real networks are much more fascinating as there are many meaningful latent structures hidden under chaos, implicitly carrying the information which reveals particular characteristics of the vertices. Amongst them, network clusters (communities) which are the reservoirs preserving rich group-wise features of cohesiveness, are one of the most significant latent structures. As many real-world applications, e.g., online social community detection [21, 26], document segmentation [2], biological module identification [7], and drug discovery [31], are directly related to network cluster analysis, how to effectively discover clusters in network data has drawn much interest in the recent [10].

Aiming at discovering sets of cohesive vertices sharing similar features, network clustering has been an everlasting challenge in the machine learning community. And there have been a number of computational approaches proposed to solve this problem. As topological information is the prerequisite for constructing a network, many approaches have been proposed to perform the task of network clustering taking into consideration the optimization of various topology-related similarities of vertices in each cluster, including edge and edge weights. Modularity [3], which measures the difference in terms of density of vertices in the same group and that when vertices are randomly grouped, is one of the most popular benchmarks. Consequently, many approaches, such as Clauset-Newman-Moore algorithm

[⋆] This research is supported by the National Research Foundation Singapore under its AI Singapore Programme [Award Number: AISG-RP-2018-004], the National Natural Science Foundation of China under Grant 61802317, and Data Science and Artifical Intelligence Research Center, Nanyang Technological University.

[*] Corresponding author
tiantian.he@ntu.edu.sg (T. He); asysong@ntu.edu.sg (Y. Ong); hupengwei@hotmail.com (P. Hu)
ORCID(s): 0000-0003-4839-681X (T. He); 0000-0002-4480-169X (Y. Ong); 0000-0001-5974-7932 (P. Hu)

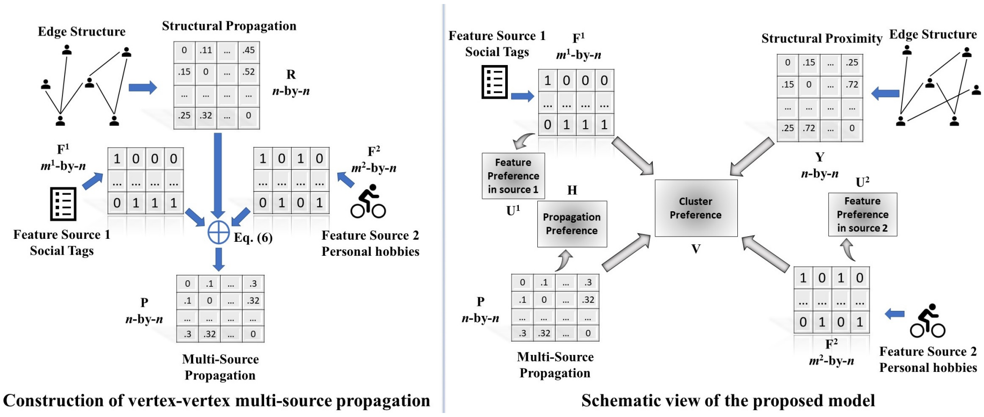


Figure 1: Left: Schematic view of how to construct multi-source feature propagation between pairwise vertices. Right: Graphical illustration of the idea behind the proposed Multi-Source Propagation Aware Network Clustering (MSPANC).

(CNM) [3], Shrink [8], and Sparse-Cut [17] are proposed to discover network clusters via immediately optimizing modularity. Besides, some model-based methods, such as Stochastic Block models [18], and Matrix-Factorization-based models [29, 30] also can discover network clusters utilizing edge structure. To further capture the local structural similarities between linked vertices, methods like Normalized cut [19], and LambdaCC [22] can perform the task through maximizing the edge weights, which represent the vertex-vertex similarities in each cluster.

Besides edge structure, content data, e.g., comments made by the users in social networks, can be supplementary features which describe vertices from the semantic perspective. Therefore, some approaches are also able to unfold network clusters utilizing vertex features rather than topological information. For example, k-means algorithm [15] is able to uncover network clusters using vertex features.

To unfold network clusters which are characterized by both topological and semantic information, many recent methods perform the task concerning both network topology and vertex features. As structure and content features are both used to describe vertices in the network, a new term, Attributed Network/Graph (AN/AG) [7] is proposed to distinguish from those networks only containing topological information. Clustering in attributed networks, attempting to learn the cluster membership for each vertex from two separate data sources, i.e., topology and vertex feature, is capable of inferring those latent groups of vertices preserving both topological and semantic properties. In general, AN clustering can be solved following two different methodologies. First, clusters can be uncovered in the weighted network where edges are assigned with proper weights concerning the similarity regarding the features of two connected vertices. Common clustering models, e.g., spectral clustering [27], expectation-maximization clustering [11], or markov clustering [32], can be used to extract the clusters in the weighted network. Second, a low-dimensional latent space shared by edge structure and vertex features can be inferred by a co-learning model and such latent space represents the cluster preference for each vertex in AN. Several topic-model-based approaches, e.g., iTopic models [20] and Relational topic models [2] have been proposed to learn low-dimensional latent spaces from network topology and vertex features. Communities from edge structure and node attributes (CESNA) [28] is a probabilistic matrix-factorization model which can learn a low dimensional latent space representing the vertex-cluster preference, from edge structure and feature similarity. Manifold regularized stochastic block model (MrSBM) [6] is another probabilistic model for uncovering clusters in ANs. Through considering the regularization of the manifold which indicates the higher-order similarity between pairwise vertices, MrSBM can learn the enhanced cluster membership for each vertex in ANs. In addition to Bayesian models, other machine learning techniques, such as matrix factorization [14, 12], have also been used to unfold clusters in the attributed network. For example, Mining interesting sub-graphs (MISAGA) [7] is an effective approach that is capable of inferring meaningful cluster memberships for each vertex from both topological and feature spaces.

In today's big data era, vertices in the network are always characterized using the features coming from multiple sources. Taking the online social network depicted in Fig. 1 as an example, an online social network can be modeled using the vertices and edges to represent social networking users and their friendship. Besides these basic elements, supplementary information like personal hobbies and social tags of each user can also be used to characterize the vertices. Under this circumstance, attributed network can be redefined as Multi-Source Featured Network, in which the features characterizing the vertices are collected from multiple distinct sources.

Despite the availability of the data, there are no effective approaches that are capable of discovering clusters in such multi-source featured networks as well as summarizing the features which characterize the uncovered clusters. Previous approaches attempt to solve this problem following two ways. First, by concatenating the multi-source vertex features as a single vector, clusters hidden in the multi-source featured network are able to be uncovered by an approach to AN clustering. Second, by constructing a set of multi-layer networks [9] in each of which vertices are connected if they share the same vertex features in a corresponding source, clusters can then be discovered by an effective approach to multi-view clustering, e.g., collective non-negative matrix factorization [1]. Though effective to some extent, both the aforementioned are incapable of fully utilizing the multi-source vertex features, or discovering source-wise latent features characterizing the clusters. In this paper, we instead propose a novel model called Multi-Source Propagation Aware Network Clustering (MSPANC), to discover meaningful clusters in multi-source featured networks. Figure 1 illustrates how MSPANC discovers clusters in a social network with two sources of vertex features. Representing multi-source vertex features and structural proximity between vertices as independent data matrices, i.e., $\mathbf{F}^i$ ($i = 1, 2$), and $\mathbf{Y}$, MSPANC assumes that the structural proximity is generated by $\mathbf{V}$, which is a latent space shared by $\mathbf{F}^i$s. Additionally, taking advantage of multi-source features and the edge structure, MSPANC constructs a weighted matrix ($\mathbf{P}$) which represents the overall propagation of multi-source features between connected vertices and attempts to find the optimal latent structure ($\mathbf{H}$) which maximizes the propagation within each cluster, according to the cluster preference $\mathbf{V}$. Making use of the above modeling scheme, MSPANC is able to uncover clusters simultaneously concerning network structure, multi-source features, and multi-source feature propagation.

The contributions of this paper can be summarized as follows:

- We propose MSPANC, which is a novel model for discovering meaningful clusters in multi-source featured networks. Different from previous approaches, MSPANC attempts to infer the cluster preference from multi-source vertex features and structural proximity. Simultaneously, the learning of cluster preference is also involved into the maximization of intra-cluster feature propagation. Concerning multiple factors that may influence the cluster preference of each vertex, MSPANC is capable of accurately inferring clusters in different types of network data.

- We propose a unified objective function for MSPANC and derive an iterative algorithm for optimizing the formulated clustering problem. In addition, we prove the convergence of the proposed model and analyze its computational complexity.

- The proposed model has been compared with a number of competitive baselines on various real applications, including social community detection, functional module identification, and document segmentation. Experimental results show that MSPANC outperforms other baselines in most clustering tasks, which validates the effectiveness of the proposed model.

The rest of this paper is organized as follows. In Section 2, we elaborately introduce how the proposed model formulates the clustering problem in the networks with multi-source vertex features as a unified objective function, and derive an iterative algorithm to optimize the formulated problem. In Section 3, we theoretically prove the convergence of the model and analyze the computational complexity of the derived algorithm for model optimization. In Section 4, we conduct a series of experiments to validate the effectiveness of the proposed model. At last, in Section 5, we summarize the contributions of the paper and propose future works that can further improve the proposed model.

2. The proposed model

In this section, we elaborate the proposed Multi-Source Propagation Aware Network Clustering model (MSPANC). First, we introduce the mathematical notations used in this paper. Second, three important components of the proposed model, including source-wise feature modeling, proximity modeling, and multi-source propagation modeling are introduced. Based on them, we formulate the objective function for MSPANC. Finally, we derive an iterative algorithm which may optimize the proposed model.

2.1. Notations

Assuming a network containing n vertices, $|E|$ edges, and m vertex features collected from d sources, each of which has m^i vertex features, we use $\mathbf{D} \in \{0,1\}^{n\times n}$ and $\mathbf{F}^i \in \{0,1\}^{m^i\times n}$ to represent whether there is an edge between two vertices and whether there is a feature from source i associated with a vertex, respectively. In addition, we use two

Table 1
Notations used by MSPANC

Notation	Meaning
n, m, $\|E\|$	Number of vertices, vertex features, and edges
m^i	Number of features in source i
d	Number of sources of vertex features
k	Number of clusters to be discovered
$\mathbf{D}$	Vertex adjacency matrix
$\mathbf{F}^i$	Node-feature matrix in source i
$\mathbf{Y}$	Vertex structural proximity matrix
$\mathbf{P}$	Vertex propagationality matrix
$\mathbf{V}$	$n \times k$ latent space representing vertex-cluster preference
$\mathbf{U}^i$	$m^i \times k$ latent space representing feature-cluster preference in feature source i
$\mathbf{H}$	$n \times k$ latent space representing vertex-propagation preference
α, β	Model parameters

non-negative real matrices $\mathbf{Y}$ and $\mathbf{P}$ with the size $n \times n$ to represent structural proximity, and feature propagationality between two connected vertices, respectively. Three non-negative matrices, $n \times k$ latent space $\mathbf{V}$, $m^i \times k$ latent space $\mathbf{U}^i$, and $n \times k$ latent space $\mathbf{H}$, which respectively represent the shared cluster preference of each vertex, feature-cluster contributions of source i, and vertex-propagation preference are used by MSPANC to model the clustering problem in multi-source featured network. $\mathbf{D}_{ij}$ denotes the (i, j)-th element of a matrix $\mathbf{D}$. $\|\cdot\|_F$ denotes the Frobenius norm, and $tr(\cdot)$ denotes the trace operator. The notations used in this paper have been summarized in Table 1.

2.2. Multi-source vertex feature clustering

Vertex features from different sources may individually influence the cluster preference of each vertex in the network. MSPANC intends to infer two latent spaces ($\mathbf{V}$ and $\mathbf{U}^i$), which respectively represent the vertex-cluster preference and feature-cluster contributions in source i, to effectively capture the latent effects from multi-source vertex features. Given $\mathbf{F}^i$ representing the vertex features from source i, MSPANC assumes that $\mathbf{F}^i$ is generated by the product of $\mathbf{U}^i$ and $\mathbf{V}^T$, i.e., $\mathbf{F}^i_{jk} = [\mathbf{U}^i\mathbf{V}^T]_{jk} + \xi_{jk}$, where ξ_{jk} denotes the error term. Thus, the problem of multi-source vertex feature clustering can be formulated as minimizing the following objective function:

$$O_1 = \sum_i \left\| \mathbf{F}^i - \mathbf{U}^i\mathbf{V}^T \right\|_F^2, \tag{1}$$

Through minimizing the objective function in Eq. (1), a shared optimal preference between n vertices and k clusters, and a source-wise preference between m^i features in source i and k clusters can be respectively learned by MSPANC.

2.3. Vertex proximity modeling

As topological information is the foundation of the network data, it is essential to appropriately model the interrelationship between pairwise vertices for network clustering. To capture the true structure of the network, MSPANC attempts to model the local structural proximity between vertices pairs. Different from traditional methods that consider the edge structure, a.k.a. first-order proximity is generated by the corresponding latent space, MSPANC concerns the second-order proximity rather than the original edge structure to avoid the noise brought by those edges connecting vertices whose structures are very distinct. Inspired by the diffusion theory [4], we propose the following method to evaluate the vertex proximity ($\mathbf{Y}$) in the network:

$$\mathbf{Y}_{ij} = \begin{cases} \frac{[\mathbf{D}^T\mathbf{D}+\mathbf{D}]_{ij}}{2d_i} + \frac{[\mathbf{D}^T\mathbf{D}+\mathbf{D}]_{ji}}{2d_j} & \text{if } \mathbf{D}_{ij} = 1, \\ 0 & \text{otherwise}, \end{cases} \tag{2}$$

where d_i denotes the degree of vertex i in the network. Compared with prevalent methods which are generally based on Cosine and Jaccard similarity, MSPANC computes the second-order proximity between connected vertices following a different way. When two vertices, say v_i and v_j are connected, MSPANC firstly computes the total number of vertices ($[\mathbf{D}^T\mathbf{D}+\mathbf{D}]_{ij}$ or $[\mathbf{D}^T\mathbf{D}+\mathbf{D}]_{ji}$) that link both v_i and v_j in the network. MSPANC then obtains two proportions by dividing the total degrees of v_i and v_j, i.e., d_i and d_j, into $[\mathbf{D}^T\mathbf{D}+\mathbf{D}]_{ij}$ and $[\mathbf{D}^T\mathbf{D}+\mathbf{D}]_{ji}$. As d_i and d_j are generally different, MSPANC averages the obtained two proportions to preserve the symmetry of $\mathbf{Y}$. As Eq. (2) shows, only those connected vertices sharing similar local structure may be obtained higher degrees of proximity, so that the noise brought by the original edge structure is reduced.

Given O_1 in Eq. (1), MSPANC attempts to learn a shared latent space $\mathbf{V}$ from multiple sources of vertex features. Additionally, we assume that the proximity between any pair of vertices is generated by $\mathbf{V}$ and $\mathbf{V}^T$, i.e., $\mathbf{Y}_{ij}=[\mathbf{V}\mathbf{V}^T]_{ij}+\gamma_{ij}$, where γ_{ij} denotes the error term. Thus, the proximity modeling used by MSPANC is formulated as minimizing the following cost function:

$$O_2=\left\|\mathbf{Y}-\mathbf{V}\mathbf{V}^T\right\|_F^2. \tag{3}$$

By minimizing O_2 in Eq. (3), the learning of the shared latent space $\mathbf{V}$ is also influenced by vertex proximity $\mathbf{Y}$ of the network.

2.4. Multi-source feature propagation modeling

Although multi-source vertex feature clustering is capable of learning the features characterizing those discovered clusters, it cannot appropriately capture the frequency that a vertex feature might be propagated within a cluster. In fact, the vertex-cluster preference might be influenced by such propagation frequency as vertices tend to form a cluster with those sharing frequently propagated features. Besides considering the modeling of multi-source features and vertex proximity, MSPANC therefore attempts to acquire the influence of feature propagationality between pairwise vertices, when inferring the vertex-cluster preference.

The propagationality of vertices in the network is always different and can be influenced by the local structure of the network. Inspired by [25], we firstly compute the structural propagationality making use of the network transition matrix. Let $\mathbf{R}$ and $\mathbf{T}$ be the propagation and transition matrix, respectively. We assume that the propagationality between any pair of vertices in the network is invariant and can be computed by:

$$\mathbf{R}_{ij}=\begin{cases}\eta[\mathbf{R}\mathbf{T}]_{ij} & \text{if } i\neq j,\\ \phi+[\mathbf{R}\mathbf{T}]_{ii} & \text{otherwise,}\end{cases}\quad \mathbf{T}_{ij}=\begin{cases}\frac{\mathbf{D}_{ij}}{d_i} & \text{if } i\neq j,\\ 1e-9 & \text{otherwise,}\end{cases} \tag{4}$$

where η is the damping coefficient of the propagation process, and ϕ is a positive constant regarding the probability that a vertex is propagating a feature to itself. Through simple mathematical transmissions, $\mathbf{R}$ can be solved by:

$$\mathbf{R}=(\phi\mathbf{I})(\mathbf{I}-\eta\mathbf{T})^{-1} \tag{5}$$

In this paper, we typically set $\phi=1$ and $\eta=0.5$ for all the experiments. Given $\mathbf{R}$ and jth feature in source i, the degree that kth vertex propagates this feature can then be obtained via $\sum_l \mathbf{F}^i_{jl}\cdot\mathbf{R}_{lk}$. As shown in Eqs. (4) and (5), $\mathbf{R}$ represents the structural propagationality of each vertex in the network. Thus, $\sum_l \mathbf{F}^i_{jl}\cdot\mathbf{R}_{lk}$ may quantify how frequently a vertex feature is propagated from one vertex to others, given the structure possessed by the vertex in the network. Having obtained vertex-feature propagationality in all the sources, the vertex-vertex propagation regarding multi-source features can be obtained by:

$$\mathbf{P}_{jk}=\begin{cases}\frac{1}{d}\sum_{i=1}^d \mathbf{P}^i_{jk} & \text{if } \mathbf{D}_{jk}=1,\\ 0 & \text{otherwise}\end{cases},\mathbf{P}^i_{jk}=\frac{[(\mathbf{F}^i\mathbf{R})^T(\mathbf{F}^i\mathbf{R})]_{jk}}{\sqrt{d_{\mathbf{F}^i_{:,j}}d_{\mathbf{F}^i_{:,k}}}}, \tag{6}$$

where $d_{\mathbf{F}^i_{:,j}}$ represents the total number of features in ith source possessed by vertex j. Given Eq. (6), it is seen that the feature propagationality between a pair of linked vertices is determined by both structural and feature similarity between a pair of connected vertices. When the corresponding vertices share similar structural propagationality ($\mathbf{R}$) and multi-source features, the values in $\mathbf{P}$ become higher, indicating higher frequencies of vertex features propagated among connected vertices. Having obtained **P**, MSPANC attempts to model the propagation of multi-source features within each cluster using **H** and **V**, i.e., the matrices of propagation preference and vertex-cluster preference. And this goal can be achieved via minimizing the following cost function:

$$O_3 = \|\mathbf{P} - \mathbf{H}\mathbf{V}^T\|_F^2, \tag{7}$$

Given Eq. (7), the learning of cluster preference is additionally influenced by vertex-vertex propagation of multi-source features, **V** is thereby forced to be embedded with more information on the vertex propagationality in each cluster. And those vertices sharing similar features that are frequently propagated in the network are possible to be assigned with analogous cluster labels by the proposed model in the learning process.

2.5. The unified objective function

Combining O_1, O_2, and O_3 with balancing parameters and constraints, the unified objective function of MSPANC can be defined as follows:

$$\begin{aligned} &\text{minimize} \\ &O = \underbrace{\sum_i \left\|\mathbf{F}^i - \mathbf{U}^i\mathbf{V}^T\right\|_F^2}_{Multi-source\ feature\ modeling} + \underbrace{\alpha\left\|\mathbf{Y} - \mathbf{V}\mathbf{V}^T\right\|_F^2}_{Network\ proximity\ modeling} + \underbrace{\beta\|\mathbf{P} - \mathbf{H}\mathbf{V}^T\|_F^2}_{Multi-source\ propagation\ modeling} \\ &\text{subject to } \mathbf{V} \geq 0, \mathbf{U}^i \geq 0, \mathbf{H} \geq 0, \end{aligned} \tag{8}$$

where α, and β are balancing parameters which are used to control the relative significance of proximity and propagation modeling. By minimizing the unified objective function in Eq. (8), the optimal cluster preferences for all the n vertices can be learned from multi-source vertex features and network topology, and the learning of **V** is also regularized by the optimization of intra-cluster feature propagation so that the group assignment for propagation optimization is also implanted into **V**.

According to Eq. (8), MSPANC demonstrates to be fundamentally different from previous approaches to AN clustering and multi-layer network clustering. Compared with previous approaches to AN clustering, the model structure of MSPANC is more generic and considers more factors that may influence the discovery of network clusters. MSPANC attempts to model the structural proximity between pairwise vertices using the latent space **V** learned from multi-source features. Meanwhile, the learning of **V** is also regularized to consider the propagationality of multi-source features between pairwise vertices. Those vertices sharing similar local structure, analogous multi-source features that are propagated with a high frequency are possible to be assigned with similar cluster labels by the proposed model. Such a learning scheme is novel to AN clustering. MSPANC is also different from approaches to multi-layer network clustering as it assumes that the cluster preference for each vertex in the network is directly learned from multi-source features and topology. Using such a modeling method, MSPANC is able to well capture the source-wise effect brought by the vertex features and directly learn the latent descriptions of clusters, which makes the discovered clusters interpretable. While, multi-layer network clustering usually does not possess the functionality of learning cluster features as it transfers multi-source features to multiple networks.

2.6. Model optimization

In general, the objective function in Eq. (8) is non-convex and does not have an analytical solution. Fortunately, it is convex with respect to **V**, **H**, or $\mathbf{U}^i$ when the others are fixed. Therefore, we are able to derive an algorithm that can iteratively optimize **V**, **H**, and $\mathbf{U}^i$ till Eq. (8) converges. The details of the algorithm are illustrated as follows.

2.6.1. *Updating* V

Letting γ_{jk} be the Lagrange multiplier for $\mathbf{V}_{jk} \geq 0$, we are able to formulate the Lagrange function for **V**:

$$L(\mathbf{V}, \gamma) = O - tr(\gamma^T\mathbf{V}). \tag{9}$$

According to KKT conditions for constrained optimization, we can obtain the following element-wise equation system:

$$
\begin{aligned}
&\frac{\partial L}{\partial \mathbf{V}_{jk}} = 4\alpha[\mathbf{V}\mathbf{V}^T\mathbf{V}-\mathbf{Y}\mathbf{V}]_{jk}+2\beta[\mathbf{V}\mathbf{H}^T\mathbf{H}-\mathbf{P}\mathbf{H}]_{jk}+2\sum_i[\mathbf{V}\mathbf{U}^{iT}\mathbf{U}^i-\mathbf{F}^{iT}\mathbf{U}^i]_{jk}-\gamma_{jk}=0,\\
&\gamma_{jk}\cdot\mathbf{V}_{jk}=0,\\
&\gamma_{jk}\geq 0.
\end{aligned}
\tag{10}
$$

Solving γ_{jk} in the first equation in (10) and substituting it in the second one, we have the following:

$$
[4\alpha\mathbf{V}\mathbf{V}^T\mathbf{V}-4\alpha\mathbf{Y}\mathbf{V}-2\beta\mathbf{P}\mathbf{H}+2\beta\mathbf{V}\mathbf{H}^T\mathbf{H}+2\sum_i\mathbf{V}\mathbf{U}^{iT}\mathbf{U}^i-\mathbf{F}^{iT}\mathbf{U}^i]_{jk}\cdot\mathbf{V}_{jk}=0.
\tag{11}
$$

Eq. (11) can be equivalently rewritten as follows through some mathematical transformations:

$$
\begin{aligned}
&[4\alpha\mathbf{V}\mathbf{V}^T\mathbf{V}+\sum_i\mathbf{V}\mathbf{U}^{iT}\mathbf{U}^i+\beta\mathbf{V}\mathbf{H}^T\mathbf{H}]_{jk}^2\cdot\mathbf{V}_{jk}^4=[\sum_i[\mathbf{V}\mathbf{U}^{iT}\mathbf{U}^i+\beta\mathbf{V}\mathbf{H}^T\mathbf{H}]_{jk}^2+8\alpha[\mathbf{V}\mathbf{V}^T\mathbf{V}]_{jk}\\
&\cdot[2\alpha\mathbf{Y}\mathbf{V}+\sum_i\mathbf{F}^{iT}\mathbf{U}^i+\beta\mathbf{P}\mathbf{H}]_{jk}]\cdot\mathbf{V}_{jk}^4.
\end{aligned}
\tag{12}
$$

Solving Eq. (12), we can obtain the rule for updating $\mathbf{V}$:

$$
\begin{aligned}
&\mathbf{V}_{jk}\leftarrow\mathbf{V}_{jk}\cdot\frac{\sqrt{\sqrt{\Delta_{jk}}-[\sum_i\mathbf{V}\mathbf{U}^{iT}\mathbf{U}^i+\beta\mathbf{V}\mathbf{H}^T\mathbf{H}]_{jk}}}{\sqrt{4\alpha[\mathbf{V}\mathbf{V}^T\mathbf{V}]_{jk}}},\\
&\Delta_{jk}=[\sum_i\mathbf{V}\mathbf{U}^{iT}\mathbf{U}^i+\beta\mathbf{V}\mathbf{H}^T\mathbf{H}]_{jk}^2+[8\alpha\mathbf{V}\mathbf{V}^T\mathbf{V}]_{jk}[2\alpha\mathbf{Y}\mathbf{V}+\sum_i\mathbf{F}^{iT}\mathbf{U}^i+\beta\mathbf{P}\mathbf{H}]_{jk}.
\end{aligned}
\tag{13}
$$

2.6.2. Updating $\mathbf{U}^i$ *and* $\mathbf{H}$

We are able to derive the updating rules for $\mathbf{U}^i$ and $\mathbf{H}$ following the similar way. Letting γ_{jk} be the Lagrange multiplier for $\mathbf{U}^i_{jk}\geq 0$, we are able to formulate the Lagrange function for $\mathbf{U}^i$ shown as follows:

$$
L(\mathbf{U}^i,\eta)=O-tr(\gamma^T\mathbf{U}^i).
\tag{14}
$$

According to KKT conditions for constrained optimization, we can obtain the following element-wise equation system:

$$
\begin{aligned}
&\frac{\partial L}{\partial \mathbf{U}^i_{jk}}=2[\mathbf{U}^i\mathbf{V}^T\mathbf{V}-\mathbf{F}^i\mathbf{V}]_{jk}-\gamma_{jk}=0,\\
&\gamma_{jk}\cdot\mathbf{U}^i_{jk}=0,\\
&\gamma_{jk}\geq 0.
\end{aligned}
\tag{15}
$$

Solving the above equation system, we are able to derive the rule for updating $\mathbf{U}^i$:

$$
\mathbf{U}^i_{jk}\leftarrow\mathbf{U}^i_{jk}\cdot\frac{[\mathbf{F}^i\mathbf{V}]_{jk}}{[\mathbf{U}^i\mathbf{V}^T\mathbf{V}]_{jk}}.
\tag{16}
$$

Letting γ_{jk} be the Lagrange multiplier for $\mathbf{H}_{jk}\geq 0$, we are able to formulate the Lagrange function for latent variables in $\mathbf{H}$ shown as follows:

$$
L(\mathbf{H},\gamma)=O-tr(\gamma^T\mathbf{H}).
\tag{17}
$$

According to KKT conditions for constrained optimization, we can obtain the following element-wise equation system:

$$
\begin{aligned}
&\frac{\partial L}{\partial \mathbf{H}_{jk}}=2\beta[\mathbf{H}\mathbf{V}^T\mathbf{V}-\mathbf{P}\mathbf{V}]_{jk}-\gamma_{jk}=0,\\
&\gamma_{jk}\cdot\mathbf{H}_{jk}=0,\\
&\gamma_{jk}\geq 0.
\end{aligned}
\tag{18}
$$

Solving the above equation system, we are able to derive the rule for updating $\mathbf{H}$:

$$\mathbf{H}_{jk} \leftarrow \mathbf{H}_{jk} \cdot \frac{[\mathbf{PV}]_{jk}}{[\mathbf{HV}^T\mathbf{V}]_{jk}}. \tag{19}$$

The objective function in Eq. (8) will finally converge, when the latent variables in $\mathbf{V}$, $\mathbf{U}^i$ and $\mathbf{H}$ are updated according to Eqs. (13), (16), and (19), respectively. The optimization procedure of MSPANC is summarized in Algorithm 1.

Algorithm 1 Multi-Source Propagation Aware Network Clustering (MSPANC)

```
Input: Multi-Source Featured Network Data: $\mathbf{Y}$, $\{\mathbf{F}^i\}_{i=1}^d$
Output: Cluster preference for each vertex: $\mathbf{V}$; Source-wise feature-cluster contribution: $\{\mathbf{U}^i\}_{i=1}^d$, Propagation-cluster preference $\mathbf{H}$
  Compute $\mathbf{P}$;
  for $i \leftarrow 1 : d$ do
      Initialize $\mathbf{U}^i$;
  end for
  Initialize $\mathbf{V}$, $\mathbf{H}$;
  $t \leftarrow 0$;
  while $t < T_{max}$ do
      $t \leftarrow t + 1$;
      Update $\mathbf{V}$ by Eq. (13);
      Update $\mathbf{H}$ by Eq. (19);
      for $i \leftarrow 1 : d$ do
          Update $\mathbf{U}^i$ by Eq. (16);
      end for
      Compute objective value $O^{(t)}$ by Eq. (8);
      if $O^{(t-1)} - O^{(t)} \leq \epsilon$ then
          break;
      end if
  end while
  Identify cluster label for each vertex using $\mathbf{V}$;
```

3. Theoretical analysis on MSPANC

In this section, we conduct the theoretical analysis on the proposed model by proving the convergence of the optimization algorithm derived in the last section and analyze the computational complexity of the model.

3.1. Convergence analysis

The convergence of the model can be proved if we utilize one property of an auxiliary function which is also used in the validation of the Expectation-Maximization algorithm [5]. The property of the auxiliary function is described as follows. If there exists an auxiliary function satisfying the conditions that $Q(x, x') \geq F(x)$ and $Q(x, x) = F(x)$, then F is non-increasing under the following updating rule:

$$x^{(t+1)} = \arg\min_x Q(x, x^{(t)}). \tag{20}$$

The equality $F(x^{(t+1)}) = F(x^{(t)})$ holds if and only if x is a local minimum of $Q(x, x')$. By iteratively updating x according to Eq. (20), F will converge to the local minimum $x^{min} = \arg\min_x F(x)$. We can show the model convergence by defining an appropriate auxiliary function for the objective function in Eq. (8).

3.1.1. Convergence analysis on $\mathbf{V}$

First we validate the convergence of O in Eq. (8) when $\mathbf{V}$ is updated according to Eq. (13). Here, only those terms in Eq. (8) related to the latent variables in $\mathbf{V}$ need considering. Thus, we have:

$$O_{\mathbf{V}}=\alpha\cdot tr(\mathbf{V}\mathbf{V}^T\mathbf{V}\mathbf{V}^T)+\sum_i tr(\mathbf{V}\mathbf{U}^{iT}\mathbf{U}^i\mathbf{V}^T)+\beta\cdot tr(\mathbf{V}\mathbf{H}^T\mathbf{H}\mathbf{V}^T)-2\beta\cdot tr(\mathbf{P}\mathbf{H}\mathbf{V}^T)-2\alpha\cdot tr(\mathbf{Y}\mathbf{V}\mathbf{V}^T)-2\sum_i tr(\mathbf{F}^{iT}\mathbf{U}^i\mathbf{V}^T), \tag{21}$$

where $O_{\mathbf{V}}$ represents the sum of terms in Eq. (8) that are related to the variables in $\mathbf{V}$. According to Lemma 2 in [23], we have:

$$\begin{aligned}
&-2\alpha\cdot tr(\mathbf{Y}\mathbf{V}\mathbf{V}^T)=-2\alpha\cdot tr(\mathbf{V}^T\mathbf{Y}\mathbf{V})\leq-2\alpha[tr(\mathbf{V}'^T\mathbf{Y}\mathbf{Z})+tr(\mathbf{Z}^T\mathbf{Y}\mathbf{V}'^T)+tr(\mathbf{V}'^T\mathbf{Y}\mathbf{V}')],\\
&-2\sum_i tr(\mathbf{F}^{iT}\mathbf{U}^i\mathbf{V}^T)\leq-2\sum_i tr(\mathbf{Z}^T\mathbf{F}^{iT}\mathbf{U}^i)+tr(\mathbf{V}'^T\mathbf{F}^{iT}\mathbf{U}^i),\\
&-2\beta\cdot tr(\mathbf{P}\mathbf{H}\mathbf{V}^T)\leq-2\beta[tr(\mathbf{Z}^T\mathbf{P}\mathbf{H})+tr(\mathbf{V}'^T\mathbf{P}\mathbf{H})],\\
&\mathbf{Z}_{jk}=\mathbf{V}'_{jk}log\frac{\mathbf{V}_{jk}}{\mathbf{V}'_{jk}}.
\end{aligned}\tag{22}$$

Apparently, the equalities hold when $\mathbf{V}_{jk}=\mathbf{V}'_{jk}$. According to Lemmas 6 and 7 in [23], we have:

$$\begin{aligned}
&tr(\mathbf{V}\mathbf{U}^{iT}\mathbf{U}^i\mathbf{V}^T)\leq tr(\mathbf{V}'\mathbf{U}^{iT}\mathbf{U}^i\mathbf{W}^T),\\
&\beta\cdot tr(\mathbf{V}\mathbf{H}^T\mathbf{H}\mathbf{V}^T)\leq\beta\cdot tr(\mathbf{V}'\mathbf{H}^T\mathbf{H}\mathbf{W}^T),\\
&\alpha\cdot tr(\mathbf{V}\mathbf{V}^T\mathbf{V}\mathbf{V}^T)\leq\alpha\cdot tr(\mathbf{V}'\mathbf{V}'^T\mathbf{V}'\mathbf{X}^T),\\
&\mathbf{W}_{jk}=\frac{\mathbf{V}^2_{jk}}{\mathbf{V}'_{jk}},\mathbf{X}_{jk}=\frac{\mathbf{V}^4_{jk}}{\mathbf{V}'^3_{jk}}.
\end{aligned}\tag{23}$$

Similar to Eq. (22), the equalities holds when $\mathbf{V}_{jk}=\mathbf{V}'_{jk}$. According to Eqs. (22) and (23), we may define the auxiliary function for proving the convergence of Eq. (21) as follows:

$$\begin{aligned}
&Q(\mathbf{V},\mathbf{V}')=\alpha\cdot tr(\mathbf{V}'\mathbf{V}'^T\mathbf{V}'\mathbf{X}^T)+\sum_i tr(\mathbf{V}'\mathbf{U}^{iT}\mathbf{U}^i\mathbf{W}^T)+2\beta[0.5tr(\mathbf{V}'\mathbf{H}^T\mathbf{H}\mathbf{W}^T)-tr(\mathbf{Z}^T\mathbf{P}\mathbf{H})-tr(\mathbf{V}'^T\mathbf{P}\mathbf{H})]\\
&-2\alpha[tr(\mathbf{V}'^T\mathbf{Y}\mathbf{Z})+tr(\mathbf{Z}^T\mathbf{Y}\mathbf{V}'^T)+tr(\mathbf{V}'^T\mathbf{Y}\mathbf{V}')]-2\sum_i tr(\mathbf{Z}^T\mathbf{F}^{iT}\mathbf{U}^i)+tr(\mathbf{V}'^T\mathbf{F}^{iT}\mathbf{U}^i).
\end{aligned}\tag{24}$$

According to Eq. (24), for any element in $\mathbf{V}$, say $\mathbf{V}_{jk}$, the auxiliary function can be written as:

$$\begin{aligned}
&Q(\mathbf{V}_{jk},\mathbf{V}'_{jk})=\alpha[\mathbf{V}'\mathbf{V}'^T\mathbf{V}']_{jk}\mathbf{X}_{jk}-4\alpha[\mathbf{Y}\mathbf{V}']_{jk}\mathbf{Z}_{jk}+\beta[\mathbf{V}'\mathbf{H}^T\mathbf{H}]_{jk}\mathbf{W}_{jk}-2\beta[\mathbf{P}\mathbf{H}]_{jk}\mathbf{Z}_{jk}\\
&+\sum_i[\mathbf{V}'\mathbf{U}^{iT}\mathbf{U}^i]_{jk}\mathbf{W}_{jk}-2\sum_i[\mathbf{F}^{iT}\mathbf{U}^i]_{jk}\mathbf{Z}_{jk}.
\end{aligned}\tag{25}$$

Since Eq. (25) is an auxiliary function of $\mathbf{V}_{jk}$, $Q(\mathbf{V}_{jk},\mathbf{V}'_{jk})\geq O_{\mathbf{V}_{jk}}$, where $O_{\mathbf{V}_{jk}}$ represents terms in Eq. (8) relevant to $\mathbf{V}_{jk}$. An updating rule for minimizing $Q(\mathbf{V}_{jk},\mathbf{V}'_{jk})$ can therefore be derived according to Eq. (20). Taking the derivative of Eq. (25) w.r.t. $\mathbf{V}_{jk}$ and letting it equal zero, we have:

$$\begin{aligned}
&\mathbf{V}^{(t+1)}_{jk}=\arg\min_{\mathbf{V}_{jk}}Q(\mathbf{V}_{jk},\mathbf{V}'_{jk})\Rightarrow\frac{\partial Q(\mathbf{V}_{jk},\mathbf{V}'_{jk})}{\partial\mathbf{V}_{jk}}=4\alpha[\mathbf{V}'\mathbf{V}'^T\mathbf{V}']_{jk}\frac{\mathbf{V}^3_{jk}}{\mathbf{V}'^3_{jk}}+2[\sum_i\mathbf{V}'\mathbf{U}^{iT}\mathbf{U}^i+\beta\mathbf{V}'\mathbf{H}^T\mathbf{H}]_{jk}\frac{\mathbf{V}_{jk}}{\mathbf{V}'_{jk}}\\
&-[4\alpha\mathbf{Y}\mathbf{V}'+2\beta\mathbf{P}\mathbf{H}+2\sum_i\mathbf{F}^{iT}\mathbf{U}^i]_{jk}\frac{\mathbf{V}'_{jk}}{\mathbf{V}_{jk}}=0.
\end{aligned}\tag{26}$$

Eq. (26) can be equivalently rewritten as:

$$\begin{aligned}&2\alpha[\mathbf{V}'\mathbf{V}'^T\mathbf{V}']_{jk}\mathbf{V}_{jk}^4+\sum_i[\mathbf{V}'\mathbf{U}^{iT}\mathbf{U}^i]_{jk}\mathbf{V}_{jk}'^2\mathbf{V}_{jk}^2+\beta[\mathbf{V}'\mathbf{H}^T\mathbf{H}]_{jk}\mathbf{V}_{jk}'^2\mathbf{V}_{jk}^2-2\alpha[\mathbf{Y}\mathbf{V}']_{jk}\mathbf{V}_{jk}'^4\\&-\beta[\mathbf{P}\mathbf{H}]_{jk}\mathbf{V}_{jk}'^4-\sum_i[\mathbf{F}^{iT}\mathbf{U}^i]_{jk}\mathbf{V}_{jk}'^4=0.\end{aligned}\tag{27}$$

Utilizing the quadratic formula, we have:

$$\begin{aligned}&\mathbf{V}_{jk}^2=\frac{\mathbf{V}_{jk}'^2\sqrt{\Delta_{jk}}-[\beta\mathbf{V}'\mathbf{H}^T\mathbf{H}+\sum_i\mathbf{V}'\mathbf{U}^{iT}\mathbf{U}^i]_{jk}\mathbf{V}_{jk}'^2}{4\alpha[\mathbf{V}'\mathbf{V}'^T\mathbf{V}']_{jk}},\\&\Delta_{jk}=[\beta\mathbf{V}'\mathbf{H}^T\mathbf{H}+\sum_i\mathbf{V}'\mathbf{U}^{iT}\mathbf{U}^i]_{jk}^2+[8\alpha\mathbf{V}'\mathbf{V}'^T\mathbf{V}']_{jk}\cdot[2\alpha\mathbf{Y}\mathbf{V}'+\beta\mathbf{P}\mathbf{H}+\sum_i\mathbf{F}^{iT}\mathbf{U}^i]_{jk}.\end{aligned}\tag{28}$$

Thus, we can obtain:

$$\mathbf{V}_{jk}^{(t+1)}\leftarrow\mathbf{V}_{jk}'\cdot\frac{\sqrt{\sqrt{\Delta_{jk}}-[\beta\mathbf{V}'\mathbf{H}^T\mathbf{H}+\sum_i\mathbf{V}'\mathbf{U}^{iT}\mathbf{U}^i]_{jk}}}{\sqrt{4\alpha[\mathbf{V}'\mathbf{V}'^T\mathbf{V}']_{jk}}}.\tag{29}$$

As shown above, the updating rule for minimizing $Q(\mathbf{V}_{jk},\mathbf{V}'_{jk})$ is equivalent to the rule shown in Eq. (13). As $Q(\mathbf{V}_{jk},\mathbf{V}'_{jk})$ is an auxiliary function of $O_{\mathbf{V}_{jk}}$, O is non-increasing when it updates $\mathbf{V}_{jk}$ according to Eq. (13).

3.1.2. Convergence analysis on $\mathbf{U}^i$ *and* $\mathbf{H}$

Following the same way, we may prove the convergence of the unified objective function when $\mathbf{U}^i$ and $\mathbf{H}$ are respectively updated according to Eqs. (16) and (19). Similar to construct the auxiliary function for $\mathbf{V}_{jk}$, we define the following auxiliary functions for $\mathbf{U}_{jk}^i$ and $\mathbf{H}_{jk}$:

$$\begin{aligned}&Q(\mathbf{H}_{jk},\mathbf{H}'_{jk})=O_{\mathbf{H}_{jk}}+O'_{\mathbf{H}_{jk}}(\mathbf{H}_{jk}-\mathbf{H}'_{jk})+\frac{[\mathbf{H}'\mathbf{V}^T\mathbf{V}]_{jk}}{\mathbf{H}'_{jk}}(\mathbf{H}_{jk}-\mathbf{H}'_{jk})^2,\\&Q(\mathbf{U}_{jk}^i,\mathbf{U}_{jk}^{i\prime})=O_{\mathbf{U}_{jk}}+O'_{\mathbf{U}_{jk}}(\mathbf{U}_{jk}^i-\mathbf{U}_{jk}^{i\prime})+\frac{[\mathbf{U}^{i\prime}\mathbf{V}^T\mathbf{V}]_{jk}}{\mathbf{U}_{jk}^{i\prime}}(\mathbf{U}_{jk}^i-\mathbf{U}_{jk}^{i\prime})^2,\end{aligned}\tag{30}$$

where $O_{\mathbf{U}_{jk}}$ and $O'_{\mathbf{U}_{jk}}$ are the terms w.r.t. $\mathbf{U}_{jk}^i$ in Eq. (8), and the first-order partial derivative of Eq. (8) w.r.t. $\mathbf{U}_{jk}^i$, respectively. As proving the above two are auxiliary functions for $\mathbf{H}_{jk}$ and $\mathbf{U}_{jk}^i$ is similar to what has been shown in Eqs. (22)-(25), we omit the details of the proof. As what are shown in Eq. (30) are auxiliary functions of $\mathbf{H}_{jk}$ and $\mathbf{U}_{jk}^i$, we can verify the updating rule w.r.t. $\mathbf{H}_{jk}$ and $\mathbf{U}_{jk}^i$ by finding the local minimum of Eq. (30):

$$\begin{aligned}&\mathbf{H}_{jk}^{(t+1)}=\arg\min_{\mathbf{H}_{jk}}Q(\mathbf{H}_{jk},\mathbf{H}'_{jk})\Rightarrow\mathbf{H}_{jk}^{(t+1)}\leftarrow\mathbf{H}'_{jk}\cdot\frac{[\mathbf{P}\mathbf{V}]_{jk}}{[\mathbf{H}'\mathbf{V}^T\mathbf{V}]_{jk}},\\&\mathbf{U}_{jk}^{i(t+1)}=\arg\min_{\mathbf{U}_{jk}^i}Q(\mathbf{U}_{jk}^i,\mathbf{U}_{jk}^{i\prime})\Rightarrow\mathbf{U}_{jk}^{i(t+1)}\leftarrow\mathbf{U}_{jk}^{i\prime}\cdot\frac{[\mathbf{F}^i\mathbf{V}]_{jk}}{[\mathbf{U}^{i\prime}\mathbf{V}^T\mathbf{V}]_{jk}}.\end{aligned}\tag{31}$$

As shown in Eq. (31), the updating rules for minimizing $Q(\mathbf{H}_{jk},\mathbf{H}'_{jk})$ and $Q(\mathbf{U}_{jk}^i,\mathbf{U}_{jk}^{i\prime})$ are equivalent to the updating rules in Eqs. (16) and (19). As $Q(\mathbf{H}_{jk},\mathbf{H}'_{jk})$ and $Q(\mathbf{U}_{jk}^i,\mathbf{U}_{jk}^{i\prime})$ are auxiliary functions of O w.r.t. $\mathbf{H}_{jk}$, and $\mathbf{U}_{jk}^i$ respectively, O is verified to be non-increasing when $\mathbf{H}_{jk}$ and $\mathbf{U}_{jk}^i$ are updated according to Eqs. (16) and (19). Based on the above proof, we may conclude that the objective function Eq. (8) is non-increasing when the latent variables in $\mathbf{V}$, $\mathbf{U}^i$, and $\mathbf{H}$ are updated according to Eqs. (13), (16) and (19), respectively.

Table 2
Statistics of the testing datasets. Soc, Bio or Doc represents the dataset is a social, biological, or document network.

Dataset	Type	n	$\lvert E\rvert$	m	d	k
Ego	Soc	4039	88234	1283	1	191
Twitter	Soc	3687	49881	20905	2	242
Gplus	Soc	107614	3755989	13966	5	463
DIP	Bio	4579	20845	4237	3	200
Wiki	Doc	2405	17981	4973	1	17

3.2. Computational complexity analysis

The computational complexity of MSPANC can be analyzed according to the updating rules shown in Eqs. (13), (16) and (19). Given the iterative algorithm which has been derived in Section 2, updating corresponding latent variables in $\mathbf{V}$, $\mathbf{U}^i$, and $\mathbf{H}$ follows the order of $O(2n^2k + 5nk^2 + mk^2 + nmk)$, $O(m^i nk + (n + m^i)k^2)$, and $O(n^2k + 2nk^2)$, respectively. Therefore, the overall complexity of the proposed model is approximately $O(n^2k + nmk)$.

4. Experimental results and analysis

In this section, we verify the effectiveness of the proposed model through comparing it with several competitive baselines on a number of real-world networks.

4.1. Experimental setup

4.1.1. Baselines for comparison

Eight computational approaches to network clustering are selected as baselines, including NCut [19], CNM [3], CoDA [29], k-means [15], CP-PI [11], CESNA [28], MISAGA [7], and CoNMF [1]. These baselines are able to perform the task of network clustering concerning different properties of the network data.

NCut, CNM, and CoDA are three effective methods for network clustering, which perform the task utilizing network topology. NCut is a spectral clustering approach for network data. It can perform the task through grouping those vertices sharing higher structural similarity. CNM is a typical heuristic approach to community detection, which performs the task via modularity maximization. CoDA is a state-of-the-art (SOTA) algorithm for uncovering clusters in the network. It is able to perform the task through symmetric probabilistic matrix factorization.

k-means is a well-known clustering model, which can detect network clusters utilizing the concatenation of vertex features from multiple sources.

CESNA, CP-PI, and MISAGA are three SOTA approaches to attributed network clustering. CESNA is a probabilistic-matrix-factorization model, which learns a shared latent space as vertex-cluster preference from edge and vertex feature matrices. CP-PI is a prevalent two-step approach to AN clustering. It first re-weights the vertex-feature matrix according to vertex transitivity in the network. Then, a mixture-model-like clustering method is used to uncover clusters in the re-weighted vertex-feature matrix. MISAGA is a prevalent approach to network clustering, which can infer a shared latent space from edge structure and pairwise similarity of vertex features as cluster preference.

Besides the aforementioned approaches to network clustering, CoNMF, which is an effective approach to multi-view clustering, is also selected as one of the baselines. As mentioned, the problem addressed in this paper can also be solved via multi-view clustering, we used CoNMF to uncover network clusters using multi-source vertex features and edge structure. Comparing the performance between MSPANC's and that obtained by CoNMF may further reveal the effectiveness of the proposed model.

In our experiments, we implemented all the compared baselines using the source codes which are publicly available. Specifically, CNM and CESNA do not require any parameter setting for clustering tasks. For the rest of the baselines, including NCut, CoDA, k-means, CP-PI, and MISAGA, we use the settings that are recommended in the corresponding papers. For the number of clusters, i.e., k, which has to be predetermined in NCut, CoDA, k-means, CoNMF, CP-PI, and MISAGA, we set it to equal the number of ground-truth clusters of the testing network. For the proposed MSPANC, we set $\alpha = 5$, and $\beta = 1$. The setting of k in MSPANC is the same as that in the baselines. All of the experiments were performed on a workstation with 6-core 3.4GHz CPU and 32GB RAM.

4.1.2. Dataset description

Five real-world networks, including three social networks, one biological network, and one document network are used as testing datasets. All these networks are with ground-truth clusters which have been verified in the previous study. As these networks are constructed based on different types of real-world data, they can verify the robustness of different methods.

$Ego-facebook$ (Ego) [16], $Twitter$ [28], and $Googleplus$ (Gplus) [16] are three social networks which are used to describe the interactions (edges) among social players (vertices) on different social networking sites. Besides, content features collected from different sources, e.g., social tags, comments, and user profiles are also used to characterize the vertices in each dataset. Specifically, there are 4039 vertices, 88234 edges, and 1283 vertex features in Ego, 3687 vertices, 49881 edges, and 20905 vertex features collected from two sources in Twitter, and 107614 vertices, 3755989 edges, and 13966 vertex features collected from five sources in Gplus, respectively. Social circles have been verified as ground-truth clusters in these datasets. Specifically, there are 191, 242, and 463 social circles in Ego, Twitter, and Gplus, respectively.

DIP [24] is a widely used biological network which describes the interactions between proteins in $Saccharomyces$ $cerevisiae$. There are 4579 vertices, and 20845 edges which represent proteins, and the protein-protein interactions, respectively. Besides, 4237 vertex features collected from three sources are used to characterize the proteins. 200 protein complexes have been verified in the laboratory experiments and they can be seen as ground-truth clusters of this dataset.

$Wiki$ [13] is an online document network collected from wikipedia. There are 2405 vertices, 17981 edges, and 4973 vertex features representing web-pages of wiki terms, hyperlinks between web-pages, and keywords of each term, respectively. 17 document classes have been verified in the previous study and they can be used as ground-truth clusters. The characteristics of these five testing datasets have been summarized in Table 2.

4.1.3. Evaluation metrics

Two widely used metrics, which are Normalized Mutual Information (NMI) and Accuracy (Acc) are considered in our experiments for evaluating the clustering performance of different approaches.

NMI measures the overall accuracy of the matches between the uncovered clusters and the ones in the ground-truth database. It is defined as follows:

$$NMI = \frac{\sum\limits_{C_i,C_j^*} Pr(C_i, C_j^*) log \frac{Pr(C_i,C_j^*)}{Pr(C_i)Pr(C_j^*)}}{\max(H(C), H(C^*))},$$
$$H(C) = -\sum_i Pr(C_i) log Pr(C_i), H(C^*) = -\sum_j Pr(C_j^*) log Pr(C_j^*), \tag{32}$$

where $\Pr(C_i, C_j^*)$ represents the probability that the vertices simultaneously exist in the uncovered cluster i and j-th cluster of ground-truth, and $\Pr(C_i)$ represents the probability that a vertex exists in cluster i. A higher value of NMI means a better matching between the uncovered clusters and the ones in the ground-truth database.

Different from NMI, Acc evaluates the best matching between each uncovered cluster and the one in the ground-truth database. It is defined as follows:

$$Acc = \sum_i \frac{|C_i|}{|C|} f(C_i, C^*), \tag{33}$$

where $|C_i|$ and $f(\cdot, \cdot)$ denote the size of i-th cluster, and the function searching for the best matching for an uncovered cluster in the ground-truth database, respectively. According to the definition of Acc, a higher value of it means a better matching between each detected cluster and the ground-truth. Since NMI and Acc take different emphases when evaluating the clustering performance, we may utilize them to comprehensively reveal the effectiveness of different approaches.

4.2. Clustering performance evaluation

Detecting clusters in various types of network data is related to a number of significant applications, such as social community detection, biological module identification, and document segmentation. In our experiment, we used the aforementioned three social networks, one biological network, and one document network to test the effectiveness of

Table 3
Clustering Performance Evaluated by NMI and Acc (%). The best performance is highlighted in bold.

Datasets / Approaches	Ego		Twitter		Gplus		DIP		Wiki	
	NMI	Acc	NMI	Acc	NMI	Acc	NMI	Acc	NMI	Acc
NCut [19]	53.646	44.689	48.421	42.121	5.122	14.921	82.687	4.695	8.638	17.588
CNM [3]	48.266	37.979	35.485	40.136	7.554	9.603	74.796	3.451	30.572	41.788
CoDA [29]	55.505	52.091	60.878	66.537	15.126	17.705	76.314	8.362	30.203	45.267
k-means [15]	40.461	29.116	14.929	28.343	6.421	10.385	80.757	8.167	35.214	44.407
CoNMF [1]	48.366	38.945	54.025	50.344	18.811	48.207	87.440	10.465	46.865	59.171
CESNA [28]	57.513	46.124	46.588	51.340	10.164	26.783	77.286	2.600	9.374	24.738
CP-PI [11]	48.981	36.915	48.538	51.179	–	–	87.499	9.528	44.164	55.385
MISAGA [7]	56.452	45.159	65.329	68.619	10.245	37.271	87.952	10.340	35.109	45.114
MSPANC	**64.794**	**52.835**	**69.179**	**72.244**	**21.154**	**61.261**	**91.372**	**15.185**	**49.833**	**61.211**
Improvement (%)	17.876	1.428	5.893	5.283	12.455	27.079	3.888	45.103	6.333	3.448

all the nine approaches. As the ground-truth clusters of all the mentioned five datasets are available, we can validate the clusters uncovered by different approaches against the ground-truth. The experimental results w.r.t. NMI and Acc of different methods are summarized in Table 3.

When the discovered clusters are evaluated by NMI, MSPANC performs the best in all the testing networks. The proposed approach may significantly outperform the second-best by at least 5% in four datasets out of five. Specifically, MSPANC is better than CESNA by 17.876% in Ego. In Twitter, MSPANC is better than MISAGA by 5.893%. When MSPANC is compared with the second-best in Wiki, the improvement in terms of NMI is 6.333%. In Gplus, which is the largest dataset used in our experiment, MSPANC outperforms CoNMF by 12.455%. It is worth noting that CP-PI cannot fulfill the clustering task in Gplus, which indicates it is not an approach designed for massive network datasets.

When the uncovered clusters are evaluated by the Acc metric, MSPANC still can obtain a robust performance. In three datasets, MSPANC is better than any other baseline by at least 5%. Specifically, in Twitter and DIP, the proposed model outperforms MISAGA by 5.283% and 45.103%, respectively. In Gplus, MSPANC is better than CoNMF by 27.079%.

According to the experimental results w.r.t. NMI and Acc, the proposed MSPANC demonstrates to be an effective model in AN clustering. Adopting the multi-source learning scheme, the proposed model is able to capture the source-wise impact on the cluster preference of each vertex. In addition, MSPANC also benefits from the modeling of multi-source propagation of vertex features, as it can regularize the model to infer similar cluster preferences for vertices sharing high propagationality regarding features from all the sources.

4.3. Parameter sensitivity test

In this section, we investigate how different combinations of α and β, which lead to different relative weights between proximity and propagation modeling, will impact the performance of network clustering. Specifically, we set $\alpha = [0.1, 0.5, 1, 5, 10, 20, 50, 100]$ and $\beta = [0.1, 0.5, 1, 5, 10, 20, 50, 100]$, and run MSPANC on all datasets. Then, the clusters discovered by MSPANC using different combinations of α and β are evaluated using NMI and Acc. We present the results obtained from Ego dataset here as an example (Fig. 2). As the figure shows, both NMI and Acc perform robustly when the value of α is relatively large, e.g., $\alpha \geq 5$. While, the proposed model performs well when β is set lower than 10. Based on the sensitivity test shown in Fig. 2, the performance of the proposed MSPANC is robust under a wide range of parameter combinations. For simplicity, we set $\alpha = 5$ and $\beta = 1$ in all our experiments.

4.4. Model convergence test

In addition to theoretically proving the model convergence, we also test whether the proposed model can achieve convergence when performing clustering tasks in real networks. Specifically, we recorded the objective function value of MSPANC for the first 300 iterations on all five datasets. As the figure shows, the objective function of the proposed model can achieve convergence in less than 200 iterations on all testing datasets, which demonstrates MSPANC can attain the desired clustering results in an efficient manner.

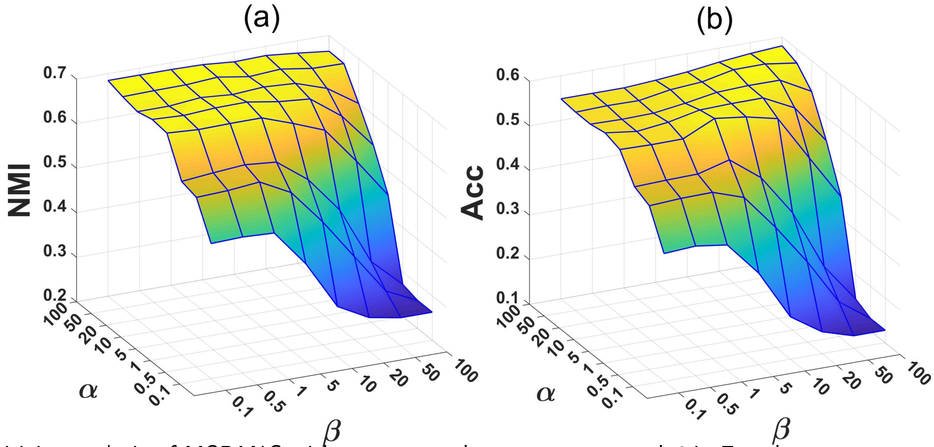


Figure 2: Sensitivity analysis of MSPANC with respect to the parameter α and β in Ego dataset

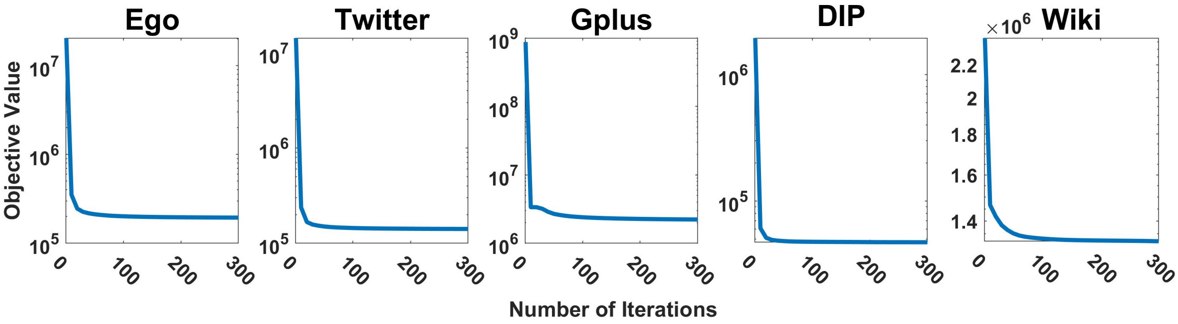


Figure 3: Model convergence on testing datasets.

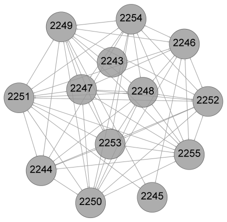
Table 4
Optimization time on different datasets

Approaches \ Datasets	Ego	Twitter	Gplus	DIP	Wiki
CoDA [29]	201s	131s	20540s	169s	118s
CESNA [28]	592s	1044s	23670s	1808s	1368s
MISAGA [7]	114s	115s	6692s	116s	103s
MSPANC	96s	171s	4878s	28s	142s

4.5. Comparison on model efficiency

Besides validating the convergence speed of the proposed model, we further investigate its scalability in real-world networks. Specifically, we recorded the optimization time of MSPANC in all the testing networks and compare it with that cost by other prevalent approaches that are also based on iterative optimization. The comparison between MSPANC and CoDA, CESNA, and MISAGA is summarized in Table 4. As the table shows, the proposed MSPANC can obtain a competitive performance in all the datasets. Its advantage on scalability becomes evident when the size of the network becomes large. For example, MSPANC only costs 4878 seconds to complete model optimization in Gplus, which is the largest dataset in our experiment. In contrast, MISAGA, which is known as an efficient approach to network clustering, has to spend 6692 seconds to complete the model optimization. Based on the results of model efficiency, we may conclude that MSPANC is able to uncover clusters in large scale networks, which demonstrates the applicability of MSPANC in real-world analytical tasks.

Table 5
Cluster structure and top-five source-wise cluster features learned by MSPANC

Cluster Structure	Cluster Features	
	Source 1	Source 2
 2249 2254 2246 2243 2251 2247 2248 2252 2253 2255 2244 2245 2250	#Eng	@WorldSportCNN
	(0.19683)	(0.63601)
	#MCFC	@alexthomascnn
	(0.15163)	(0.61757)
	#mufc	@AmandaJDavies
	(0.14937)	(0.38807)
	#Olympic	@EldergillCNN
	(0.13929)	(0.35443)
	#CL	@HarryReekieCNN
	(0.12414)	(0.33517)

4.6. Detailed analysis on the discovered cluster

To investigate whether MSPANC may learn the vertex-cluster preference simultaneously concerning network topology, multi-source vertex features, and their corresponding propagation between pairwise vertices, we perform a detailed analysis on the uncovered clusters and provide an example here.

In Twitter dataset, a 13-member social group is detected by MSPANC. This uncovered cluster perfectly matches a social circle in the ground-truth database. Its structure is shown in the first column of Table 5. As the figure depicts, all the vertices in this cluster are densely connected, indicating they have a locally cohesive network topology. Besides analyzing the structure of the discovered cluster, we also investigate whether the vertex features are frequently propagated and whether the features learned by the proposed model might well characterize the unfolded cluster. It is found that the average propagationality between vertices in this cluster is 0.0937, which is about 70 times higher than that in the whole network. This indicates more similar features are possible to be spread frequently among vertices in this cluster. Such a high intra-cluster propagationality also ensures the features learned by the model may well characterize the unfolded cluster. The top-five cluster features learned by MSPANC in each source are also listed in Table 5. As the table shows, the themes of the two sources are soccer, and CNN editors of soccer news, respectively. It makes sense that social network users who are soccer fans spread more soccer-related contents and consequently tend to form an online social community. Considering the optimization of propagation w.r.t. multi-source vertex features within each cluster, the proposed model is able to uncover meaningful clusters as well as their interpretable features like the example shown in this section. Such findings may indicate what multi-source vertex features potentially influence the cluster formation, and these learned features might impact a number of real applications, such as online personalized recommendation, and marketing.

5. Conclusion

In this paper, we propose a novel model dubbed Multi-source propagation aware network clustering (MSPANC). Different from previous approaches to attributed network clustering, MSPANC is considering to infer the vertex-cluster preference from multi-source vertex features and vertex structural proximity. As the latent space representing the cluster preference also involves into the optimization of intra-cluster propagation regarding multi-source vertex features, the latent information on cluster-propagation preference also influences the learning of cluster membership. The proposed model has been tested on five real-world datasets and has been compared with eight competitive baselines. The experimental results show that MSPANC outperforms all the baselines on all the datasets. In future, we will attempt to further improve the effectiveness of MSPANC via allowing the model to adaptively learn the relative significance of multi-source vertex features.

References

[1] Akata, Z., Thurau, C., Bauckhage, C., 2011. Non-negative matrix factorization in multimodality data for segmentation and label prediction, in: 16th Computer vision winter workshop.

[2] Chang, J., Blei, D., 2009. Relational topic models for document networks, in: Artificial Intelligence and Statistics, pp. 81–88.

[3] Clauset, A., Newman, M.E., Moore, C., 2004. Finding community structure in very large networks. Physical review E 70, 066111.

[4] Coussi-Korbel, S., Fragaszy, D.M., 1995. On the relation between social dynamics and social learning. Animal behaviour 50, 1441–1453.

[5] Dempster, A.P., Laird, N.M., Rubin, D.B., 1977. Maximum likelihood from incomplete data via the em algorithm. Journal of the royal statistical society. Series B (methodological) , 1–38.

[6] He, T., Bai, L., Ong, Y.S., 2019. Manifold regularized stochastic block model, in: 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI), IEEE. pp. 800–807.

[7] He, T., Chan, K.C., 2018. Misaga: An algorithm for mining interesting subgraphs in attributed graphs. IEEE transactions on cybernetics 48, 1369–1382.

[8] Huang, J., Sun, H., Han, J., Deng, H., Sun, Y., Liu, Y., 2010. Shrink: a structural clustering algorithm for detecting hierarchical communities in networks, in: Proceedings of the 19th ACM international conference on Information and knowledge management, ACM. pp. 219–228.

[9] Kim, J., Lee, J.G., 2015. Community detection in multi-layer graphs: A survey. ACM SIGMOD Record 44, 37–48.

[10] Leskovec, J., Lang, K.J., Mahoney, M., 2010. Empirical comparison of algorithms for network community detection, in: Proceedings of the 19th international conference on World wide web, ACM. pp. 631–640.

[11] Liu, L., Xu, L., Wangy, Z., Chen, E., 2015. Community detection based on structure and content: a content propagation perspective, in: Data Mining (ICDM), 2015 IEEE International Conference on, IEEE. pp. 271–280.

[12] Liu, Z., Luo, X., Wang, Z., 2020. Convergence analysis of single latent factor-dependent, nonnegative, and multiplicative update-based nonnegative latent factor models. IEEE Transactions on Neural Networks and Learning Systems .

[13] Lu, Q., Getoor, L., 2003. Link-based classification, in: Proceedings of the 20th International Conference on Machine Learning (ICML-03), pp. 496–503.

[14] Luo, X., Liu, Z., Li, S., Shang, M., Wang, Z., 2018. A fast non-negative latent factor model based on generalized momentum method. IEEE Transactions on Systems, Man, and Cybernetics: Systems .

[15] MacKay, D.J., Mac Kay, D.J., 2003. Information theory, inference and learning algorithms. Cambridge university press.

[16] Mcauley, J., Leskovec, J., 2014. Discovering social circles in ego networks. ACM Transactions on Knowledge Discovery from Data (TKDD) 8, 4.

[17] Miyauchi, A., Kakimura, N., 2018. Finding a dense subgraph with sparse cut, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, ACM. pp. 547–556.

[18] Qiao, M., Yu, J., Bian, W., Li, Q., Tao, D., 2018. Adapting stochastic block models to power-law degree distributions. IEEE transactions on cybernetics , 1–12.

[19] Shi, J., Malik, J., 2000. Normalized cuts and image segmentation. IEEE Transactions on pattern analysis and machine intelligence 22, 888–905.

[20] Sun, Y., Han, J., Gao, J., Yu, Y., 2009. itopicmodel: Information network-integrated topic modeling, in: Data Mining, 2009. ICDM'09. Ninth IEEE International Conference on, IEEE. pp. 493–502.

[21] Sun, Z., Wang, B., Sheng, J., Yu, Z., Zhou, R., Shao, J., 2019. Community detection based on information dynamics. Neurocomputing 359, 341–352.

[22] Veldt, N., Gleich, D.F., Wirth, A., 2018. A correlation clustering framework for community detection, in: Proceedings of the 2018 World Wide Web Conference on World Wide Web, International World Wide Web Conferences Steering Committee. pp. 439–448.

[23] Wang, F., Li, T., Wang, X., Zhu, S., Ding, C., 2011. Community discovery using nonnegative matrix factorization. Data Mining and Knowledge Discovery 22, 493–521.

[24] Xenarios, I., Salwinski, L., Duan, X.J., Higney, P., Kim, S.M., Eisenberg, D., 2002. Dip, the database of interacting proteins: a research tool for studying cellular networks of protein interactions. Nucleic acids research 30, 303–305.

[25] Xiang, B., Liu, Q., Chen, E., Xiong, H., Zheng, Y., Yang, Y., 2013. Pagerank with priors: An influence propagation perspective, in: Twenty-Third International Joint Conference on Artificial Intelligence.

[26] Xu, M., Li, Y., Li, R., Zou, F., Gu, X., 2019. Eadp: An extended adaptive density peaks clustering for overlapping community detection in social networks. Neurocomputing 337, 287–302.

[27] Xu, Z., Ke, Y., 2016. Effective and efficient spectral clustering on text and link data, in: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, ACM. pp. 357–366.

[28] Yang, J., McAuley, J., Leskovec, J., 2013. Community detection in networks with node attributes, in: Data Mining (ICDM), 2013 IEEE 13th international conference on, IEEE. pp. 1151–1156.

[29] Yang, J., McAuley, J., Leskovec, J., 2014. Detecting cohesive and 2-mode communities indirected and undirected networks, in: Proceedings of the 7th ACM international conference on Web search and data mining, ACM. pp. 323–332.

[30] Ye, F., Chen, C., Zheng, Z., 2018. Deep autoencoder-like nonnegative matrix factorization for community detection, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, ACM. pp. 1393–1402.

[31] Zhang, Y., Liu, Y., Li, Q., Jin, R., Wen, C., 2020. Lilpa: A label importance based label propagation algorithm for community detection with application to core drug discovery. Neurocomputing .

[32] Zhou, Y., Cheng, H., Yu, J.X., 2010. Clustering large attributed graphs: An efficient incremental approach, in: Data Mining (ICDM), 2010 IEEE 10th International Conference on, IEEE. pp. 689–698.