Modularity Based Community Detection with Deep Learning

Liang Yang,[1,2,3] **Xiaochun Cao,**[1] **Dongxiao He,**[3,*] **Chuan Wang,**[1] **Xiao Wang,**[4] **Weixiong Zhang**[5,6]
[1]State Key Laboratory of Information Security,
Institute of Information Engineering, Chinese Academy of Sciences
[2]School of Information Engineering, Tianjin University of Commerce
[3]School of Computer Science and Technology,
Tianjin Key Laboratory of Cognitive Computing and Application, Tianjin University
[4]Department of Computer Science and Technology, Tsinghua University
[5]College of Math and Computer Science, Institute for Systems Biology, Jianghan University
[6]Department of Computer Science and Engineering, Washington University in St. Louis
{yangliang,caoxiaochun,wangchuan}@iie.ac.cn, hedongxiao@tju.edu.cn,
wxiaotju@gmail.com, weixiong.zhang@wustl.edu

Abstract

Identification of module or community structures is important for characterizing and understanding complex systems. While designed with different objectives, i.e., stochastic models for regeneration and modularity maximization models for discrimination, both these two types of model look for low-rank embedding to best represent and reconstruct network topology. However, the mapping through such embedding is linear, whereas real networks have various nonlinear features, making these models less effective in practice. Inspired by the strong representation power of deep neural networks, we propose a novel nonlinear reconstruction method by adopting deep neural networks for representation. We then extend the method to a semi-supervised community detection algorithm by incorporating pairwise constraints among graph nodes. Extensive experimental results on synthetic and real networks show that the new methods are effective, outperforming most state-of-the-art methods for community detection.

1 Introduction

Real-world systems often appear in networks, e.g., social networks in Facebook media, protein interaction networks, power grids and the Internet. Real-world networks often consist of functional units, which manifest in the form of network modules or communities, subnetworks with nodes more tightly connected with respect to the rest of the networks. Finding network communities is, therefore, critical for characterizing the organizational structures and understanding complex systems.

A great deal of effort has been devoted to developing network community finding methods, among which, two are widely adopted and thus worth mentioning. The stochastic model [Psorakis *et al.*, 2011] focuses on deriving generative models of networks. Such a generative model in essence maps a network to an embedding in a low-dimensional latent space. The mapping can be done by, e.g., nonnegative matrix factorization (NMF) [Wang *et al.*, 2008]. Unlike the stochastic model, the modularity maximization model [Newman, 2006], as the name suggests, attempts to maximize a modularity function on network substructures. The optimization can be done by eigenvalue decomposition (EVD), which is equivalent to reconstructing a low-rank modularity matrix.

In short, while appeared in different forms, the stochastic model and modularity maximization model share an essential commonality, i.e., mapping a network to a low-dimensional, latent space embedding. Motivated by the strong discrimination power of the modularity maximization model and the relationship between maximizing modularity and reconstructing *modularity matrix* by finding low-dimensional embedding, we aim to seek a more effective reconstruction algorithm for modularity optimization. However, all these types of embedding adopted in the two popular models are linear, e.g., NMF in the stochastic model and EVD in the modularity maximization model. This is in sharp contrast to the fact that real-world networks are full of nonlinear properties, e.g., a relationship (e.g., distance) among nodes may not necessarily be linear. As a result, the representation power of these linear mapping based models is limited on real-world networks.

Neural networks, particularly that with deep structures, are well known to provide nonlinear low-dimensional representations [Bourlard and Kamp, 1988]. They have been successfully applied to complex problems and systems in practice, such as image classification, speech recognition, and playing an ancient strategic board game of Go. To the best of our knowledge, however, deep neural networks have not yet been successfully applied to community detection.

Taking advantage of the nonlinear representation power of deep neural networks, we propose in this paper a nonlinear reconstruction (DNR) algorithm for community detection using deep neural networks.

*Corresponding author.

It is known that information beyond network topology can greatly aid network community identification [Zhang, 2013; Yang *et al.*, 2015]. Examples of such information include semantics on nodes, e.g., names and labels, and constraints on relationships among nodes, e.g., community membership constraints between adjacent nodes (pairwise constraints). In the current study, we extend our DNR method to a semi-supervised DNR (semi-DNR) algorithm to explicitly incorporates pairwise constraints among nodes to further improve community detection.

2 Reconstruction based Community Detection

We consider an undirected and unweighted graph $G = (V, E)$, where $V = \{v_1, v_2, ..., v_N\}$ is the set of N vertices, and $E = \{e_{ij}\}$ the set of edges each of which connects two vertices in V. The adjacency matrix of G is a nonnegative symmetric matrix $\mathbf{A} = [a_{ij}] \in \mathbb{R}_+^{N\times N}$ where $a_{ij} = 1$ if there is an edge between vertices i and j, or $a_{ij} = 0$ otherwise, and $a_{ii} = 0$ for all $1 \leq i \leq N$. The degree of vertex i is defined as $k_i = \sum_j a_{ij}$. The problem of community detection is to find K modules or communities $\{V_i\}_{i=1}^K$ that are subgraphs whose vertices are more tightly connected with one another than with outside vertices. Here, we consider disjoint communities, i.e., $V_i \cap V_j = \emptyset$ for $i \neq j$.

2.1 Stochastic Model

In stochastic model [Psorakis *et al.*, 2011; He *et al.*, 2015; Jin *et al.*, 2015], a_{ij} can be viewed as the probability that vertices i and j are connected. This probability can be further considered to be determined by the probabilities that vertices i and j generate edges belonging to the same community. We introduce latent variables $\mathbf{H} = [h_{ik}] \in \mathbb{R}_+^{N\times K}$ with h_{ik} representing the probability that node i generates an edge belonging to community k. This latent variable also captures the probability that node i belongs to community k, and each row of $\mathbf{H}$ can be considered as a community membership distribution of a vertex. The probability that vertices i and j is connected by a link belonging to community k is then $h_{ik}h_{jk}$, and the probability that they are connected is:

$$\hat{a}_{ij} = \sum_{k=1}^{K} h_{ik}h_{jk}.$$

As a result, the community detection problem can be formulated as a nonnegative matrix factorization $\mathbf{A} \approx \hat{\mathbf{A}} = \mathbf{H}\mathbf{H}^T$. The NMF-based community detection approaches [Psorakis *et al.*, 2011] aim to find a nonnegative membership matrix $\mathbf{H}$ to reconstruct adjacency matrix $\mathbf{A}$. There are two common objective (loss) functions to quantify the reconstruction error. The first is based on the square loss function [Wang *et al.*, 2008; Zhang *et al.*, 2007] which is equivalent to the square of the Frobenius norm of the difference between two matrices

$$\mathcal{L}_{LSE}(\mathbf{A}, \mathbf{H}\mathbf{H}^T) = ||\mathbf{A} - \mathbf{H}\mathbf{H}^T||_F^2.$$

The second is based on the Kullback-Leibleer divergence (KL-divergence) between two matrices

$$\mathcal{L}_{KL}(\mathbf{A}, \mathbf{H}\mathbf{H}^T) \quad = \quad KL(\mathbf{A}||\mathbf{H}\mathbf{H}^T).$$

The index of the largest element in the i^{th} row of $\mathbf{H}$ indicates the community that node i belongs to.

There are many variations to the stochastic model, such as nonnegative matrix tri-factorization and stochastic block model. Nearly all of these models can be intuitively viewed as finding new representations in a low-dimensional space that can best represent and reconstruct the adjacency matrix.

2.2 Modularity Maximization Model

This model was introduced by Newman [Newman, 2006] to maximize a modularity function *Q*, which is defined as the difference between the number of edges within communities and the expected number of such edges over all pairs of vertices. For example, consider a network with two communities, then

$$Q = \frac{1}{4m}\sum_{ij}\Big(a_{ij} - \frac{k_i k_j}{2m}\Big)(h_i h_j),$$

where h_i equals to 1 (or -1) if vertex i belongs to the first (or second) group, $\frac{k_i k_j}{2m}$ is the expected number of edges between vertices i and j if edges are placed randomly, k_i is the degree of vertex i and $m = \frac{1}{2}\sum_i k_i$ is the total number of edges in the network. By defining modularity matrix $\mathbf{B} = [b_{ij}] \in \mathbb{R}^{N\times N}$ whose element is $b_{ij} = a_{ij} - \frac{k_i k_j}{2m}$, modularity Q can be written as

$$Q = \frac{1}{4m}\mathbf{h}^T\mathbf{B}\mathbf{h}, \tag{1}$$

where $\mathbf{h} = [h_i] \in \mathbb{R}^N$ is a community membership indicator vector. Maximizing Eq. (1) is NP-hard, for which many optimization algorithms have been proposed, such as extremal optimization [Duch and Arenas, 2005]. In practice, we can relax the problem by allowing variable h_i to take any real value and $\mathbf{h}^T\mathbf{h} = N$. To generalize Eq. (1) to $K > 2$ communities, we can define an indicator matrix $\mathbf{H} = [h_{ij}] \in \mathbb{R}^{N\times K}$ and obtain

$$\mathcal{L}_{MOD}(\mathbf{H}, \mathbf{B}) = Q = \mathrm{Tr}(\mathbf{H}^T\mathbf{B}\mathbf{H}),$$
$$s.t.\ \mathrm{Tr}(\mathbf{H}^T\mathbf{H}) = N,$$

where $\mathrm{Tr}(.)$ is the trace of a matrix. Based on Rayleigh Quotient, the solution to this problem is the largest K eigenvectors of the modularity matrix $\mathbf{B}$. Each row of matrix $\mathbf{H}$ can be regarded as a new representation of the corresponding vertex in the latent space, and clustering algorithms, such as *k*-means, can be used to classify the nodes into disjoint groups in the latent space.

The Eckart-Young-Mirsky Theorem [Eckart and Young, 1936] explores the relationship between the reconstruction and singular value decomposition (SVD).

Theorem 1. [Eckart and Young, 1936] *For a matrix* $\mathbf{D} \in \mathbb{R}^{m\times n}(m \geqslant n)$*, if* $\mathbf{D} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ *is the singular value decomposition of* $\mathbf{D}$*, and* $\mathbf{U}$*,* $\mathbf{V}$*,* $\mathbf{\Sigma} = diag(\sigma_1, \sigma_2, ..., \sigma_m)$ *where* $\sigma_1 \geqslant \sigma_2 \geqslant ... \geqslant \sigma_m$ *are as follows:*

$$\mathbf{U} = [\mathbf{U_1}\mathbf{U_2}], \mathbf{\Sigma} = \begin{bmatrix}\mathbf{\Sigma_1} & \mathbf{0}\\ \mathbf{0} & \mathbf{\Sigma_2}\end{bmatrix}, \mathbf{V} = [\mathbf{V_1}\mathbf{V_2}],$$

where $\mathbf{\Sigma}_1$ *is* $r \times r$*,* $\mathbf{U}_1$ *is* $m \times r$ *and* $\mathbf{V}_1$ *is* $n \times r$*, then the optimal solution to the following problem*

$$\underset{\hat{\mathbf{D}}\in\mathbb{R}^{m\times n}\ \mathrm{rank}(\hat{\mathbf{D}})\leqslant r}{\mathrm{argmin}} \quad ||\mathbf{D} - \hat{\mathbf{D}}||_F$$

is $\hat{\mathbf{D}}^* = \mathbf{U_1}\mathbf{\Sigma_1}\mathbf{V_1}^T$ *and*

$$||\mathbf{D} - \hat{\mathbf{D}}^*||_F = \sqrt{\sigma_{r+1}^2 + ... + \sigma_m^2}.$$

The above theorem means that the matrix reconstruction from the singular vectors corresponding to the K largest singular values is the best rank-K approximation to the original matrix under the Frobenuis norm. Since modularity matrix $\mathbf{B}$ is symmetric, there exists orthogonal decomposition $\mathbf{B} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, where $\mathbf{\Lambda}$ is diagonal matrix with the eigenvalues of $\mathbf{B}$ as the diagonal elements and $\mathbf{U^T U} = \mathbf{I}$. Thus, the matrix reconstructed by the eigenvectors corresponding to the K largest eigenvalues is also the best approximation to the input matrix $\mathbf{B}$ with rank K. Therefore, the modularity maximization problem can be regarded as reconstructing the modularity matrix using a low-rank approximation.

Put together, stochastic model and modularity maximization can be intuitively interpreted as to find low-dimensional representations to best reconstruct given network structures. It is important to note that both of them only reconstruct original networks by linear reconstruction, using, e.g., NMF or SVD, and ignore nonlinear properties of the networks. It is unclear how NMF and SVD based approaches can be extended to accommodate nonlinear low-dimensional embedding. We aim to overcome this difficulty by using deep neural networks, the focus of the current paper.

3 Deep Nonlinear Reconstruction Model

We now present a novel deep nonlinear reconstruction (DNR) model for community detection. We first introduce an Auto-Encoder, which is a key building block of the model, and then describe a stacked Auto-Encoder. While in the following discussion we focus on finding a nonlinear embedding that best reconstructs the modularity matrix $\mathbf{B}$, as inspired by the SVD-based modularity maximization (Section 2.2), our method can be readily applied to other network input forms, such as adjacency and Laplacian matrices.

3.1 Reconstruction based on Auto-Encoder

Auto-Encoder is a special neural network that is used to learn a new representation that can best approximate the original data [Bourlard and Kamp, 1988; Hinton and Zemel, 1994]. We adopt modularity matrix $\mathbf{B} = [b_{ij}] \in \mathbb{R}^{N\times N}$ as the input to the Auto-Encoder. Here, the elements of $\mathbf{B}$ are $b_{ij} = a_{ij} - \frac{k_i k_j}{2m}$, and the i^{th} column $\mathbf{b}_i$ of $\mathbf{B}$ represents vertex i. The Auto-Encoder consists of two key components: encoder and decoder. The encoder maps the original data $\mathbf{B}$ to a low-dimensional embedding $\mathbf{H} = [h_{ij}] \in \mathbb{R}^{d\times N}$ where $d < N$ and the i^{th} column of $\mathbf{H}$, i.e., $\mathbf{h}_i$, represents vertex i in the latent space

$$\mathbf{h_i} = f(\mathbf{b_i}) = s(\mathbf{W}_H\mathbf{b_i} + \mathbf{d}_H), \tag{2}$$

where $\mathbf{W}_H \in \mathbb{R}^{d\times N}, \mathbf{d}_H \in \mathbb{R}^{d\times 1}$ are the parameters to be learned in the encoder, and $s(.)$ is an element-wise nonlinear mapping, such as sigmoid function $s_{sigmoid}(x) = \frac{1}{1+e^{-x}}$ or tanh function $s_{tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$. The decoder maps the latent representation $\mathbf{H}$ back into the original data space, i.e., reconstructs the original data from the latent representation:

$$\mathbf{m_i} = g(\mathbf{h_i}) = s(\mathbf{W}_M\mathbf{h_i} + \mathbf{d}_M),$$

where $\mathbf{W}_M \in \mathbb{R}^{N\times d}, \mathbf{d}_M \in \mathbb{R}^{N\times 1}$ are the parameters to be learned in the decoder and $g(.)$ is another element-wise nonlinear mapping similar to $s(.)$. Auto-Encoder aims at learning a low-dimensional nonlinear representation $\mathbf{H}$ that can best reconstruct the original data $\mathbf{B}$, i.e. minimize the difference between the original data $\mathbf{B}$ and reconstruction data $\mathbf{M}$ under parameters $\theta = \{\mathbf{W}_H, \mathbf{d}_H, \mathbf{W}_M, \mathbf{d}_M\}$

$$\begin{aligned}\hat{\theta} &= \operatorname*{argmin}_\theta L_\theta(\mathbf{B}, \mathbf{M}) = \operatorname*{argmin}_\theta \sum_{i=1}^N L_\theta(\mathbf{b_i}, \mathbf{m_i}) \\ &= \operatorname*{argmin}_\theta \sum_{i=1}^N L_\theta(\mathbf{b_i}, g(f(\mathbf{b_i}))),\end{aligned} \tag{3}$$

where $L_\theta(\mathbf{b_i}, \mathbf{m_i})$ is a distance function that measures the reconstruction error. Here we adopt the Euclidean distance and sigmoid cross-entropy distance as distance functions. The sigmoid cross-entropy distance maps elements in $\mathbf{b_i} = [b_{ji}] \in \mathbb{R}^{N\times 1}$ and $\mathbf{m_i} = [m_{ji}] \in \mathbb{R}^{N\times 1}$ to $[0, 1]$ using sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$, and then computes the cross-entropy of them as

$$\sum_{j=1}^N \big(\sigma(m_{ji})\log(\sigma(b_{ji})) + (1 - \sigma(m_{ji}))\log(1 - \sigma(b_{ji}))\big).$$

After training the Auto-Encoder, $\mathbf{W}_H$ and $\mathbf{d}_H$ are obtained and Eq. (2) can be used to generate the new representations for all vertices.

3.2 Optimization

Eq. (3) can be solved by back-propagation with stochastic gradient descent. In each iteration, the parameters $\theta = \{\mathbf{W}_H, \mathbf{d}_H, \mathbf{W}_M, \mathbf{d}_M\}$ are updated as follows

$$W_\alpha^{ji} = W_\alpha^{ji} - \gamma\frac{\partial}{\partial W_\alpha^{ji}}L_\theta(\mathbf{B}, \mathbf{M}),$$
$$d_\alpha^j = d_\alpha^j - \gamma\frac{\partial}{\partial d_\alpha^j}L_\theta(\mathbf{B}, \mathbf{M}),$$

where $\alpha \in \{\mathbf{H}, \mathbf{M}\}$. By defining $\mathbf{z}_\alpha = \mathbf{W}_\alpha\mathbf{x} + \mathbf{d}_\alpha$, we have

$$\begin{aligned}\frac{\partial}{\partial W_\alpha^{ji}}L_\theta(\mathbf{X}, g(f(\mathbf{X}))) &= \sum_{i=1}^N \frac{\partial}{\partial W_\alpha^{ji}}L_\theta(\mathbf{x_i}, g(f(\mathbf{x_i}))) \\ = \sum_{i=1}^N \frac{\partial}{\partial z_\alpha^j}L_\theta(\mathbf{x_i}, g(f(\mathbf{x_i})))\frac{\partial}{\partial W_\alpha^{ji}}z_\alpha^j &= \sum_{i=1}^N \delta_\alpha^j\mathbf{x}_i^T,\end{aligned}$$

$$\begin{aligned}\frac{\partial}{\partial d_\alpha^j}L_\theta(\mathbf{X}, g(f(\mathbf{X}))) &= \sum_{i=1}^N \frac{\partial}{\partial d_\alpha^{ji}}L_\theta(\mathbf{x_i}, g(f(\mathbf{x_i}))) \\ = \sum_{i=1}^N \frac{\partial}{\partial z_\alpha^j}L_\theta(\mathbf{x_i}, g(f(\mathbf{x_i})))\frac{\partial}{\partial d_\alpha^j}z_\alpha^j &= \sum_{i=1}^N \delta_\alpha^j,\end{aligned}$$

where $\delta_\alpha^j = \frac{\partial}{\partial z_\alpha^j} L_\theta(\mathbf{x_i}, g(f(\mathbf{x_i})))$ denotes the contribution of a node to the overall reconstruction error. For Euclidean distance based $L_\theta(\mathbf{B}, \mathbf{M})$,

$$\delta_M^j = -\sum_{i=1}^{N}(b_{ij} - m_{ij})s'(z_M^j), \delta_H^j = (\sum_{i=1}^{N} W_H^{ji}\delta_M^i)s'(z_H^j),$$

where $s'(x)$ is the derivative of $s(x)$.

3.3 Stacked Auto-Encoder

Recently, deep learning has been successful on various problems in many fields, such as image classification, semantic segmentation [Liu *et al.*, 2015; Liang *et al.*, 2015]. However, as the number of layers increases, the space of parameters grows exponentially, making optimization inefficient. A compromising strategy is to train the network layer by layer [Vincent *et al.*, 2010].

To take advantage of a deep architecture, we stack a series of Auto-Encoders to form a DNR model. For a deep Auto-Encoder network, we train the first Auto-Encoder by reconstructing the original data, i.e., the modularity matrix $\mathbf{B}$ and obtain a new representation $\mathbf{H}^1 \in \mathbb{R}^{N\times t_1}$. We then train the i^{th} Auto-Encoder by reconstructing the output of the $(i-1)^{th}$ Auto-Encoder and obtain a representation $\mathbf{H}^i \in \mathbb{R}^{N\times t_i}$, where $t_i < t_{i-1}$. The number of Auto-Encoders we stack and the dimensions of new representations, i.e., t_is', are discussed in Section 5.2.

4 Pairwise Constrained Semi-supervised Community Detection

We now incorporate pairwise constraints on vertices into the proposed DNR model and introduce a novel reconstruction space graph regularization for semi-supervised DNR (semi-DNR). If we have *a priori* knowledge that vertices i and j belong to the same community, we can make use of this knowledge in two different ways. First, to classify two nodes into the same community, the new representations of nodes i and j, i.e., $\mathbf{h_i}$ and $\mathbf{h_j}$, should be similar, since the new representations are used to cluster the two vertices after encoding. Second, this *a priori* information should also be encoded into the DNR model to further affect the embedding of other vertices. Therefore, instead of just modifying the embedding representation, we incorporate the pairwise constraints into the loss function of the Auto-Encoder in Eq. (3).

To measure the similarity of their latent representations, we can adopt either Euclidean distance or Kullback-Leibleer divergence (KL-divergence), i.e.

$$\mathcal{D}_{LSE}(\mathbf{h}_i, \mathbf{h}_j) = ||\mathbf{h}_i - \mathbf{h}_j||_2^2 = \sum_{k=1}^{K}(h_{ik} - h_{jk})^2,$$

$$\mathcal{D}_{KL}(\mathbf{h}_i||\mathbf{h}_j) = \sum_{k=1}^{K}\Big(h_{ik}\log\Big(\frac{h_{ik}}{h_{jk}}\Big) - h_{ik} + h_{jk}\Big).$$

If nodes i and j are known to belong to the same community, we then try to minimize the difference between their new representations $\mathcal{D}_{LSE}(\mathbf{h}_i, \mathbf{h}_j)$. We define a pairwise constraint matrix $\mathbf{O} = [o_{ij}] \in \mathbb{R}_+^{N\times N}$, where $o_{ij} = 1$ if nodes i and j are known to be in the same community, or $o_{ij} = 0$ otherwise. Thus, we can write the pairwise constraints as

$$\mathcal{R}_{LSE}(\mathbf{O}, \mathbf{H}) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} o_{ij}||\mathbf{h}_i - \mathbf{h}_j||_2^2$$
$$= \mathrm{Tr}(\mathbf{H}^T\mathbf{D}\mathbf{H}) - \mathrm{Tr}(\mathbf{H}^T\mathbf{O}\mathbf{H}) = \mathrm{Tr}(\mathbf{H}^T\mathbf{L}\mathbf{H}),$$

where $\mathrm{Tr}(.)$ is the trace of a matrix, $\mathbf{D} = [d_{ij}] \in \mathbb{R}_+^{N\times N}$ a diagonal matrix whose entries are row summation of $\mathbf{O}$, i.e., $d_{ii} = \sum_{j=1}^{N} o_{ij}$, and $\mathbf{L} = \mathbf{D} - \mathbf{O}$ the graph regularization matrix (Laplacian matrix) of a priori information $\mathbf{O}$. Similarly, the KL-divergence based constraints can be written as:

$$\mathcal{R}_{KL}(\mathbf{O}, \mathbf{H}) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} o_{ij}\Big(\mathcal{D}_{KL}(\mathbf{h}_i||\mathbf{h}_j) + \mathcal{D}_{KL}(\mathbf{h}_j||\mathbf{h}_i)\Big),$$

which takes into account the asymmetry of KL-divergence and averages $\mathcal{D}_{KL}(\mathbf{h}_i||\mathbf{h}_j)$ and $\mathcal{D}_{KL}(\mathbf{h}_j||\mathbf{h}_i)$. By minimizing $\mathcal{R}_{LSE}(\mathbf{O}, \mathbf{H})$ or $\mathcal{R}_{KL}(\mathbf{O}, \mathbf{H})$, we expect the new representations of two nodes i and j are similar if we have some information indicating that these two nodes belong to the same community, i.e., the corresponding element $o_{ij} = 1$.

By incorporating the pairwise constraints in Eq (4) with the reconstruction error function in Eq (3), we obtain the overall loss function for semi-supervised DNR (semi-DNR) as

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}}\, L_\theta(\mathbf{B}, \mathbf{M}) + \lambda\,\mathrm{Tr}(\mathbf{H}^T\mathbf{L}\mathbf{H}), \tag{4}$$

where λ is a parameter for making a tradeoff between the reconstruction error (the first term $L(\mathbf{B}, \mathbf{M})$) and the consistency of the new representations with *a priori* information (the second term $\mathrm{Tr}(\mathbf{H}^T\mathbf{L}\mathbf{H})$). For semi-DNR, we can impose a graph regularization term on the reconstruction layers of all Auto-Encoders, and then evaluate the performance improvement as discussed in Section 5.3.

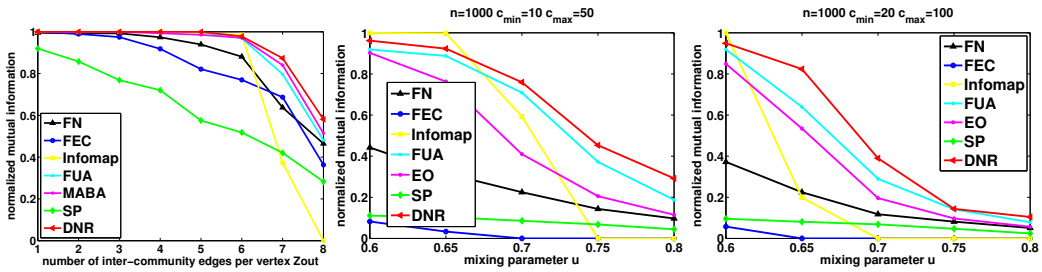
Eq. (4) can also be solved using back-propagation. Since the reconstruction space graph regularization term $\mathrm{Tr}(\mathbf{H}^T\mathbf{L}\mathbf{H})$ is independent of reconstruction of $\mathbf{M}$, it does not affect the update of $\mathbf{W}_M$ and $\mathbf{b}_M$. Therefore, we only need to modify δ_H^i as $\delta_{H(semi)}^i = \delta_H^i + \delta_{graph}^i$, where

$$\delta_{graph\_lse} = \mathbf{H}(\mathbf{L} + \mathbf{L}^T) \odot s'(H),$$
$$\delta_{graph\_kl} = \mathbf{H}(\mathbf{L} + \mathbf{L}^T) \div H \odot s'(H),$$

denote how much a node is responsible for the inconsistency of the derived representation and the pairwise constraints based on the Euclidean distance and KL-divergence, respectively. Here $\odot$ and $\div$ denote the element-wise multiplication and division, respectively.

5 Experimental Analysis

To evaluate the proposed DNR and semi-DNR methods, we analyzed their performance on widely used synthetic benchmarks and real-world networks. We first compared DNR with seven state-of-the-art approaches, which are divided into two categories based on whether they are based on modularity maximization. The modularity based methods include the spectral (SP) algorithm [Newman, 2006], the external optimization



(a) GN network (b) LFR network (small community) (c) LFR network (large community)

Figure 1: Comparison of DNR with six competing methods on GN and LFR networks.

Table 1: Performance on Real-world Networks (the best performance is in bold and the second best performance is in italics)

Datasets	***N***	***M***	***K***	**SP**	**EO**	**FN**	**FUA**	**DNR_L2**	**DNR_CE**
Karate [Zachary, 1977]	34	78	2	**1.000**	0.587	0.692	0.587	**1.000**	**1.000**
Dolphins [Lusseau and Newman, 2004]	62	159	2	0.753	0.579	0.572	0.516	**0.889**	*0.818*
Friendship6 [Xie *et al.*, 2013]	68	220	6	0.418	**0.952**	0.727	0.852	0.888	*0.924*
Friendship7 [Xie *et al.*, 2013]	68	220	7	0.477	*0.910*	0.762	0.878	0.907	**0.932**
Football [Girvan and Newman, 2002]	115	613	12	0.334	0.885	0.698	0.890	**0.927**	*0.914*
Polbooks [Newman, 2006]	105	441	3	*0.561*	0.557	0.531	0.574	0.552	**0.582**
Polblogs [Adamic and Glance, 2005]	1,490	16,718	2	*0.511*	0.501	0.499	0.375	0.389	**0.517**
Cora [Yang *et al.*, 2009]	2,708	5,429	7	0.295	0.441	*0.459*	0.260	**0.463**	0.421

(EO) algorithm [Duch and Arenas, 2005], the FUA algorithm [Blondel *et al.*, 2008], the MABA algorithm [He *et al.*, 2012] and the (FN) algorithm [Newman, 2004]. The remaining methods include the (FEC) algorithm [Yang *et al.*, 2007] and the Infomap algorithm [Rosvall and Bergstrom, 2008].

To assess the effectiveness of using pairwise constraints, we compared semi-DNR with two existing semi-supervised community detection algorithms, ModLink [Zhang, 2013] and GraphNMF [Yang *et al.*, 2015]. The former transforms pairwise constraints into information of network topology, modifies the adjacency matrix and uses a conventional algorithm to find communities on new label-refined networks. The latter incorporates the labels by constraining the nodes belonging to the same community to have similar membership representations. We adopted the Normalized Mutual Information (NMI) for performance measure.

5.1 Experiment Setup

Table 2: Deep Nonlinear Reconstruction Network Setting

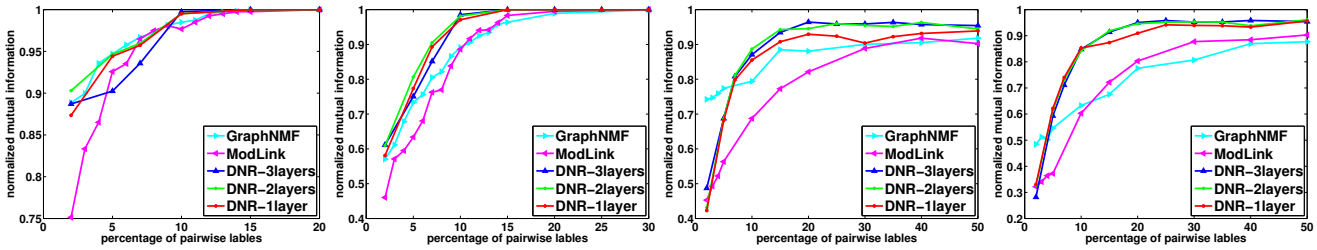
Datasets	***N***	**Layers Configuration**
Karate	34	34-32-16
Dolphins	62	62-32-16
Friendship6	68	68-32-16
Friendship7	68	68-32-16
Football	115	115-64-32-16
Polbooks	105	105-64-32-16
Polblogs	1,490	1,490-256-128-64
Cora	2,708	2,708-512-256-128
GN Network	128	126-64-32-16
LFR Network	1,000	1,000-512-256-128

The layer configurations of the deep neural networks for different problems tested are shown in Table 2. The networks have at most 3 stacked Auto-Encoder, and the dimension of each latent space is less than that of its input and output spaces. For example, the stacked Auto-Encoder network for the LFR network consists of three Auto-Encoders, where the first is 1,000-512-1,000, the second 512-256-512 and the third 256-128-256. All Auto-Encoders were trained separately. We took a modularity matrix as the input to the first Auto-Encoder, and trained the it to minimize the reconstruction error, then took the embedding result as the input to the second Auto-Encoder, and so on. We set the training batch to the size of the network and ran at most 100,000 iterations. For each network we trained a DNR model with 10 random initializations, and took the latent representations from three Auto-Encoders for clustering. Here, we adopted the k-means for clustering and returned the results with the maximum modularity.

5.2 Community Detection Results

We considered two types of synthetic networks, Girvan-Newman (GN) networks [Girvan and Newman, 2002] and Lancichinetti-Fortunato-Radicchi (LFR) networks [Lancichinetti *et al.*, 2008]. Each GN network consists of 128 nodes divided into 4 communities of 32 nodes each. Each node has on average 16 edges, among which Z_{out} edges are inter-community edges. The results are shown in Figure 1(a). As shown, DNR outperforms all competing methods, especially when $Z_{out} > 6$.

The LFR networks are more complicated than the GN networks. As suggested in [Lancichinetti *et al.*, 2008], we set the number of nodes to 1000, the average degree to 20, the exponent of a vertex degree and the community size to -2



(a) GN newtwork with Zout=7 (b) GN newtwork with Zout=8 (c) LFR network with $\mu = 0.75$ (d) LFR network with $\mu = 0.8$

Figure 2: Comparison of semi-DNR and two competing methods on GN and LFR networks.

and -1 respectively, and varied the mixing parameter μ from 0.6 to 0.8. To fully evaluate the performance on networks with different community sizes, we generated two groups of networks: one with community sizes from 10 to 50 while the other from 20 to 100. The results on these two groups of networks are shown in Figs 1(b) and 1(c). As shown, DNR can successfully detect small and large communities. It achieves the best performance when $\mu > 0.65$ on networks with small community size (middle figure, Figure 1) and $\mu > 0.6$ on networks with large community sizes (right figure, Figure 1). While Infomap performs slightly better than DNR when is small, it completely fails when $\mu \geqslant 0.75$.

In summary, the experimental results on synthetic networks showed that the deep nonlinear model is more effective on difficult networks with vague community structures and is competitive on easy networks with clear community structures. While the factors for such a superb performance compared with other modularity-based methods remains to be further investigated, it may be partially attributed to the nonlinear structure of our new model, which helps to mitigate the resolution limit [Fortunato and Barthelemy, 2007] and the extreme degeneracy [Good *et al.*, 2010] problems, both of which are often suffered from modularity optimization.

Nine widely-used real networks, listed in Table 1, were used for evaluation. As shown in previous work [Psorakis *et al.*, 2011], no single unified loss function seemed to exist that can successfully detect communities in all networks. In our experiments, we also adopted the Euclidean distance and sigmoid cross-entropy as error functions. We used sigmoid cross-entropy instead of KL-divergence because the former can be integrated with the sigmoid function for back-propagation.

The results are shown in Table 1. Here, we compared DNR with other well-known modularity-based community detection methods. As shown in the table, DNR with the L2 norm (DNR_L2) and with the cross-entropy distance (DNR_CE) outperforms most of the competing modularity-based optimization algorithms.

5.3 Semi-supervised Community Detection Results

To evaluate the new semi-supervised deep nonlinear reconstruction (semi-DNR) method, we used networks on which it is difficult to find satisfactory community structures without label information. As shown in Figure 2, the performance is mediocre on GN network with $Z_{out} = 8$ and LFR network with $\mu = 0.75$ and 0.8. Besides, we chose GN network with $Z_{out} = 7$ where the methods without labels can also achieve better results for comparison. Here, we set the balancing parameter $\lambda = 1000$. We also verifed the effects of pairwise constraints on more than one layer i.e., only the bottom layer, both bottom and middle layers and all the three layers.

The results are shown in Figure 2. On the GN with $Z_{out} = 7$, all the methods have similar improvements by enforcing the same percent of labels. On the networks where unsupervised methods cannot obtain satisfactory results, in comparison, the semi-DNR achieves much better performance with the same number of labels. For example, with 20% pairwise constraints, the NMI of semi-DNR achieves 0.95 while that of GraphNMF and ModLink only achieve 0.76 and 0.79, respectively, on LFR network with $\mu = 0.8$ (right figure in Figure 2). This means the semi-DNR is much more efficient on the use of pairwise constraints than other methods. Furthermore, the performance of enforcing pairwise constraints on multi-layers has similar improvements. It illustrates that semi-DNR can fully explore pairwise constraints by only one layer graph regularization.

6 Conclusion

In order to overcome the serious drawback of linear low-rank embedding used by the widely adopted stochastic model and modularity maximization model for network community identification, we proposed a nonlinear model in deep neural networks to gain representation power for large complex networks; developed an algorithm using the model for network community detection; and further extended this method to a semi-supervised deep nonlinear reconstruction algorithm by incorporating pairwise constraints. Extensive experimental results on synthetic and real networks illustrate that our new methods outperform the existing state-of-the-art methods for network community identification. In the future, we plan to study model selection using the latent space embedding from DNR to make the model and methods more robust.

7 Acknowledgments

This work was supported by National Natural Science Foundation of China (No. 61503281, 61502334, 61303110, 61422213), "Strategic Priority Research Program" of the Chinese Academy of Sciences (XDA06010701), Open Funding Project of Tianjin Key Laboratory of Cognitive Computing and Application, Foundation for the Young Scholars by Tianjin University of Commerce (150113), the Talent Development Program of Wuhan, the municipal government of Wuhan, Hubei, China (2014070504020241), and an internal research

grant of Jianghan University, Wuhan, China, as well as by United States National Institutes of Health (R01GM100364).

References

[Adamic and Glance, 2005] Lada A Adamic and Natalie Glance. The political blogosphere and the 2004 us election: divided they blog. In *Proceedings of the 3rd international workshop on Link discovery*, pages 36–43. ACM, 2005.

[Blondel *et al.*, 2008] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[Bourlard and Kamp, 1988] Hervé Bourlard and Yves Kamp. Auto-association by multilayer perceptrons and singular value decomposition. *Biological cybernetics*, 59(4-5):291–294, 1988.

[Duch and Arenas, 2005] Jordi Duch and Alex Arenas. Community detection in complex networks using extremal optimization. *Phys. Rev. E*, 72:027104, Aug 2005.

[Eckart and Young, 1936] Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

[Fortunato and Barthelemy, 2007] Santo Fortunato and Marc Barthelemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104(1):36–41, 2007.

[Girvan and Newman, 2002] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[Good *et al.*, 2010] Benjamin H Good, Yves-Alexandre de Montjoye, and Aaron Clauset. Performance of modularity maximization in practical contexts. *Physical Review E*, 81(4):046106, 2010.

[He *et al.*, 2012] Dongxiao He, Jie Liu, Bo Yang, Yuxiao Huang, Dayou Liu, and Di Jin. An ant-based algorithm with local optimization for community detection in large-scale networks. *Advances in Complex Systems*, 15(08):1250036, 2012.

[He *et al.*, 2015] Dongxiao He, Dayou Liu, Di Jin, and Weixiong Zhang. A stochastic model for detecting heterogeneous link communities in complex networks. In *AAAI Conference on Artificial Intelligence*, 2015.

[Hinton and Zemel, 1994] Geoffrey E Hinton and Richard S Zemel. Autoencoders, minimum description length, and helmholtz free energy. *Advances in neural information processing systems*, pages 3–3, 1994.

[Jin *et al.*, 2015] Di Jin, Zheng Chen, Dongxiao He, and Weixiong Zhang. Modeling with node degree preservation can accurately find communities. In *AAAI Conference on Artificial Intelligence*, 2015.

[Lancichinetti *et al.*, 2008] Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. Benchmark graphs for testing community detection algorithms. *Physical Review E*, 78(4):046110, 2008.

[Liang *et al.*, 2015] Xiaodan Liang, Si Liu, Xiaohui Shen, Jianchao Yang, Luoqi Liu, Jian Dong, Liang Lin, and Shuicheng Yan. Deep human parsing with active template regression. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(12):2402–2414, 2015.

[Liu *et al.*, 2015] S. Liu, X. Liang, L. Liu, X. Shen, J. Yang, C. Xu, L. Lin, Xiaochun Cao, and S. Yan. Matching-cnn meets knn: Quasi-parametric human parsing. In *CVPR 2015*, pages 1419–1427, June 2015.

[Lusseau and Newman, 2004] David Lusseau and Mark EJ Newman. Identifying the role that animals play in their social networks. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 271(Suppl 6):S477–S481, 2004.

[Newman, 2004] Mark EJ Newman. Fast algorithm for detecting community structure in networks. *Physical Review E*, 69(6):066133, 2004.

[Newman, 2006] Mark EJ Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006.

[Psorakis *et al.*, 2011] Ioannis Psorakis, Stephen Roberts, Mark Ebden, and Ben Sheldon. Overlapping community detection using bayesian non-negative matrix factorization. *Physical Review E*, 83(6):066114, 2011.

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

[Vincent *et al.*, 2010] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencodernmfs: Learning useful representations in a deep network with a local denoising criterion. *The Journal of Machine Learning Research*, 11:3371–3408, 2010.

[Wang *et al.*, 2008] Rui-Sheng Wang, Shihua Zhang, Yong Wang, Xiang-Sun Zhang, and Luonan Chen. Clustering complex networks and biological networks by nonnegative matrix factorization with various similarity measures. *Neurocomputing*, 72(1):134–141, 2008.

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

[Yang *et al.*, 2007] Bo Yang, William K Cheung, and Jiming Liu. Community mining from signed social networks. *Knowledge and Data Engineering, IEEE Transactions on*, 19(10):1333–1348, 2007.

[Yang *et al.*, 2009] Tianbao Yang, Rong Jin, Yun Chi, and Shenghuo Zhu. Combining link and content for community detection: A discriminative approach. In *Proceedings of the 15th ACM SIGKDD '09*, pages 927–936, 2009.

[Yang *et al.*, 2015] Liang Yang, Xiaochun Cao, Di Jin, Xiao Wang, and Dan Meng. A unified semi-supervised community detection framework using latent space graph regularization. *Cybernetics, IEEE Transactions on*, 45(11):2585–2598, Nov 2015.

[Zachary, 1977] W Zachary. An information flow modelfor conflict and fission in small groups. *Journal of anthropological research*, 33(4):452–473, 1977.

[Zhang *et al.*, 2007] Shihua Zhang, Rui-Sheng Wang, and Xiang-Sun Zhang. Uncovering fuzzy community structure in complex networks. *Physical Review E*, 76(4):046103, 2007.

[Zhang, 2013] Zhong-Yuan Zhang. Community structure detection in complex networks with partial background information. *EPL (Europhysics Letters)*, 101(4):48005, 2013.