Community Detection in Graphs through Correlation

Lian Duan
New Jersey Institute of Technology
Newark, NJ 07102, USA
lian.duan@njit.edu

W. Nick Street
University of Iowa
Iowa City, IA 52242, USA
nick-street@uiowa.edu

Yanchi Liu
New Jersey Institute of Technology
Newark, NJ 07102, USA
yl473@njit.edu

Haibing Lu
Santa Clara University
Santa Clara, CA 95053, USA
hlu@scu.edu

ABSTRACT

Community detection is an important task for social networks, which helps us understand the functional modules on the whole network. Among different community detection methods based on graph structures, modularity-based methods are very popular recently, but suffer a well-known resolution limit problem. This paper connects modularity-based methods with correlation analysis by subtly reformatting their math formulas and investigates how to fully make use of correlation analysis to change the objective function of modularity-based methods, which provides a more natural and effective way to solve the resolution limit problem. In addition, a novel theoretical analysis on the upper bound of different objective functions helps us understand their bias to different community sizes, and experiments are conducted on both real life and simulated data to validate our findings.

Categories and Subject Descriptors

H.2.8 [**DATABASE MANAGEMENT**]: Database Applications – Data Mining

Keywords

community detection; correlation analysis; modularity; leverage; likelihood ratio

1. INTRODUCTION

The modern science of graphs has significantly helped us understand complex systems. One important feature of graphs is community structure where nodes in the same community have a higher chance to be connected to each other than that of nodes in different communities. Such communities can be considered as relatively independent components and play a role in the system. Community detection, which attempts to identify the modules by using the graph topology, has a long history in sociology, biology, and computer science where systems are often represented as graphs. The first research on community detection was made by Weiss and Jacobson [39] to study the working relationships between members of a government agency. Nowadays, there are many different community detection methods, such as spectral-based methods [24], density-based methods [26], modularity-based methods [9, 35], divisive methods [19], statistical-inference-based methods [28], etc. Generally speaking, their progress can be categorized into the following three procedures: (1) feature selection (2) objective function (3) search procedure.

Feature selection selects relevant features, and removes irrelevant noisy information. Spectral-based methods [24] use the eigenvectors of the adjacency matrix for community detection. The Laplacian is by far the most used matrix in spectral-based methods. Though no unique matrix is exactly called the graph Laplacian [24], one commonly used Laplacian is calculated as follows. Given the adjacency matrix W of the graph G, we calculate the matrix D where the diagonal element d_{ii} is equal to $\sum_{j=1}^{n}(w_{ij})$ and non-diagonal elements are 0. The Laplacian matrix L is equal to $D - W$. We choose k eigenvectors corresponding to the k smallest eigenvalues to transform the original adjacency matrix, and then apply the traditional clustering methods like K-mean [25] on the transformed matrix. The spectral-based methods are popular because the change of representation induced by eigenvectors makes the community structure more obvious.

Objective function is the function to express our goal in mathematical terms. No matter how the community is defined, the commonly accepted goal for community detection can be boiled down to two objectives: (1) More connections are inside each community. (2) Fewer connections are across different communities. Since there are two objectives, people have proposed many different methods to strike a different balance between them. There are various kinds of objective functions for community detection [1]. For example, density-based methods [26] try to find the communities within which nodes are tightly connected with each other. We define the internal-community density $\delta_{int}(S)$ of the subgraph S as the ratio of the number of internal edges of S to the number of all possible internal edges, i.e. $\delta_{int}(S) = \frac{k_{int}(S)}{n_s*(n_s-1)/2}$ where $k_{int}(S)$ is the number of internal edges of S. Similarly, the external-community density $\delta_{ext}(S) = \frac{k_{ext}(S)}{n_s*(n-n_s)}$. For S to be a community, we expect large $\delta_{int}(S)$ and small $\delta_{ext}(S)$.

KDD'14, August 24–27, 2014, New York, NY, USA.
Copyright 2014 ACM 978-1-4503-2956-9/14/08 ...$15.00.
http://dx.doi.org/10.1145/2623330.2623629 .

Searching for the best tradeoff between $\delta_{int}(S)$ and $\delta_{ext}(S)$ is the goal of density-based methods. A simple way of doing that is to maximize the sum of the difference $\delta_{int}(S)-\delta_{ext}(S)$ [26]. Another example is modularity-based methods which search the partition that maximizes the modularity function [9]. Although the modularity function is originally introduced as a stopping criterion for hierarchical methods [19], it has rapidly become an essential element of many clustering methods by searching the partitions that maximize it.

Search procedure is the way we search the optimal solution according to the objective function. However, for the most objective functions, finding the optimal value is very slow. For example, it has been proved that modularity optimization is an NP-complete problem [7]. Therefore, many different heuristic search algorithms, including greedy search [9, 38], simulated annealing [27], extremal optimization [6, 14], and genetic algorithms [31], are used.

According to different survey studies [16, 40] on community detection, modularity-based methods are considered as one classical type of methods. Despite the vast amount of expert endeavor spent on different optimization techniques to maximize the modularity function, there is little analysis on the modularity function itself, and the most analysis on the modularity function is related to the calculation of the edge probability [17] or the extension to complicated networks [3]. In addition, modularity-based methods suffer the well-known resolution limit problem [22]. This problem also cannot be solved by multi-resolution methods [33] as two concurrent biases, the tendency to merge small communities and to split large communities, are introduced.

The modularity function [9] searches for partitions that the actual number of edges is larger than the expected number of edges inside communities under the assumption of random partition, while correlation analysis on itemset mining [13, 18, 36] searches for itemsets that occur more than the expected occurrence if items are independent from each other. Since both modularity on community detection and correlation analysis on itemset mining search for patterns that actually happens more than what is expected under the assumption of independence, this paper builds the connection between the modularity function and correlation measures, and changes the modularity function from the correlation perspective, through which the resolution limit problem of the original modularity function can be solved and performance can be improved. The focus in this paper is on the improvements over the modularity function from the correlation perspective. In order to have a fair comparison within the framework of correlation, the original modularity function is treated as a baseline for the performance evaluation rather than other commonly used objective functions such as density [26] and conductance [21]. Although the modularity function can be combined with fuzzy logic for overlapping community detection [29], we limit our performance evaluation to non-overlapping community detection in this paper. The rest of the paper is organized as follows. Section 2 introduces the basic notation of correlation analysis and modularity-based community detection. We build the connection between correlation analysis and modularity-based community detection by subtly reformatting their math formulas, and introduce a novel theoretical analysis on the bias of different correlation measures by analyzing their upper bounds in Section 3. Experiments on both real life and simulated datasets are conducted to test different methods in Section 4. Finally, we draw a conclusion and point out future directions in Section 5.

2. BASIC NOTATION

Since we try to improve modularity-based methods from the correlation analysis perspective, the basic concepts of correlation and modularity-based community detection will be introduced first before connecting them.

2.1 Correlation Analysis

Most of the correlation analysis in the data mining area is conducted on the context of itemset mining. Therefore, we follow the routine to introduce the basic concept of correlation. Given an itemset $S = \{I_1, I_2, ..., I_m\}$ with m items in a dataset with sample size n, the true probability is $tp = P(S)$, the expected probability under the assumption of independence among items is $ep = \prod_{i=1}^{m} P(I_i)$. Many functions have been proposed to measure correlation [12, 13, 18, 36]. Here, we only introduce four typical correlation measures, Simplified χ^2, Probability Ratio, Leverage, and Likelihood Ratio, which are derived from the simple statistical theory and enough for generality.

2.1.1 Simplifie χ^2

The χ^2 test is arguably the most popular statistical check for correlation, and is specifically designed for use with categorical data. It is calculated as $\chi^2 = \sum_i \sum_j (r_{ij} - E(r_{ij}))^2/E(r_{ij})$. If an itemset contains m items, 2^m cells in the contingency table must be considered for the above χ^2 statistic. The computation of the statistic itself is intractable for high-dimensional data. However, we can still use the basic idea behind χ^2 to create Simplified χ^2 [20]: $\chi'^2 = (r-E(r))^2/E(r)$, i.e., $n \cdot (tp - ep)^2/ep$, where the cell r corresponds to the exact itemset S and n is the total number of records. Since Simplified χ^2 is more computationally desirable, we only discuss the properties and experimental results of Simplified χ^2. The value of Simplified χ^2 is always larger than 0 and cannot differentiate positive from negative correlation. Therefore, we take advantage of the comparison between tp and ep. If $tp > ep$, it is a positive correlation. Then Simplified χ^2 is equal to $n \cdot (tp-ep)^2/ep$. If $tp < ep$, it is a negative correlation. Then Simplified χ^2 is equal to $-n \cdot (tp-ep)^2/ep$. This transformed Simplified χ^2 is mathematically favorable. Larger positive numbers indicate stronger positive correlation, 0 indicates no correlation, and larger (in magnitude) negative numbers indicate stronger negative correlation.

2.1.2 Probability Ratio/Lift/Interest Factor

Probability Ratio (also known as Lift or Interest Factor) [8] is the ratio of an itemset's true probability to its expected probability under the assumption of independence. It is calculated as follows: $ProbabilityRatio(S) = tp/ep$. This measure is straightforward and means how many times the itemset S happens more than expected. However, this measure might not be a good correlation measure to use. The problem is that it favors the itemsets containing a large number of items rather than significant trends in the data.

2.1.3 Leverage

An itemset S with higher occurrence and low Probability Ratio may be more interesting than an alternative itemset S' with low occurrence and high Probability Ratio. Introduced by Piatesky-Shapiro [30], $Leverage(S) = tp - ep$. It measures the difference between the true probability of an itemset S and its expected probability if all the items in S are independent from each other. Since ep is always no less than 0, $Leverage(S)$ can never be bigger than tp. Therefore, Leverage is biased to high-occurrence itemsets.

2.1.4 Likelihood Ratio

Likelihood Ratio is similar to a statistical test based on the loglikelihood ratio described by Dunning [15]. We take the ratio of the likelihood under our hypothesis of independence to the likelihood of the best "explanation" overall. To apply the likelihood ratio test as a correlation measure, we use the binomial distribution $Pr(p, o, n) = \left(\begin{array}{c} n \\ o \end{array}\right) p^o(1-p)^{(n-o)}$, where p is the probability of a given itemset S, o is the occurrence of the itemset S, and n is the total number of transactions. Given our assumption of independence of all items, we predict that each trial has a probability of success ep. Therefore, the chance for us to observe o out n transactions contain S is $Pr(ep, o, n)$ under the assumption of independence. However, the best possible explanation for the single trial probability is tp instead of ep according to the observed data. In order to measure to what extent our assumption of item independence was violated in practice, we comparing the null hypothesis of independence with the best possible explanation. Formally, the Likelihood Ratio in this case is $LikelihoodRatio(S) = Pr(tp, o, n)/Pr(ep, o, n)$. The Likelihood Ratio strikes a balance between the Probability Ratio and the actual occurrence o. It favors itemsets with both high Probability Ratio and high occurrence. For the itemsets containing a small number of items, their occurrence tends to be high, but the Probability Ratio tends to be low, while, for the itemsets containing a large number of items, their Probability Ratio tends to be high, but the actual occurrence tends to be low. Likelihood Ratio favors middle-sized itemsets which can strike a balance between the Probability Ratio and the actual occurrence. The numerator of the Likelihood Ratio is the maximal likelihood of the real situation, so the Likelihood Ratio is always larger than 1 and cannot differentiate positive from negative correlation. Therefore, we conduct the similar transformation we do for Simplified χ^2 by comparing tp with ep.

2.2 Modularity-based Community Detection

The modularity function has several variants, but these variants share the same idea. Without the loss of generality, we introduce the original modularity-based method [9]. Given a graph with n nodes and m links represented by the adjacency matrix W, the expected number of edges falling between two nodes i and j is $k_i \cdot k_j/(2m)$ under the assumption of independence where k_i is the degree of node i. The modularity Q is calculated as $\frac{1}{2m}\sum_{ij}(w_{ij} - \frac{k_i \cdot k_j}{2m}) \cdot \delta(v_i, v_j)$. It is the sum of the difference between the actual number of edges and the expected number of edges over all the pairs of nodes in the same community. $\delta(v_i, v_j)$ is the Kronecker delta function whose value is equal to 1 if v_i and v_j are in the same community and 0 otherwise. Initially, each node is the only member of its own community. The original algorithm iteratively joins the two communities that increase the modularity most in the current round. The original algorithm will stop if the best merge cannot further increase modularity.

3. CORRELATION ANALYSIS AND MODULARITY

3.1 Connecting Modularity-based Community Detection with Correlation Analysis

In this section, we subtly transform the modularity function and connect it with correlation measures. Given a partition with l groups $\{G_1, G_2, ..., G_l\}$ for the graph G with n nodes and m links, the modularity Q is $\frac{1}{2m}\sum_{ij}(w_{ij} - \frac{k_i \cdot k_j}{2m}) \cdot \delta(v_i, v_j)$. For the node v_q in the group G_p, k_q^{int} is the number of the nodes in the group G_p that connect to v_q. The partial modularity Q_p, which all the nodes in the group G_p contribute to the overall modularity function, is $\frac{1}{2m}\sum_{i \in G_p, j \in G}(w_{ij} - \frac{k_i \cdot k_j}{2m}) \cdot \delta(v_i, v_j)$.

Therefore,

$$\begin{aligned}
Q_p &= \sum_{i \in G_p, j \in G} \frac{w_{ij} \cdot \delta(v_i, v_j)}{2m} - \sum_{i \in G_p, j \in G} \frac{k_i \cdot k_j \cdot \delta(v_i, v_j)}{(2m)^2} \\
&= \sum_{i \in G_p} \frac{\sum_{j \in G} w_{ij} \cdot \delta(v_i, v_j)}{2m} - \sum_{i \in G_p} \frac{k_i \cdot \sum_{j \in G} k_j \cdot \delta(v_i, v_j)}{(2m)^2} \\
&= \sum_{i \in G_p} \frac{k_i^{int}}{2m} - \sum_{i \in G_p} \frac{k_i \cdot \sum_{j \in G_p} k_j}{(2m)^2} \\
&= \frac{\sum_{i \in G_p} k_i^{int}}{2m} - \frac{\sum_{i \in G_p} k_i}{2m} \cdot \frac{\sum_{j \in G_p} k_j}{2m}.
\end{aligned}$$

It is easy to calculate that the total number of links inside G_p is $\sum_{i \in G_p} k_i^{int}/2$ and the total number of links in the graph G is m. If we randomly select a link from the graph G, the probability of the link inside G_p is $\frac{\sum_{i \in G_p} k_i^{int}/2}{m}$. Similarly, the probability of the link with at least one end inside G_p is $\frac{\sum_{i \in G_p} k_i}{2m}$ when we randomly select a link from the graph G. If the partition with l groups $\{G_1, G_2, ..., G_l\}$ for the graph G is totally random, the probability of the link with the other end inside G_p from the links with one end already inside G_p is $\frac{\sum_{i \in G_p} k_i}{2m} \cdot \frac{\sum_{j \in G_p} k_j}{2m}$. Therefore, given a partition with l groups $\{G_1, G_2, ..., G_l\}$ for the graph G, if we randomly select a link from the graph G, the true probability of the link being inside G_p, tp, is $\frac{\sum_{i \in G_p} k_i^{int}}{2m}$, and the expected probability of the link being inside G_p under the assumption of independent partition, ep, is $\frac{\sum_{i \in G_p} k_i}{2m} \cdot \frac{\sum_{j \in G_p} k_j}{2m}$. Therefore, the partial modularity function Q_p can be rewritten as: $Q_p = tp - ep$. By comparing the correlation measure $Leverage(S) = tp - ep$, we can see the modularity function shares the same idea with the correlation measure Leverage. Since the other correlation measures are also functions of tp and ep, we can change the partial modularity function Q_p by using the formula of other correlation measures. In the rest of the paper, instead of using the term modularity, we use Simplified χ^2, Probability Ratio, Leverage, and Likelihood Ratio referring to the corresponding changed partial modularity function Q_p, and Leverage is the original modularity community detection method.

3.2 Upper Bound Analysis

The performance differences among different correlation measures have been recognized since 2004 by two very influential papers [18, 36] in the data mining area. They categorized measures according to their different property satisfaction. By categorizing measures, users only need to check the performance of the typical measure in each category instead of all the possible measures. However, two measures can still generate different results even if they satisfy the same set of properties. Instead, one recent paper [37] categorized measures directly according to their final result similarity. No matter how categorizing measures, the fundamental question is still not answered. If there is a difference between results of two measures, what is the difference? In this section, we provide a novel way to understand the performance difference by analyzing the upper bound of different partial modularity functions Q_p inferred from different correlation measures. Simplified χ^2, Probability Ratio, Leverage, and Likelihood Ratio all satisfy the third correlation property proposed by Piatesky-Shapiro [30]: The correlation Measure M monotonically decreases with the increase of *ep* when *tp* remains the same. According to the above correlation property, the measures reach their upper bound when *tp* is fixed and *ep* reaches its lower bound.

THEOREM 1. *Simplified χ^2, Probability Ratio, Leverage, and Likelihood Ratio monotonically decreases with the increase of ep when tp remains the same.*

PROOF. **Simplified χ^2:** When $tp \geq ep$, $\chi^2 = n \cdot (tp - ep)^2/ep$. If we consider Simplified χ^2 as a function of *ep*, then $\chi^{2'} = n \cdot (ep^2 - tp^2)/ep^2$. Since $0 \leq ep \leq tp \leq 1$, $ep^2 \leq tp^2$. Therefore, $\chi^{2'}(S) \leq 0$. Similarly, when $tp < ep$, $\chi^2 = -n \cdot (tp - ep)^2/ep$ and $\chi^{2'} = -n \cdot (ep^2 - tp^2)/ep^2 < 0$. In all, Simplified χ^2 decreases with the increase of *ep*.

Probability Ratio: When *tp* is fixed, Probability Ratio, *tp*/*ep*, decreases with the increase of *ep*.

Leverage: When *tp* is fixed, Leverage, $tp - ep$, decreases with the increase of *ep*.

Likelihood Ratio: When $tp > ep$,

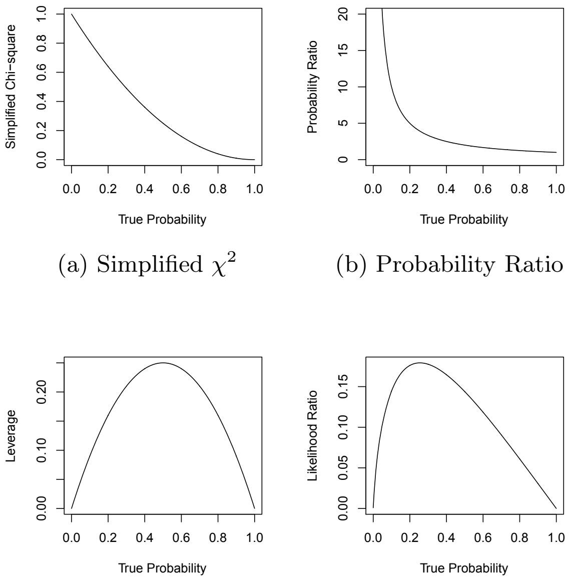
$$\begin{aligned} ln(LR(S)) &= n \cdot tp \cdot (ln(tp) - ln(ep)) \\ &\quad + n \cdot (1 - tp) \cdot (ln(1 - tp) - ln(1 - ep)) \\ &= n \cdot tp \cdot ln(tp) - n \cdot tp \cdot ln(ep) \\ &\quad + n \cdot ln(1 - tp) - n \cdot ln(1 - ep) \\ &\quad - n \cdot tp \cdot ln(1 - tp) + n \cdot tp \cdot ln(1 - ep) \\ &= n \cdot tp \cdot ln\frac{tp}{1 - tp} + n \cdot ln(1 - tp) \\ &\quad - n \cdot ln(1 - ep) + n \cdot tp \cdot ln\frac{1 - ep}{ep}. \end{aligned}$$

If we consider $ln(LR(S))$ as a function of *ep*, then

$$\begin{aligned} ln(LR(S))' &= \frac{n}{1 - ep} - \frac{n \cdot tp}{(1 - ep) \cdot ep} \\ &= \frac{n \cdot (ep - tp)}{(1 - ep) \cdot ep}. \end{aligned}$$

Since $tp > ep$, then $ln(LR(S))' < 0$. In other words, Likelihood Ratio decreases with the increase of *ep* when $tp > ep$. Similarly, when $tp < ep$, we can prove Likelihood Ratio decreases with the increase of *ep*. In all, Likelihood Ratio decreases with the increase of *ep*.

□



(a) Simplified χ^2 (b) Probability Ratio

(c) Leverage (d) Likelihood Ratio

Figure 1: Upper bounds of different measures for a single community

Given a partition with l groups $\{G_1, G_2, ..., G_l\}$ for the graph G, the true probability of a link being inside G_p, *tp*, is $\frac{\sum_{i \in G_p} k_i^{int}}{2m}$, and the expected probability, *ep*, is $\frac{\sum_{i \in G_p} k_i}{2m} \cdot \frac{\sum_{j \in G_p} k_j}{2m}$. If $\sum_{i \in G_p} k_i^{int}$ is fixed, the lowest possible value for $\sum_{i \in G_p} k_i$ is $\sum_{i \in G_p} k_i^{int}$ because $k_i^{int} \leq k_i$. In other words, when *tp* for the group G_p is fixed, the lowest possible value for *ep* is tp^2. When $ep = tp^2$, the measures reach their upper bound. Figure 1 shows the upper bounds of the various measures with respect to different *tp* for a single community. It is easy to see that different measures favor groups within different *tp* ranges. The upper bound of Simplified χ^2 increases to 1 and that of Probability Ratio increase to infinity when *tp* is close to 0, which means they favor extremely small groups rather than large groups. Leverage and Likelihood Ratio reach their highest upper bound when *tp* is between 0 and 1. According to the graph, Leverage does not favor the group which contains more than half of the edges in the graph since its upper bound starts to decrease even when the group size increases. Similarly, Likelihood Ratio does not favor the group which contains more than roughly one quarter of the edges in the graph. In all, Probability Ratio favors the smallest groups, followed by Simplified χ^2, Likelihood Ratio, and Leverage.

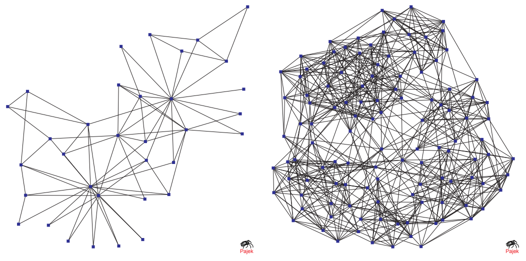
4. EXPERIMENTS

4.1 Experiment Settings

Most prior research on modularity-based methods sets modularity as their objective function and uses different optimization techniques to search for the partition that generates the highest value. In this paper, instead of exploring better optimization techniques for the same objective function, we change the objective function and study what kinds of difference the various objective functions make. In order to conduct the fair comparison for different objective functions, we choose greedy search, the simplest optimization technique, which is also used by the original modularity method and generates reasonably good results [9]. Initially, each node is the only member of its own community. The algorithm iteratively join the two communities that increase the objective function most in the current round. The algorithm will stop if the best merge cannot further increase the objective function. In this section, we conduct experiments on both real life and simulated datasets.

4.1.1 Real Life Data

The two real life datasets include Karate Club [41] and College Football [19], as shown in Figure 2 [1]. As we mentioned in Section 1, we only conduct the performance evaluation on non-overlapping community detection. The existing large datasets with ground truth that we can find are all for overlapping community detection, so we can only do the performance evaluation on large datasets of simulated data. The karate dataset contains friendships between 34 members of a karate club at a US university in the 1970s. There was a disagreement between the administrator and the instructor in the club, which resulted in two communities in this graph. The football dataset records games between Division IA colleges during regular season Fall 2000. There were 115 teams in 12 different conferences.



(a) Karate (b) Football

Figure 2: Real data sets

4.1.2 Simulated Data

With regard to graph simulation, the first related work is called the planted L-partition model [10]. The model simulates a graph with $n = g * l$ nodes in l groups with g nodes each. Nodes of the same group are linked with a probability $p_{internal}$, whereas nodes of different groups are linked with a probability $p_{external}$. If $p_{internal} > p_{external}$, the intra-cluster edge density exceeds the inter-cluster edge density. Then the graph has a community structure which is quite intuitive. However, by using the planted L-partition model, all nodes have approximately the same degree and all communities have exactly the same size. In the real social network data, degree distributions are usually skewed, with many nodes with low degree and a few nodes with high degree. A similar heterogeneity is also observed in the distribution of community size. Recently, Lancichinetti et al. [23] introduced a very popular LFR model. They assume that the distributions of degree and community size are power laws with τ_1 and τ_2 respectively. Each node shares a fraction of $1 - u$ of its edges with the nodes in the same community and a fraction of u with the nodes in the other communities, where u is the mixing parameter. Since the LFR model is much more realistic, we will use the graphs generated by this model to test our algorithms. The simulation procedure is as follows:

1 A set of community sizes s_j following the predefined power law parameter τ_2 is generated.

2 A set of node degrees k_i following the predefined power law parameter τ_1 is generated. The internal degree of each node is $(1-u)k_i$ where u is the mixing parameter.

3 In the beginning, nodes are not assigned to any community. Each node is assigned to a randomly-chosen community which has empty spots to accept a new node. If the community size exceeds the internal degree of the node, the node enters the community; otherwise, it enters a waiting list.

4 For each node in the waiting list, we let the node enter a random community whose size exceeds the node's internal degree and randomly kick one node in the selected community out to the waiting list. We do this step iteratively until the waiting list is empty.

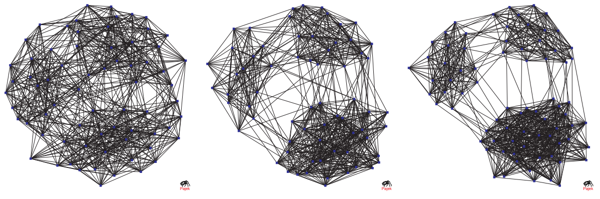
5 We enforce the condition on the fraction of internal degree and external degree. The rewiring procedure in [4] is performed when needed.

However, the constraint used in the LFR model to assign the internal degree of each node in the second step is problematic because the condition imposed by a fixed u cannot guarantee $p_{internal} > p_{external}$ which must be satisfied for a community structure. For a node A in a community with n' nodes in a graph with n nodes, u must be smaller than $1 - n'/n$ to guarantee $p_{internal} > p_{external}$. Therefore, we use the following constraint to assign the internal degree of each node in the second step: $p_{internal} = \beta \cdot p_{external}$, where β is the ratio to control the community structure and must be greater than 1.

There are 8 parameters related to the LFR simulation model: the total number of nodes, the minimal node degree, the maximal node degree, the power law parameter for node degree, the minimal community size, the maximal community size, the power law parameter for community size, and the ratio β for community structure. In order to avoid bias to different objective functions, we choose greedy search, the simplest optimization technique. However, greedy search can only handle 2,000 nodes within a reasonable amount of time. Therefore, we also use a fast unfolding search technique [5] to handle the network with more than 1 million nodes, which is closer to the real world network size.

Given the number of nodes is 2,000, we conduct 9 sets of experiments and the parameter values are shown in Table 1. In order to check the performance difference on different community sizes and tightness of community structure, we only change the minimal community size and the ratio β for community structure to generate different graphs. When the minimal community size is 5, there are a lot of small communities, some mid-size communities, and a few large communities, while the graph only contains large communities

[1] All the network visualization in this paper is visualized by Pajek Software



(a) β is 5 (b) β is 10 (c) β is 20

Figure 3: Simulated data sets

when the minimal community size is 100. The community structure is fuzzy when β is 5, while it is clear when β is 20. How the β affects the community structure is shown in Figure 3. Given the number of nodes is 300,000, we conduct a similar set of 9 experiments and the parameter values are also shown in Table 1.

4.1.3 Evaluation Measures

After finding communities in a given graph, we need to compare our search results with the "actual communities" (the ground truth). For two partitions $X = (X_1, X_2, ..., X_{n_x})$ and $Y = (Y_1, Y_2, ..., Y_{n_y})$ of a graph, X is determined by the algorithm with n_x communities and Y is the ground truth with n_y communities. We need a criterion to measure how similar the partition result of the algorithm is to the partition we hope to find. Many different measures, such as Normalized Mutual Information [11], Jaccard, Rand Index [32], and F-measure [34], have been proposed, and they can be divided into three categories: pair counting, community matching, and information theory [2]. Since different measures have different bias, we show the experimental results on different measures, but focus our analysis on the most widely accepted measure, Normalized Mutual Information. First, it calculates Mutual Information: $MI = \sum_i \sum_j P(X_i \cap Y_j) log \frac{P(X_i \cap Y_j)}{P(X_i)P(Y_j)}$. MI measures the amount of information by which our knowledge about the community in one partition increases when we are told what the community in the other partition is. The minimum of MI is 0 if the X partition is random with respect to the Y partition. However, given a partition Y, all partitions derived from Y by further partitioning have the same mutual information with Y, even though they are different from each other. In this case, the mutual information is equal to the entropy $H(Y) = -\sum_j P(Y_j) log(P(Y_j))$. To avoid that, Danon et al. [11] proposed the normalized mutual information: $NMI = \frac{2*MI}{H(X)+H(Y)}$. It is currently often used and reaches its maximal value 1 if X partition is identical to Y partition.

4.2 Evaluation on Real Life Datasets

The result on real life datasets is shown in Table 2 [2]. The best method for Karate dataset is Leverage. The result is consistent with our theoretical analysis. The karate dataset only contains two large communities, and Leverage is the method having the most bias to large communities. Both Simplified χ^2 and Likelihood Ratio are good for the football dataset. This dataset contains 12 almost equal size communities. According to Figure 1, Likelihood Ratio has the bias to middle-size communities, and Simplified χ^2 has the bias, but not the extreme bias, to small communities. It is why Simplified χ^2 and Likelihood Ratio work better on the football dataset.

4.3 Evaluation on Simulated Datasets

For each parameter setting, we generate the graph 10 times to test each method. We calculate the average value for each evaluation measure shown in Table 3 and 4. Since the result of 2,000 nodes is very similar to that of 300,000 nodes, we focus our analysis on the result of 2,000 nodes. In addition, different evaluation measures provide different information. Since NMI is the most widely-accepted measure, we focus our discussion on NMI in the following. The average NMI and the average number of partitioned communities generated by each method for each parameter setting are shown in Figures 4 - 7.

Figure 4 shows the NMI achieved by each method and Figure 5 shows the number of partitioned communities when fixing the minimal community size and changing the ratio β. No matter what the minimal community size and the ratio β are, the NMI and the number of partitioned communities for both Simplified χ^2 and Probability Ratio are almost the same. They always detect more than 500 communities. Since the total number of nodes is 2,000, most of the communities they detect contain 2 or 3 nodes. That supports our observation in Section 3 that Simplified χ^2 and Probability Ratio favor small size communities. No matter what the minimal community size is, both Leverage and Likelihood Ratio achieve better NMI when the community structure becomes clearer. Only when the community structure is clear and the whole graph only contains large communities, the NMI of Leverage is better than that of Likelihood Ratio. Leverage has more bias towards large communities than Likelihood Ratio according to our upper bound analysis; therefore, we are expecting Leverage is better than Likelihood Ratio when the graph only contains large communities. In practice, social networks contain a lot of small communities; therefore, Likelihood Ratio is better in the common case. The number of partitioned communities by Leverage is almost the same no matter how we change the minimal community size and the ratio β, while the number of partitioned communities by Likelihood Ratio get closer to the ground truth when the community structure becomes clearer. Another interesting observation related to Leverage is that its NMI is very low when the minimal community size is large under the fuzzy community structure. Even under the fuzzy community structure, Leverage detects the same number of large communities. Such a partition assigns many nodes in different real communities to the same partitions, which results in the low NMI. Generally speaking, the partition generated by Likelihood Ratio is better and more adaptive to the different types of graphs than that of Leverage.

Figure 6 shows the NMI achieved by each method and Figure 7 shows the number of partitioned communities when fixing the ratio β and changing the minimal community size. Since both Simplified χ^2 and Probability Ratio favor small-size communities, their NMI decreases with the increase of the minimal community size no matter the community structure is fuzzy or clear. The NMI of Likelihood Ratio decreases with the increase of the minimal community size when the

[2] RI: Rand Index; DNC: the detected number of communities; ANC: the actual number of communities; χ^2: Simplified χ^2; PR: Probability Ratio; and LR: Likelihood Ratio

Parameter	Data set A	Data set B
The total number of nodes	2000	300000
The minimal node degree	5	50
The maximal node degree	300	3000
The power law parameter for node degree	2.5	2.5
The minimal community size	5, 50, or 100	50, 500, or 5000
The maximal community size	300	10000
The power law parameter for community size	1.5	1.5
The ratio β for community structure	5, 10, or 20	5, 10, or 20

Table 1: Parameter Setting for Simulated Graphs

Data Set	Measure	NMI	Jaccard	RI	F-measure	DNC	ANC
Karate	χ^2	0.4852	0.2842	0.6453	0.4426	7	2
	PR	0.3868	0.0945	0.5561	0.1728	14	2
	Leverage	**0.6925**	**0.6833**	**0.8414**	**0.8118**	3	2
	LR	0.5385	0.3958	0.6952	0.5671	5	2
Football	χ^2	**0.9141**	0.7571	0.9793	0.8618	14	12
	PR	0.6864	0.0829	0.9240	0.1531	55	12
	Leverage	0.6977	0.3622	0.8807	0.5317	6	12
	LR	0.9086	**0.7897**	**0.9812**	**0.8825**	12	12

Table 2: Results on real life datasets

Data Set	Measure	NMI	Jaccard	RI	F-measure	DNC	ANC
MCS=5 β=5	χ^2	**0.5868**	0.0122	**0.9391**	0.0240	629.5	50.8
	PR	0.5856	0.0062	0.9390	0.0124	903.3	50.8
	Leverage	0.1222	**0.0809**	0.7749	**0.1481**	9.4	50.8
	LR	0.5515	0.0272	0.9388	0.0530	300.5	50.8
MCS=5 β=10	χ^2	**0.6023**	0.0146	0.9397	0.0289	604.8	51.2
	PR	0.5937	0.0068	0.9394	0.0136	905	51.2
	Leverage	0.2741	**0.1523**	0.7900	**0.2605**	7.5	51.2
	LR	0.5992	0.0462	**0.9406**	0.0883	263.4	51.2
MCS=5 β=20	χ^2	0.6212	0.0196	0.9436	0.0385	564.6	51.8
	PR	0.6035	0.0089	0.9432	0.0177	855.7	51.8
	Leverage	0.5349	0.2265	0.8215	0.3658	6.8	51.8
	LR	**0.7545**	**0.5136**	**0.9714**	**0.6699**	139.5	51.8
MCS=50 β=5	χ^2	**0.4775**	0.0091	**0.9177**	0.0181	586.2	16
	PR	0.4765	0.0054	0.9176	0.0107	777.6	16
	Leverage	0.1172	**0.1016**	0.7754	**0.1820**	9.2	16
	LR	0.4314	0.0194	0.9172	0.0381	283.7	16
MCS=50 β=10	χ^2	**0.5075**	0.0125	0.9240	0.0246	554.3	16.5
	PR	0.4969	0.0069	0.9237	0.0137	747.9	16.5
	Leverage	0.4318	**0.2523**	0.8302	**0.3983**	6.2	16.5
	LR	0.5040	0.0507	**0.9259**	0.0958	243.3	16.5
MCS=50 β=20	χ^2	0.5280	0.0154	0.9211	0.0303	533	15.8
	PR	0.5065	0.0076	0.9205	0.0151	758.4	15.8
	Leverage	0.7375	0.4098	0.8886	0.5773	6.5	15.8
	LR	**0.7663**	**0.6430**	**0.9703**	**0.7778**	67.9	15.8
MCS=100 β=5	χ^2	0.4210	0.0073	**0.8978**	0.0145	568.9	10.7
	PR	**0.4243**	0.0044	0.8977	0.0088	750	10.7
	Leverage	0.1471	**0.1330**	0.7710	**0.2336**	8.5	10.7
	LR	0.3727	0.0156	0.8972	0.0306	280.2	10.7
MCS=100 β=10	χ^2	0.4538	0.0092	0.9038	0.0183	571.9	11.3
	PR	0.4514	0.0053	0.9036	0.0106	774.2	11.3
	Leverage	**0.5587**	**0.3423**	0.8532	**0.5038**	6.2	11.3
	LR	0.4458	0.0287	**0.9046**	0.0558	250.8	11.3
MCS=100 β=20	χ^2	0.4844	0.0128	0.9016	0.0253	522.6	11.2
	PR	0.4657	0.0069	0.9010	0.0138	721.9	11.2
	Leverage	**0.8318**	0.5755	0.9291	0.7300	6.8	11.2
	LR	0.7614	**0.6550**	**0.9638**	**0.7851**	52.2	12

Table 3: Results on simulated datasets with 2,000 nodes

Data Set	Measure	NMI	Jaccard	RI	F-measure	DNC	ANC
MCS=50 β=5	χ^2	**0.5567**	0.0007	**0.9882**	0.0057	87651.0	428.2
	PR	0.5294	0.0002	0.9882	0.0068	117151.1	428.2
	Leverage	0.0698	**0.0173**	0.6044	**0.0642**	5.6	428.2
	LR	0.5145	0.0006	0.9882	0.0088	56275.7	428.2
MCS=50 β=10	χ^2	**0.5731**	0.0007	**0.9889**	0.0060	89529.1	458.4
	PR	0.5413	0.0002	0.9889	0.0071	118370.6	458.4
	Leverage	0.0941	**0.0191**	0.6488	**0.0715**	7.4	458.4
	LR	0.5349	0.0008	0.9889	0.0094	55654.8	458.4
MCS=50 β=20	χ^2	0.5793	0.0007	0.9880	0.0056	91573.8	433.7
	PR	0.5427	0.0002	0.9880	0.0067	119961.7	433.7
	Leverage	0.3588	0.0486	0.7721	0.2688	11.1	433.7
	LR	**0.6270**	**0.5480**	**0.9947**	**0.2805**	38997.4	433.7
MCS=500 β=5	χ^2	**0.5009**	0.0005	**0.9860**	0.0021	86258.5	135.7
	PR	0.4730	0.0001	0.9860	0.0018	103355.6	135.7
	Leverage	0.0534	**0.0182**	0.6446	**0.0661**	4.7	135.7
	LR	0.4625	0.0005	0.9860	0.0024	56536.5	135.7
MCS=500 β=10	χ^2	**0.5156**	0.0005	**0.9863**	0.0022	87168.8	139.2
	PR	0.4816	0.0002	0.9863	0.0018	104363.0	139.2
	Leverage	0.2004	**0.0354**	0.7473	**0.2048**	7.3	139.2
	LR	0.4828	0.0006	0.9863	0.0026	56810.1	139.2
MCS=500 β=20	χ^2	0.5234	0.0006	0.9854	0.0021	88439.6	131.2
	PR	0.4854	0.0002	0.9854	0.0018	104308.4	131.2
	Leverage	0.5731	0.0866	0.8531	0.4551	20.4	131.2
	LR	**0.7106**	**0.8050**	**0.9972**	**0.6000**	18350.4	131.2
MCS=5000 β=20	χ^2	**0.3747**	0.0001	**0.9751**	0.0006	81634.5	41.7
	PR	0.3427	0.0001	0.9751	0.0005	68326.2	41.7
	Leverage	0.0099	**0.0226**	0.8037	**0.0632**	10.0	41.7
	LR	0.3456	0.0001	0.9751	0.0007	58845.6	41.7
MCS=5000 β=20	χ^2	0.3885	0.0002	**0.9748**	0.0007	81365.6	41.4
	PR	0.3521	0.0001	0.9748	0.0006	65978.2	41.4
	Leverage	**0.8037**	**0.2315**	0.9099	**0.6057**	19.2	41.4
	LR	0.3726	0.0003	0.9748	0.0010	58409.5	41.4
MCS=5000 β=20	χ^2	0.4053	0.0002	0.9749	0.0008	80492.0	41.4
	PR	0.3737	0.0002	0.9749	0.0008	64004.6	41.4
	Leverage	0.8964	0.4267	0.9635	0.7332	24.8	41.4
	LR	**0.9971**	**0.9962**	**0.9999**	**0.9964**	214.4	41.4

Table 4: Results on simulated datasets with 300,000 nodes

community structure is not clear. However, when the community structure is clear, Likelihood Ratio achieves almost the same performance with the increase of the minimal community size. The NMI of Leverage always increases with the increase of the minimal community size since it has the bias to large communities.

5. CONCLUSION

In this paper, we connect modularity-based methods with correlation analysis by subtly reformatting their math formulas, and make smart use of different correlation measures to change the objective function of modularity-based methods. A novel theoretical analysis on upper bounds is conducted to analyze the bias of different objective functions and the bias is validated by our experiments. Using the widely-accepted Normalized Mutual Information to compare the partitions determined by the algorithm with the ground truth, Likelihood Ratio is better and more robust. However, different measures can be used for different purposes. For example, Probability Ratio can be used if we want to fairly partition the students in the class into small groups for class projects, and we might use Leverage to find relatively large groups for marketing campaigns. As shown above, our finding provides a more natural and effective way to solve the resolution limit problem of the original modularity function by modifying it through different correlation measures. In the future, we will investigate more correlation measures, and test performance differences for detecting overlapping communities.

6. REFERENCES

[1] H. Almeida, D. O. G. Neto, W. M. Jr., and M. J. Zaki. Is there a best quality metric for graph clusters? In *15th European Conference on Principles and Practice of Knowledge Discovery in Databases*, Sep 2011.

[2] E. Amigó, J. Gonzalo, J. Artiles, and F. Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval*, 12:461–486, August 2009.

[3] A. Arenas, A. FernÂÍÂćndez, and S. GÂÍÂőmez. Analysis of the structure of complex networks at different resolution levels. *New Journal of Physics*, 10(5):053039, 2008.

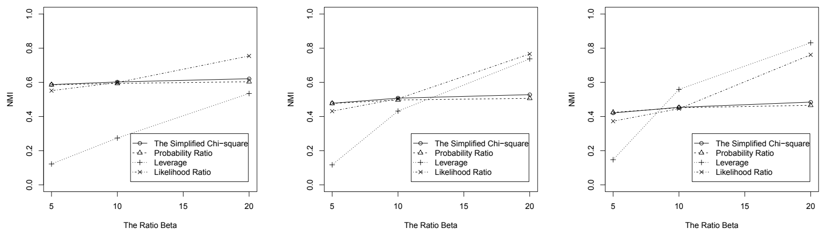
[4] J. P. Bagrow. Evaluating local community methods in networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(05):P05001+, 2008.

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

[6] S. Boettcher and A. G. Percus. Extremal optimization for graph partitioning. *PHYS.REV.E*, 64:026114, 2001.

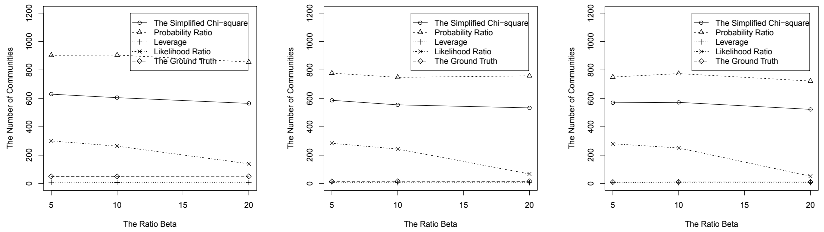
[7] U. Brandes, D. Delling, M. Gaertler, R. Gorke, M. Hoefer, Z. Nikoloski, and D. Wagner. On modularity clustering. *IEEE Trans. on Knowl. and Data Eng.*, 20:172–188, February 2008.

[8] S. Brin, R. Motwani, J. D. Ullman, and S. Tsur. Dynamic itemset counting and implication rules for market basket data. In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, pages 255–264, 1997.



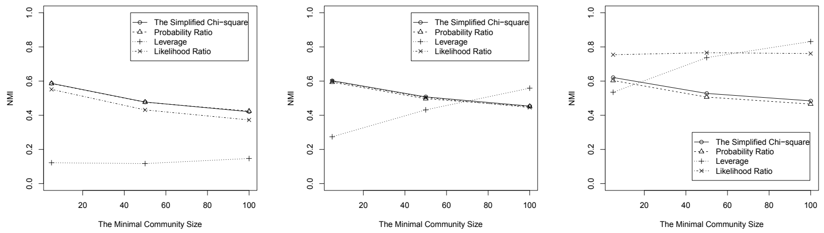
(a) The minimal community size is 5 (b) The minimal community size is 50 (c) The minimal community size is 100

Figure 4: NMI when fixing the minimal community size



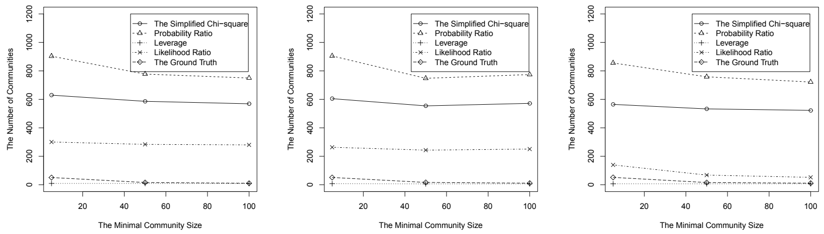
(a) The minimal community size is 5 (b) The minimal community size is 50 (c) The minimal community size is 100

Figure 5: The number of communities when fixing the minimal community size



(a) The ratio β is 5 (b) The ratio β is 10 (c) The ratio β is 20

Figure 6: NMI when fixing the ratio β



(a) The ratio β is 5 (b) The ratio β is 10 (c) The ratio β is 20

Figure 7: The number of communities when fixing the ratio β

[9] A. Clauset, M. E. J. Newman, and C. Moore. Finding community structure in very large networks. *Physical Review E*, 70(6):066111+, Dec. 2004.

[10] A. Condon and R. M. Karp. Algorithms for graph partitioning on the planted partition model. *Random Struct. Algorithms*, 18:116–140, March 2001.

[11] L. Danon, A. D. Guilera, J. Duch, and A. Arenas. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(9):P09008–09008, Sept. 2005.

[12] L. Duan and W. N. Street. Finding maximal fully-correlated itemsets in large databases. In *ICDM '09: Proc. Int. Conf. on Data Mining*, pages 770–775, Miami, FL, USA, 2009.

[13] L. Duan and W. N. Street. Selecting the right correlation measure for binary data. *http://ssrn.com/abstract=2035491*, 2012.

[14] J. Duch and A. Arenas. Community detection in complex networks using extremal optimization. *Physical Review E*, 72(2):027104+, Aug. 2005.

[15] T. Dunning. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74, 1993.

[16] S. Fortunato. Community detection in graphs. *Physics Reports*, 486:75–174, 2010.

[17] M. Gaertler, R. Görke, and D. Wagner. Significance-driven graph clustering. In *Proceedings of the 3rd international conference on Algorithmic Aspects in Information and Management*, AAIM '07, pages 11–26, Berlin, Heidelberg, 2007. Springer-Verlag.

[18] L. Geng and H. J. Hamilton. Interestingness measures for data mining: A survey. *ACM Computing Surveys*, 38(3):9, 2006.

[19] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, June 2002.

[20] C. Jermaine. Finding the most interesting correlations in a database: How hard can it be? *Information Systems*, 30(1):21–46, 2005.

[21] R. Kannan, S. Vempala, and A. Veta. On clusterings-good, bad and spectral. In *Proceedings. 41st Annual Symposium on Foundations of Computer Science*, pages 367–377, 2000.

[22] A. Lancichinetti and S. Fortunato. Limits of modularity maximization in community detection. *Phys. Rev. E*, 84:066122, Dec 2011.

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

[24] U. Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17:395–416, December 2007.

[25] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. L. Cam and J. Neyman, editors, *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.

[26] S. Mancoridis, B. S. Mitchell, and C. Rorres. Using automatic clustering to produce high-level system organizations of source code. In *Proc. 6th Intl. Workshop on Program Comprehension*, pages 45–53, 1998.

[27] C. P. Massen and J. P. K. Doye. Identifying communities within energy landscapes. 71(046101), 2005.

[28] M. E. J. Newman. Community detection and graph partitioning. *CoRR*, abs/1305.4974, 2013.

[29] V. Nicosia, G. Mangioni, V. Carchiolo, and M. Malgeri. Extending the definition of modularity to directed graphs with overlapping communities. *Journal of Statistical Mechanics: Theory and Experiment*, 2009:P03024, 2009.

[30] G. Piatetsky-Shapiro. *Discovery, Analysis, and Presentation of Strong Rules.* AAAI/MIT Press, 1991.

[31] C. Pizzuti. Community detection in social networks with genetic algorithms. In *Proceedings of the 10th annual conference on Genetic and evolutionary computation*, GECCO '08, pages 1137–1138, New York, NY, USA, 2008. ACM.

[32] W. M. Rand. Objective Criteria for the Evaluation of Clustering Methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.

[33] J. Reichardt and S. Bornholdt. Statistical mechanics of community detection. *Phys. Rev. E*, 74:016110, Jul 2006.

[34] C. V. Rijsbergen. Foundation of Evaluation. *Journal of Documentation*, 30:365–373, 1974.

[35] H. Shiokawa, Y. Fujiwara, and M. Onizuka. Fast algorithm for modularity-based graph clustering. In M. desJardins and M. L. Littman, editors, *AAAI*. AAAI Press, 2013.

[36] P.-N. Tan, V. Kumar, and J. Srivastava. Selecting the right objective measure for association analysis. *Information Systems*, 29(4):293–313, 2004.

[37] C. Tew, C. Giraud-Carrier, K. Tanner, and S. Burton. Behavior-based clustering and analysis of interestingness measures for association rule mining. *Data Mining and Knowledge Discovery*, pages 1–42, 2013.

[38] K. Wakita and T. Tsurumi. Finding community structure in mega-scale social networks. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 1275–1276, New York, NY, USA, 2007. ACM.

[39] R. S. Weiss and E. Jacobson. A method for the analysis of the structure of complex organizations. *American Sociological Review*, 20(6):pp. 661–668, 1955.

[40] J. Xie, S. Kelley, and B. K. SZYMANSKI. Overlapping community detection in networks: the state of the art and comparative study. *ACM Computing Surveys*, 45, 2013.

[41] W. W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33:452–473, 1977.