

## Research Article

# Semisupervised Community Preserving Network Embedding with Pairwise Constraints

Dong Liu <sup>1,2</sup>, Yan Ru <sup>1</sup>, Qinpeng Li,<sup>1</sup> Shibin Wang,<sup>1,2</sup> and Jianwei Niu<sup>3</sup>

<sup>1</sup>School of Computer and Information Engineering, Henan Normal University, Xinxiang, Henan, China

<sup>2</sup>Big Data Engineering Laboratory for Teaching Resources and Assessment of Education Quality, Xinxiang, Henan, China

<sup>3</sup>School of Computer Science and Engineering, Beihang University, Beijing, China

Correspondence should be addressed to Dong Liu; [liudong@htu.edu.cn](mailto:liudong@htu.edu.cn)

Received 30 May 2020; Revised 15 October 2020; Accepted 26 October 2020; Published 10 November 2020

Academic Editor: Xianggui Guo

Copyright © 2020 Dong Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Network embedding aims to learn the low-dimensional representations of nodes in networks. It preserves the structure and internal attributes of the networks while representing nodes as low-dimensional dense real-valued vectors. These vectors are used as inputs of machine learning algorithms for network analysis tasks such as node clustering, classification, link prediction, and network visualization. The network embedding algorithms, which considered the community structure, impose a higher level of constraint on the similarity of nodes, and they make the learned node embedding results more discriminative. However, the existing network representation learning algorithms are mostly unsupervised models; the pairwise constraint information, which represents community membership, is not effectively utilized to obtain node embedding results that are more consistent with prior knowledge. This paper proposes a semisupervised modularized nonnegative matrix factorization model, SMNMF, while preserving the community structure for network embedding; the pairwise constraints (must-link and cannot-link) information are effectively fused with the adjacency matrix and node similarity matrix of the network so that the node representations learned by the model are more interpretable. Experimental results on eight real network datasets show that, comparing with the representative network embedding methods, the node representations learned after incorporating the pairwise constraints can obtain higher accuracy in node clustering task and the results of link prediction, and network visualization tasks indicate that the semisupervised model SMNMF is more discriminative than unsupervised ones.

## 1. Introduction

Most systems in the real world exist in the form of networks, such as protein networks in biological systems, logistics networks in transportation systems, and the most common social networks such as Facebook and WeChat, and the research and analysis of these complex networks' information have high application value [1–3]. Network analysis is highly dependent on the representation of network data. Most traditional representation methods are based on adjacency matrix, but the adjacency matrix is high dimensional and has the problem of sparsity. This representation has limitations in statistical learning tasks, and when processing large-scale data, it will result in high-computational load and operation time. With the development and application of representation learning technology in the field of natural

language processing, more and more scholars began to explore how to represent network nodes with low-dimensional dense vectors [4].

The network embedding methods learn effective low-dimensional node representation vectors while preserving the network structure and inherent attributes. After that, through vector-based machine learning algorithms, the node representation vectors can be used as features of nodes to perform network analysis tasks, such as community detection, node classification, link prediction, and network visualization. DeepWalk [5] is the most representative network embedding algorithm. It uses the random walk strategy to obtain node sequence of the network and then uses the famous Skip-gram model to train the node representation vectors. Line [6] describes the first-order and second-order similarity of nodes with two different

functions from the perspective of network topology and then combines them to get the final node representation vectors. Furthermore, GraRep [7], NEU [8], and AROPE [9] all capture the higher-order approximation except the first-order and the second-order similarity.

However, most network embedding methods only preserve the local microstructure of the network. Community structure is one of the most common and important topological properties of the network. Revealing community structure of the network is very important for analyzing network function and discovering hidden pattern of the network [10, 11]. In response to this research direction, MNMF [12] imposes a higher level of constraint on the node representation when considering network proximity, and the representation of nodes in the same community should be more similar than nodes in different communities. Considering the community structure for network embedding can provide more effective and rich information, which not only solves the problem of data sparsity in the microstructure but also makes the learned node embedding vectors more discriminative. Another network representation method which preserving community structure is CARE [13], this model first discovers the community structure of a given network and then uses the community-aware random walk when learning the node representation vectors. Therefore, the walk path contains not only the nodes with the same neighbor structures but also the nodes in the same community so that nodes in the same community can have similar representations. Later, NECS [14] is proposed to learn the network embedding with community structural information, which preserves the high-order proximity and incorporates the community structure in vertex representation learning.

However, the above network embedding models whether preserving local microstructure or integrating community structure are all unsupervised methods. In recent years, some semisupervised network representation learning methods have also appeared, and they can integrate prior information such as node labels into the process of network embedding, thus can improve the experimental effect of representation vectors in subsequent tasks. Based on this view, MMDW [15] optimizes the maximum interval classifier and the objective matrix factorization model based on matrix decomposition. The learned node representation vectors can reflect not only the topological structure characteristics of the network but also the node label information. Other semisupervised network embedding algorithms include GCN [16] based on label propagation rules and Planetoid [17] for joint forecasting neighbor nodes and class labels.

Nevertheless, the existing semisupervised network representation learning algorithms failed to impose the constraints of community structure on the node representation vectors, and almost all scholars chose individual labels to assist the learning process of node representation vectors. In complex network field, besides individual labels, there are also widely used priori information-pairwise constraints, which are used to specify the community relationship of two nodes. If two nodes must be in the same community, their

relationship is must-link. If two nodes must be in different communities, their relationship is cannot-link [11].

In recent years, there are also some algorithms which utilize “pairwise constraints” to improve the performance of network representation learning algorithms. For example, SDNE [18], DNE-SBP [19], and DNE-APP [20] were all proposed to incorporate “pairwise constraints” into unsupervised stacked autoencoder to make the directly connected nodes have more similar representations. However, the pairwise constraints defined in these models represent the pairwise similarity between nodes. If two nodes are similar, their relationship is must-link; if two nodes are dissimilar, their relationship is cannot-link. However, the pairwise constraints in this paper are the prior information which represents the community affiliation about two nodes. According to whether two nodes belong to the same community, researchers define their relationship as must-link and cannot-link.

Of course, a deep nonlinear reconstruction model DNR [21] uses pairwise constraints which represent community membership to improve the performance of network representation learning algorithm, but this paper only uses the must-link constraint and ignores the cannot-link constraint. Moreover, this paper is driven by the community detection task and does not use other network analysis tasks to prove the effectiveness of incorporating pairwise constraints. Therefore, this paper attempts to use the pairwise constraints which represent the community membership to assist the traditional unsupervised network representation learning algorithm and enhance the discrimination of node representation vectors on different network analysis tasks while preserving the community structure.

Specifically, this paper proposes a novel semisupervised modularity nonnegative matrix factorization model, which not only preserves the network microstructure (pairwise similarity of nodes) and mesoscopic structure (community) for network embedding but also effectively integrates the pairwise constraints with the network adjacency matrix and node similarity matrix so that the final learned node representation vectors are more distinguishable.

The reason why this paper uses pairwise constraints to conduct the research on network representation learning of preserving community structure is as follows. As we know, the use of individual labels must know the specific community number of a given network in advance, that is, when dividing nodes, researchers must know how many kinds of nodes can be divided ahead of time; therefore, individual labels are more suitable for node classification tasks, and different types of nodes can be divided more clearly. However, the use of pairwise constraints is relatively simple. Scholars do not need to know how many communities this network contain, but only need to know the community relationship of two nodes. Therefore, pairwise constraints are more suitable for network clustering mode. Because this paper is a study of network representation learning to preserve the community structure, it is very necessary to make the nodes in the same community closer together.

The main contributions of this paper are as follows:

- (i) This paper fuses the pairwise constraints must-link and cannot-link with the adjacency matrix and node similarity matrix of the network, therefore makes the final generated node representation vectors more discriminative
- (ii) This paper proposes an iterative optimization algorithm for SMNMF model
- (iii) This paper uses node clustering, link prediction, and visualization experiments on eight real datasets to prove that the semisupervised model SMNMF is more effective than several existing network embedding algorithms

The structure of this paper is as follows. Section 2 briefly reviews the previous works related to unsupervised network representation learning methods, semisupervised network representation learning methods, and the use of pairwise constraints. Section 3 details the semisupervised modularity nonnegative matrix factorization model SMNMF and the complexity analysis of this algorithm. Section 4 compares the experimental results of SMNMF with some representative network representation learning algorithms on eight real networks. Section 5 summarizes this article.

## 2. Related Work

*2.1. Unsupervised Representation Learning Method Based on Micro-/Macrotopological Structure.* DeepWalk [5] is the most representative network representation learning algorithm. This model proves that the frequency of node distribution in the truncated random walk sequence is similar to the frequency of words distribution appearing in the text. Therefore, the Skip-gram model in the field of word vector representation is used to learn representations for nodes in the network. Line [6] is first proposed to optimize the objective function that preserves the first-order and the second-order proximity of nodes during node representation learning. The direct connection of nodes characterizes the first-order proximity, and the co-neighbors of nodes without direct connection characterize the second-order proximity.

In addition to the first-order and second-order proximity, the higher-order node proximity can also be used to preserve the topology of networks. GraRep [7] captures different  $k$ -order proximities by defining different loss functions and then merges the node representations learned by each function. NEU [8] first proves that the existing network representation learning methods include two steps of proximity matrix construction and dimensionality reduction. Constructing a  $k$ -order proximity matrix preserves higher-order node similarity and improves network embedding performance. AROPE [9] preserves the proximity of any order based on the Singular Value Decomposition (SVD) framework to obtain the final node representation vectors. However, the above network embedding methods do not consider the constraints imposed by the community structure when learning the node representation vectors, and all of them are unsupervised models.

*2.2. Unsupervised Representation Learning Method Integrating Mesoscopic Community Structure.* Identifying communities in networks can help us better understand the network structure and perform network analysis tasks more efficiently. There are many types of community detection algorithms, such as clustering-based methods, modularity-based methods, and spectral methods. Combining network representation learning methods with community structure modeling imposes a higher level of constraint on the node representation vectors.

MNMF [12] uses matrix decomposition to learn the similarity of nodes and models the community structure by maximizing the modularity of the network, and then using an auxiliary community representation matrix, it jointly optimized these two parts. MNMF preserves not only the microscopic structure of the network through node similarity but also the mesoscopic structure of the network by using the community structure. CARE [13] uses the Louvain method to detect community structure of a given network and then generates random walks by considering the community structure when extracting the neighborhood structure of nodes; finally, it uses the Skip-gram model to get node representation vectors. NECS [14] utilizes the high-order proximity to learn the vertex representation vectors while incorporating the network community structure. It uses matrix factorization to approximate the high-order proximity and utilizes homophily principle to incorporate the community structure for guiding the vertex representation learning.

However, the above network embedding methods combined with community structure are also essentially unsupervised learning, and the prior information is not effectively utilized in these models.

*2.3. Semisupervised Network Representation Learning Method.* Unsupervised network representation learning lacks distinctiveness in machine learning tasks such as node clustering and classification. A representative semisupervised network embedding model node2vec [22] expands DeepWalk by changing the generation mode of random walk sequence. By introducing two parameters  $p$  and  $q$ , the breadth first search and depth first search are introduced into the generation process of random walk sequences, and the network representation learning of local and global information is carried out.

Other semisupervised network representation learning methods can effectively use some labeled nodes, which can be added to the process of network embedding. By using these semisupervised information, the final node representations can improve the accuracy of node classification and clustering tasks. MMDW [15] designs a unified network representation learning framework based on the matrix decomposition. This model is affected by the maximum interval classifier, and the learned node representation vectors can reflect not only the structural characteristics of the network but also the label information of nodes. GCN [16] designs a convolutional neural network which acts on the network structure and uses label propagation rules to

realize semisupervised network representation learning. Planetoid [17] jointly predicts neighbor nodes and class labels of nodes, eventually achieves semisupervised network representation.

In order to further improve the performance of network representation learning algorithms, some semisupervised network representation learning methods also utilize the “pairwise constraints” information which represents node similarity to design models. SDNE [18] is proposed to incorporate a pairwise constraint into an unsupervised stacked autoencoder (SAE) to map each two directly connected vertices near to each other in the embedding space and successfully captures highly nonlinear network structure. However, the SDNE model can only preserve the first-order and second-order node approximation. DNE-APP [20] is proposed to employ a semisupervised stacked autoencoder to make the node pairs possessing higher aggregated proximities have more similar hidden vector representations. The DNE-APP model designs different pairwise constraint matrices for different tasks and combines pairwise constraints with the stacked autoencoder.

In addition to the above semisupervised network representation learning algorithms in unsigned networks, the DNE-SBP model [19] employs a semisupervised stacked autoencoder to reconstruct the adjacency connection of a given signed network. To preserve the structural balance property of signed networks, they design the pairwise constraints to make the positively connected nodes much closer than the negatively connected nodes in the embedding space.

However, most of the existing semisupervised network representation learning methods focus on node labels; in addition, the pairwise constraint information they used represent the pairwise similarity between nodes rather than the community membership of networks. In the field of community detection, there is also a kind of prior information called pairwise constraints, which represents the community relationship of two nodes. At present, there are few network representation learning algorithms using such pairwise constraints.

*2.4. Pairwise Constraints.* In practical applications, researchers can obtain some prior information about the network in advance, and they are usually in the form of individual labels and pairwise constraints. Individual labels are used to specify the community number which one node belongs to. Pairwise constraints are used to indicate the relationship between the communities which the two nodes belong to. If the two nodes must belong to the same community, their relationship is defined as must-link; if they belong to different communities, their relationship is defined as cannot-link [11].

The information of pairwise constraints is used more frequently. In past studies, scholars have used the combination of pairwise constraints and network topology to guide the process of community detection. Zhang demonstrates that integrating pairwise constraint matrices  $C_{ML}$  and  $C_{CL}$  with the adjacency matrix and then using unsupervised

learning such as Nonnegative Matrix Factorization (NMF), spectral clustering, and InfoMap (Information Map Algorithm) for community structure detection can improve the performance of community detection to some extent [23]. SNMF-SS [24] combines sparse Nonnegative Matrix Factorization (SNMF) with semisupervised clustering for community detection and uses pairwise constraints to guide the clustering process. PSSNMF [25] combines the idea of graph regularization with pairwise constraints and proposes a semisupervised nonnegative matrix factorization model. In recent years, a deep nonlinear reconstruction model DNR [21] is presented for community detection. This model proves that the stochastic model and modularity maximization can be intuitively interpreted as to find the low-dimensional representation to best reconstruct the given network structure. Then, it stacks a series of autoencoders and incorporates pairwise constraints on vertices into the proposed DNR model. However, this semisupervised model only utilizes the must-link information and ignores the cannot-link information.

In view of the above research, this paper fuses the pairwise constraints information which represent the community membership with the network adjacency matrix and node similarity matrix and affects the results of node representation vectors in the process of network embedding. In this way, the generated node representation vectors are more consistent with prior knowledge, and the performance of node representations on node clustering, link prediction, and visualization tasks are improved to some extent.

### 3. The Proposed Model

The semisupervised model SMNMF mainly includes the following modules: firstly, the adjacency matrix of the network is modified by the pairwise constraints information obtained in advance; secondly, the community structure of network is modeled by the modified adjacency matrix; thirdly, the microstructure of network is modeled by preserving the first-order and second-order similarity of nodes; finally, the microstructure modeling and community structure modeling are integrated by an auxiliary community representation matrix to form the final objective function. Through the optimization and training of the objective function, the model SMNMF can get the most representative vectors of node features. Table 1 summarizes the frequently used notations in this paper.

For an undirected and unweighted network  $G = (V, E)$ , the number of nodes is  $n$ , the number of edges is  $e$ , the adjacency matrix is represented by  $A = [A_{ij}] \in E^{n \times n}$ , and the elements in the adjacency matrix are

$$A_{ij} = \begin{cases} 1, & \text{if } (i, j) \in E, \\ 0, & \text{otherwise,} \end{cases} \quad (1)$$

where  $(i, j) \in E$  indicates that there is a connection edge between node  $i$  and node  $j$ ; otherwise, there is no edge between them.  $A$  is a symmetric matrix. The goal of model SMNMF is to learn the node representation matrix  $U \in R^{n \times d}$ ,  $d$  ( $d \ll n$ ) is the dimension of node representation.

TABLE 1: The frequently used notations.

Notation	Meaning
$G$	An unweighted and undirected network
$V$	The nodes set of $G$
$E$	The edges set of $G$
$A$	The adjacency matrix of $G$
$D$	The degree matrix of $G$
$S$	The node similarity matrix
$U$	The node representation matrix
$C_{ml}$	The must-link constraint matrix
$C_{cl}$	The cannot-link constraint matrix
$n$	The number of nodes
$e$	The number of edges
$k$	The number of communities
$d$	The dimension of node representation vectors

### 3.1. Integrating Priori Information into the Adjacency Matrix.

The adjacency matrix of real network is usually very sparse, but in many practical applications, some background information can be obtained in advance, which can be used as priori knowledge to improve the node embedding results obtained by network representation learning. Specifically, this paper considers the following types of pairwise constraints [26]:

- (i) Must-link constraints  $C_{ml}$ :  $(i, j) \in C_{ml}$  means that node  $i$  and node  $j$  must belong to the same community
- (ii) Cannot-link constraints  $C_{cl}$ :  $(i, j) \in C_{cl}$  means that node  $i$  and node  $j$  must belong to different communities

Then, integrate  $C_{ml}$  and  $C_{cl}$  with the adjacency matrix to get a new matrix  $B$ :

$$B_{ij} = \begin{cases} 1, & \text{if } (i, j) \in C_{ml}, \\ 0, & \text{if } (i, j) \in C_{cl}, \\ A_{ij}, & \text{otherwise.} \end{cases} \quad (2)$$

According to the above integration, not only the basic topology structure of the network are obtained but also the network structure are denoised and modified according to the pairwise constraints matrices  $C_{ml}$  and  $C_{cl}$ . In other words, for two nodes  $i$  and  $j$ , if there is an edge connected them and they belong to the same community, the value of  $B_{ij}$  is 1; if there is an edge connected them but  $i$  and  $j$  belong to different communities, the value of  $B_{ij}$  is 0. If there is no edge connected them and they belong to different communities, the value of  $B_{ij}$  is 0; if there is no edge connected them but they belong to the same community, the value of  $B_{ij}$  is 1. Therefore, this paper uses the pairwise constraints as priori information to modify the adjacency matrix according to the community structure. For nodes that are not connected but belong to the same community, model SMNMF add connection edge for them; for nodes with connection edge but belong to different communities, model SMNMF disconnect their edge so that the edges in the network are more consistent with prior knowledge.

3.2. *Community Structure Modeling.* Newman [27] proposed community detection based on modularity maximization in 2006, which is the most widely used the community structure modeling method. According to the modified adjacency matrix  $B$ , if the network has two communities, this paper redefine the modularity as follows:

$$Q = \frac{1}{4e} \sum_{ij} \left( B_{ij} - \frac{k_i k_j}{2e} \right) h_i h_j, \quad (3)$$

where  $k_i$  is the degree of node  $i$ ,  $h_i = 1$  represents that node  $i$  belongs to the first community,  $h_i = -1$  represents that node  $i$  belongs to another community, and  $(k_i k_j / 2e)$  represents the case that edges of network are randomly allocated, so the modularity measures the difference between the network under certain community division and the random network. Because the random network does not have community structure, the greater the difference, the better the community division. This paper define the modularity matrix as  $Y \in R^{n \times n}$  and the elements of  $Y$  are  $Y_{ij} = B_{ij} - (k_i k_j / 2e)$ , so the modularity  $Q = (1/4e) h^T Y h$ ,  $h = [h_i] \in R^n$  is the community indicator.

When the number of communities is extended to  $k > 2$ , the community indicator factor  $h$  can be extended to the community indicator matrix  $H \in R^{n \times k}$ , each column represents one community. For each row of  $H$ , only one element is 1 representing the community which  $i$  belongs to, all other elements are 0. So, the model can get the constraint  $\text{tr}(H^T H) = n$ ; since the constant has no effect on the modularity maximization, the constant can be compressed to obtain

$$Q = \text{tr}(H^T Y H), \quad \text{s.t. } H \geq 0, H1 = 1, \quad (4)$$

where  $\text{tr}(X)$  is the trace of matrix  $X$ .

3.3. *Microstructure Modeling.* This paper uses the first-order and second-order proximity to model local microstructure of the network. The first-order proximity is defined as follows.

The first-order proximity  $S^{(1)} = [S_{ij}^{(1)}] \in R^{n \times n}$  represents the observable pairwise similarity between two nodes [12]. The modified adjacency matrix  $B$  is used to express the first-order proximity between nodes. If  $B_{ij} > 0$ , there is a positive first-order proximity between node  $i$  and node  $j$ , otherwise the first-order similarity is 0. The first-order proximity is the most direct representation of the network. If two nodes have a directly connected edge, the representations of these two nodes in low-dimensional vector space should be similar. By revising the adjacency matrix, although this paper improves the edge sparsity problem to a certain extent, for two nodes without connected edge, it does not mean that there is no similarity between them. Therefore, it is not enough to calculate the similarity between two nodes only by using the first-order proximity. The second-order proximity can be defined by the common neighbors of two nodes. If there are a large number of common neighbors for two nodes, even if they have no directly connected edge, the two nodes are still similar.

The second-order proximity is  $S^{(2)} = [S_{ij}^{(2)}] \in R^{n \times n}$ , since  $B_i = (B_{i,1}^{(1)}, \dots, B_{i,n}^{(1)})$  represents the first-order proximity between node  $i$  and other nodes, so the similarity between  $B_i$  and  $B_j$  can represent the consistency of co-neighbors between node  $i$  and node  $j$  [12], that is, the second-order proximity between  $i$  and  $j$ . Using cosine similarity as a measure,  $S_{ij}^{(2)} = (B_i B_j / \|B_i\| \|B_j\|)$ ,  $\|X\|$  represents the norm of vector  $X$ . Therefore, the second-order proximity ranges from 0 to 1.

In order to preserve the microstructure of the network, the first-order proximity and the second-order proximity are combined to obtain the final node similarity matrix  $S = S^{(1)} + \mu S^{(2)}$ ,  $\mu > 0$  is the weight of the second-order proximity; according to the MNMF model, the value of  $\mu$  is set to 5.

In order to combine node similarity with network embedding, the nonnegative matrix factorization method is widely used. The formula is as follows:

$$\begin{aligned} \min \|S - VU^T\|_F^2 \\ \text{s.t. } V \geq 0, U \geq 0, \end{aligned} \quad (5)$$

where the nonnegative matrix  $V, U \in R^{n \times d}$  represents the base matrix and the node representation matrix, respectively,  $d$  is the representation dimension of nodes, and the  $i$ th row of  $U$  represents the representation of node  $i$ . The objective function expects the node representation and the node similarity matrix  $S$  to be as close as possible.

**3.4. Semisupervised Community Preserving Network Embedding Model.** In order to combine the mesoscopic level of community structure with the microscopic level of node similarity to obtain the final representation matrix  $U$ , a nonnegative matrix  $W \in R^{k \times d}$  is defined for the community representation matrix and the  $r$ th row represents the representation of the community  $r$ . Thus, for the representation  $U_i$  of node  $i$ , the probability that node  $i$  belongs to the community  $r$  can be defined as  $U_i W_r^T$ . Since the community indicator matrix  $H$  provides the community affiliation of all nodes, model SMNMF needs to make  $UW^T H$  as consistent as possible, and the objective function is

$$\begin{aligned} \min \|H - UW^T\|_F^2 \\ \text{s.t. } W \geq 0. \end{aligned} \quad (6)$$

Through modification of the adjacency matrix, the pairwise constraints are integrated into the modularity-based community detection process, which makes the learned community members more consistent with the real community structure. At the same time, the pairwise constraints are used to adjust the connected edges between nodes. Therefore, the connected edges of network are more conducive to the judgment of node similarity. There should be more edges between nodes in the same community, so the representation of nodes in the same community should be similar in low-dimensional space. The final node representation is obtained by joint optimization of  $U$  and  $H$ .

Therefore, the total objective function of the semisupervised modularity nonnegative matrix model SMNMF is

$$\begin{aligned} \min_{V,U,W,H} \|S - VU^T\|_F^2 - \beta \text{tr}(H^T YH) + \alpha \|H - UW^T\|_F^2 \\ \text{s.t. } V \geq 0, U \geq 0, H \geq 0, W \geq 0, H1 = 1. \end{aligned} \quad (7)$$

The constraint on  $H$  makes the above formula difficult to solve, so this model relaxes the constraint on  $H$  to an orthogonal constraint, i.e.,  $H^T H = I$  [28]. The above objective function can be rewritten as follows:

$$\begin{aligned} \min_{V,U,W,H} \|S - VU^T\|_F^2 - \beta \text{tr}(H^T YH) + \alpha \|H - UW^T\|_F^2 \\ + \lambda \|H^T H - I\|_F^2 \\ \text{s.t. } V \geq 0, U \geq 0, H \geq 0, W \geq 0, \end{aligned} \quad (8)$$

where the positive parameters  $\alpha$  and  $\beta$  are used to adjust the contribution of the corresponding term to the objective function. The positive parameter  $\lambda$  needs to be large enough to ensure the satisfaction of the orthogonal condition. In the experiment, it is fixed to  $10^9$ .

**3.5. Iterative Optimization Algorithm.** Since the total objective function of SMNMF model is nonconvex, all variables  $U, V, W$ , and  $H$  cannot be optimized at the same time. In order to solve the optimization problem of the objective function, the optimization part is divided into four subproblems for iterative optimization. When one of the variables is optimized, other variables are fixed to ensure that each subproblem can converge to the local minimum.

**3.5.1. Update Rules for  $U$ .** For example,  $H, W$ , and  $V$  are fixed to update the node representation matrix  $U$ . When  $H, W$ , and  $V$  are fixed, the objective function is a convex function for the matrix  $U$ , and the problem of updating  $U$  is a joint NMF problem [29], and by removing the independent term for  $U$ , the objective function can be rewritten as follows:

$$\min_{U \geq 0} \|S - VU^T\|_F^2 + \alpha \|H - UW^T\|_F^2. \quad (9)$$

The Lagrangian function of formula (9) is

$$L(U) = \|S - VU^T\|_F^2 + \alpha \|H - UW^T\|_F^2 - \text{tr}(\Lambda_U U^T), \quad (10)$$

where  $\Lambda_U$  is the Lagrangian multiplier for the constraint  $U \geq 0$ . By calculating the derivative of  $L(U)$  versus  $U$  and setting it to 0, model can obtain

$$\Lambda_U = -S^T V + UV^T V - \alpha HW + \alpha UW^T W. \quad (11)$$

According to the Karush–Kuhn–Tucker (KKT) condition for nonnegative nature of  $U$ , i.e.,  $\Lambda_U U = 0$ , the following conclusion can be obtained:

$$(-S^T V + UV^T V - \alpha HW + \alpha UW^T W)U = 0. \quad (12)$$

Therefore, the update rule for  $U$  is

$$U \leftarrow U \sqrt{\frac{S^T V + \alpha H W}{U(V^T V + \alpha W^T W)}}. \quad (13)$$

3.5.2. *Update Rules for  $W$ ,  $V$ , and  $H$ .* Similar to the update process of matrix  $U$ ,  $W$ ,  $H$ , and  $V$  can be updated by the following rules:

$$\begin{aligned} W &\leftarrow W \sqrt{\frac{H^T U}{W U^T U}}, \\ V &\leftarrow V \sqrt{\frac{S U}{V U^T U}}. \end{aligned} \quad (14)$$

When fixing other parameters to update  $H$ , the objective function can be rewritten as follows:

$$\min_{H \geq 0} L(H) = \alpha \|H - U W^T\|_F^2 - \beta \operatorname{tr}(H^T (B - C)H) + \lambda \|H^T H - I\|_F^2. \quad (15)$$

The above formula can be further written as follows:

$$\begin{aligned} \min_{H \geq 0} L(H) &= \alpha \|H - U W^T\|_F^2 - \beta \operatorname{tr}(H^T B H) + \beta \operatorname{tr}(H^T C H) \\ &+ \lambda \|H^T H - I\|_F^2. \end{aligned} \quad (16)$$

According to the Lagrange multiplier method and the KKT condition, the update rules for  $H$  are as follows:

$$H \leftarrow H \sqrt{\frac{-2\beta C H + \sqrt{\Phi}}{8\lambda H H^T H}}, \quad (17)$$

where  $\Phi = 2\beta(CH) \odot 2\beta(CH) + 16\lambda(HH^T H) \odot (2\beta B H + 2\alpha U W^T + (4\lambda - 2\alpha)H)$ .

The above updating procedure will continue until convergence.

3.6. *Complexity Analysis.* The computational scale of SMNMF mainly depends on the update of  $U$ ,  $V$ ,  $H$ , and  $W$  in each iteration. The modification of the adjacency matrix and the node similarity matrix by using prior information do not increase the complexity of model optimization. The scales of iterative update for matrices  $U$ ,  $V$ ,  $H$ , and  $W$  are  $O(nd^2 + n^2d + d^2k)$ ,  $O(nd^2 + n^2d)$ ,  $O(nk^2 + n^2k + dnk)$ , and  $O(n dk)$ , since  $d \leq n$ , the total computational complexity of the SMNMF model is  $O(n^2d + n^2k)$ , which is consistent with the order of magnitude of the standard nonnegative matrix.

## 4. Experiments and Results

4.1. *Datasets and Baseline Methods.* To verify the effectiveness of the proposed method, this paper conducts experiments on the following eight real network datasets, including Polbooks, WebKB (Cornell, Texas, Washington,

and Wisconsin) [12], email, Citeseer [15], and Cora [30], the details of these datasets are shown in Table 2.

This paper compares the semisupervised method SMNMF with the following representative network representation learning algorithms and unsupervised MNMF without prior information:

DeepWalk [5]: learning node representations using random walks and Skip-gram model

Line [6]: learning node representation by optimizing the objective function of the first-order and second-order proximity of nodes

node2vec [22]: learning the node representation by maximizing the possibility of saving the neighborhoods of nodes

GraRep [7]: learning node representations in conjunction with global structure information

MNMF [12]: integrating node similarity and modularity-based community detection to learn node representation

NECS [14]: preserving the high-order proximity and incorporating the community structure in vertex representation learning

DNR [21]: incorporating must-link constraints into a series of stacked autoencoders to find nonlinear embeddings which can best reconstruct modularity matrix  
SMNMF: integrating pairwise constraints information based on the unsupervised model MNMF to guide the preservation of community structure and node similarity and to affect the learned node representation results

In order to make a fair comparison, the representation dimension  $d$  is set to 100 uniformly; the values of  $\alpha$  and  $\beta$  are set to 0.1 and 1, respectively. In order to verify the effect of different amounts of prior information on the representation of nodes, the ratio of prior information is set to 1%, 2%, 5%, and 10%, named as SMNMF (1), SMNMF (2), SMNMF (5), and SMNMF (10), respectively. When the ratio of prior information is set to 0, the method SMNMF is equivalent to the unsupervised model MNMF. Similarly, the ratio of prior information in the DNR model is also set to 1%, 2%, 5%, and 10%, which is convenient for comparison under the same parameters. The reason why this paper uses different proportion of pairwise constraints is to prove that the semisupervised network representation learning model with pairwise constraints can learn more discriminative node representation vectors than the unsupervised network representation learning model, and the more pairwise constraints are used, the learned node representation vectors perform better in subsequent network analysis tasks.

4.2. *The Usage of Prior Information.* Given an undirected graph  $G$  that has  $n$  nodes and  $k$  communities, there are  $N = (n * (n - 1) / 2)$  pairs of pairwise constraints, and the constraints are divided into two types: must-link and cannot link.

TABLE 2: Statistics of real-world datasets used.

Datasets	Polbooks	Cornell	Texas	Washington	Wisconsin	E-mail	Citeseer	Cora
$ V $	105	183	183	215	251	986	2128	2485
$ e $	441	277	279	365	450	16064	3699	5069
Avg. degree	4.20	1.51	1.52	1.70	1.79	16.29	1.74	2.04
$k$	3	5	5	5	5	42	6	7

The total number of must-link is  $N_{ml} = \sum_{c=1}^k (n_c(n_c - 1)/2)$ , where  $n_c$  represents the number of nodes included in the  $c$ th community. The total number of cannot-link is  $N_{cl} = N - N_{ml}$ . When choosing pairwise constraints, this paper randomly selects two nodes from set  $V$ . Different percentages 1%, 2%, 5%, and 10% represent the random selection of a certain proportion of edge relationships from  $N = (n * (n - 1)/2)$  pairwise constraints, and the  $C_{ml}$  and  $C_{cl}$  matrices of nodes are constructed according to the selected pairwise constraints.

It is worth mentioning that the prior information selected in this paper is all valid prior information. For node  $v_i$  and node  $v_j$  in the same community, if they already have an edge, the selected must-link information is invalid; for node  $v_i$  and node  $v_j$  in different communities, if they do not have an edge, the selected cannot-link information is invalid, too. In order to avoid this situation and select as many effective prior information as possible, when this paper selected the prior information, two nodes that belong to the same community but do not have an edge is given a must-link; the nodes that belong to different communities but have an edge is given a cannot-link. In this way, this paper can select as many valid prior information as possible to guide the learning process of node representation vectors.

### 4.3. Experiment Setup

**4.3.1. Node Clustering Experiment.** In order to evaluate the effect of node representation, this paper first uses node clustering experiments to verify the superiority of our model. First, the learned node representation vectors are clustered by the K-means method, and then accuracy [31] is used to compare the node clustering results.

Accuracy is used to measure the clustering performance. Given a node  $v_i$ , let  $r_i$  be the obtained cluster label and  $s_i$  be the label provided by the datasets that express community number which the node belongs, and the accuracy is defined as follows:

$$\text{accuracy} = \frac{\sum_{i=1}^n \delta(s_i, \text{map}((r_i)))}{n}, \quad (18)$$

where  $n$  is the total number of nodes in the network,  $\delta(x, y)$  is the delta function that equals one if  $x = y$  and equals zero otherwise, and  $\text{map}(r_i)$  is the permutation mapping function that maps each cluster label  $r_i$  to the equivalent label from the community datasets. The best mapping can be found by using the Kuhn–Munkres algorithm [32].

In view of the sensitivity of K-means method to the initial values, this paper repeats the clustering experiments 20 times, each time with a new initial centroid, taking the average of 20 experimental results, as shown in Table 3.

It can be seen from Table 3 that on most datasets such as Polbooks, Cornell, Texas, Washington, Wisconsin, and email, the clustering results of MNMF and NECS, preserving community structure, are better than those of the representative algorithms which only preserve microstructure of networks such as DeepWalk, Line, node2vec, and GraRep. It shows that the integration of community structure can effectively improve the effect of network representation learning algorithms on node clustering task. Only on two larger datasets Citeseer and Cora, the clustering accuracy of unsupervised model MNMF and NECS are inferior to GraRep, but after SMNMF added the prior information, the clustering accuracy can be improved immediately. And in general, the higher proportion of prior information is, the higher node clustering accuracy is, especially in Polbooks dataset, when the ratio of prior information is set to 10% and the clustering accuracy can reach to 100%.

In addition, for the baseline DNR, when the proportion of prior information is 1%, the semisupervised algorithm SMNMF are slightly inferior to DNR model on two datasets Texas and Washington; however, when the proportion of prior information is increased to 2%, 5%, and 10%, the clustering accuracy of SMNMF on eight datasets are all higher than those of DNR. Especially, on two large datasets Citeseer and Cora, the clustering accuracy of SMNMF is much higher than those of other baseline algorithms. It is worth noting that the clustering accuracy of semisupervised algorithm DNR decreases slightly with the increase of prior information. This maybe because DNR cannot effectively improve the clustering accuracy when adding a small amount of prior information. However, for our SMNMF algorithm, the clustering accuracy of nodes is getting higher and higher with the increase of the prior information ratio. In a word, the semisupervised SMNMF which uses pairwise constraints can successfully improve the accuracy of node clustering. It is proved that the node representation vectors learned by the SMNMF model can better represent the characteristics of nodes and more conform to the real attributes of nodes, in other words, they are more interpretable.

Figure 1 more intuitively shows the effect that clustering accuracy increases with the proportion of prior information in the SMNMF model on different datasets.

Figure 1 indicates that, with the increase of prior information ratio, the clustering accuracy is significantly improved, which shows that the more pairwise constraints are used, the node representation vectors are more discriminative in node clustering task, so it is more helpful to gather the nodes in the same community together.

**4.3.2. Link Prediction Experiment.** For the link prediction task, this paper randomly samples a node pair which is not connected as a negative instance for each edge, while the

TABLE 3: Accuracy (%) of node clustering.

Methods	Polbooks	Cornell	Texas	Washington	Wisconsin	Email	Citeseer	Cora
DeepWalk	52.29	33.29	40.44	39.12	35.26	30.18	48.01	20.64
Line	41.33	32.24	33.83	32.61	33.55	24.14	24.30	23.44
node2vec	67.77	32.62	33.81	34.37	31.79	21.97	23.99	26.95
GraRep	42.86	35.15	38.98	37.67	38.22	49.81	43.16	50.94
MNMF	71.91	39.73	45.96	52.09	43.49	50.41	27.68	35.60
NECS	70.38	45.41	55.03	50.37	47.21	14.08	25.75	29.62
DNR (1)	72.65	38.40	55.04	59.32	41.80	20.15	31.87	38.28
DNR (2)	73.08	36.97	48.38	50.39	43.73	17.49	40.17	31.68
DNR (5)	75.03	35.60	48.05	58.64	40.36	15.33	47.46	41.05
DNR (10)	72.63	33.63	39.75	43.06	47.42	16.34	34.77	29.53
SMNMF (1)	75.05	41.58	47.10	55.26	45.98	58.95	70.82	63.12
SMNMF (2)	78.38	43.82	51.04	57.21	48.45	65.85	75.85	65.16
SMNMF (5)	91.43	68.52	55.74	65.16	69.08	67.86	79.24	69.95
SMNMF (10)	100.00	76.61	72.19	71.35	74.78	70.93	82.12	74.12

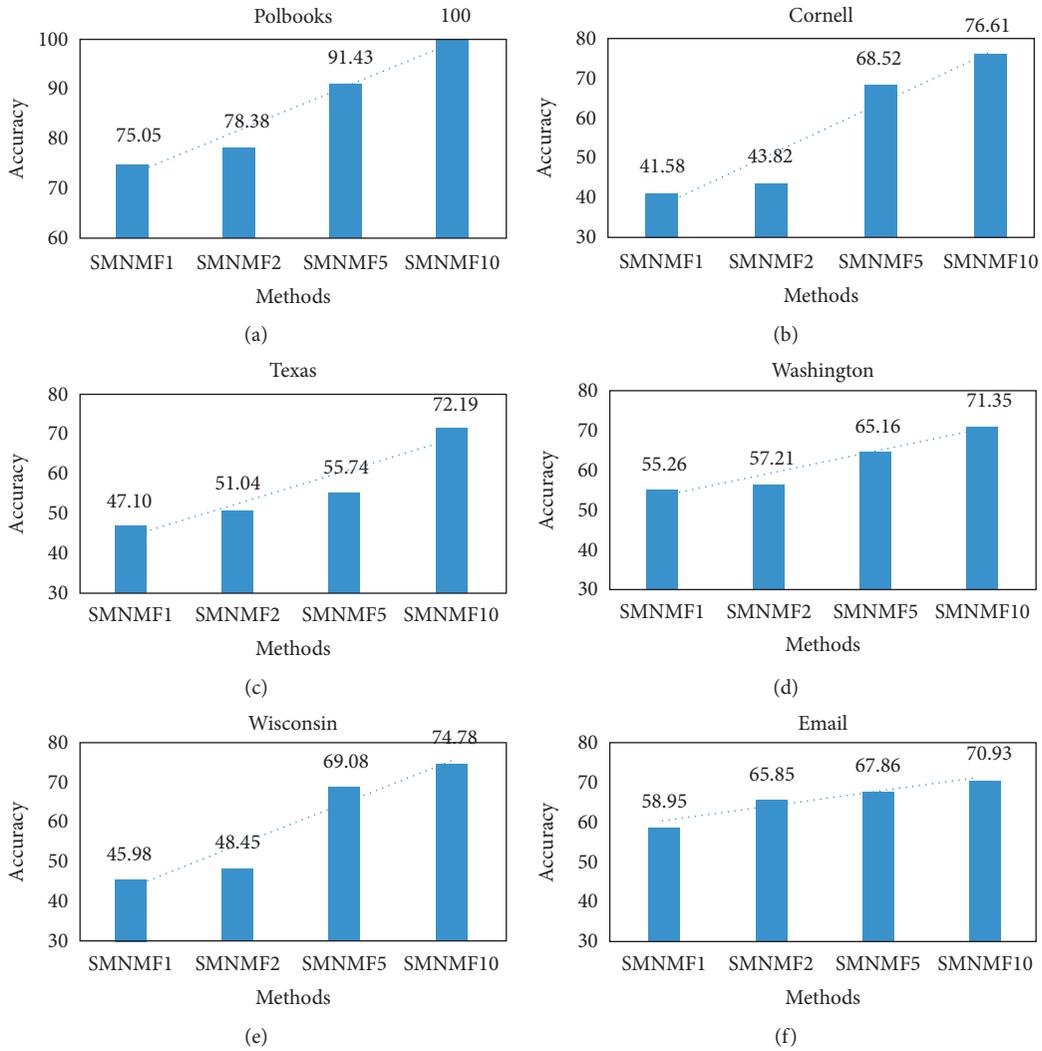


FIGURE 1: Continued.

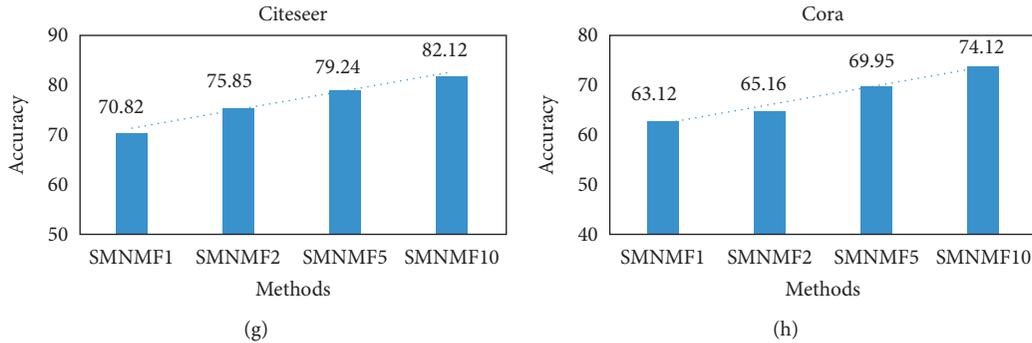


FIGURE 1: The accuracy of SMNMF with different proportion prior information on eight datasets.

TABLE 4: AUC-ROC of link prediction (90% training set and 10% test set).

Methods	Polbooks	Cornell	Texas	Washington	Wisconsin	Email	Citeseer	Cora
DeepWalk	0.6266	0.8917	0.8757	0.6776	0.5381	0.6363	0.6611	0.5849
Line	0.8769	0.9203	0.8979	0.8621	0.8649	0.6664	0.6535	0.6282
node2vec	0.5745	0.9046	0.9481	0.8565	0.8112	0.6319	0.5514	0.5892
GraRep	0.9615	0.8481	0.8979	0.8351	0.8638	0.6613	0.6487	0.6754
MNMF	0.9941	0.9967	0.9978	0.9983	0.9907	0.9822	0.9916	0.9806
NECS	0.9998	0.8988	0.8942	0.8710	0.8445	0.8132	0.5972	0.6347
DNR (1)	0.9016	0.9070	0.9224	0.8566	0.8724	0.9176	0.7412	0.8367
DNR (2)	0.8946	0.9034	0.9133	0.8410	0.8730	0.8905	0.6191	0.7520
DNR (5)	0.8944	0.8979	0.9142	0.8171	0.8662	0.9053	0.5192	0.6881
DNR (10)	0.8907	0.9160	0.8466	0.7797	0.8314	0.9244	0.5510	0.6362
SMNMF (1)	0.9884	0.9956	0.9999	0.9882	0.9851	0.9745	0.9630	0.9586
SMNMF (2)	0.9963	0.9890	0.9972	0.9940	0.9877	0.9616	0.9416	0.8889
SMNMF (5)	0.9912	0.9787	0.9546	0.9660	0.9787	0.9293	0.9187	0.9323
SMNMF (10)	0.9914	0.9556	0.9653	0.9299	0.9525	0.9009	0.8972	0.9177

links are considered positive instances. Then, this paper randomly splits different proportions of instances as training set and the remaining instances as the test set, in order to get more sufficient link prediction experimental setup, and the training set proportions are set to 90%, 80%, and 70%; correspondingly, the proportions of test sets are 10%, 20%, and 30%, respectively. Then, the node embedding is learned on the training set and edge embedding are generated by concatenating the two node embedding of links. Finally, the embedding of edges is treated as features and whether or not a node pair has edges as the ground truth. This paper trains a simple logistic regression classifier on the training set and adopts AUC-ROC (Area under the ROC (Receiver Operating Characteristic) Curve), which have been used in the previous work [31] to evaluate the performance on the test set. The results of the experiment are shown in Tables 4–6 compared with the baseline methods in link prediction task.

From the results of link prediction tasks (under different proportion of training set and test set) on eight datasets, our semisupervised model SMNMF is generally superior to DeepWalk, Line, node2vec, GraRep, NECS, and DNR algorithms, which proves that the node representation vectors generated by SMNMF are effective on link prediction task. However, the AUC-ROC values of the semisupervised model SMNMF are slightly lower than those of MNMF in

general. This is probably because the SMNMF model in this paper modifies the topology of original network by combining pairwise constraints information when generating node representation vectors, such as adding and subtracting edges according to whether two nodes belong to the same community; therefore, in link prediction task, it may cause the prediction of two nodes which belong to the same community but do not have connection edge to have an edge, resulting in the decrease of the AUC-ROC value. However, on the whole, the link prediction results of our model SMNMF are also good.

**4.3.3. Visualization Experiment.** In order to prove that the node representation vectors learned by semisupervised SMNMF model in this paper are more interpretable, this paper uses *t*-SNE (*t*-distributed stochastic neighbor embedding) visualization tool to show the effect of node embedding vectors learned by different network representation learning models on Polbooks dataset. In all graphs, each point represents a node in the network, and each color represents a category. The results shown in Figure 2 indicate that the integration of pairwise constraints in model SMNMF can distinguish the nodes of different communities more clearly. On the contrary, the representation learned by other baseline methods tends to mix together.

TABLE 5: AUC-ROC of link prediction (80% training set and 20% test set).

Methods	Polbooks	Cornell	Texas	Washington	Wisconsin	Email	Citeseer	Cora
DeepWalk	0.6257	0.8034	0.8539	0.6571	0.5340	0.6314	0.6436	0.5648
Line	0.7749	0.8356	0.8494	0.8276	0.7789	0.6685	0.6110	0.5920
node2vec	0.6747	0.8615	0.8408	0.7689	0.7650	0.6289	0.5442	0.5642
GraRep	0.9712	0.7864	0.8398	0.7643	0.8070	0.6614	0.6099	0.6563
MNMF	0.9922	0.9968	0.9707	0.9937	0.9911	0.9807	0.9907	0.9874
NECS	0.9993	0.8339	0.8675	0.8313	0.7949	0.7145	0.5718	0.6180
DNR (1)	0.9070	0.8758	0.8998	0.8634	0.8460	0.9134	0.7058	0.8103
DNR (2)	0.9013	0.8726	0.8937	0.8540	0.8484	0.8843	0.6378	0.7365
DNR (5)	0.8851	0.8697	0.8930	0.8479	0.8261	0.9011	0.5639	0.6718
DNR (10)	0.8831	0.8865	0.8582	0.8297	0.7976	0.9196	0.5511	0.6101
SMNMF (1)	0.9926	0.9934	0.9945	0.9978	0.9886	0.9725	0.9591	0.9629
SMNMF (2)	0.9846	0.9504	0.9848	0.9780	0.9829	0.9586	0.9419	0.9046
SMNMF (5)	0.9903	0.9518	0.9772	0.9774	0.9603	0.9273	0.9150	0.9408
SMNMF (10)	0.9753	0.9331	0.9448	0.9431	0.9255	0.8923	0.8825	0.9207

TABLE 6: AUC-ROC of link prediction (70% training set and 30% test set).

Methods	Polbooks	Cornell	Texas	Washington	Wisconsin	Email	Citeseer	Cora
DeepWalk	0.6146	0.7829	0.7621	0.6374	0.5612	0.6322	0.6203	0.5568
Line	0.7750	0.8221	0.8112	0.7858	0.7651	0.6703	0.6171	0.5902
node2vec	0.6411	0.8359	0.8372	0.7398	0.6430	0.6309	0.5451	0.5395
GraRep	0.9640	0.6956	0.8159	0.7440	0.7762	0.6623	0.6179	0.6450
MNMF	0.9900	0.9987	0.9804	0.9960	0.9985	0.9820	0.9910	0.9855
NECS	0.9974	0.8021	0.8148	0.7839	0.7549	0.7135	0.5762	0.6150
DNR (1)	0.8914	0.8802	0.8718	0.8369	0.8720	0.9118	0.6911	0.7884
DNR (2)	0.8944	0.8760	0.8700	0.8482	0.8736	0.8843	0.6174	0.7182
DNR (5)	0.8744	0.8761	0.8750	0.8307	0.8616	0.9012	0.5455	0.6790
DNR (10)	0.8703	0.8891	0.8442	0.8196	0.8334	0.9176	0.5449	0.6281
SMNMF (1)	0.9916	0.9889	0.9955	0.9923	0.9892	0.9733	0.9630	0.9579
SMNMF (2)	0.9843	0.9606	0.9908	0.9780	0.9791	0.9596	0.9401	0.9004
SMNMF (5)	0.9822	0.9475	0.9548	0.9662	0.9486	0.9258	0.9037	0.9399
SMNMF (10)	0.9758	0.9278	0.9615	0.9183	0.9390	0.8940	0.8814	0.9203

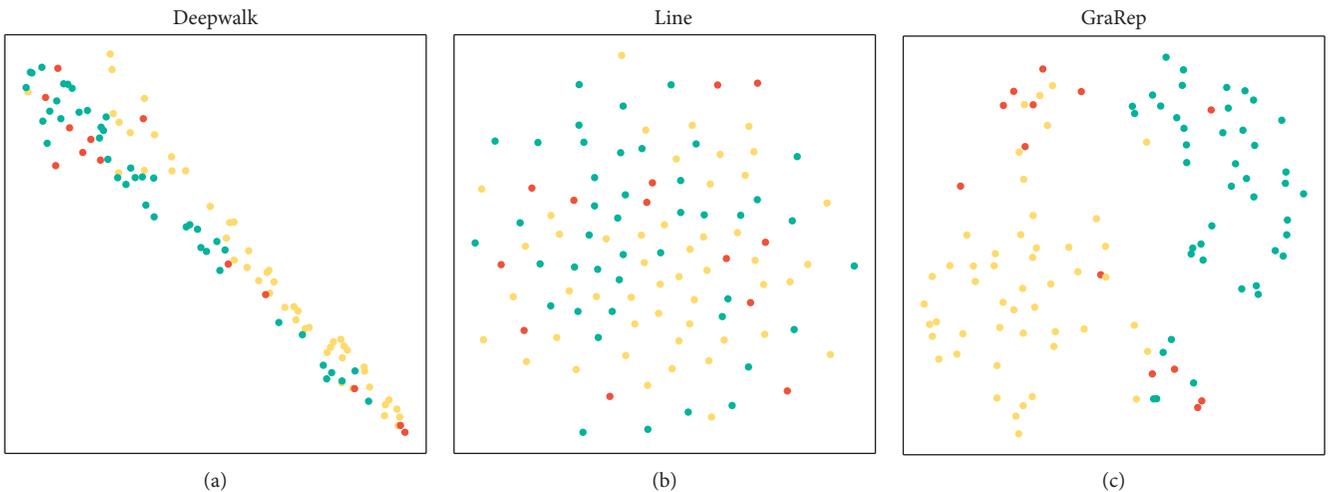


FIGURE 2: Continued.

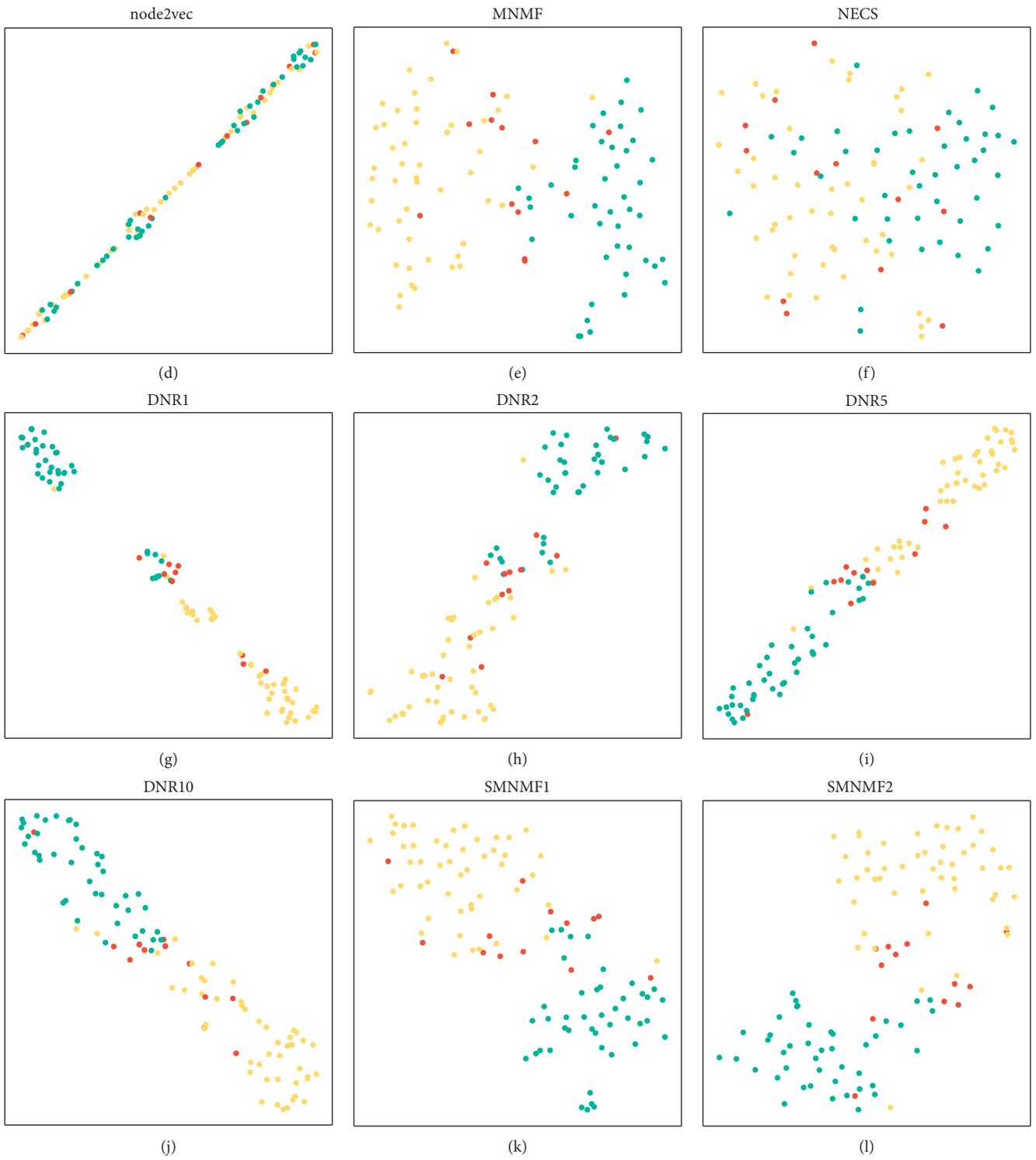


FIGURE 2: Continued.

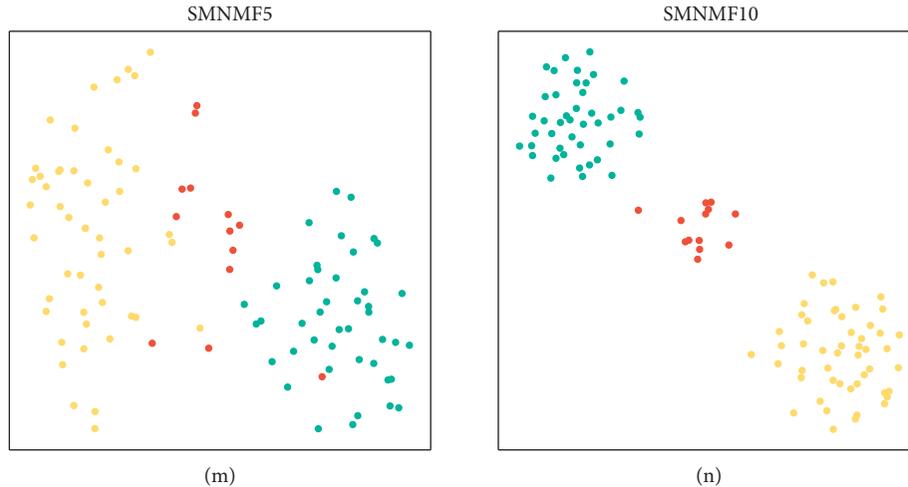


FIGURE 2: *t*-SNE 2D representations of different methods on polbooks.

## 5. Conclusion

In this paper, a semisupervised modularity nonnegative matrix factorization model SMNMF is proposed, which combines the prior information must-link and cannot-link on the basis of the unsupervised network embedding algorithm MNMF, preserving the community structure. First, this paper uses the prior knowledge of must-link and can-link to modify the adjacency matrix of the network and affect the node similarity matrix. Then, the modified adjacency matrix is used to maximize the modularity to model the community structure. The modified node similarity matrix is used to preserve the similarity of the nodes on topology. The experimental results show that the integration of prior information makes the final node representation vectors more discriminative; from the results of node clustering task, the accuracy obtained by the semisupervised model SMNMF is further improved; from the results of link prediction experiments, the AUC-ROC values of the semisupervised SMNMF model are also good; and from the results of visualization task, the semisupervised SMNMF model can distinguish the nodes in different communities more obviously. In the future, we strive to explore whether the priori information can effectively guide the random walk process of traditional representation learning algorithms to generate node representation results that are more consistent with prior knowledge.

## Abbreviations

DeepWalk [5]:	A novel approach for learning latent social representations of vertices
Line [6]:	Large-scale information network embedding
GraRep [7]:	Learning graph representations with global structural information
NEU [8]:	Network embedding update algorithm
AROPE [9]:	Arbitrary-order proximity preserved network embedding
MNMF [12]:	Modularized nonnegative matrix factorization model
CARE [13]:	Community aware random walk for network embedding

NECS [14]:	Network embedding with community structural information
node2vec [22]:	An algorithmic framework for learning continuous feature representations for nodes in networks
MMDW [15]:	Max-margin deepwalk discriminative learning of network representation
GCN [16]:	Semisupervised classification with graph convolutional networks
Planetoid [17]:	Predicting labels and neighbors with embeddings transductively or inductively from data
SDNE [18]:	Structural deep network embedding
DNE-SBP [19]:	Deep network embedding with structural balance preservation
DNE-APP [20]:	Deep network embedding model with aggregated proximity preserving
DNR [21]:	Deep nonlinear reconstruction
SNMF-SS [24]:	A framework which combining SNMF (symmetric nonnegative matrix factorization) and a semisupervised clustering approach
PSSNMF [25]:	A unified semisupervised framework to integrate network topology with prior information.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported by the National Key Research and Development Program of China (2018YFB1701402) and National Natural Science Foundation of China (62072160).

## References

- [1] M. E. J. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, no. 2, pp. 167–256, 2003.
- [2] D. Liu, Y. Jing, and B. Chang, "Identifying influential nodes in complex networks based on expansion factor," *International Journal of Modern Physics C*, vol. 27, no. 9, Article ID 1650105, 2016.
- [3] D. Liu, "A fast and efficient algorithm for mining top-k nodes in complex networks," *Scientific Reports*, vol. 7, p. 43330, 2017.
- [4] C. Peng, "A survey on network embedding," *IEEE Transactions on Knowledge and Data Engineering*, vol. 1, 2018.
- [5] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: online learning of social representations," 2014.
- [6] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: large-scale information network embedding," in *Proceedings of the WWW 2015 Proceedings & Companion*, Florence, Italy, 2015.
- [7] S. Cao, W. Lu, and Q. Xu, *GraRep: Learning Graph Representations with Global Structural Information*, Association for Computing Machinery, New York, NY, USA, 2015.
- [8] C. Yang, "Fast network embedding enhancement via high order proximity approximation," in *Proceedings of the International Joint Conference on Artificial Intelligence AAAI Press*, Melbourne, Australia, 2017.
- [9] Z. Zhang, P. Cui, X. Wang, J. Pei, X. Yao, and W. Zhu, "Arbitrary-order proximity preserved network embedding," in *Proceedings of KDD*, pp. 2778–2786, London, UK, 2018.
- [10] M. Girvan, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, no. 12, 2002.
- [11] F. Lili, "Semi-supervised community detection based on distance dynamics," *IEEE Access*, vol. 1, 2018.
- [12] X. Wang, P. Cui, J. Wang, J. Pei, and Z. Wenwu, "Community preserving network embedding," in *Proceedings of AAAI*, San Francisco, CA, USA, 2017.
- [13] M. M. Keikha, M. Rahgozar, and M. Asadpour, "Community aware random walk for network embedding," *Knowledge-Based Systems*, vol. 148, pp. 47–54, 2018.
- [14] L. Yu, Y. Wang, T. Zhang, and J. Zhang, "Learning network embedding with community structural information," in *Proceedings of the IJCAI*, Macao, China, 2019.
- [15] C. Tu, "Max-margin DeepWalk: discriminative learning of network representation," in *Proceedings of the International Joint Conference on Artificial Intelligence AAAI Press*, New York, NY, USA, July 2016.
- [16] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016.
- [17] Z. Yang, W. Cohen, and R. Salakhutdinov, "Revisiting semi-supervised learning with graph embedding," in *Proceedings of the ICML*, New York, NY, USA, 2016.
- [18] D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," in *Proceedings of the SIGKDD*, pp. 1225–1234, San Francisco, CA, USA, August 2016.
- [19] X. Shen and F.-L. Chung, "Deep network embedding for graph representation learning in signed networks," *IEEE Transactions on Cybernetics*, vol. 50, no. 4, pp. 1556–1568, 2020.
- [20] X. Shen and F.-L. Chung, "Deep network embedding with aggregated proximity preserving," in *Proceedings of the ASONAM*, pp. 40–44, Sydney Australia, 2017.
- [21] L. Yang, X. Cao, D. He, C. Wang, X. Wang, and W. Zhang, "Modularity based community detection with deep learning," in *Proceedings of the IJCAI*, pp. 2252–2258, New York, NY, USA, July 2016.
- [22] A. Grover and J. Leskovec, "node2vec: scalable feature learning for networks," 2016.
- [23] Z. Y. Zhang, "Community structure detection in complex networks with partial background information," 2012.
- [24] X. Ma, X. Yong, and L. Fu, "Semi-supervised clustering algorithm for community structure detection in complex networks," *Physica A: Statistical Mechanics and Its Applications*, vol. 389, no. 1, pp. 187–197, 2010.
- [25] L. Gao, "A unified semi-supervised community detection framework using latent space graph regularization," *IEEE Transactions on Cybernetics*, vol. 45, no. 11, 2015.
- [26] Z. Y. Zhang, K. D. Sun, and S. Q. Wang, "Enhanced community structure detection in complex networks with partial background information," *Scientific Reports*, vol. 3, no. 1, p. 3241, 2013.
- [27] E. J. M. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Physical Review E*, vol. 74, no. 3, Article ID 36104, 2006.
- [28] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.
- [29] Z. Akata, C. Thurau, and C. Bauckhage, "Non-negative matrix factorization in multimodality data for segmentation and label prediction," in *Proceedings of the 16th Computer Vision Winter Workshop*, Mitterberg, Austria, 2011.
- [30] A. Kachites McCallum, K. Nigam, J. Rennie, and K. Seymore, "Automating the construction of internet portals with machine learning," *Information Retrieval*, vol. 3, no. 2, Article ID 1278C163, 2000.
- [31] D. X. Cai, J. Han, and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1548–1560, 2011.
- [32] L. Lovasz and M. Plummer, *Matching Theory*, Akademiai Kiado, Budapest, Hungary, 1986.