A Non-negative Symmetric Encoder-Decoder Approach for Community Detection

Bing-Jie Sun, Huawei Shen, Jinhua Gao, Wentao Ouyang, Xueqi Cheng
{sunbingjie,gaojinhua}@software.ict.ac.cn,{shenhuawei,ouyangwt,cxq}@ict.ac.cn
CAS Key Laboratory of Network Data Science and Technology, Institute of Computing Technology,
Chinese Academy of Sciences, Beijing, China

ABSTRACT

Community detection or graph clustering is crucial to understanding the structure of complex networks and extracting relevant knowledge from networked data. Latent factor model, e.g., non-negative matrix factorization and mixed membership block model, is one of the most successful methods for community detection. Latent factor models for community detection aim to find a distributed and generally low-dimensional representation, or coding, that captures the structural regularity of network and reflects the community membership of nodes. Existing latent factor models are mainly based on reconstructing a network from the representation of its nodes, namely network *decoder*, while constraining the representation to have certain desirable properties. These methods, however, lack an *encoder* that transforms nodes into their representation. Consequently, they fail to give a clear explanation about the meaning of a community and suffer from undesired computational problems. In this paper, we propose a non-negative symmetric encoder-decoder approach for community detection. By explicitly integrating a decoder and an encoder into a unified loss function, the proposed approach achieves better performance over state-of-the-art latent factor models for community detection task. Moreover, different from existing methods that *explicitly* impose the sparsity constraint on the representation of nodes, the proposed approach *implicitly* achieves the sparsity of node representation through its symmetric and non-negative properties, making the optimization much easier than competing methods based on sparse matrix factorization.

CCS CONCEPTS

• **Information systems** → **Clustering**; • **Computing methodologies** → *Machine learning*;

KEYWORDS

Community detection; Latent factor model; Encoder-decoder

CIKM'17 , November 6–10, 2017, Singapore, Singapore

ACM ISBN 978-1-4503-4918-5/17/11...$15.00
https://doi.org/10.1145/3132847.3132902

1 INTRODUCTION

Many real networks, such as social networks, citation networks, and biological networks, are found to divide naturally into communities, i.e., groups of nodes with relatively denser connections within groups but sparser connections between them [1]. Detecting such communities in real networks, has attracted much interest among scientists from various fields in the last decade [2–7]. Many methods for community detection have been proposed and successfully applied to several specific networks [8–10]. Traditional community detection methods are based on network partition that is optimal in terms of certain criteria, e.g., modularity [10] and average length of random walk [5], or partition the whole network by certain heuristics, e.g., iteratively deleting edges with the highest betweenness [1] or bridgeness [11]. For these methods, each node is only allowed to belong to one community, and the community membership of each node is represented as a one-hot representation over communities.

Latent factor model (LFM) is proposed as a new kind of approaches for community detection. Latent factor models for community detection aim to find a distributed and generally low-dimensional representation, or coding, that reflects the community membership of nodes. These models are mainly based on reconstructing the network from the representation of nodes while constraining the representation to have certain desirable properties (e.g., non-negativity). Typical methods include: (1) mixed membership block models [12–15], and (2) non-negative matrix factorization and its variants [16–23]. Mixed membership block models follow a probabilistic framework, where the link probability between nodes is fully determined by their community membership, i.e., their representation. The representation of each node is defined as a multinomial distribution over communities. Such a normalization constraint on node representation indicates that a higher probability of membership to one community implies a lower probability of membership to other communities. Consequently, these models are inappropriate for modeling the mixed membership of nodes [24]. Non-negative matrix factorization methods (NMF) take each node as a dimension of network, and regard the task of community detection as finding a concise and low-dimensional representation of network. However, it doesn't guarantee that the obtained representation corresponds to communities, and thus it generally requires some heuristic constraints to improve its interpretability.

The strength and limitation of latent factor models pose an important and tricky question: for community detection,

what property a good representation of nodes should possess? Intuitively, a desirable representation should be: (1) **distributed**: each node is allowed to belong to multiple communities; (2) **non-negative**: the representation of a node should be non-negative in order to reflect its propensity to each community. This ensures the interpretability of representation and is valuable to understanding the physical meaning of communities; (3) **sparse**: a sparse representation of nodes implies that each node only belongs to a few communities. This property also improves the identifiability of each community, i.e., each community has clear identity relative to other communities.

In this paper, we propose a non-negative symmetric encoder-decoder approach to learn a good representation for community detection. An encoder transforms the original representation of network (e.g., adjacency matrix) into a low-dimensional representation (also known as coding), and a decoder reconstructs the network from the representation. Different from NMF which only considers the loss of the decoder, the proposed approach integrates both decoder and encoder into a unified loss function. The obtained representation of nodes exactly satisfies the aforementioned three desirable properties. First, the representation is distributed, being located in a low-dimensional space. Second, non-negativity constraint is explicitly imposed. This property distinguishes the proposed approach from other encoder-decoders, e.g., autoencoder and SESM [25–28], guaranteeing that the obtained representation corresponds to communities while other encoder-decoders do not. Finally, the symmetry between the encoder and decoder naturally imposes an implicit orthogonality constraint. This constraint, together with the non-negativity constraint, ensures that the representation is sparse. As such, the obtained representation offers us a clear explanation about community membership of each node: the representation of a node reflects its propensity to each community, characterized by its coordinate in the basis vector associated with the community.

Extensive experiments on synthetic and real-world networks demonstrate that the proposed approach achieves a better performance over state-of-the-art latent factor models for community detection task. Furthermore, the proposed approach could be used to detect hierarchical community structure by simply stacking the encoder-decoder into a deep architecture. Moreover, we apply our model to various types of inputs, including cascade data, bipartite networks and directed networks, and the results demonstrate the wide applicability of our model. We also evaluate the generalization capability of the proposed encoder-decoder approach by applying it to the task of link prediction.

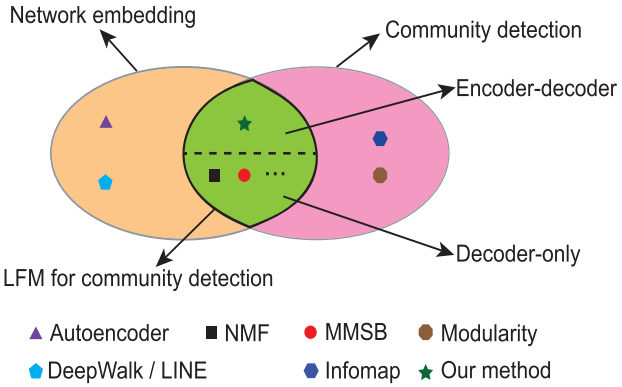


Figure 1: The relationship between our method and the traditional methods of community detection and node representation.

2 RELATED WORK

In this paper, we study the problem of community detection using latent factor models or network representation learning. Therefore, we classify the related works into three categories: community detection methods based on network partition, network embedding methods, and latent factor models for community detection. For clarity, we give a diagram to illustrate the relationship between related work and our work (Figure 1).

2.1 Community detection methods based on network partition

Traditional methods for community detection are mostly based on network partition, taking each component of a partition as a community. For example, Girvan and Newman proposed to find network partition by iteratively deleting the edges with the highest edge betweenness [1]. Rosvall and Bergstrom detected communities by looking for a partition that minimizes the expected description length of a random walk [5]. Shen et al. investigated spectral clustering methods on various types of matrices that characterize the structure of networks [29]. For community detection methods based on network partition, the key is how to evaluate the quality of a network partition. Typical criteria of network partition include the number of edges across communities (or cut), ratio cut [29, 30], normalized cut or conductance [31]. The most widely-used criterion is modularity [8], which is defined as the difference between the fraction of edges within communities and the expected fraction of edges within communities when all edges are randomly placed. Many methods are proposed to find communities through modularity optimization, such as simulated annealing, greedy algorithm, and spectral optimization [8, 10, 32, 33].

For community detection based on network partition, each node is only allowed to belong to one community, i.e., the

community membership of each node is a **one-hot representation** over communities. However, in real world networks one node often belongs to multiple communities. Motivated by the mixed community membership of nodes, researchers resort to finding **distributed representations** of nodes, also known as network embedding.

2.2 Network embedding

Network embedding aims to find a distributed and generally low-dimensional representation of nodes that maximize the likelihood of network. Typical network embedding methods include principal component analysis (PCA) [34], Autoencoder [25, 26], DeepWalk [35], and LINE [36]. PCA is a linear transformation, projecting each node into a low-dimensional space spanned by the top eigenvectors of certain type of matrix of network. Autoencoder is a non-linear dimension reduction technique, which finds a distributed representation of nodes by stacking multi-layer restricted Boltzmann machines (RBMs) [26]. DeepWalk [35] learns a distributed representation of nodes by exploiting the co-occurrence of nodes in truncated random walks. LINE [36] extends DeepWalk by simultaneously modeling both the first-order proximity and the second-order proximity between nodes. Node2Vec [37] was proposed to explore both width-first and depth-first random walk to learn node embedding that are well calibrated with end tasks. Cui et al. [38] proposed to use regularized autoencoders to learn network embedding by exploring high-order and non-linear linking patterns in network.

Although network embedding offers an effective way to obtain a distributed representation of nodes, it fails to give a good community detection method. The reason is that the obtained representation does not correspond to the community structure of network. As a result, we need to conduct clustering on the obtained representation to detect communities. In this paper, we aim to directly learn a distributed representation that corresponds to the community structure of network, without requiring clustering as a post processing.

2.3 Latent factor models for community detection

Latent factor models are used as principled and effective methods for community detection. These models are mainly based on reconstructing the network from the representation of nodes while constraining the representation to have certain desirable properties (e.g., non-negativity). Typical methods include: (1) mixed membership block models [12–15], and (2) non-negative matrix factorization methods and its variants [16–23]. Mixed membership block models follow a probabilistic framework, where the link probability between nodes is fully determined by their representation. The representation of each node is defined as a multinomial distribution over communities. Such a normalization constraint on node representation indicates that a higher probability of membership to one community implies a lower probability of membership to other communities. Consequently, these models are inappropriate for modeling the mixed membership of nodes [24]. Non-negative matrix factorization methods take each node as a dimension of network, and regard the task of community detection as finding a low-dimensional representation of network. However, they don't guarantee that the obtained representation corresponds to communities, thus requiring some heuristic constraint to improve their interpretability.

Existing latent factor models for community detection generally adopt a decoder architecture, i.e., they only consider the reconstruction loss of the original network through the representation of nodes. In this paper, we propose a non-negative symmetric encoder-decoder approach for community detection. The proposed approach integrates the decoder and encoder into a unified loss function. In this way, the obtained representation of a node clearly corresponds to its community membership.

3 NON-NEGATIVE SYMMETRIC ENCODER-DECODER FOR COMMUNITY DETECTION

In this section, we propose a non-negative symmetric encoder-decoder for community detection. Generally speaking, the proposed method could be taken as a kind of non-negative matrix factorization. Before diving into the details of the proposed method, we first introduce some preliminary materials that are useful to understanding it.

Non-negative matrix factorization (NMF) is equivalent to probabilistic latent factor models [14, 17], e.g., the PLSA model [14] and the mixed membership block model [12], except that a normalization constraint is required in probabilistic latent factor models. NMF is a very flexible framework for clustering and exhibits highly competitive performance at an array of scenarios, such as text analysis, recommendation and image clustering. In NMF, the original input is represented as a matrix A, where each row corresponds to a feature and each column corresponds to a sample or an instance over these features. NMF assumes that each feature is a dimension of the original space and aims to project the original space into a new one, anticipating the dimension of the new space could capture the redundance or correlation between the dimensions of the original space. For this purpose, NMF factorizes the original input matrix A into a basis matrix W and a code matrix Z such that $A \approx WZ$. The columns of W span a new space, and the columns of Z corresponds to the representation of samples in the new space.

NMF is a decoder that reconstructs the original input (A) from the representation (Z) in a new space (W). However, this decoder-only architecture suffers from two limitations: (1) It requires an expensive auxiliary algorithm to find the representation associated with a new sample. This problem is also called the fold-in issue in the PLSA model [14]. (2) Even for the samples used in the factorization process of NMF, their representation cannot be obtained by directly projecting the original input into the new space, i.e., Z cannot be obtained by W^TA. To combat these issues, researchers

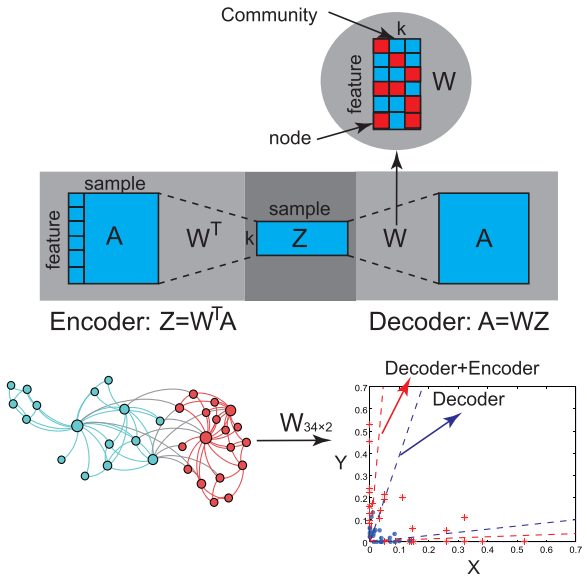


Figure 2: Upper panel: our encoder-decoder model. The decoder obtains the best basis matrix W and code Z to reconstruct the original data. The encoder fine-tunes W and Z to ensure the sparsity and consistency of the decoder and encoder under W. Lower panel: performance on karate club network illustrates the difference of our encoder-decoder model and the decoder-only method in the learnt basis matrix W. Each node in the network is mapped as 2-dimension node representation. Nodes on the axis imply that they belong to only one community or overlapping nodes otherwise. The dash line is the direction of the corresponding communities. Different colors of the dash lines are related to different methods.

naturally resort to an encoder-decoder architecture, e.g., autoencoder [25, 26]. However, existing encoder-decoder methods are inappropriate for community detection since the obtained basis matrix does not satisfy the aforementioned three desirable properties that community detection task requires, i.e., the representation of node over communities should be distributed, non-negative, and sparse. This motivates us to design a non-negative symmetric encoder-decoder for community detection.

3.1 Non-negative symmetric encoder-decoder

The architecture of our proposed non-negative symmetric encoder-decoder is shown in Figure 2. It distinguishes itself from other encoder-decoders via its three properties: (1) the basis matrix W is non-negative, (2) the code matrix Z is non-negative, and (3) the decoder and the encoder is symmetric, i.e., the basis matrix is shared in the processes of both decoder and encoder.

For community detection, each row of the original input A corresponds to a node and each column corresponds to a sample defined over nodes. When we use the adjacency matrix of an undirected network as the input matrix, columns represent the neighborhood of nodes. For directed networks, the neighborhood could be defined by incoming edges or outgoing edges, giving two different adjacency matrices. Accordingly, the obtained community membership differs, reflecting the correlations between nodes at characterizing their incoming neighborhood and outgoing neighborhood. For unweighted network, $A_{ij} = 1$ if there is an edge between node i and node j and $A_{ij} = 0$ otherwise. For weighted networks, A_{ij} can be generalized to represent the weight of edge between i and j. For cascade data, each column of A corresponds to a diffusion cascade and $A_{ij} = 1$ if the user denoted by node i involves in cascade j.

3.1.1 Decoder. In the proposed method, an decoder reconstructs the original data A through a $n \times k$ basis matrix W and a $k \times m$ code matrix Z such that $A \approx WZ$, where n, m and k indicate the number of nodes, the number of training samples and the desired number of communities respectively. The learnt basis matrix W reveals the community membership of nodes, with each column representing a community and the value in one column indicating the propensity of the corresponding node to that community [39]. Minimizing the loss function $\|A - WZ\|_F^2$ gives a pair of W and Z that reconstruct the original data best.

3.1.2 Encoder. An encoder transforms the original data A into the distributed representation using the basis matrix W, i.e., $Z = W^T A$. We expect the code obtained in the encode process to be consistent with the code obtained in the decode process, guaranteeing W to be capable of revealing community structure.

In order to enable the learnt W to indicate node community assignment, we add simple but principled constraints to the parameters. First, the values in W are expected to be non-negative so that they can indicate the propensity of a node to a community. Second, since each node only belongs to one or a few communities at most, we expect the rows in W to be sparse. Contrast to traditional latent factor models whose sparsity constraints are achieved through an explicit sparse regularization term on W or orthogonality constraints that $WW^T = I$, we simply require that the decoder and the encoder share the same basis matrix W. This symmetry constraint guarantees soft orthogonality constraint naturally. Specifically, when the loss function is well optimized, the following two equations hold

$$A \approx WZ, \tag{1}$$

$$Z \approx W^T A. \tag{2}$$

Substituting (2) into (1), we have

$$A \approx WW^T A, \tag{3}$$

which implies that ensuring Eqs. (1) and (2) naturally give rise to an orthogonality constraint on W, which is one of

the most important implication of the symmetric architecture. Consequently, the symmetry constraint together with the non-negativity constraint naturally ensures a sparsity constraint on W in our encoder-decoder architecture.

3.2 Loss function and optimization

The loss function of our method can be written as

$$\begin{aligned} L &= \|A - WZ\|_F^2 + \|Z - W^T A\|_F^2 \\ s.t. &\quad W \geq 0,\ Z \geq 0. \end{aligned} \tag{4}$$

$\|\cdot\|_F^2$ is the square of the Frobenius norm. We adopt multiplicative updating rules [16] to update the basis matrix W as,

$$W \leftarrow W \otimes \frac{AZ^T}{WZZ^T + AA^TW}. \tag{5}$$

In the same way, the low dimension representation Z can be updated as,

$$Z \leftarrow Z \otimes \frac{W^TA}{W^TWZ + Z}. \tag{6}$$

We stop the iteration when the change in the loss is less than 10^{-6}.

After optimization, the basis matrix W stores the community structure information, with each column corresponding to a community. For each node, we choose the column index of the largest value in its representation to be its community label.

3.3 Connection with methods based on Non-negative Matrix factorization

The proposed encoder-decoder method is a kind of matrix factorization method. Here, we focus on clarifying the difference between the proposed method and existing matrix factorization methods, including standard matrix factorization, non-negative matrix factorization (NMF) [17], orthogonal NMF (Orth_NMF) [18], and projective NMF (PNMF) [21].

First, matrix factorization is an effective technique for representation learning by reconstructing the original data. Difference among various matrix factorization methods lies in the constraints imposed on the basis matrix and the code matrix. Non-negative matrix factorization is distinguished from other matrix factorization by its use of non-negativity constraints. These constraints lead to a parts-based representation because they allow only additive, not subtractive, combinations. However, as pointed out in [25], non-negativity does not guarantee the sparsity of representation. Researchers have to add explicit sparsity constraint on NMF to achieve the sparsity of representation. Subsequently, Orth_NMF is proposed, requiring that the column of basis matrix or the row of code matrix is orthogonal. Combining the orthogonality and non-negativity, the sparsity of representation is hard-coded in Orth_NMF. PNMF is another variant of Orth_NMF, directly obtaining the subspace spanned by the basis matrix with a loss function $A - WW^TA$. In our method, the sparsity is achieved by the non-negativity constraint and the symmetry between the decoder and encoder.

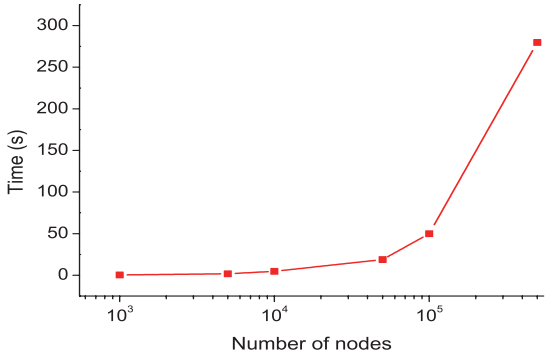


Figure 3: Scalability performance of our method on benchmark networks at different scales. The scale of networks varies from $1,000$ nodes to $500,000$ nodes. Benchmark networks are generated by the "benchmark" tool [40] with similar parameter setting as illustrated in section 4.1.2. For all the networks, our model runs a fixed number of iterations to collect the time consumption information.

Second, from the perspective of optimization, Orth_NMF and PNMF both require a hard orthogonality constraint while our method achieves this objective implicitly. As such, during the optimization process, our method does not need to normalize the basis matrix, and thus is more efficient than the Orth_NMF and PNMF.

3.4 Time complexity analysis

Bulk of the computation depends on the matrix multiplication in the updating rules. The computations of updating rules in (5) and (6) run in $O(n^2k + nk^2 + n^2k)$ and $O(n^2k + nk^2)$ respectively. Since $k \ll n$, consequently, the overall time complexity of the encoder-decoder method is $O(n^2k)$, which is the same order of magnitude as NMF methods. We test the scalability of our method on various benchmark networks at different scales. The result is shown in Figure 3.

4 EXPERIMENTS

To validate the effectiveness of our method, we apply it to detecting communities on both synthetic networks and real-world networks. The results demonstrate the superiority of our method over baseline models. We also apply our method to certain tasks involving various types of network data, including cascade data, bipartite networks, hierarchical networks and directed networks, demonstrating the wide applicability of our method. Furthermore, we validate the learnt representations of nodes through link prediction task, and the results demonstrate the learnt basis matrix can better capture the structure of input networks.

4.1 Community detection in networks

We apply our methods to both synthetic networks where the ground truth of community assignments is known, and

real-world networks where node labels instead of community memberships are provided. Six baseline models are adopted for comparison, and classic evaluation measures are utilized in those two datasets.

4.1.1 Baseline models. We adopt six representation-based methods as our baseline models. As categorized in Figure 1, we choose DeepWalk [35] and LINE [36] as the representatives of network embedding. For latent factor models, we select MMSB [13] which adopts statistical model to maximize the likelihood of generating the original network, and three NMF-based methods which aim to reconstruct the original input of networks using matrix decomposition, including NMF [19], Orth_NMF [18] and PNMF [21].

- DeepWalk: this model uses local information obtained from truncated random walks on the network to learn latent representations by treating walks as the equivalent of sentences as done in word2vec [41].
- LINE: this model proposes to learn better representations of nodes by designing an objective function which preserves both the first-order and the second-order proximities between nodes.
- MMSB: in this model, whether two nodes are connected is determined by the mixed community membership of nodes, i.e. the representation of nodes.
- NMF: this model reconstructs the original data using matrix factorization with non-negative constraints on the basis matrix W.
- Orth_NMF: this model reconstructs the original data based on NMF and constrain each dimension of the basis matrix W to be orthogonal.
- PNMF: this model projects the original data to a subspace by minimizing $\|A - WW^T A\|_F^2$.

For DeepWalk and LINE, we first learn the embedding of the nodes, and then apply standard k-means algorithm [42] to identify the communities in the networks.

4.1.2 Expriments on synthetic networks. We adopt the "benchmark" tool introduced in [40] to generate synthetic networks. Three key parameters have to be set when generating benchmark networks, i.e., the power exponent of the degree distribution α, the power exponent of the community size distribution β, and the mixing parameter μ. The mixing parameter controls the fraction of a node's links that connect to nodes in other communities. When $\mu = 0.2$, the structures of generated networks are very close to that of the real-world networks. Thus we set $\mu = 0.2$ in our experiments, and α and β are set to be 2 and 1 respectively. Besides, we set the maximum degree as $k_{max} = n^{1/\alpha}$, i.e., $k_{max} = \sqrt{n}$ in our case, and the average degree of nodes as $k_{avg} = 0.6 \times k_{max}$, where n is the desired number of nodes of generated network. For community size, we set the minimum and maximum community size to be $[n/50, n/20]$ as in Gopalan et al. [13]. We generate three networks under the above settings, with the number of nodes being set to be 500, 5000 and 10,000 respectively. The adjacency matrices of generated networks are provided as the original inputs.

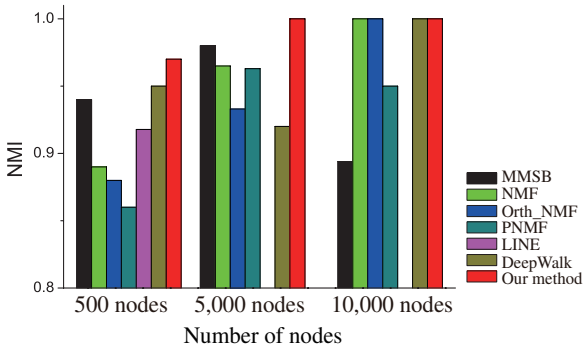


Figure 4: Performance comparison on synthetic networks using NMI. The results of LINE on networks with 5,000 and 10,000 nodes are hard to be seen in this figure, as it achieves quite low results (<1e-6).

For evaluation, as community assignments of nodes are known, we adopt normalized mutual information (NMI) [40], which measures the extent to which the true and discovered community labels are consistent with each other, to measure the performance of different models. For each algorithm we run it 10 times and take the best NMI value among all the 10 runs to alleviate the effect of local optimum caused by random initialization.

As shown in Figure 4, our method outperforms or at least perform fairly well as the baseline methods in most cases. The difference in performance between our method and NMF-based methods can be explained by considering the optimization step. Orth_NMF and PNMF restrict the sparsity through orthogonality constraint which is difficult to optimize and easy to get trapped in local optimum. In fact, for Orth_NMF, if we set the basis matrix W to be the one obtained by our method, and conduct the optimization process, we can get lower loss value than that obtained by itself, which indicates that Orth_NMF is hard to be optimized.

In contrast, our method ensures the sparsity through parameter symmetry (or parameter sharing) of the decode and encode process which is more flexible and easier to obtain results close to global optimum. We also check the performance of our method on synthetic networks with μ varying from 0.1 to 0.5, in which our method obtain consistent performance over different settings of μ.

4.1.3 Experiments on real world networks. In real-world networks, the underlying community structure is usually unknown. Instead, there are substantial labels related to the nodes, indicating certain attributes of the nodes, e.g. hobbies, research areas, and social groups [43].

We adopt the Amazon dataset and the DBLP dataset provided in the SNAP project [44] for our experiments. The statistics of these two datasets are shown in Table 1. The network of Amazon dataset characterizes the co-purchased products on the Amazon website, with each node representing a product, and each link indicating that those two products are frequently co-purchased. The product category provided

Table 1: Real-world networks with node labels

Dataset	Nodes	Edges	Average community size
Amazon	334,863	925,872	99.86
DBLP	317,080	1,049,866	429.79

Table 2: Performance of community purity on Amazon and DBLP network

	Method	Amazon	DBLP
Network embedding method	LINE	0.159	0.024
	DeepWalk	0.340	0.102
Decoder-only LFM	MMSB	0.169	0.023
	NMF	0.360	0.130
Encoder-decoder method	Our method	**0.612**	**0.170**

by Amazon website is also included in the dataset, which is treated as the ground-truth label for each product. The DBLP network is a collaboration network. The nodes are authors, and two authors are connected if they have published at least one paper together. The publication venues, e.g., the journal or conference, are treated as the labels of nodes. The adjacency matrices are adopted as the original inputs for both networks. As the ground-truth of community structure is unknown in real-world settings, community purity [45] is utilized to evaluate our method, which measures the ratio of the dominant label in the communities discovered by those latent factor models.

Purity, as a measure to evaluate the results of community detection methods, is defined as

$$P = \frac{1}{k}\sum_{i=1}^{k}\max_{1\le j\le r}\frac{|C_{ij}|}{|C_i|}, \tag{7}$$

where k is the number of detected communities, r is the number of distinct labels and $|C_{ij}|$ is the number of label j in community i. Higher purity score indicates better performance.

Following the results on synthetic datasets, we adopt four models as our baselines in real-world settings, i.e., LINE and DeepWalk as network embedding works, and MMSB and NMF as latent factor models. The number of communities is set to be $k = 30$ and $k = 100$ respectively for the Amazon dataset and the DBLP dataset. For DeepWalk and LINE, we adopt the official toolkits and conduct experiments in default settings. The size of latent representations is set to be 64 for both models, which is a default setting in DeepWalk toolkit. For our method, we run 500 iterations for optimization. The results are shown in Table 2.

Our method significantly outperforms other baselines. For network embedding models, DeepWalk outperforms LINE and has a competitive performance compared to NMF. The improvements of our method over NMF demonstrates the effectiveness of introducing the encoder architecture. Besides,

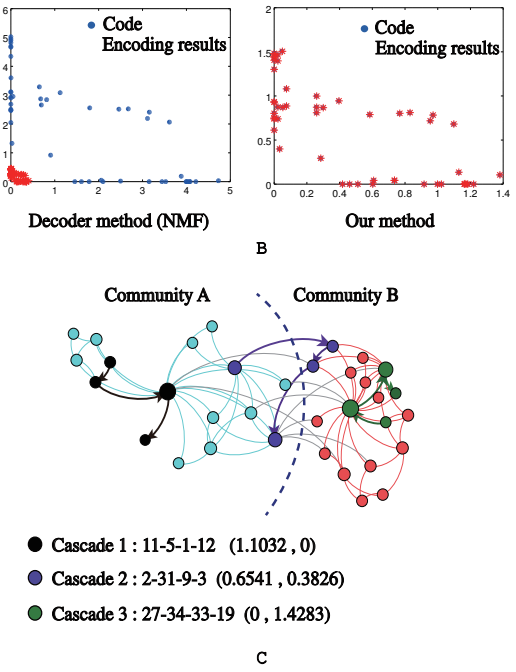


Figure 5: Performance comparison of encoding ability on karate club network. Panel (a): The comparison of embedding of the encoding results (W^TA) and the learnt code (Z) in new space. The blue circle represents the learnt code Z and the red star represents the code W^TA obtained from encoding. Nodes on the axis imply that the cascades locate inside one community, while the others imply these cascades skip between communities. Panel (b): The new cascades are obtained from the communities discovered by our method. Cascade 1 and 3 are random walks inside community A and B separately and cascade 2 skips between the two communities. The code of these new cascades are shown in the brackets.

the significant performance difference between those two datasets is resulted from the intrinsic nature of those two datasets. The link of the DBLP network characterizes the co-authorship while the label indicates the publication venue. Authors in the same community, i.e., authors who co-author with each other, are not likely to publish in the same venue, which results in the low community purity in the DBLP network. Meanwhile, for the Amazon dataset, products which are frequently co-purchased are more likely to belong to the same category.

4.2 Applications

In this section, various types of network data, including cascade data, bipartite networks, hierarchical networks, and directed networks, are used to validate our method, and the results demonstrate the wide applicability of our method.

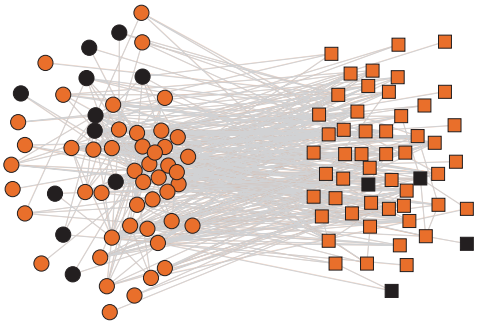


Figure 6: The adjacency network of English words. Node groups corresponding to adjectives and nouns are respectively denoted by circle and square. The dark nodes correspond to the mistaken nodes.

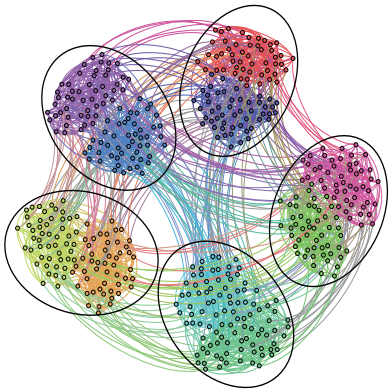


Figure 7: The hierarchical community structure discovered through stacking our method. The colors of nodes illustrate the detected communities in the first layer, with nodes in a community sharing the same color. The circles illustrate the detected communities of the second layer.

Furthermore, we validate the learnt representations of nodes using link prediction task.

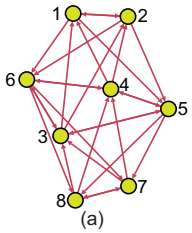
4.2.1 Learning community membership from cascade data. To further illustrate the performance of our methods, we adopt random walk cascades on networks as our original data. The samples of our input are cascades with nodes serving as features. We collect $1,500$ cascades and each cascade is a 4-step random walk on the karate club network. We run our algorithm on the original data to obtain the basis matrix W and the code of the samples Z. To demonstrate the encoding ability of our method compared with NMF which considers only the decode process, we choose 50 cascades randomly from the original data and encode them into the low dimension space through the basis matrix W learnt by NMF and our method respectively. The number of communities is set to be 2 for the convenience of visualization. The results of NMF and our method are shown in Figure 5 (a).

The encoded representations using W trained by our method almost perfectly match the codes learnt during the training process, while for NMF, the encoded representations differ much. It indicates that the basis matrix W can better serve to encode the original data. Furthermore, we encode some new cascades to validate the encoding ability. The result is shown in Figure 5 (b). The learnt representations perfectly match the structures of cascades, as cascade 1 diffuses in the left community, cascade 3 wanders in the right community, and cascade 2 walks across the communities.

4.2.2 Community detection in bipartite networks. Now we illustrate our model's capability of detecting bipartite community structure. The test network is the adjacent network of English words taken from [46]. In this network, 112 nodes represent commonly occurring adjectives and nouns in the novel *David Copperfield* by Charles Dickens, with edges connecting any pair of words that appear adjacent to each other at any place in the text. Generally, adjectives appear next to nouns in English. Thus most edges in the network connect an adjective and a noun, and the network is approximately bipartite, i.e., this network possesses anti-community structure. This can be illustrated in Figure 6, where the adjectives and nouns are respectively represented by circles and squares.

We fit our model to this network with the number of communities being set to be 2, and assign each node to its most preferred group. As shown in Figure 6, the resulted two disjoint groups well separate the adjectives from nouns. In fact, 97 of all the 112 nodes are correctly classified. This indicates that our method is effective at the detection of anti-community structure.

4.2.3 Detecting hierarchical community structure. We stack layers of our model, by taking the basis matrix learnt in the previous layer as the input for the next layer, to detect the hierarchical community structure. To illustrate the performance of our method, we generate hierarchical synthetic networks using LFR "benchmark" tool [40]. The number of nodes in the generated network is set to be $n = 500$, and the maximum degree and average degree of the nodes are set to be 10. The mixing parameters of the first level community and the second level community are both set to be 0.1. The minimum and maximum community size of the first level is 50 and the second level is 100. In the first layer, we use the adjacent matrix of the synthetic network as input, the basis matrix W is learnt as output. We set $k = 10$ to discover the 10 communities planted in the network. In the second layer, we use the learnt W in the first layer as input. The first level communities are features in this layer, and we intend to learn another W_2 with each dimension indicating the relative importance of first-level communities. In this case, each dimension of W_2 represents the higher level community structure of hierarchical network. The detected hierarchical community structure of the synthetic network by stacking our models is shown in Figure 7 .



Out \ In	{1,2,5,6}	{3,4,7,8}
{1,2,3,4}	Connected	-
{5,6,7,8}	-	Connected

(b)

Figure 8: The structure and the construction rules of the directed network.

4.2.4 Detecting communities in directed networks. In directed networks, nodes can always serve dual roles. For example, in online social networks, users serve as followers who follow other users, as well as followees who are followed by others. From the followers' perspective, we can detect communities of users who have followed similar users. From the followees' view, we can group together users who are followed by similar users.

To demonstrate our model's capability of detecting dual-role communities in directed networks, we apply our model to a toy-network constructed by Shen et al. [15]. The network and the construction rules are shown in Figure 8. For connected *Out-set* and *In-set*, we use directed edges to link each node in the *Out-set* to all the nodes in the connected *In-set*, with self links being excluded. It can tell from the construction rules that the network can be divided into different partitions from the incoming link perspective and the outgoing link perspective. Thus we construct two types of inputs to account for the dual roles of nodes. In the incoming-view input, each node is represented as a n-dimensional vector using its incoming neighbors, i.e. the j-th entry of node i's representation is 1 if there exists a link from node j to node i. And for the outgoing-view input, the j-th entry of node i's representation is 1 if there exists a link from node i to node j. We apply our model to these two types of inputs and obtain expected results, i.e., the nodes in the incoming-view input are grouped as $\{\{1,2,5,6\},\{3,4,7,8\}\}$ and the nodes in the outgoing-view input are divided into $\{\{1,2,3,4\},\{5,6,7,8\}\}$.

4.2.5 Link prediction. As community detection is an unsupervised task, to evaluate the generalization capability of latent factor models for community detection, it is valuable to illustrate the performance of these models at link prediction task. Link prediction task aims to predict the missing links by calculating the similarities between nodes. Representations learnt by latent factor models should both well reveal the community structure and capture the similarities between nodes. Thus we adopt link prediction task to further demonstrate the generalization capability of our model.

We sample two sub-networks from the Amazon dataset and the DBLP dataset respectively using the same sampling strategy as in [47]. The subgraphs of Amazon and DBLP has $2,060$ nodes $6,138$ edges and $2,846$ nodes $8,172$ edges respectively. 15% of existing links in each graph are selected as test data and eliminated from the network, and the resulted network serves as training data. *Precision@k* (*P@k*) is adopted to validate the performance [38, 48]. The results of our models and other three baselines are shown in Table 3. Our model achieves the best or competitive results on both datasets, which demonstrates the learnt representations can better capture the structures of input networks.

Table 3: Performance of link prediction on Amazon and DBLP network

Data	Method	P@k(%)		
		P@1000	P@2000	P@3000
Amazon	LINE	4.9	3.0	2.37
	DeepWalk	**8.8**	7.65	6.6
	NMF	5.7	5.7	5.4
	Our method	8.7	**8.1**	**6.7**
DBLP	LINE	4.4	2.9	2.13
	DeepWalk	5.7	5.0	4.13
	NMF	10.6	**7.45**	5.6
	Our method	**11.8**	7.3	**5.63**

5 CONCLUSIONS

In this paper, we proposed a non-negative symmetric encoder-decoder approach to combat the problem of community detection. The proposed approach achieves a distributed, non-negative and sparse representation of nodes. Such a representation clearly reveals the intrinsic community structure of networks. It distinguishes itself from existing latent factor models by its joint optimization of decoder and encoder loss. Meanwhile, it outperforms other encoder-decoder methods by its non-negativity and symmetry property. Extensive experiments on synthetic and real-world networks demonstrate that the proposed approach achieves better performance over state-of-the-art latent factor models at community detection task. Furthermore, the proposed approach is also applicable to detecting hierarchical community structure and various structural regularities in bipartite networks and directed networks.

ACKNOWLEDGMENTS

This work was funded by the National Basic Research Program of China (973 Program) under grant numbers $2013CB329606$ and $2014CB340401$, and the National Natural Science Foundation of China under grant numbers 61472400, 61425016, 61202215, 61433014 and 61602439. Huawei Shen is also funded by K.C.Wong Education Foundation, Youth Innovation Promotion Association CAS and the CCF-Tencent RAGR (No. 20160107).

REFERENCES

[1] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.
[2] S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3):75–174, 2010.

[3] B. J. Sun, H. W. Shen, and X. Q. Cheng. Detecting overlapping communities in massive networks. *EPL*, 108(6):68001, 2014.

[4] G. Palla, I. Derényi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, 2005.

[5] M. Rosvall and C. T. Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118–1123, 2008.

[6] H. W. Shen, X. Q. Cheng, K. Cai, and M. B. Hu. Detect overlapping and hierarchical community structure in networks. *Physica A*, 388(8):1706–1712, 2009.

[7] H. W. Shen, X. Q. Cheng, and B. X. Fang. Covariance, correlation matrix, and the multiscale community structure of networks. *Physical Review E*, 82(1):016114, 2010.

[8] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(2):026113, 2004.

[9] M. E. J. Newman and E. A. Leicht. Mixture models and exploratory analysis in networks. *Proceedings of the National Academy of Sciences*, 104(23):9564–9569, 2007.

[10] V. D. Blondel, J. L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics*, P10008, 2008.

[11] X. Q. Cheng, F. X. Ren, H. W. Shen, Z. K. Zhang, and T. Zhou. Bridgeness: a local index on edge significance in maintaining global connectivity. *Journal of Statistical Mechanics*, P10011, 2010.

[12] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9:1981–2014, 2008.

[13] P. K. Gopalan and D. M. Blei. Efficient discovery of overlapping communities in massive networks. *Proceedings of the National Academy of Sciences*, 110(36):14534–14539, 2013.

[14] B. Ball, B. Karrer, and M. E. J. Newman. Efficient and principled method for detecting communities in networks. *Physical Review E*, 84(3):036103, 2011.

[15] H. W. Shen, X. Q. Cheng, and J. F. Guo. Exploring the structural regularities in networks. *Physical Review E*, 84(5):056111, 2011.

[16] S. Choi. Algorithms for orthogonal nonnegative matrix factorization. In *IEEE International Joint Conference on Neural Networks*, pp. 1828–1832, 2008.

[17] C. Ding, X. F. He, and H. D. Simon. On the equivalence of nonnegative matrix factorization and spectral clustering. In *Proceedings of the 2005 SIAM International Conference on Data Mining*, pp. 606–610, 2005.

[18] C. Ding, T. Li, W. Peng, and H. Park. Orthogonal nonnegative matrix t-factorizations for clustering. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge Discovery and Data mining*, pp. 126–135, 2006.

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

[20] W. Ren, G. Y. Yan, X. P. Liao, and L. Xiao. Simple probabilistic algorithm for detecting community structure. *Physical Review E*, 79(3):036111, 2009.

[21] Z. J. Yuan and E. Oja. Projective nonnegative matrix factorization for image compression and feature extraction. *Scandinavian Conference on Image Analysis*, Springer, 333–342, 2005.

[22] J. Yang and J. Leskovec. Overlapping community detection at scale: a nonnegative matrix factorization approach. In *Proceedings of the sixth ACM international conference on Web Search and Data Mining*, pp. 587–596, 2013.

[23] D. Jin, D. X. He, Q. H. Hu, C. Baquero, and B. Yang. Extending a configuration model to find communities in complex networks. *Journal of Statistical Mechanics*, P09013, 2013.

[24] M. Kim and J. Leskovec. Latent multi-group membership graph model. In *Proceedings of the 29th International Conference on Machine Learning*, pp. 947–954, 2012.

[25] Y. Boureau and Y. LeCun. Sparse feature learning for deep belief networks. In *Advances in Neural Information Processing Systems*, pp. 1185–1192, 2008.

[26] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[27] A. Lemme, R. F. Reinhart, and J. J. Steil. Online learning and generalization of parts-based image representations by non-negative sparse autoencoders. *Neural Networks*, 33:194–203, 2012.

[28] S. Michele Rajtmajer, B. Smith, and S. Phoha. Non-negative sparse autoencoder neural networks for the detection of overlapping, hierarchical communities in networked datasets. *Chaos*, 22(4):043141, 2012.

[29] H. W. Shen and X. Q. Cheng. Spectral methods for the detection of network community structure: a comparative analysis. *Journal of Statistical Mechanics*, P10020, 2010.

[30] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W Mahoney. Statistical properties of community structure in large social and information networks. In *Proceedings of the 17th international conference on World Wide Web*, 695–704, 2008.

[31] J. B. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[32] M. E. J. Newman. Spectral methods for community detection and graph partitioning. *Physical Review E*, 88(4):042822, 2013.

[33] H. W. Shen, X. Q. Cheng, and J. F. Guo. Quantifying and identifying the overlapping community structure in networks. *Journal of Statistical Mechanics*, P07042, 2009.

[34] I. Jolliffe. 2002. *Principal component analysis*. Wiley Online Library.

[35] B. Perozzi, R. Al-Rfou, and S. Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge Discovery and Data mining*, pp. 701–710, 2014.

[36] J. Tang, M. Qu, M. Z. Wang, M. Zhang, J. Yan, and Q. Z. Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, pp. 1067–1077, 2015.

[37] A. Grover and J. Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 855–864, 2016.

[38] D. X. Wang, P. Cui, and W. W. Zhu. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1225–1234, 2016.

[39] W. Xu, X. Liu, and Y. H. Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and Development in Informaion Retrieval*, pp. 267–273, 2003.

[40] A. Lancichinetti and S. Fortunato. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Physical Review E*, 80(1):016118, 2009.

[41] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pp. 3111–3119, 2013.

[42] J. A. Hartigan and M. A. Wong. Algorithm AS 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C*, 28(1):100–108, 1979.

[43] L. Peel, D. B. Larremore, and A. Clauset. The ground truth about metadata and community detection in networks. *Science Advances*, 3(5):e1602548, 2017.

[44] J. Yang and J. Leskovec. Defining and evaluating network communities based on ground-truth. *Knowledge and Information Systems*, 42(1):181–213, 2015.

[45] Z. Y. Zhao, S. Z. Feng, Q. Wang, J. Z. Huang, G. J. Williams, and J. P. Fan. Topic oriented community detection through social objects and link analysis in social networks. *Knowledge-Based Systems*, 26:164–173, 2012.

[46] M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74(3):036104, 2006.

[47] L. F. He, C. T. Lu, J. Q. Ma, J. P. Cao, L. L. Shen, and P. S. Yu. Joint community and structural hole spanner detection via harmonic modularity. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 875–884, 2016.

[48] D. Liben-Nowell and J. Kleinberg. The link-prediction problem for social networks. *Journal of the Association for Information Science and Technology*, 58(7):1019–1031, 2007.