Evaluation of Community Detection Methods

Xin Liu, Hui-Min Cheng, and Zhong-Yuan Zhang

Abstract—Community structures are critical towards understanding not only the network topology but also how the network functions. However, how to evaluate the quality of detected community structures is still challenging and remains unsolved. The most widely used metric, normalized mutual information (NMI), was proven to have finite size effect, and its improved form relative normalized mutual information (rNMI) has reverse finite size effect. Corrected normalized mutual information (cNMI) was thus proposed and has neither finite size effect nor reverse finite size effect. However, in this paper, we show that cNMI violates the so-called proportionality assumption. In addition, NMI-type metrics have the problem of ignoring importance of small communities. Finally, they cannot be used to evaluate a single community of interest. In this paper, we map the computed community labels to the ground-truth ones through integer linear programming, and then use kappa index and F-score to evaluate the detected community structures. Experimental results demonstrate the advantages of our method.

Index Terms—Normalized mutual information, kappa index, F-score, community structures detection

1 INTRODUCTION

There are often community structures in complex networks, which often refer to groups of nodes that are intra-connected tightly and inter-connected loosely. Community structures are very important towards understanding not only the network topology but also how the network functions, and many methods have been proposed to detect community structures from different perspectives [1]. However, how to evaluate them is still a challenging problem and remains unsolved.

The most widely used approach is testing and comparing different methods on synthetic benchmark networks with known community labels of nodes. Their performance is determined by calculating the similarity between computed labels and the ground-truth ones. The metrics are borrowed heavily from the field of clustering, including purity [2], rand index [3], etc., and the most widely used one among them is normalized mutual information (NMI) [4], [5]. However, recent research showed that NMI has finite size effect due to finite size of network [6]. Hence relative normalized mutual information (rNMI) was proposed to handle this issue, but it has reverse finite size effect due to the same reason. To overcome the biases of NMI and rNMI, corrected normalized mutual information (cNMI) was proposed. Actually, cNMI is not sufficient either because it violates the proportionality assumption, as will be discussed later.

The main challenge when designing reasonable indices for evaluation of community structures detection methods is the limited amount of information available from the community labels. The problem is a kind of unsupervised learning, and the results can only provide the information of which nodes are clustered together and which ones are not, making no sense to point-wise compare between computed labels and ground-truth ones. For example, the two community label vectors \( [1, 1, 1, 1, 1, 2, 2, 3, 3] \) and \( [2, 2, 2, 2, 2, 3, 3, 3, 1, 1] \) are actually identical. Furthermore, the indices cannot be used to evaluate the ability to detect a single community of interest due to the same reason, which is often very important in real applications such as understanding and assessing terror organizations, mining biological networks for unknown pathways, etc. The above problems can be largely mitigated in the field of supervised learning since the computed labels are relatively more informative, making point-wise label comparison reasonable. In this paper, we map the computed community labels to the ground-truth ones under the framework of linear programming, and then use kappa index and F-score for classification to evaluate community structures. Experimental results demonstrate the advantages of our method.

The rest of the paper is organized as follows: Section 2 briefly reviews the representative indices for performance evaluation in clustering and classification; Section 3 proposes the linear programming method, mapping the computed community labels to the ground-truth ones; Section 4 empirically demonstrates the advantages of our method over others, and Section 5 concludes.

2 RELATED WORK

In this section, we will briefly review the representative indices for performance evaluation in clustering and classification.

2.1 Indices of Clustering Validity in Unsupervised Learning

There are two types of indices for clustering evaluation: internal ones and external ones, and in this section we offer a brief analysis of some representative ones.
2.1.1 Internal Indices

Internal indices evaluate the clustering result without any external information and are only based on the information intrinsic to the data, such as sum of squares, Calinski-Harabasz index and Silhouette index, etc.

i) Sum of squares: As the simplest and most intuitive evaluation index, it computes the sum of within-cluster sum of squares as follows:

\[ \sum_{r=1}^{n} \sum_{i=1}^{n} w_{ir} \| s_i - \bar{s}(r) \|^2, \]  

where there are \( n \) samples \( s = \{s_1, s_2, \ldots, s_n\} \) with \( c \) clusters, and \( \bar{s}(r), r = 1, 2, \ldots, c \) is the cluster center, \( w_{ir} \) is the clustering membership indicator, i.e., \( w_{ir} = \begin{cases} 1, & s_i \in \text{cluster } r \\ 0, & \text{otherwise.} \end{cases} \)

ii) Calinski-Harabasz index [8]: This index considers not only internal compactness of clusters, but also separability of clusters. It is very similar with F-test in ANOVA, and is calculated as the following ratio of between-cluster mean and sum of within-cluster sum of squares

\[ \frac{BGSS/(c-1)}{WGSS/(n-c)}. \]  

where

\[ BGSS = \sum_{r=1}^{c} n_r \| \bar{s}(r) - \bar{s} \|^2, \]

and \( WGSS \) is defined in Eq. (1), \( n, c \) are the numbers of samples and clusters, respectively, \( c-1 \) and \( n-c \) are degrees of freedom, respectively, \( n_r \) is the number of samples in cluster \( r \), and \( \bar{s} \) is the center of the \( n \) samples.

The larger the index, the better the clustering result.

iii) Silhouette index [9]: The index for sample \( i \) calculates how similar the sample is to its own cluster compared with other ones, and is defined as follows:

\[ S(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \]

where \( a(i) \) is the average distance of sample \( i \) to the ones in its own cluster \( r \), \( b(i) = \min\{b(ik), k = 1, 2, \ldots, c, k \neq r\} \), and \( b(ik) \) is the average distance of sample \( i \) to those in the \( k \)-th cluster \( (k \neq r) \).

The Silhouette index of clustering result is the average of \( S(i) \), \( i = 1, 2, \ldots, n \), and ranges from \(-1\) to \(+1\).

Since almost all internal indices are actually measuring the compactness of clusters, they naturally prefer the clustering result with a larger cluster number and fewer samples in each cluster. For example, Calinski-Harabasz index, which is considered as one of the most effective indices [10], chooses 5 as the optimal cluster number on the famous Iris dataset, whose structures are simple and has 3 clusters [11]. Furthermore, it is hard to extend the above indices to network data since there are no coordinates for the nodes, making it difficult to calculate the distance between nodes. Specially designed indices include modularity, partition density, etc.

iv) Modularity [12]: The basic idea of modularity is very similar to hypothesis testing: given a network partition, it compares the fraction of links inside each module of the partition with the so-called null model, i.e., the expected fraction of links inside the corresponding module in degree preserving counterpart, and sums the differences over all of the modules of the partition. Higher modularity value indicates more reasonable and meaningful partition. For a complex network \( G = (V, E) \), if it is divided into \( c \) communities, the modularity is defined as follows:

\[ Q = \frac{1}{2m} \sum_{i,j=1}^{n} \left( A_{ij} - \frac{k_i k_j}{2m} \right) \delta(g_i, g_j), \]

where \( m \) is the number of edges in the network, \( g_i \) is the community label of node \( i, g_i \in \{1, 2, \ldots, c\}, \) \( k_i \) is degree of node \( i, A = (A_{ij})_{n \times n} \) is the adjacency matrix.

Modularity not only can be used for evaluation of community structures, but can also be optimized directly. However, modularity-optimization strategy has a resolution limit [13]. Modularity maximization usually tends to merge small communities into large ones in network. Further studies have found more disadvantages of modularity [14], [15], [16]. These drawbacks may lead to wrong estimation of the number of communities [17]. New community quality metrics, such as modularity density, were proposed to overcome these problems [18].

v) Partition density [19]: It only considers the compactness of communities, i.e., local information, and is defined as follows:

\[ D = \frac{2}{m} \sum_{u=1}^{c} m_u \frac{m_u - (n_u - 1)}{(n_u - 2)(n_u - 1)}, \]

where \( m_u \) and \( n_u \) are the numbers of edges and vertices in the community \( u \), respectively.

Although partition density is free from the resolution limit of modularity, it suffers from the problem of inverse resolution limit and has preference towards triangles [20].

In addition, Yang et al. [21] studied the internal indices in depth for evaluation of community detection methods, too.

In order to evaluate the clustering results more objectively, a large number of external evaluation indices were proposed.

2.1.2 External Indices

Generally speaking, external indices evaluate the clustering result by comparing it with the external supplied true labels. For the sake of clarity, we assume that there are \( n \) samples with the true labels \{1, 1, 1, 1, 1, 2, 2, 2, 3, 3\}.

i) Purity [2]: This index maps each cluster obtained by clustering method to the ground-truth one with highest overlap, and tries to calculate the percentage of “corrected” clustered samples, leading to the following formula:
where $n$ is the number of samples, $B = \{B_1, B_2, \ldots, B_c\}$ and $A = \{A_1, A_2, \ldots, A_r\}$ are sets of clusters obtained by clustering method and the ground-truth ones, respectively.

For example, if the labels obtained by clustering method are $[2, 2, 2, 2, 1, 1, 1, 3, 3]$, purity will match 2 to 1, 1 to 2, 3 to 3, and the purity value is 0.80. Note that the mapping is not one-to-one, but many-to-one. For example, if the labels obtained by clustering method are $[1, 1, 1, 4, 4, 4, 2, 2, 3, 3]$, purity will match 1 to 1, 4 to 1, 2 to 2 and 3 to 3. The purity value is 1, which is apparently unreasonable.

Rossetti et al. [22] proposed community precision, community recall and F1-measure to evaluate the performance of community detection methods. However, the mapping in the method is not one-to-one either, leading to the same problem that purity has. For example, if the labels obtained by clustering method are $[1, 1, 1, 1, 1, 1, 1, 1, 2, 2]$, the mapping will be: $1 \mapsto 1, 2 \mapsto 3$, and the value of F1 is 0.93. On the other hand, if the labels obtained by clustering method are $[1, 1, 1, 1, 2, 2, 2, 3, 3]$, the mapping will be: $1 \mapsto 1, 2 \mapsto 2, 3 \mapsto 3$, and the value of F1 is 0.90. The results are apparently unreasonable.

ii) Rand index [3]: The most important information we can get from clustering result is not the labels, but rather which samples are clustered together and which ones are not. This motivates people to define Rand index as follows:

$$\text{Rand}(A, B) = \frac{2(a + d)}{n(n - 1)},$$

where $n$ is the number of samples, $B = \{B_1, B_2, \ldots, B_c\}$ and $A = \{A_1, A_2, \ldots, A_r\}$ are sets of detected clusters and the ground-truth ones, respectively. $a$ is the number of pairs of samples which are in the same cluster of both $A$ and $B$, and $d$ is the number of pairs of samples which are not in the same cluster of both $A$ and $B$.

For example, if the labels obtained by clustering method are $[1, 1, 1, 4, 4, 4, 2, 2, 3, 3]$, the Rand index is 0.8, performing better than purity. However, Rand index does not notice the importance of small clusters in imbalanced data. For example, both of the clustering results $[1, 1, 1, 1, 1, 1, 1, 2, 2, 3, 3]$ and $[1, 1, 1, 1, 1, 1, 2, 2, 2, 3, 3]$ have the same value of Rand index 0.84, which is not informative.

iii) Normalized mutual information (NMI, [4], [5]): NMI derives from entropy in information theory. For a discrete random variable $X$, its entropy is defined as

$$H(X) = -E[\log p(x)] = -\sum_x p(x) \log p(x).$$

For example, the entropy of the true labels is

$$H_1 = -\left(\frac{6}{10} \log \frac{6}{10} + \frac{2}{10} \log \frac{2}{10} + \frac{2}{10} \log \frac{2}{10}\right) = 1.37.$$
In addition, the probabilities in NMI are approximated by frequency, resulting in finite size effect and preferring large number of partitions, and rNMI was proposed to overcome this issue [6]. However, rNMI has reverse finite size effect also due to finite size of network, making it prefer small number of partitions. Thus, cNMI was proposed to remove both finite size effect of NMI and reverse finite size effect of rNMI [7]. We use Fig. 2 [7] to better demonstrate the above conclusions: we calculate NMI, rNMI and cNMI between two partitions $A$ and $B$, where $B$ is obtained from $A$ by flipping the labels of a fraction of nodes. One can observe that: 1) The values of NMI between two independent partitions are much larger than zero due to finite size effect; 2) The values of rNMI between two identical partitions are not one due to reverse finite size effect; 3) cNMI can handle both finite size effect and reverse finite size effect, but the lines are not straight, indicating that the decline of cNMI is not proportional to the fraction of flipping labels, which is counter-intuitive.

Finally, it is worth mentioning that neither rNMI nor cNMI can solve the problem of ignoring importance of small communities. For example, for the clustering results $[1, 1, 1, 1, 1, 2, 2, 2, 3]$ and $[1, 1, 1, 1, 1, 2, 2, 3, 3]$, the values of rNMI are 0.57 and 0.47, respectively, and those of cNMI are 0.76 and 0.64, respectively. The results are still counter-intuitive.

More discussions of random effect on clustering similarity can be found in [23].

Besides the problems discussed above, the indices of clustering validity in unsupervised learning cannot be used to evaluate a specific community of interest. However, under many circumstances, we prefer the methods who can detect specific communities or structures in the network even if their overall performance is not good.

In the next section, we will briefly describe how to evaluate classification results in supervised learning.

### 2.2 Indices of Classification Validity in Supervised Learning

We start with binary classification case. Confusion matrix displaying the classification results intuitively, can be defined as Table 1. Based on confusion matrix, one can define accuracy, sensitivity, specificity, precision, recall and F-score as follows [24, 25, 26, 27]:

\[
\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \tag{14}
\]

\[
\text{Sensitivity} = \frac{TP}{TP + FN} \tag{15}
\]

\[
\text{Specificity} = \frac{TN}{TN + FP} \tag{16}
\]

\[
\text{Precision} = \frac{TP}{TP + FP} \tag{17}
\]

\[
\text{Recall} = \frac{TP}{TP + FN} \tag{18}
\]

\[
\text{F-score} = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}} \tag{19}
\]

Accuracy is the percentage of correctly classified samples, and does not consider the significance of small class. Sensitivity/recall (specificity) calculates how good a method is at detecting the positives (avoiding false positives), and can be maximized by labeling all of the samples with positive (negative). Precision calculates the fraction of samples labeled with positive that are really positives, and can be maximized by only labeling one positive sample with positive. In summary, the above metrics may be misleading, and F-score is proposed to solve the problems by combining precision and recall instead of just using one.

Furthermore, kappa index [26] considers the influence of random effects, and is defined as

\[
\text{kappa} = \frac{\text{accuracy} - \text{expected accuracy}}{1 - \text{expected accuracy}}. \tag{20}
\]

Kappa index can overcome the challenge of ignoring small class mentioned in Section 2.1.2: External indices, i.e., if the true labels are $[1, 1, 1, 1, 1, 2, 2, 3, 3]$, the kappa indices for the two classification results $[1, 1, 1, 1, 1, 2, 2, 2, 3]$ and $[1, 1, 1, 1, 2, 2, 2, 3, 3]$ are 0.82 and 0.83, respectively. This meets our expectations.
The metrics can be naturally extended for evaluation of multi-class classification problems. Note that the indices mentioned above can be used to evaluate specific class of interest.

In next section, we connect the performance evaluation problem of clustering to classification through integer linear programming. Then we use kappa index for overall performance evaluation of community detection methods, and F-score for evaluation of detecting a single community of interest.

3 MODEL DESCRIPTION

In this section, we map the computed labels to the ground-truth ones through integer linear programming. The mapping is one-to-one.

We assume without loss of generality that \( c \leq c^* \), and introduce the following decision variables:

\[
    x_{ij} = \begin{cases} 
        1, & \text{computed label } i \text{ is mapped to ground truth } j. \\
        0, & \text{otherwise.}
    \end{cases}
\]

The cost of the mapping is \( l_{ij} = |B_i \cup A_j| - |B_i \cap A_j| \), where \( A = \{A_1, A_2, \ldots, A_c\} \) and \( B = \{B_1, B_2, \ldots, B_c\} \) are set of ground-truth communities and set of computed communities, respectively. The complete integer linear programming model is

\[
    \min \ z = \sum_{i=1}^{c} \sum_{j=1}^{c'} l_{ij} x_{ij}
    \quad \text{s.t.} \quad \begin{cases} 
    \sum_{j=1}^{c} x_{ij} \leq 1, & j = 1, 2, \ldots, c^*. \\
    \sum_{i=1}^{c} x_{ij} = 1, & i = 1, 2, \ldots, c. \\
    x_{ij} = 0 \text{ or } 1.
    \end{cases}
\]

Actually, it can be reformulated as standard assignment problem by creating the cost matrix as \( L' = (l'_{ij})_{c \times c^*} \) such that

\[
    l'_{ij} = \begin{cases} 
        l_{ij}, & \text{if } i \leq c \\
        0, & \text{otherwise}
    \end{cases}
\]

i.e.,

\[
    L' = \begin{bmatrix}
    l_{11} & l_{12} & \cdots & \cdots & \cdots & l_{1c^*} \\
    l_{21} & l_{22} & \cdots & \cdots & \cdots & l_{2c^*} \\
    \vdots & \vdots & \ddots & \cdots & \cdots & \vdots \\
    l_{c1} & l_{c2} & \cdots & \cdots & \cdots & l_{cc^*} \\
    0 & 0 & \cdots & \cdots & \cdots & 0 \\
    \vdots & \vdots & \ddots & \cdots & \cdots & \vdots
    \end{bmatrix}_{c \times c^*}.
\]

And the model is updated

\[
    \min \ z = \sum_{i=1}^{c} \sum_{j=1}^{c'} l'_{ij} x_{ij}
    \quad \text{s.t.} \quad \begin{cases} 
    \sum_{j=1}^{c} x_{ij} = 1, & j = 1, 2, \ldots, c^*. \\
    \sum_{i=1}^{c} x_{ij} = 1, & i = 1, 2, \ldots, c^*. \\
    x_{ij} = 0 \text{ or } 1,
    \end{cases}
\]

which can be solved by Hungarian method.

The following theorem proves that the optimal solution of (22) is guaranteed by solving (25).

**Theorem 1.** If \( c \leq c^* \) and \( X'' = (x''_{ij})_{c \times c^*} \) is the optimal solution of (25), then \( X' = (x'_{ij})_{c \times c^*} \) is the optimal solution of (22), where \( x'_{ij} = x''_{ij} \), \( i = 1, 2, \ldots, c, j = 1, 2, \ldots, c^* \).

**Proof.** Obvious if \( c = c^* \).

Otherwise, since \( X'' \) is the optimal solution of problem (25), one has: \( \forall \) matrix \( X \) of size \( c \times c^* \), if \( X \) is the feasible solution of problem (22), one can create \( X' \) as follows:

\[
    x'_{ij} = \begin{cases} 
        x_{ij}, & \text{if } i \leq c, \\
        0, & \text{if } i > c, \text{ and } \sum_{j=1}^{c} x_{ij} = 1 \\
        1, & \text{if } i > c, \text{ and } \sum_{j=1}^{c-1} x_{ij} = 0, j = 1, 2, \ldots, c^* - 1 \\
        0, & \text{if } i > c, \text{ and } \sum_{j=1}^{c} x_{ij} = 0, \sum_{i=1}^{c-1} x_{ii} = 1.
    \end{cases}
\]

\( X' \) is obviously the feasible solution of problem (25), hence

\[
    \sum_{i=1}^{c} \sum_{j=1}^{c'} l'_{ij} x'_{ij} \leq \sum_{i=1}^{c} \sum_{j=1}^{c'} l'_{ij} x''_{ij}.
\]

In other words

\[
    \sum_{i=1}^{c} \sum_{j=1}^{c'} l'_{ij} x'_{ij} \leq \sum_{i=1}^{c} \sum_{j=1}^{c'} l'_{ij} x''_{ij},
\]

based on the definition of \( l'_{ij} \), meaning that \( X' \) is the optimal solution of (22). \( \square \)

For example, if the ground-truth labels and the computed labels are \( [1, 1, 1, 1, 4, 4, 2, 2, 3, 3] \) and \( [3, 3, 3, 3, 3, 3, 3, 1, 1, 2] \), respectively, then \( l_{11} = 6, l_{12} = 3 = 1 - 2, l_{13} = 3 - 1 = 2, l_{14} = 4, l_{21} = 5, l_{22} = 3, l_{23} = 1, l_{24} = 3, l_{31} = 7 - 4 = 3, l_{32} = 8 - 1 = 7, l_{33} = 3, l_{34} = 7 - 2 = 5 \), and the resulting map is: \( 1 \mapsto 2, 2 \mapsto 3, 3 \mapsto 1 \), making it possible to use indices in the field of supervised learning, such as kappa index and F-score for evaluation of community detection methods. It is worth mentioning that the method also applies to overlapping community structures detection. For example, if the ground truth labels are \( [1, 1, 1, 1, (1, 2), (1, 2), 2, 2, 3, 3] \), where nodes 5 and 6 have multiple labels, and the computed labels are \( [3, 3, 3, 3, 1, (1, 3), 1, 1, 2, 2] \), then the costs of mapping are: \( l_{11} = 8 - 2 = 6, l_{12} = 4 - 4 = 0, l_{13} = 6 - 0 = 6, l_{21} = 8, \ldots \).

### Table 2

| Networks       | Parameters | Num vertices | Num Edges | Num communities |
|----------------|------------|--------------|-----------|-----------------|
| Email-Eu-core  | 1,005      | 25,571       | 42        |                 |
| Orkut          | 1,247      | 15,511       | 43        |                 |
| Political blogs| 1,490      | 19,025       | 2         |                 |
| Amazon         | 1,644      | 5,022        | 131       |                 |
| Cora           | 2,708      | 5,429        | 6         |                 |
| Youtube        | 3,088      | 6,695        | 595       |                 |
| Citeseer       | 3,312      | 4,732        | 6         |                 |
| DBLP           | 5,831      | 18,733       | 676       |                 |
| Livejournal    | 17,969     | 434,075      | 668       |                 |
l_{22} = 6, \ l_{23} = 2 - 2 = 0, \ l_{31} = 6 - 5 = 1, \ l_{32} = 8 - 1 = 7, \ l_{33} = 7, \text{ and the resulting mapping is } 1 \leftrightarrow 2, 2 \leftrightarrow 3, 3 \leftrightarrow 1.

However, how to evaluate overlapping community detection methods is not the focus of this paper, and we will leave it for further study.

Time complexity of obtaining the cost matrix in the integer linear programming is \(O(\frac{cc}{c})\), where \(c\) and \(c'\) are the numbers of computed communities and the ground-truth ones, respectively. The integer linear programming model can be solved by Hungarian method (also known as Kuhn-Munkres algorithm), whose time complexity is \(O(c^2 c')\) with \(c \leq c'\).

Actually, the numbers of communities do not increase proportionally with growing network sizes. For example, the numbers of communities in LFR networks used in the paper are about \(0.004n\), where \(n\) is the network size, as is shown in Fig. 10; the numbers of communities in real networks used in our paper and Ref. [28] are also very small compared with the network sizes, as is summarized in Table 2. Note that the networks used in Ref. [28] are obtained after preprocessing the data from the Stanford Large Network Dataset Collection (SNAP)\(^2\) and are not released. In addition, effective algorithms for large scale assignment problems were proposed [29]. Hence, the proposed method can be applied to large scale networks.

4 Experimental Results

In this section, we will first demonstrate the performance advantages of using kappa and F-score compared with NMI, then systematically evaluate several popular community detection methods on both synthetic benchmark networks and real networks using our method.

1. http://zafar.cc/2017/7/19/hungarian-algorithm/
2. http://snap.stanford.edu/

4.1 Description of Synthetic Networks

The Lancichinetti-Fortunato-Radicchi (LFR) benchmark network model [30] is the most popular one used for evaluation of community detection algorithms because it is a good proxy for real networks. The LFR network is generated by several parameters, including number of total nodes \(n\), average degree \(k\), maximal degree \(k_{max}\), minimal community size \(s_{min}\) and maximal community size \(s_{max}\), exponent of power-law distributions of nodes degree \(\gamma\) and community size \(\beta\), respectively, and a mixing ratio of external links \(\mu\). Specifically, the higher the mixing parameter \(\mu\) of a network is, the more difficult it is to detect the community.

4.2 Description of Real Networks

Four real networks, including email-Eu-core network [31], [32], Political blogs network [33], CiteSeer network [34], [35] and Cora network [34], [35] are used for our experiments. Details are described below:

1) email-Eu-core network: This data set is the e-mail network of members in a European research institute. The members are represented as nodes, and a directed edge \((i, j)\) means member \(i\) sent to member \(j\) at least one email. There are 1,005 nodes and 25,571 edges. Each member belongs to one of 42 departments at the institute.

2) Political blogs network: This data set is a network of hyperlinks between weblogs on US politics. There are 1,490 nodes and 19,025 edges. Each node is either conservative or liberal.

3) CiteSeer network: This data set is a citation network. The nodes of the network are publications and the directed edges denote citations. There are 3,312 nodes and 4,732 edges. Each publication belongs to one of six classes, including Agents, Artificial Intelligence,
4) Cora network: This data set is also a citation network. There are 2,708 scientific publications and 5,429 edges. Each publication belongs to one of seven classes, including Case-based Reasoning, Genetic Algorithms, Neural Networks, Probabilistic Methods, Reinforcement Learning, Rule Learning, and Theory.

4.3 Advantages of Kappa Index Compared with NMI and rNMI

As previously mentioned, NMI has finite size effect, and rNMI has reverse finite size effect. Finite size effect can be demonstrated by calculating NMI between two independent random partitions $A$ and $B$, and reverse finite size effect can be demonstrated by calculating rNMI of $A$ against itself. The results and the advantages of kappa index are demonstrated in Figs. 3a, 3b and 4a, 4b. One can observe that: 1) NMI between two independent random partitions is much larger than zero, and is farther away from zero for partitions with larger number of communities due to finite size effect. This phenomenon is more evident in smaller networks. 2) rNMI of a partition against itself is not one, and is farther away from one for partitions with larger number of communities due to reverse finite size effect. This phenomenon is also more evident in smaller networks. 3) The values of kappa are not strictly constant in Fig. 3b, especially for large numbers of communities.

Actually, the index value between two random and independent partitions $A$ and $B$ depends on the network size $n$.
and the community number $c$, and should not be zero, whatever the index is. For example, there are three samples with two clusters, possible partitions include $P = \{(1, 2), (1, 1, 2), (1, 2, 1), (2, 1, 2), (2, 2, 1)\}$, and none of the values between $A$ and $B$ that are randomly selected from the set $P$ are zero. In extreme circumstances where $c = n$, the reasonable value should be 1.

Fig. 3c calculates the values of kappa, NMI, rNMI and cNMI between two random and independent partitions $A$ and $B$ of networks with $n = 2000$. Figs. 3a and 3b are parts of Fig. 3c. The results are averages over ten trials. One can observe that: 1) The values of kappa increase near linearly with growing community numbers. 2) The gap between NMI and kappa is vast due to finite size effect of NMI. 3) The values of rNMI and cNMI are always zero, and are apparently unreasonable.

In addition, we also calculate the values of kappa, NMI, rNMI and cNMI between two random and independent partitions $A$ and $B$ of networks with $n = 2000$, where $A$ has 200 communities and $B$ has a varying number of communities. The results are averages of ten trials and are shown in Fig. 5. One can observe that: 1) NMI increases gradually with growing community numbers. 2) Kappa achieves its maximum value at $c = 200$, where $B$ is most compatible with $A$ since they have the same community number. 3) The values of rNMI and cNMI are still zero, and are apparently unreasonable.

To further observe the behaviors of kappa and NMI, we calculate the values of partition $A$ versus $B$, where $A$ is (a) that in Fig. 3 with network size $n = 2000$ and community number $c = 200$; (b) that in Fig. 3 with network size $n = 4000$ and community number $c = 200$, and $B$ is obtained through randomly flipping the labels of a fraction of $A$, as is shown in Fig. 6. From this figure, one can see that: 1) The gap between NMI and the theoretical line is vast. 2) Generally, kappa is well consistent with theoretical values, except when the flipping fraction is considerable large.

In summary, 1) The results of the proposed method are more reasonable. 2) The behaviors of NMI, rNMI and cNMI cannot meet our expectations, and are not suitable for evaluating community detection methods. More discussions about cNMI can be found in the next section.

4.4 Advantages of Kappa Index Compared with cNMI

Actually, the discussions on finite size effect and reverse finite size effect are not new, and cNMI has been proposed to handle these issues, as we can see from Figs. 3c and 4c. Except for the unreasonable results in Fig. 3c, there are other problems with cNMI, i.e., violating the proportionality assumption, as is shown in Fig. 7. We calculate cNMI and kappa index between two partitions $A$ and $B$, where $A$ contains the ground-truth labels of LFR networks, and $B$ is obtained through randomly flipping the community labels of a fraction of nodes in $A$. One can conclude that: 1) The slope of cNMI keeps decreasing as the similarity of $A$ and $B$ decreases, which is consistent with the theoretical line. 2) Kappa is well consistent with theoretical values, except when the flipping fraction is considerably large.
is decreased, indicating that cNMI is not proportionally decreased. 2) The line of kappa index is straight, satisfying proportionality assumption.

In addition, all of NMI, rNMI and cNMI suffer from the problem of ignoring importance of small communities.

In summary, kappa index is advantageous over NMI, rNMI and cNMI.

4.5 Advantages of F-Score Compared with NMI

Even if ignoring the above discussions, NMI-type metrics cannot be used to evaluate the quality of a single community of interest. For example, given 103 nodes with ground-truth labels $W$, and the computed labels $X, Y$ and $Z$, as is shown in Fig. 8, the results of NMI, rNMI, cNMI and kappa index are similar, i.e., $\text{NMI}(W, X) > \text{NMI}(W, Y) > \text{NMI}(W, Z)$, $\text{rNMI}(W, X) > \text{rNMI}(W, Y) > \text{rNMI}(W, Z)$, $\text{cNMI}(W, X) > \text{cNMI}(W, Y) > \text{cNMI}(W, Z)$, $\kappa(W, X) > \kappa(W, Y) > \kappa(W, Z)$. These results cannot reveal the methods’ ability to detect a specific community of interest. Thus we here use F-score to evaluate the two communities separately. The results are $F(W, X) > F(W, Y) > F(W, Z)$ for class orange, and $F(W, Y) > F(W, X) > F(W, Z)$ for class green.

4.6 Evaluation of Community Structures Detection Methods Using Kappa Index

In this section, we use normalized mutual information and kappa index to rank three methods, InfoMap [36], Louvain [37] and Modularity BP [38] on LFR benchmark networks. The parameters of networks are: average degree $k = 8$, mixing parameter $\mu = 0.45$, exponent of degree distribution $\gamma = 2$, exponent of community size distribution $\beta = 1$, maximum degree is 50, and community sizes range from 200 to 400.

![Fig. 11. Distributions of community sizes detected by three methods, InfoMap, Louvain, and Modularity BP on LFR benchmark with network size $n = 29000$. The parameters of the network are: Average degree $k = 8$, mixing parameter $\mu = 0.45$, exponent of degree distribution $\gamma = 2$, exponent of community size distribution $\beta = 1$, maximum degree is 50, and community sizes range from 200 to 400.](image-url)
In addition, we have conducted more experiments on real networks, and the numerical results of NMI and kappa are given in Fig. 12, from which one can observe that: 1) The order of the three methods based on kappa is different compared with that based on NMI. 2) The order of InfoMap and ModBP is reversed, and overall, Louvain does well in the ranking. 3) The values of ModBP are better than InfoMap on real networks, and this is consistent with the results on synthetic networks with small network sizes $n$.

5 Conclusion

In this paper, we first review the metrics for evaluation of clustering methods, especially the ones for community structures detection. In summary, NMI-type metrics have systematic bias, ignore importance of small communities, and cannot evaluate the methods’ ability to detect a single specific community of interest. To overcome these challenges, our proposed method maps the computed community labels to the ground-truth ones through integer linear programming, and then uses kappa index and F-score as evaluation metrics, instead of NMI. Systematic experimental results demonstrate the advantages of the method. There are several interesting problems for future work: Combining internal indices with external ones for performance evaluation in unsupervised learning and supervised learning; Extending the method to evaluate general clustering results and to evaluate overlapping and hierarchical community structures, etc.

Acknowledgments

The authors highly appreciate the anonymous reviewers for their insightful comments and suggestions on their manuscript. This work was supported by the Program for Innovation Research in the Central University of Finance and Economics and the disciplinary funding of the Central University of Finance and Economics.

References

[1] S. Fortunato, “Community detection in graphs,” Physics Rep., vol. 486, no. 3–5, pp. 75–174, 2010.
[2] H. Schütze, C. D. Manning, and P. Raghavan, Introduction to Information Retrieval, vol. 39. Cambridge, U.K.: Cambridge Univ. Press, 2008.
[21] J. Yang and J. Leskovec, “Defining and evaluating network communities based on ground-truth,” Knowl. Inf. Syst., vol. 42, no. 1, pp. 181–213, 2015.

[22] G. Rossetti, L. Pappalardo, and S. Rinzivillo, “A novel approach to evaluate community detection algorithms on ground truth,” in Complex Networks VII. Berlin, Germany: Springer, 2016, pp. 133–144.

[23] A. J. Gates and Y.-Y. Ahn, “The impact of random models on clustering similarity,” J. Mach. Learn. Res., vol. 18, no. 1, pp. 3049–3076, 2017.

[24] C. E. Metz, “Basic principles of ROC analysis,” Seminars Nucl. Med., vol. 8, no. 4, pp. 283–298, 1978.

[25] D. M. Powers, “Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation,” J. Mach. Learn. Technol., vol. 2, no. 1, pp. 37–63, 2011.

[26] G. Galton, Finger Prints. New York, NY, USA: Macmillan and Company, 1892.

[27] C. Van Rijsbergen, Inf. Retrieval. Butterworths, 1979. [Online]. Available: https://books.google.com/books?id=t-pTAAAAMAAJ

[28] S. Harenberg, G. Bello, L. Gjeltema, S. Ranshous, J. Harlalka, R. Seay, K. Padmanabhan, and N. Samatova, “Community detection in large-scale networks: A survey and empirical evaluation,” Wiley Interdisciplinary Rev.: Comput. Statist., vol. 6, no. 6, pp. 426–439, 2014.

[29] H. Cui, J. Zhang, C. Cui, and Q. Chen, “Solving large-scale assignment problems by Kuhn-Munkres algorithm,” 2nd Int. Conf. Advances Mech. Eng. Ind. Inform. (AMEII 2016), Jan. 2016, doi: 10.2991/ameii-16.2016.160

[30] A. Lancichinetti, S. Fortunato, and F. Radicchi, “Benchmark graphs for testing community detection algorithms,” Phys. Rev. E, vol. 78, no. 4, 2008, Art. no. 046110.

[31] H. Yin, A. R. Benson, J. Leskovec, and D. F. Gleich, “Local higher-order graph clustering,” in Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2017, pp. 555–564.

[32] J. Leskovec, J. Kleinberg, and C. Faloutsos, “Graph evolution: Densification and shrinking diameters,” ACM Trans. Knowl. Discovery Data, vol. 1, no. 1, 2007, Art. no. 2.

[33] L. A. Adamic and N. Glance, “The political blogosphere and the 2004 U.S. election: Divided they blog,” in Proc. 3rd Int. Workshop Link Discovery, 2005, pp. 36–43.

[34] Q. Lu and L. Getoor, “Link-based classification,” in Proc. 20th Int. Conf. Mach. Learn., 2003, pp. 496–503.

[35] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, “Collective classification in network data,” AI Mag., vol. 29, no. 3, 2008, Art. no. 93.

[36] M. Rosvall and C. Bergstrom, “Maps of random walks on complex networks reveal community structure,” Proc. Nat. Acad. Sci. USA, vol. 105, no. 4, pp. 1118–1123, 2008.

[37] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, “Fast unfolding of communities in large networks,” J. Statistical Mech.: Theory Experiment, vol. 2008, no. 10, 2008, Art. no. P10008.

[38] P. Zhang and C. Moore, “Scalable detection of statistically significant communities and hierarchies, using message passing for modularity,” Proc. Nat. Academy Sci. United States America, vol. 111, no. 51, pp. 18144–18149, 2014.

Xin Liu received the bachelor’s degree in statistics from Northeastern University at Qinhuangdao. She is working toward the post-graduate degree majoring in applied statistics at the Central University of Finance and Economics, P.R. China. Her interests are within the areas of data mining and complex social network analysis.

Hui-Min Cheng received the master’s degree from the Central University of Finance and Economics, P.R. China, in 2017. She is working toward the PhD degree in the School of Statistics and Mathematics, Central University of Finance and Economics, P.R. China. Her current research interest include complex social network analysis.

Zhong-Yuan Zhang received the PhD degree from the Academy of Mathematics and Systems Science, Chinese Academy of Sciences, in 2008. He is a full professor with the School of Statistics and Mathematics, Central University of Finance and Economics, P.R. China. His research interests include data mining and complex social network analysis.