Extending a Configuration Model to Find Communities in Complex Networks

Di Jin[1,2,4], Dongxiao He[3,4], Qinghua Hu[1,2], Carlos Baquero[5]
[1] School of Computer Science and Technology, Tianjin University, Tianjin 300072, China
[2] Tianjin Key Laboratory of Cognitive Computing and Application, Tianjin University, Tianjin 300072, China
[3] College of Computer Science and Technology, Jilin University, Changchun 130012, China
[4] Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun, 130012, China
[5] HASLab, INESC TEC & University of Minho, Braga, Portugal

E-mail: jindi@tju.edu.cn, hedongxiaojlu@gmail.com,
huqinghua@tju.edu.cn, and cbm@di.uminho.pt

Abstract. Discovery of communities in complex networks is a fundamental data analysis task in various domains. Generative models are a promising class of techniques for identifying modular properties from networks, which has been actively discussed recently. However, most of them cannot preserve the degree sequence of networks, which will distort the community detection results. Rather than using a blockmodel as most current works do, here we generalize a configuration model, namely, a null model of modularity, to solve this problem. Towards decomposing and combining sub-graphs according to the soft community memberships, our model incorporates the ability of describing community structures, something the original model does not have. Also, it holds the property, as with the original model, that fixes the expected degree sequence to be the same as that of the observed network. We combine both the community property and degree sequence preserving into a single unified model, which gets better community results compared with other models. Thereafter, we learn the model using a technique of nonnegative matrix factorization and determine the number of communities by applying consensus clustering. We test this approach both on synthetic benchmarks and on real-world networks, and compare it with two similar methods. The experimental results demonstrate the superior performance of our method over competing methods in detecting both disjoint and overlapping communities.

Keywords: analysis of algorithms, network dynamics, random graphs, networks, clustering techniques

PACS: 89.75.Hc, 89.75.Fb, 02.10.Ox, 02.60.Cb

1. INTRODUCTION

Many networked systems, including social networks, information networks and biological networks, are found to divide naturally into modules or communities, i.e. groups of vertices with relatively dense connections within groups but sparser connections between them [1]. Depending on the source context, the groups may be disjoint or overlapping. A fundamental problem in the theory of networks, and one that has attracted substantial interest among researchers in the last decade is how to detect such communities in empirical network data [2]. There are many methods that have been proposed for the task of community detection, such as [1, 3, 4] for disjoint communities and some others [5-7] for overlapping communities. For review, the readers can refer to Ref. [2].

In particular, due to its good performance and sound theoretical principles, generative model methods constitutes a promising class of techniques for identifying modular properties from networks, and thus are actively being researched and developed [8]. Recently, several model-based methods have been proposed [9-16]. Most of them are based on the stochastic blockmodel [17] that is a popular tool for detecting community structures in networks. In the simplest stochastic blockmodel, each of n vertices in the network is assigned to one of c blocks, groups, or communities, and undirected edges are placed independently between vertex pairs with probabilities that are a function only of the group memberships of the vertices. If we denote by g_i the group to which vertex i belongs, then we can define a $c \times c$ matrix ψ of probabilities such that the matrix element ψ_{gigj} is the probability of an edge occurring between vertices i and j. While simple to describe, however, this blockmodel ignores variations in vertex degrees, making it unsuitable for application in real-world networks, that typically display a broad range of degree distributions that can significantly distort the community results [15]. Most of the current work that is based on the blockmodel, take some steps to mitigate the inherent, above described, drawback but these steps are only effective to some extent. These strategies cannot solve this problem at a fundamental level, as they are unable to theoretically fix the expected degree sequence to be the same as that of the observed network. Preserving the degree sequence of networks is, however, especially important for the task of community detection in the opinion of [15], which may improve the community detection results.

In this work, we try to solve this problem by taking a somewhat different stance. Instead of using blockmodel as most current works did, here we generalize a configuration model, namely the null model of modularity [18, 19], to incorporate communities. With the idea of decomposing and combining the color-z sub-graphs from the perspective of soft clustering, this extended model captures a new ability of describing communities that is lacking in the original null model; meanwhile, it maintains the property, preserved from the original model, that fixes the expected degree sequence to be the same as that of the observed network. Under this model, when we describe graphs, besides community structure, we also consider the fact that vertices with high degree are, all other things being equal, more likely to be connected than those with low degree, simply because they have more edges. Intuitively, as we combine both these two properties into one model, we can get a better community result compared with the blockmodels that tend to display broad degree distributions. Therefore, in order to implement learning in this model, we define a fitness function of squared loss, and solve it by using nonnegative matrix factorization. Finally, we introduce a method of model selection to determine the number of communities when it is not a prior input parameter.

Furthermore, as our model provides a soft community membership, it can find both disjoint and overlapping communities in networks. Also, because we adopt the squared loss as an objective function and use nonnegative matrix factorization as the optimization technique, that makes our method suitable for weighted networks.

This paper is organized as follows. We first review the related work in Section 2. Then in Section 3, we introduce our method, namely ENMM, meaning 'Extending Null Model of Modularity', which includes four parts: the generative model, model properties, parameter learning, and model selection. We report on experiments in Section 4. Finally, we present the conclusions and discussion in Section 5.

2. RELATED WORK

Recently, some model-based methods have been proposed, most of which are based on the stochastic blockmodel or its variations, and employed different types of optimization algorithms to learn models.

For instance, Ref. [9-12] all extend the basic blockmodel from the perspective of soft membership, and take nonnegative matrix factorization (NMF) as the optimization method to learn the parameters. To be specific, Wang et al. [9] used the squared loss, and introduced an algorithm of symmetric nonnegative matrix factorization (SNMF) to minimize their loss function. Psorakis et al. [10, 11] adopted the generalized KL-divergence as the loss function, and proposed an algorithm of Bayesian nonnegative matrix factorization (BNMF) as the optimization method. Also, they used priors that penalized their model for including too many nonzero parameter values and hence created a balance between numbers of communities and goodness of fitting to the network data. Zhang et al. [12] removed the normalized constraint that the sum of probabilities for each vertex belonging to different communities equals to 1, to better model overlapping communities. Further, they used both the squared loss and generalized KL-divergence as the loss functions, and designed a method called bounded nonnegative matrix tri-factorization (BNMTF) to solve them. Although these methods are all based on NMF (like ours), they did not consider the problem of degree sequence preserving, which may distort the community detection results.

Moreover, there are some other works [13-16] that have adopted similar models. But rather than using loss function, they adopted the likelihood probability as the goal, and take a different algorithmic approach such as the Expectation-Maximization (EM) algorithm to learn their models. Of particular note is Newman's degree-corrected stochastic blockmodel [15]. They first studied the question of why degree heterogeneity in blockmodels is a good idea, and this led them to a so-called degree-corrected blockmodel and a heuristic algorithm for community detection by inferring model parameters. To the best of our knowledge, this is the only work having the same ability, as with our model, that theoretically preserves the degree sequence of networks. However, these two works have some key differences.

Firstly, Newman's model is an extension of blockmodel to correct degree sequence, while our model is a generalization of the null model of modularity to incorporate communities. Thus, they have different mechanisms to describe the community structure of networks. Furthermore, Newman's model is based on the assumption of hard clustering which can only detect disjoint communities, but our model is a more flexible one, which is under the assumption of soft membership and has the ability of finding both disjoint and overlapping communities in networks. Last but not least, our model can deal with weighted networks, while Newman's model does not have this ability.

3. THE METHOD

In this section, we first generalize a configuration model, namely null model of modularity [18, 19], to incorporate communities; and then describe some interesting properties of it; thereafter, we present an approach based on nonnegative matrix factorization, to learn the parameters; and finally, we introduce a model selection method to determine the best number of communities.

3.1 Generative Model

Consider an undirected graph $G = (V, E)$ with adjacency matrix A, having a given number n of vertices divided among a given number c of communities. Assume that its community structure is given by S, where S_{iz} denotes the fraction by which vertex i has a community with index (or say color) z, subject to $\sum_z S_{iz} = 1$. This characterizes a *soft* community membership, since a vertex can belong to more than one community.

According to S, we can decompose the given network G into c color-z sub-graphs G_z's with z=1,2,…,c. Each G_z has a node set V, and takes $d_{iz} = d_iS_{iz}$ as the node degree of each $i \in V$, where d_i denotes node i's degree in G. Then, G_z can be regarded as a soft community of G, which is a completely random graph without community structure. The null model of modularity describes random graphs with the given degree sequence and with edges rewired at random among the vertices, having no communities [18, 19]. Thus, we take the option here to describe each color-z sub-graph G_z, with the given degree sequence $\{d_{1z}, d_{2z},\dots,d_{nz}\}$. Following this null model, the expected number of links (or expected link weight) between nodes i and j in G_z can be evaluated as

$$\hat{A}_{ij}^{z} = \frac{d_{iz}d_{jz}}{\sum_k d_{kz}}. \tag{1}$$

If we combine all the color-z sub-graphs into an ensemble one, then the expected number of links between nodes i and j in the original network G can be written as

$$\hat{\hat{A}}_{ij} = \sum_z A_{ij}^{z} = \sum_z \frac{d_{iz}d_{jz}}{\sum_k d_{kz}}. \tag{2}$$

Actually, we do not know the soft community membership S in advance; on the contrary, S should be inferred to solve the problem of community detection. Under model, above, with model parameters d_{iz}, we apply it under the following optimization problem:

$$\min_{d_{iz}\geq 0} L_{sq}\left(A, \hat{A}\right) = \left\|A - A\right\|_F^2 = \sum_{ij}\left(A_{ij} - \sum_z \frac{d_{iz}d_{jz}}{\sum_k d_{kz}}\right)^2, \qquad (3)$$
$$s.t. \quad \sum_z d_{iz} = d_i$$

where L_{sq} is the squared loss function. The best fit between the given network G and its expected graph, following (2), can be achieved by optimizing (3), which will be introduced in Section 3.3. When we get the model parameters d_{iz}, the community membership S_{iz} can then be inferred by

$$S_{iz} = \frac{d_{iz}}{\sum_r d_{ir}} = \frac{d_{iz}}{d_i}. \qquad (4)$$

Actually, as S_{iz} provides a soft community membership, it can, not only give a hard partition, but also provide the overlapping communities in the network. To be specific, if one wants to derive a hard partition, we can simply assign each node i to group r satisfying $r=arg\max_z\{S_{iz}, z=1,2,\ldots,c\}$. But if one wishes to get an overlapping community structure, we first scale the entries in each row of the membership matrix S to [0~1] to make the membership value of the community that each node prefers best to be 1. We then vary a threshold from 0 to 1 and set all those entries in S that exceed the threshold to 1 and 0 otherwise, and then we select the best result as the final overlapping community structure.

3.2 Model Properties

Our above model is a generalization of the null model of modularity, embodying the idea of decomposing and combining the color-z sub-graphs from the perspective of soft clustering, to incorporate the ability of describing the community structure in networks. Also, it holds the property, as with the original null model, that fixes the expected degree sequence to be the same as that of the observed network (*proofed in the appendix*). Intuitively, this may enable our model, compared with the blockmodels that tend to display broad degree distributions, to be more fitted to characterize community structures in the real world.

Here we offer an example to illustrate the ability of our model to describe communities, which is shown in Figure 1 and in Table 1. The red and blue sub-graphs of G are indexed by 'z=1' and 'z=2' in Table 1. First, we consider the within-community nodes, such as 3 and 4. As d_{31} and d_{41} are both large, the expected number of red links between them is dominant ($\hat{A}^1_{3,4}$= 0.76), which causes the expected number of links between nodes 3 and 4 to be large ($\hat{A}_{3,4}$= 0.76). On the contrary, let us consider node from different communities, such as 3 and 7. As d_{71} is very small, although d_{31} is large, the expected number of red links between them is still small ($\hat{A}^1_{3,7}$< 1.0e-2); similarly, the expected number of blue links between them is also small ($\hat{A}^2_{3,7}$< 1.0e-2). This leads to the expected number of links between nodes 3 and 7 being much smaller than that between nodes 3 and 4 ($\hat{A}_{3,7}$ << $A_{3,4}$). This is exactly the common knowledge of communities, which has dense intra-connections and sparse inter-connections.

Besides, our model allows for the fact that vertices with high degree are, all other things being equal, more likely to be connected than those with low degree, simply because they have more edges. For example, there is $\hat{A}_{5,4}$ > $A_{3,4}$ in Figure 1(d), as the degree of node 5 is larger than that of node 3. From an information-theoretic viewpoint, an edge between two high-degree vertices is less surprising than an edge between two low-degree vertices, and thus intuitively we may get better community results as we incorporate this observation in our model.

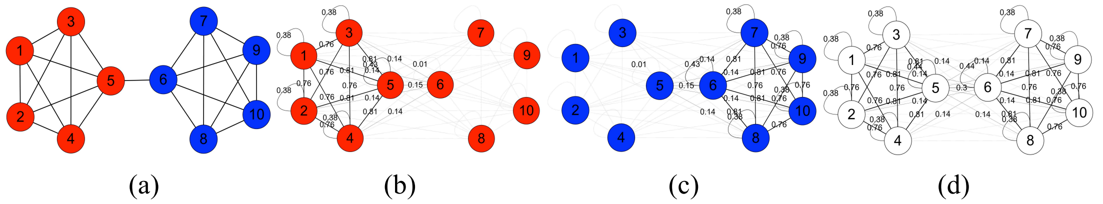


Figure 1. Example that illustrates some interesting properties of our model. The parameters d_{iz} are learned by NMF in Section 3.3, which are shown as Table 1. **(a)** The observed network G with two communities in red and blue respectively. **(b)** The expected red sub-graph of G, which is described by the null model of modularity. **(c)** The expected blue sub-graph of G. **(d)** The expected graph of G, which is an ensemble of its red sub-graph in (b) and blue sub-graph in (c). Note that the width of a link corresponds to its expected values, and that values smaller than 1.0e-2 are omitted.

Table 1. The learned model parameters d_{iz} used in Figure 1

d_{iz}	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$	$i = 6$	$i = 7$	$i = 8$	$i = 9$	$i = 10$
$z = 1$	3.999949	3.999974	3.999975	3.999972	4.253827	0.748089	7.14E-04	7.39E-04	7.02E-04	6.89E-04
$z = 2$	5.12E-05	2.63E-05	2.45E-05	2.76E-05	0.746173	4.251911	3.999286	3.999261	3.999298	3.999311

3.3 Parameter Learning

Our above model is a relatively ideal one, which mainly considers disjoint communities. However, in the real world, many networks contain communities that overlap [5]. In the opinion of [12], in order to better model overlapping community structures, the sum of probabilities for each vertex belonging to different communities may not be constrained to 1. For instance, a vertex may belong to the community ‘Politics’ with probability 0.9 and to the community ‘Economics’ with probability 0.8, due to the strong relationship between these two communities. An entity can be very active in multiple communities, but this scenario cannot be modeled well if we impose the constraint $\sum_z d_{iz} = d_i$.

Based on the above idea, we remove the normalized constraints, and this leads to a generalized loss function as

$$\min_{d_{iz} \geq 0} L_{sq}\left(A, \hat{A}\right) = \left\| A - A \right\|_F^2 = \sum_{ij} \left(A_{ij} - \sum_z \frac{d_{iz} d_{jz}}{\sum_k d_{kz}} \right)^2 . \qquad (5)$$

As the search space of (3) is inside that of (5), if our model is sound, and if a network contains only disjoint communities, the constraint $\sum_z d_{iz} = d_i$ will be well satisfied. Further, (5) will have a better ability when compared with (3) when describing networks with overlapping communities. Then, (3) can be considered as a special case of the generalized loss function in (5). Therefore, we will take (5) instead of (3) as the objective function here, which is much simpler to solve.

Now, we begin to discuss how to efficiently infer the model parameters d_{iz}. We first define an auxiliary matrix X, where X_{iz} is evaluated as

$$X_{iz} = \frac{d_{iz}}{\sqrt{\sum_j d_{jz}}} . \qquad (6)$$

Then, the optimization of (5) can be transformed into an equivalent problem of nonnegative matrix factorization, such as

$$\min_{X \geq 0} L_{sq}\left(A, XX^{\mathrm{T}}\right) = \left\| A - XX^{\mathrm{T}} \right\|_F^2 . \tag{7}$$

According to [20], X can be solved by the following multiplicative update rule

$$X_{iz} = X_{iz}\left(\frac{1}{2} + \frac{(AX)_{iz}}{\left(2XX^{\mathrm{T}}X\right)_{iz}}\right). \tag{8}$$

When it converges, using (6) we can infer the model parameters d_{iz} by

$$d_{iz} = X_{iz}\sqrt{\sum\nolimits_j d_{jz}} = X_{iz}\sum\nolimits_j X_{jz}, \tag{9}$$

Subsequently, one may wish to know how releasing the normalized constraint $\sum_z d_{iz} = d_i$ will affect the community results. For this purpose, here we offer a simple example to illustrate how much off is this expression for overlapping structures and for non-overlapping structures, when the algorithm converges. We first define a quality metric to measure how much off is this constraint for our result:

$$d_{off} = \frac{\sum_i \left| d_i - \sum_z d_{iz} \right|}{\sum_j d_j} . \tag{10}$$

Then we use two well-known networks as the testbeds. The first one is the American college football network which is widely used when detecting disjoint community structures [1], and the second one is the Les Miserables network which has become a *de facto* testbed for overlapping community detection [21]. When the algorithm converges, the d_{off} value of the American college football network is 0.0681, which is very small and much smaller that that of the Les Miserables network, with d_{off} = 0.2141. More importantly, the clustering accuracy of the American college football network is NMI=92.42%, and the clustering quality of the Les Miserables network are Q=0.4903 and L=4.7753, which are both better than that of the compared methods (see Section 4.2 for detailed comparisons).

3.4 Model Selection

Recall that our model needs to have the community number c as input, which is a priori unknown for many cases. This is the so-called model selection problem. Several approaches to model selection have been proposed [11, 13, 22], but most of them are not suitable for our model. Fortunately, the model selection method designed by Brunet et al. [23], which is based on the idea of consensus clustering, is effective for our model. Therefore, we use it to determine the number of communities when this quantity is not of prior knowledge.

Our NMF may or may not converge to the same solution on each run, depending on the random initial conditions. If a clustering into k communities is strong, we would expect that node assignment to communities would vary little from run to run. For each run, the node assignment can be defined by a connectivity matrix C_k of size n×n, with entry $C_k(i,j)$ = 1 if nodes i and j belong to the same community, and $C_k(i,j)$ = 0 if they belong to different communities, where k is the given number of communities. We can then compute the consensus matrix, $\bar{C}_k$, defined as the average connectivity matrix C_k over some runs (50 runs is generally sufficient to stabilize $\bar{C}_k$). The entries of $\bar{C}_k$ range from 0 to 1 and reflect the probability that nodes i and j cluster together. If a community structure is stable, we would expect that C_k tends not to vary among runs, and that the entries of $\bar{C}_k$ will be close to 0 or 1. Consequently, the general consistency quality of $\bar{C}_k$ is summarized by the dispersion coefficient defined as

$$\rho_k = \frac{1}{n^2} \sum_{ij} 4\left(\overline{C}_k(i,j) - \tfrac{1}{2}\right)^2 \tag{11}$$

where $0 \leq \rho_k \leq 1$, and $\rho_k = 1$ represents a perfectly consistent assignment.

A straightforward way to find the best community number is to enumerate all possible k to get the one with the maximum ρ_k [23]. This exhaustive search may become computationally expensive for large networks. Here we offer an alternative, with an effective heuristic, to this problem. We first use Louvain method [3], which is regarded as one of the best algorithms for community detection by [2], to determine an approximate community number c_s. Thereafter, we decrease k starting from c_s until $\rho_k < \rho_{k+1}$ and set $c_d = k+1$, and then increase k starting from c_s until $\rho_k < \rho_{k-1}$ and set $c_u = k-1$. Finally, we determine the best community number $c = arg\max_k \{\rho_k | k=c_d,\ldots,c_u\}$.

4. EXPERIMENTS

In order to evaluate the performance of our method ENMM, we test it on both benchmark computer-generated networks and on some widely used real-world networks. In addition, we compared it with two related NMF-based methods mentioned in Section 2, SNMF [9] and BNMTF [12], which also use the squared loss as objective function and adopt the nonnegative matrix factorization as optimization algorithm. Since each of these methods provides a soft community membership, it can not only give a hard partitioning but also provide knowledge on overlapping communities. Thus, in this section we will test their performance from both of these perspectives.

All experiments are done on a single Dell Server (Intel(R) Xeon(R) CPU 5130 @ 2.00GHz 2.00GHz processor with 4Gbytes of main memory). The source codes of the algorithms used here can all be obtained from the authors. Especially, our code is available in [24]. As the NMF-based methods do not necessarily converge to the global optimization, we repeated each algorithm 10 times with random initial conditions and have chosen the result that gives the smallest squared loss.

4.1 Synthetic Networks

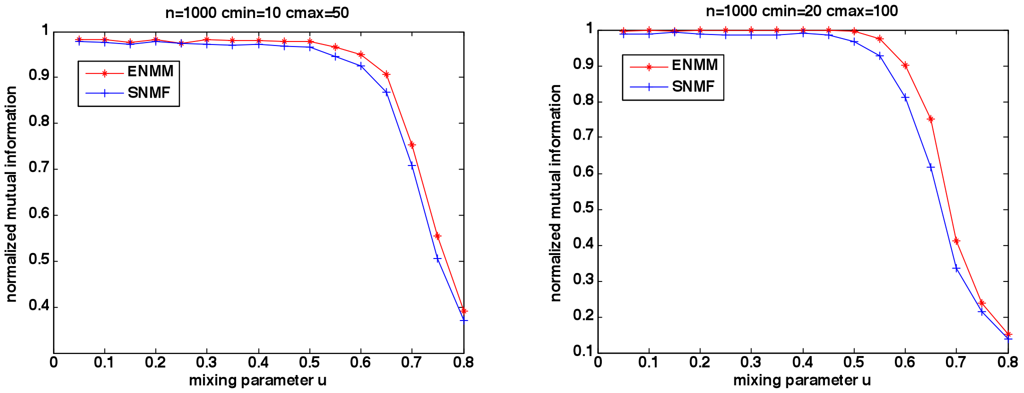
Two different types of the synthetic benchmarks, one with disjoint community structure [25] and the other with overlapping community structure [26], were proposed by Lancichinetti, Fortunato and Radicchi (LFR). Here we use both of them to test the ability of each algorithm to detect known communities under controlled conditions. In the LFR benchmark graphs, both the degree and the community size distributions are power law, which is a statistical property that most real-world networks seem to share. Notice that we did not compare the results of BNMTF, since its high computation cost meant that it could not provide results within 100 hours for each set of the tests.

4.1.1 LFR benchmark with disjoint communities

This type of benchmark is proposed by Lancichinetti et al. [25], which is designed for testing the ability of algorithms to detect disjoint communities. Here, we employ the widely used Normalized Mutual Information (NMI) index as the accuracy measure [27]. The NMI index, which makes use of information theory, is regarded as a relatively fair metric compared with the other ones [27].

Following the experiment designed by Lancichinetti et al. in [25], the parameters setting for the LFR benchmark networks are as follows. The network size n is set to 1000, the minimum community size c_{min} is set to either 10 or 20, and the mixing parameter μ (each vertex shares a fraction μ of its edges with vertices in other communities) varies from 0 to 0.8 with an interval of 0.05. We keep the remaining parameters fixed: the average degree d is 20, the maximum degree d_{max} is 2.5×d, the maximum community size c_{max} is 5×c_{min}, and the exponents of the power-law distribution of vertex degrees τ_1 and community sizes τ_2 are -2 and -1 respectively. This design space leads to two sets of benchmarks.

In Figure 2, we show that the NMI accuracy attained by each algorithm is as a function of the mixing parameter μ. As we can see, the performance of our method ENMM is slightly better than that of SNMF, especially when μ is in the range of 0.5 to 0.7.



(a) (b)

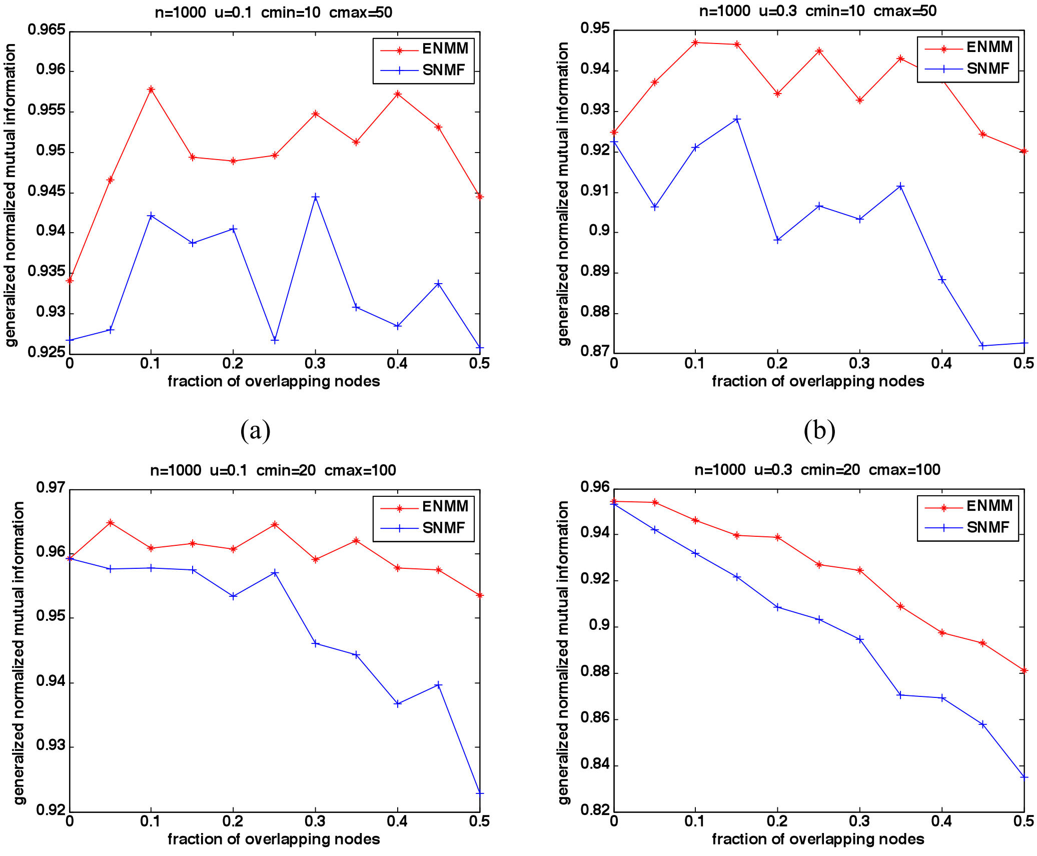
Figure 2. Comparison of ENMM and SNMF in terms of NMI accuracy on the LFR benchmark networks of disjoint version. Each data point in the figure is an average over 10 graph instances. **(a)** Results on networks with small communities (n = 1000, c_{min} = 10, c_{max} = 50). **(b)** Results on networks with large communities (n = 1000, c_{min} = 20, c_{max} = 100).

4.1.2 LFR benchmark with overlapping communities

This type of benchmark is also proposed by Lancichinetti et al. [26], but designed for testing the ability of algorithms to detect overlapping communities. Considering the accuracy measure, the standard NMI index does not work in this case. Fortunately, there is a new variant of it, namely Generalized Normalized Mutual Information (GNMI), which is extended to handle overlapping communities [6]. Thus we adopt this GNMI index as the accuracy measure for this experiment.

Like the experiment designed by Lancichinetti et al. in [26], the parameters setting for this LFR benchmarks are as follows. The network size n is 1000, the minimum community size c_{min} is set to either 10 or 20, the mixing parameter μ (each vertex shares a fraction μ of its edges with vertices in other communities) is set to either 0.1 or 0.3, the fraction of overlapping vertices (o_n/n) varies from 0 to 0.5 with interval 0.05. We keep the remaining parameters fixed: the average degree d is 20, the maximum degree d_{max} is 2.5×d, the maximum community size c_{max} is 5×c_{min}, the number of communities, each overlapping vertex belongs to (denoted o_m), is 2, and the exponents of the power-law distribution of vertex degrees τ_1 and community sizes τ_2 are -2 and -1, respectively. This design space leads to four sets of benchmarks.

Figure 3 shows the results that compare our method ENMM with SNMF in terms of the GNMI index on the heterogeneous artificial networks with overlapping communities. As we can see, the performance of ENMM is clearly better than that of SNMF for all the four samples. In particular, when the fraction of overlapping vertices (o_n/n) is larger, the superiority of our method becomes even more obvious.



(a)

(b)

(c)

(d)

Figure 3. GNMI accuracy of each algorithm as a function of the fraction of overlapping nodes. Each point is an average result over 10 graphs. **(a)** Comparison on synthetic networks with small mixing parameter and small communities (μ = 0.1, c_{min} = 10, c_{max} = 50). **(b)** Comparison on synthetic networks with big mixing parameter and small communities (μ = 0.3, c_{min} = 10, c_{max} = 50). **(c)** Comparison on synthetic networks with small mixing parameter and big communities (μ = 0.1, c_{min} = 20, c_{max} = 100). **(d)** Comparison on synthetic networks with big mixing parameter and big communities (μ = 0.3, c_{min} = 20, c_{max} = 100).

4.2 Real-World Networks

As real networks may have some different topological properties that distinguish them from the synthetic ones, we now consider some widely used real-world networks to further evaluate the performance of these algorithms. First, we test these algorithms in terms of accuracy on several networks whose community structures are known. However, networks that have a known community structures are rare. Thus, we also evaluate these different methods in terms of the community quality on networks without known community structures. Note that, all the real-world networks we used here are obtained from Newman's website [28].

4.2.1 Accuracy comparison

The real networks we used here, whose ground-truths of community structures is known are listed in Table 2, and the comparisons of different algorithms on these networks are shown in Table 3. Notice that, ‘Friendship6’ and ‘Friendship7’ denote the same network, but they used different community ground-truths. Here we also adopt the NMI index as the accuracy measure. As we can see from the results, our method ENMM has the best (or co-best) performance on five of the seven networks, and it is also competitive with SNMF and BNMTF on the other two networks. In terms of efficiency, ENMM is competitive with that of SNMF, and much faster than BNMTF.

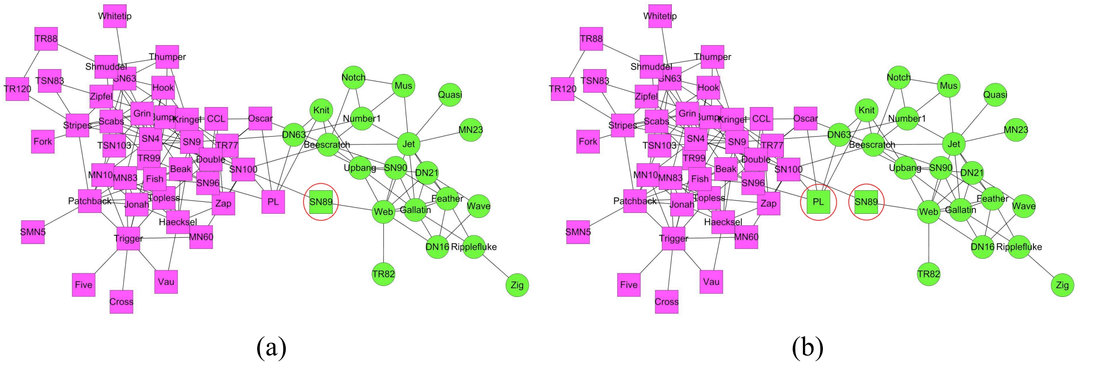
Table 2. Some real-world networks with known community structures

Datasets	*n*	*m*	*c* (ground-truths)
Zachary’s karate club	34	78	2
Dolphin social network	62	160	2
High school friendship network	69	220	6 7
Political books	105	441	3
American college football	115	613	12
Political blogs	1,490	16,717	2

Table 3. The comparison of each method on real networks in Table 2

	NMI index (%)			Run time (seconds)		
	ENMM	SNMF	BNMTF	ENMM	SNMF	BNMTF
Karate	**100**	**100**	**100**	0.0118	0.0071	1.3152
Dolphin	**88.88**	81.41	81.41	0.0069	0.0084	2.5602
Friendship6	**79.30**	78.64	71.22	0.0163	0.0222	8.3985
Friendship7	*84.26*	82.11	**84.30**	0.0286	0.0266	9.5503
Polbooks	54.04	**56.48**	51.18	0.0219	0.0484	7.0164
Football	**92.42**	90.38	**92.42**	0.0532	0.0612	30.7691
Polblogs	**71.07**	70.95	70.78	2.4307	2.2343	2498.1

Furthermore, we take the dolphin social network as an example to further compare these algorithms. Figure 4(a) is the community result obtained by our method ENMM, and Figure 4(b) is what is obtained by SNMF and BNMTF. Different shapes denote the actual communities, and different colors express the community results got by the algorithms. As we can see, we mismatch ‘SN98’. However, this node has only one neighbor ‘SN100’ in the square group and one neighbor ‘Web’ in the cycle group. Accordingly, it seems difficult to decide which group ‘SN98’ should belong to. But as the degree of ‘SN100’ is seven and that of ‘Web’ is eight, it looks more reasonable to assign ‘SN98’ to the cycle group by using only the information on network topology. This is the, correct, result got by our method ENMM. However, for SNMF and BNMTF, they both mismatch ‘PL’, which has three neighbors in the square group and two neighbors in the cycle group. Thus, it seems as a mistake to assign this node to the green group in the results of these two algorithms.



(a) (b)

Figure 4. Comparison of different algorithms on the dolphin social network. Note that, Different shapes denote the actual communities, and different colors express the community results got by the algorithms. **(a)** Community result got by our method ENMM. **(b)** Community result got by SNMF and BNMTF.

4.2.2 Quality comparison

We also test these algorithms on real-world networks whose community structures are unknown (see Table 4). As they do not have ground-truths, their community numbers are all obtained by our method of model selection which was introduced in Section 3.4. Because each of the algorithms can find both disjoint and overlapping community structures, here we used two widely used quality metrics: one is modularity Q [18] for evaluating hard partitions, and the other is the generalized map equation L for evaluating overlapping communities [29].

Table 4. Some real-world networks without known community structures

Datasets	n	m	c (model selection)
Les Miserables	77	254	12
Word adjacencies	112	425	37
Jazz musicians collaborations	198	2,742	3
C. Elegans neural	297	2,148	28
E. coli metabolic	453	2,025	18
E-mail network URV	1,133	5,451	27
Network science collaborations	1,589	2,742	277
Power grid	4,941	6,594	42
Word association	5,017	29,148	48

Table 5. Comparison of each algorithm on the real networks in Table 4. Here, the greater the better, for Q-value, and the smaller the better, for L-value. The notation '–' denotes time > 100 hours.

Datasets	Modularity Q (disjoint)			Map equation L (overlaps)			Run time (seconds)		
(abbr)	ENMM	SNMF	BNMTF	ENMM	SNMF	BNMTF	ENMM	SNMF	BNMTF
Lesmis	**0.4903**	0.4434	0.4803	**4.7753**	4.8863	4.8299	6.2094	15.4238	23.2728
Adjnoun	**0.1704**	0.1693	0.1698	**6.9684**	7.0538	7.0446	0.8085	22.1005	122.1150
Jazz	0.4398	0.4406	**0.4410**	6.8117	6.8099	**6.8091**	1.7072	4.1945	113.4448
Neural	**0.2492**	0.2465	0.2433	**8.0017**	8.0307	8.0411	18.2420	34.0775	1721.7
Metabolic	0.3724	**0.3857**	0.3745	**7.3202**	7.3536	7.3388	4.6194	29.2601	1982.2
Email	**0.4772**	0.4760	0.4699	**8.3767**	8.4717	8.5269	50.2086	38.4809	13705
Netscience	**0.8187**	0.7954	0.7407	**5.2338**	5.2922	6.7284	88.8224	90.7758	337010
Power	**0.8786**	0.8493	–	**7.9611**	8.2633	–	125.3689	131.1286	–
Word	**0.4142**	0.4078	–	**10.7608**	10.8191	–	261.5206	300.9315	–

Table 5 shows the result that compares our method ENMM with SNMF and BNMTF on the real-world networks described in Table 4. As we can see, in terms of modularity Q, ENMM has the best performance on seven of the nine networks; and in terms of the map equation L, ENMM has the best performance on eight of the nine networks. The superiority of ENMM is more obvious when a network is larger or when it has more communities. Notice that, Netscience is not an ergodic network, which is not originally supported by the map equation. But a non-ergodic process on a network can be made ergodic by introducing a small teleportation probability (such as 0.15) [4, 29]. Therefore, the map equation can be used for Netscience as well. Again, in terms of efficiency, our method ENMM is competitive with SNMF, and much faster than BNMTF.

5. CONCLUSION

In this work, we generalize a configuration model, namely the null model of modularity [18, 19], to incorporate communities from the perspective of soft clustering. Then, we define a function of squared loss based on this extended model, and solve it by using a technique of nonnegative matrix factorization. Thereafter, we introduce a method of model selection to determine the number of communities for this model. Our model can theoretically preserve the degree sequence of networks. It provides both overlapping and disjoint communities, and is also suitable for weighted networks. We have evaluated our above method ENMM both on synthetic benchmarks and on some real-world networks, and compared it with two similar methods for community detection. The experimental results demonstrated the superior performance of ENMM over the competing ones in detecting both disjoint and overlapping communities.

Nonetheless, our model selection method is not perfect. For large networks, it still needs some assistance from other methods, such as Louvain method [3] used here, to improve the efficiency. But this problem is also patent in many model-based methods for community detection, and it is still an open problem whether a reliable method of model selection can be developed that runs in reasonable time over large networks [16]. Thus in the future, we wish to improve our current method of model selection, to mitigate this problem, by following some heuristics hints.

It is noteworthy that, our model function can be also regarded as a generalization of Newman's function of modularity [18] to incorporate overlapping communities. Recently, several extensions of modularity have been proposed [2]. Of particular note is the work of Nicosia et al. [30], which also supports overlapping communities. It seems that Nicosia's modularity has a high similarity with the one proposed here by us, but they have some key differences. Firstly, Nicosia extended the basic function of modularity by introducing a belonging factor to support overlapping communities. But our extension is indeed a function of generative model which, instead of directly detecting communities, describes how such overlapping structures are generated in the first place. Secondly, Nicosia optimized their function by using a genetic-based algorithm, but we used a statistical inference method, i.e. nonnegative matrix factorization, to learn our model function. Last but not least, Nicosia's modularity obtains the number of communities by itself, but our model function needs a model selection method to determine this quantity.

Since our model function is an extension of modularity, it may also suffer from the resolution limit problem [31], not being able to discern the quality of modules smaller than a certain size. As we have noted recently, Traag et al. [32] extended the modularity to support multi-resolutions, and also, they introduced a notion of "significance" of a partition, based on subgraph probabilities, to state what partitions are significant. Based on this elicitation, we may also be able to introduce a similar resolution parameter to our model to incorporate multi-resolutions, and then we can use a suitable metric, such as Traag's significance, to determine the good resolution parameters. But we will leave it as our future work.

APPENDIX

Proposition 1. Under our model, the expected graph $\hat{G}$ preserves the same degree sequence as the observed graph *G*.

Proof. Let d_i be the degree of node *i* in *G*. Given an arbitrary set of variables d_{iz}, subject to $\sum_z d_{iz} = d_i$, which express our model parameters, then using (2) the degree of any node *i* in the expected graph $\hat{G}$ can be inferred as

$$\begin{aligned}\hat{d}_i &= \sum_j \hat{A}_{ij} = \sum_j \sum_z \frac{d_{iz} d_{jz}}{\sum_k d_{kz}} \\ &= \sum_z \frac{d_{iz} \sum_j d_{jz}}{\sum_k d_{kz}} = \sum_z d_{iz} \quad , \\ &= d_i\end{aligned} \tag{A1}$$

correctly matching the degree of node *i* in the observed graph *G*.

ACKNOWLEDGMENTS

This work is supported by Major State Basic Research Development Program of China (2013CB329301), Open Project Program of Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education (93K172013K02), Innovation Foundation of Tianjin University (2013XQ-0136), and the China Scholarship Council (award to Dongxiao He for one year's study abroad at Washington University in St Louis).

REFERENCES

[1] Girvan, M., and Newman, M. E. J. 2002. Community structure in social and biological networks. *Proc. Natl. Acad. Sci.* 99 (Jun. 2002), 7821-7826.

[2] Fortunato, S. 2010. Community detection in graphs. *Phys. Rep.* 486 (2010), 75-174.

[3] Blondel, V. D., Guillaume, J. L., Lambiotte, R., and Lefebvre, E. 2008. Fast unfolding of communities in large networks. *J. Stat. Mech.* (Oct. 2008), P10008.

[4] Rosvall, M., and Bergstrom, C. T. 2008. Maps of random walks on complex networks reveal community structure. *Proc. Natl. Acad. Sci.* 105 (Jan. 2008), 1118-1123.

[5] Palla, G., Derenyi, I., Farkas, I., and Vicsek, T. 2005. Uncovering the overlapping community structures of complex networks in nature and society. *Nature* 435 (Jun. 2005), 814-818.

[6] Lancichinetti, A., Fortunato, S., and Kertesz, J. 2009. Detecting the overlapping and hierarchical community structure in complex networks. *New J. Phys.* 11 (Mar. 2009), 033015.

[7] Ahn, Y., Bagrow, J. P., and Lehmann, S. 2010. Link communities reveal multiscale complexity in networks. *Nature* 466 (Aug. 2010), 761-764.

[8] Newman, M. E. J. 2012. Communities, modules and large-scale structure in networks. *Nature Physics* 8 (Jan. 2012), 25-31.

[9] Wang, F., Li, T., Wang, X., Zhu, S., and Ding, C. H. Q. 2011. Community discovery using nonnegative matrix factorization. *Data Mining and Knowledge Discovery* 22 (May 2011), 493-521.

[10]Psorakis, I., Roberts, S., and Sheldon, B. 2010. Soft partitioning in networks via Bayesian non-negative matrix factorization. *In Proceedings of the 24th Annual Conference on Neural Information Processing Systems (NIPS), Workshop on Networks Across Disciplines: Theory*

and Applications (Whistler, BC, Canada, December 11, 2010). NETWORKSNIPS '10. ACM, New York, NY.

[11]Psorakis, I., Roberts, S., Ebden, M., and Sheldon, B. 2011. Overlapping community detection using Bayesian non-negative matrix factorization. *Phys. Rev.* E 83 (Jun. 2011), 066114.

[12]Zhang, Y., and Yeung, D. 2012. Overlapping community detection via bounded nonnegative matrix tri-factorization. *In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining* (Beijing, China, August 12-16, 2012). KDD '12. ACM, New York, NY, 606-614.

[13]Ren, W., Yan, G., Liao, X., and Xiao, L. 2009. Simple probabilistic algorithm for detecting community structure. *Phys. Rev.* E 79 (Mar. 2009), 036111.

[14]Shen, H., Cheng, X., and Guo, J. 2011. Exploring the structural regularities in networks. *Phys. Rev.* E 84 (Nov. 2011), 056111.

[15]Karrer, B., and Newman, M. E. J. 2011. Stochastic blockmodels and community structure in networks. *Phys. Rev.* E 83 (Jan. 2011), 016107.

[16]Ball, B., Karrer, B., and Newman, M. E. J. 2011. Efficient and principled method for detecting communities in networks. *Phys. Rev.* E 84 (Sep. 2011), 036103.

[17]Nowicki, K., and Snijders, T. A. B. 2001. Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association* 96 (Sep. 2001), 1077-1087.

[18]Newman, M. E. J., and Girvan, M. 2004. Finding and evaluating community structure in networks. *Phys. Rev.* E 69 (Feb. 2004), 026113.

[19]Nadakuditi, R. R., and Newman, M. E. J. 2013. Spectra of random graphs with arbitrary expected degrees. *Phys. Rev.* E 87 (Jan. 2013), 012803.

[20]Wang, D., Li, T., Zhu, S., and Ding, C. H. Q. 2008. Multi-document summarization via sentence-level semantic analysis and symmetric matrix factorization. *In: Proceedings of the 31st annual international ACM SIGIR conference on research and development in information retrieval* (Singapore, 20-24 July 2008), SIGIR '08. ACM, New York, NY, 307-314.

[21]Knuth, D. E. 1994. The Stanford GraphBase: A Platform for Combinatorial Computing, (ACM Press, New York).

[22]Tan, V. Y. F., and Févotte, C. 2012. Automatic relevance determination in nonnegative matrix factorization with the *β*-divergence. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35 (Jul. 2012), 1592-1605.

[23]Brunet, J-P., Tamayo, P., Golun, T. R., and Mesirov, J. P. 2004. Metagenes and molecular pattern discovery using matrix factorization. *Proc. Natl. Acad. Sci.* 101 (Mar. 2004), 4164-4169.

[24]The software of our method ENMM can be found in ftp://jindi:dd@59.72.0.62:2121

[25]Lancichinetti, A., Fortunato, S., and Radicchi, F. 2008. Benchmark graphs for testing community detection algorithms, *Phys. Rev.* E 78 (Oct. 2008), 046110.

[26]Lancichinetti, A., and Fortunato, S. 2009. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Phys. Rev.* E 80 (Jul. 2009), 016118.

[27]Danon, L., Duch, J., Diaz-Guilera, A., and Arenas, A. 2005. Comparing community structure identification. *J. Stat. Mech.* (Sep. 2005), P09008.

[28]The real-world networks we used here are available in http://www-personal.umich.edu/~mejn/netdata/

[29]Esquivel, A. V., and Rosvall, M. 2011. Compression of flow can reveal overlapping-module organization in networks. *Phys. Rev.* X 1 (Dec. 2011), 021025.

[30]Nicosia, V., Mangioni, G., Carchiolo, V., and Malgeri, M. 2009. Extending the definition of modularity to directed graphs with overlapping communities. *J. Stat. Mech.* (Mar. 2009), P03024.

[31]Fortunato, S., and Barthélemy, M. 2007. Resolution limit in community detection. *Proc. Natl. Acad. Sci.* 104 (Jan. 2007), 36-41.

[32] Traag, V. A., Krings, G., and Van Dooren, P. 2013. Significant scales in community structure. arXiv:1306.3398