On the Detectability of Node Grouping in Networks

Chi Wang[‡], Hongning Wang[‡], Jialu Liu[‡], Ming Ji[‡], Lu Su[‡], Yuguo Chen[†], Jiawei Han[‡]
[‡]Department of Computer Science, [†]Department of Statistics
University of Illinois at Urbana-Champaign
{chiwang1, wang296, jliu64, mingji1, lusu2, yuguo, hanj}@illinois.edu

Abstract

In typical studies of node grouping detection, the grouping is presumed to have a certain type of correlation with the network structure (*e.g.*, densely connected groups of nodes that are loosely connected in between). People have defined different fitness measures (modularity, conductance, *etc.*) to quantify such correlation, and group the nodes by optimizing a certain fitness measure. However, a particular grouping with desired semantics, as the target of the detection, is not promised to be detectable by each measure. We study a fundamental problem in the process of node grouping discovery: Given a particular grouping in a network, whether and to what extent it can be discovered with a given fitness measure. We propose two approaches of testing the detectability, namely ranking-based and correlation-based randomization tests. Our methods are evaluated on both synthetic and real datasets, which shows the proposed methods can effectively predict the detectability of groupings of various types, and support explorative process of node grouping discovery.

1 Introduction

Node grouping is an important problem in network analysis. People partition nodes into different groups based on their linkage patterns, and often find the partitions are meaningful in semantic, functional or social perspectives. Every grouping detection algorithm is based on some underlying assumptions of the correlation between the network structure and the grouping to be detected. As one example, most community detection algorithms assume high edge connectivity within each group and low edge connectivity between groups [10]. Typically, detectors define a fitness measure such as *modularity* [18] and *conductance* [12, 15] to quantify such correlation, and then optimize the fitness measure to detect the grouping. However, a particular grouping with certain semantics, which we referred to as the *target grouping*, is not promised to be detectable under every fitness measure. Although a large amount of grouping detection algorithms have been developed, a fundamental problem on the other hand has hardly been studied: whether and to what extent a particular grouping in the network can be recovered by a certain fitness measure.

Study of this problem will support the knowledge discovery process in network analysis in many ways, including (i) finding appropriate fitness measures for different grouping views (*e.g.*, grouping researchers by their research areas, affiliation or roles); and (ii) when heterogeneous types of links are present, finding relevant relations for a specific detection task. A prevalidation of the fitness measures saves effort from attempts of designing algorithms with mismatched assumptions and avoids improper choices of evaluation benchmark. When exploring new fitness measures or tackling with new detection tasks, one can also test the detectability before substantial development of new algorithms.

Prior work and limitations: To the best of our knowledge, the (un)detectability of node grouping is rarely studied. Some theoretical work is along this line, yet focused on analysis to specific random graph models as well as special detection algorithms [7, 21, 22]. We aim for a general algorithmic test that can be applied to real networks and groupings of generic types.

Main idea of our recipe: We propose two approaches for performing such detectability test, namely ranking-based and correlation-based tests. The first is to compare the fitness of the target grouping to all other possible groupings under the given measures. The second is to test if a grouping with higher fitness measure is closer to the target grouping. The main challenge of these tests is how to efficiently and accurately assess the ranking of the target grouping among all groupings, and the correlation between a group's fitness and proximity to target grouping. For both tests, complete enumeration of all groupings is not feasible for large networks. Therefore, we propose to use a randomization method, a.k.a. resampling technique, to sample a much smaller number of random groupings, and estimate the ranking-based and correlation-based statistics for the detectabil-

ity tests.

Contributions:

• We propose two different approaches of quantifying detectability and several detectability statistics. We find that the two approaches are complementary, yet correlation-based test supports analysis in finer granularity than ranking-based test in general.

• We design one resampling algorithm for the ranking-based test, and three alternative algorithms for the correlation-based test, for comparative study.

• We apply our approach in both synthetic and real networks, demonstrating the success of our detectability test: i) without relying on any inference algorithm, we can predict the detectability of known groupings with given measures; and ii) when some grouping is undetectable by the known fitness measures, we can try new measures and test the detectability with them before developing inference algorithms.

2 Detectability Test of Node Grouping

In this section, we formally define the problem of Detectability Test of Node Grouping (DeTeNG). We use the two terms grouping and partition interchangeably as we focus on non-overlapped grouping in this work.

Let $N = (V, E)$ be an undirected network comprising a set V=$\{v_1, v_2, \ldots, v_n\}$ of nodes together with a set $E = \{(v_i, v_j)_{i \neq j}\}$ of edges, which are 2-element subsets of V. A k-way partition (grouping) is a function defined on the node set V of network N, i.e., $g(v) \in \{1, 2, \ldots, k\}$ for $v \in V$. It divides the node set into k disjoint subsets, $\mathcal{C}_i = \{v | g(v) = i\}, i = 1, \ldots, k$, each named a group. *Partition size vector* is defined to be $\mathbf{s}^g = (s_1^g, \ldots, s_k^g)$, where $s_i^g = |\mathcal{C}_i|$.

A target partition is often assumed to satisfy a certain criterion. Such a criterion can be formulated as a fitness measure.

Definition 1. (Partition Fitness Measure) *A partition fitness measure is defined to be a real-valued function on a partition g over a network N: $h(N, g) \to \mathcal{R}$. It measures the fitness of g over N under a certain criterion, so that a partition which fits the criterion better in N receives a higher assessment.*

Without loss of generality, we assume one would maximize such measures.

With a given fitness measure, one can design an optimization algorithm (a.k.a. *inference* algorithm) to find a grouping with optimal fitness score. However, a grouping with high fitness score may not always be highly similar to the target grouping. To prevent waste of effort due to mismatched fitness measures, we propose the idea of detectability test.

Definition 2. (Detectability Test) *A detectability test of node grouping takes a target partition g_0 over a network N, and a fitness measure $h(N, g)$ as input, and predicts whether an inference algorithm optimizing h over g can detect g_0 (or groupings similar to g_0), without knowing or running the algorithm.*

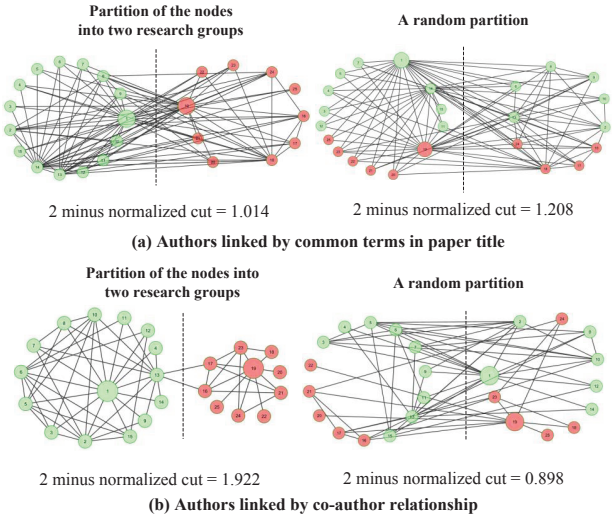

Figure 1: Detectability of the same two data mining research groups (shown in green & red colors) in two different networks. The target grouping is detectable in network (b), but undetectable in (a) under the specific fitness measure of 2 minus normalized cut. For network (a), many other partitions qualify the criterion equally well or better; whereas for network (b), a random partition does not quite fit the criterion.

In a practical point of view, the network N can be a representative example from a family of similar networks, such that we can obtain the target grouping for N. For example, in the co-author network, we can label some research groups and extract a subnetwork with the labeled nodes. Figure 1 exhibits two instances of our DeTeNG problem with the same input of target grouping and fitness measures, but in different networks. A good detectability test should predict that the grouping is less detectable in graph (a) than in graph (b).

Our detectability test does not take any inference algorithm as input, and thus the test can be conducted independently of inference algorithm design. The reason for the separation is two-folds: i) for most fitness measures which already have inference algorithms, the algorithms can only perform approximate or heuristic optimization, which may introduce bias; and ii) for new fitness measures which have no optimization algorithm yet, we want to evaluate its goodness of fit before devoting effort to the optimization problem. We propose two different approaches of framing the problem in Section 2.1.

2.1 Ranking-based and correlation-based tests

The first approach of quantifying the detectability is to examine the rank of the target grouping among all the groupings ordered by their fitness. Previous work has found that many fitness measures are significantly influenced by the group size [10, 15]. For sake of clarity, we only consider all the groupings with the same size vector as g_0, in order to remove the size effect. An ideal fitness measure should rank the target grouping topmost, so that an inference algorithm can perfectly detect the target grouping by optimizing the fitness measure. If the target grouping is ranked low, one is unlikely to detect the target grouping by searching for highly ranked groupings. With that intuition, we have the following definition of ranking-based detectability.

Definition 3. (Ranking-based Detectability) *Given a target partition g_0 on a network N, and a fitness measure $h(N, g)$, the ranking-based detectability $DetRank(N, g_0, h)$ is defined as the fraction of all groupings on network N with lower fitness measure than g_0.*

This definition has another explanation in statistical language: the ranking characterizes how significantly g_0 can be distinguished from a random partition g under the measure h. A high ranking-based detectability is equivalent to a low probability of observing a random partition with equal or higher fitness score, i.e., $P(h(N, g) \geq h(N, g_0))$.

The second approach is to examine whether the fitness measure and the proximity to the target grouping of an arbitrary grouping is positively correlated. The idea is that if a fitness measure is good for detecting a certain target grouping, a grouping with a higher fitness measure should tend to be similar to the target grouping, so that the better an inference algorithm achieves in optimizing the measure, the closer it approaches to the target grouping. Let $\Omega(g_1, g_2)$ be a function to measure the proximity between two groupings g_1 and g_2, we have:

Definition 4. (Correlation-based Detectability) *Given a target partition g_0 on a network N, and a fitness measure $h(N, g)$, the correlation-based detectability $DetCorr(N, g_0, h)$ is defined as the correlation between the fitness measure $h(N, \cdot)$ and the proximity to the target grouping $\Omega(g_0, \cdot)$ for all groupings g on N.*

The particular definition of the proximity and the correlation will be discussed in Section 4. Intuitively, correlation-based detectability involves more comprehensive analysis than rank-based detectability: the ranking only compares the target grouping with the other candidate groupings, whereas the correlation also compares among those groupings.

In the following two sections, we will present our approaches to these two kinds of tests respectively.

3 Ranking-based Test

We want to test whether the target grouping is ranked in top $\alpha\%$ among all possible groupings, where α is a small threshold such as 0.1. The ranking of the target grouping can be estimated as:

$$DetRank(N, g_0, h) = 1 - P(h(N, g) \geq h(N, g_0))$$

So testing if $DetRank(N, g_0, h) > 1 - \alpha\%$ is equivalent to testing if $P(h(N, g) \geq h(N, g_0)) = p < \alpha\%$.

In order to calculate the p-value that a random grouping has higher fitness measure than the target grouping, we propose to use a *randomization test*. The idea is to generate many replicates of the original data with rearranged labels on the observed data points, and estimate the distribution of a statisticial measure with these samples. The advantage is that it does not rely on a specific form of the data distribution, which satisfies our requirement for the general applicability.

In our scenario, we can permute the group labels $g_0(v)$ while maintaining the size of each group. Since there are too many possible partitions over the network to allow for complete enumeration, we resort to an asymptotically equivalent permutation test by *Monte Carlo sampling* [20]. It takes a small fraction of the total number of possible permutations as samples.

Algorithm 1: UniRank

Input: Network $N = (V, E)$, k-way target grouping g_0, fitness measure $h(N, g)$, size of sample M
Output: ranking-based detectability

```
Initialize x ← 0;
// # of samples with better fitness
Compute h_0 ← h(N, g_0);
for iter = 1..M do
    g ← a grouping by randomly permuting the
    labels of g_0;
    Evaluate h(N, g);
    if h(N, g) ≥ h_0 then
        x ← x + 1;
    end
end
return DetRank(N, g_0, h) = 1 − x/M ;
```

We name our algorithm UniRank as it estimates ranking via uniform sampling. It is illustrated in Algorithm 1: M samples are uniformly drawn from all possible partitions, and their fitness measures are calculated and compared to the fitness measure h_0 for the target grouping g_0. This process can be explained as a Bernoulli trial with success probability p, where the

success is defined to be observing a partition with equal or higher fitness measure than h_0. Thus the probability of x successes in the M trials $P(x)$ follows a binomial distribution. The binomial confidence interval of p can be calculated by Clopper-Pearson method [6].

4 Correlation-based Test

In this test, we study the correlation between two variables: the fitness score $h(N, g)$ of a grouping g, and the proximity $\Omega(g_0, g)$ of g to the target grouping g_0.

There exist several proximity measures that can quantify the proximity of the partition identified by an algorithm to the target grouping [10]. In this paper, we pick the "accuracy" of cluster matching, which aims at finding the largest overlaps between pairs of clusters in different partitions:

$$\Omega(g_0, g) = Acc(g_0, g) = \frac{1}{|V|} \max_{\pi} |\{i | g_0(i) = \pi(g(i))\}|$$

where π ranges over all mappings from the grouping indices of g to g_0. One can alter the measure towards the need of more strict or loose notion of "detectable".

For the same reason mentioned in Section 3, we resort to resampling technique to estimate the correlation. For each sampled grouping g_i, we calculate the fitness score $h_i = h(N, g_i)$ and the proximity to target grouping $\Omega_i = \Omega(g_0, g_i)$ to get a set of paired variables (h_i, Ω_i). Then we use *correlation coefficients* to measure correlation-based detectability. Below we present three different methods of sampling and computing the correlation coefficients.

4.1 A baseline method

As a baseline method, we modify the Monte Carlo sampling algorithm described in Section 3: instead of counting the number of groupings with higher fitness score than the target grouping, we use the samples to estimate the correlation between $h(N, \cdot)$ and $\Omega(g_0, \cdot)$.

Two commonly used correlation measures in statistics are Pearson's correlation coefficient and Kendall's τ rank correlation coefficient. The former is salient only to a linear relationship between two variables and the latter is free of that assumption. In our case, there could be a nonlinear relationship between fitness and proximity. Hence we choose Kendall's τ rank correlation to measure the extent to which, as the fitness score h increases, the proximity Ω tends to increase.

$$\tau = \frac{|\mathcal{S}^+| - |\mathcal{S}^-|}{|\mathcal{S}^+| + |\mathcal{S}^-|} \tag{4.1}$$

where

$$\mathcal{S}^+ = \{(i, j), i < j | (h_i - h_j)(\Omega_i - \Omega_j) > 0\}$$

is the set of concordant pairs, and

$$\mathcal{S}^- = \{(i, j), i < j | (h_i - h_j)(\Omega_i - \Omega_j) < 0\} \cup \{(i, j) | h_i = h_j, \Omega_i < \Omega_j\}$$

is the set of discordant pairs. Note that we customize the Kendall's rank correlation according to our need: we count two groupings with the same fitness but different proximity to the target grouping as a discordant pair because such a pair increases the uncertainty of finding the grouping closer to the target grouping.

Finally, we predict the detectability by comparing the chosen coefficient with a threshold β. We name this baseline approach UniCorr as it uses uniform sampling to estimate the correlation.

4.2 Weighted correlation

The baseline method employs traditional correlation coefficients in statistics but ignores an important aspect of our problem. We assume an inference algorithm will find groupings with high fitness score, and a grouping with low or medium fitness score is unlikely to be chosen. Thus, we should focus more on evaluating the ability of the fitness measure in differentiating groupings near the high end, not the whole space. However, the classic Kendall's rank correlation coefficient defined in Eq. (4.1) is based on the whole spectrum of the fitness measure and it gives every sample equal importance.

To address that issue, we propose to weight the samples according to their fitness score when calculating the correlation. The higher fitness score a grouping has, the larger weight it should receive. A common functional form for these weights is exponential weighting.

$$w(g, t) = exp\left(\frac{h(N, g)}{t}\right) \tag{4.2}$$

where t is a positive constant, and $\frac{1}{t}$ is called the exponential decay factor. We will emphasize more of the samples with high fitness score when t is smaller. We choose exponential function to decrease the weight drastically when fitness decreases, but other types of weighting can be used as well.

The weight of a pair is the product of the weight of the paired samples, such that more heavily weighted samples will contribute more to both concordant and discordant pairs. The weighted correlation coefficient τ_w is computed as:

$$\tau_w = \frac{\sum_{(i,j)\in\mathcal{S}^+} w(g_i)w(g_j) - \sum_{(i,j)\in\mathcal{S}^-} w(g_i)w(g_j)}{\sum_{(i,j)\in\mathcal{S}^+} w(g_i)w(g_j) + \sum_{(i,j)\in\mathcal{S}^-} w(g_i)w(g_j)} \tag{4.3}$$

To compute the weighted correlation, we can use the above formula based on the samples from the same

Monte Carlo algorithm. We name this approach UniWCorr.

4.3 Weighted sampling One potential problem with UniWCorr is that it samples groupings uniformly, regardless of their fitness score, from a huge space of candidate groupings. However, it is usually the case that the samples with high fitness score only take a small portion of the whole set. Consequently, it requires the sample size M to be extremely large in order to obtain a sufficient number of samples with high fitness for accurate correlation estimation.

To achieve a better estimate with affordable sampling complexity, we propose a sampling method based on the Metropolis-Hastings algorithm [5]: we design an auxiliary distribution over all the candidate groupings according to Eq.(4.2), such that the chance for each grouping to be sampled is proportional to its weight. Then, applying Eq.(4.1) with these samples, we can directly obtain the weighted correlation. This is an asymptotically equivalent way of estimating the weighted correlation.

In order to efficiently collect the samples and avoid being trapped by some low probability barriers [16], we appeal to the parallel tempering method for sampling [9] (we refer to it as PT sampling). In PT sampling, instead of only sampling from one auxiliary distribution (specified by the decay factor $\frac{1}{t}$), we simulate T replicas of the original distribution of interest, with each replica associating a different setting of t (referred to as *temperature*). As we have discussed in Section 4.2, higher values of t generally lead us to explore large volumes of grouping space with less emphasis on the samples of high fitness scores; whereas lower values of t promise precise sampling in a local region of high fitness scores, but may trap the samples in local fitness score maximum. PT sampling achieves good sampling results by allowing the auxiliary distributions with different decay factors to exchange complete configurations, so that the distribution with lower temperatures can access a representative set of grouping space.

The algorithm of our PT sampling method for estimating weighted correlation (PTWCorr) is illustrated in Algorithm 2. The target decay factor is $\frac{1}{t_I}$, where t_I is a selected temperature in the input set of temperatures. The example exchange rate γ controls how often we swap samples across temperatures. The new grouping candidate within each temperature is proposed by a symmetric proposal of randomly swapping the group labels of two nodes from different groups.

The key to the success of PT Sampling is to find an appropriate temperature setting. Kofke [13] showed that a geometric progression of temperatures ($\frac{t_i}{t_{i+1}} = const$) results in equal acceptance ratios and good performance.

Algorithm 2: PTWCorr

Input: Network $N = (V, E)$, k-way target grouping g_0, fitness measure $h(N, g)$, size of samples M, set of temperatures $\{t_1, \ldots, t_T\}$, sample exchange rate γ, and target decay factor index I, $1 \leq I \leq T$.

Output: correlation-based detectability

```
while n < M do
    if Unif(0,1) > γ then
        for j = 1...T do
            Propose a new grouping g^new;
            if Unif(0,1) ≤ w(g^new, t_j) / w(g_n^(j), t_j) then
            g_{n+1}^(j) ← g^new;
            else g_{n+1}^(j) ← g_n^(j);
            h_{n+1}^(j) ← h(N, g_{n+1}^(j));
            Ω_{n+1}^(j) ← Ω(g_0, g_{n+1}^(j));
        end
        n ← n + 1;
    end
    else
        Draw j from Unif(1, 2, ..., T − 1);
        if Unif(0,1) ≤ w(g_n^(j+1), t_j) w(g_n^(j), t_{j+1}) / w(g_n^(j), t_j) w(g_n^(j+1), t_{j+1}) then
        swap(g_n^(j), g_n^(j+1));
    end
end
return τ according to Eq. (4.1) using h^(I) and Ω^(I)
```

5 Related Work

The most relevant work we found is in theoretical computer science and physics literatures. Condon and Karp studied the planted partitioning problem [7]. The goal is to discover a special built-in cluster structure: every group has an equal size and a pair of nodes are linked with probability p_{in} if they belong to the same group, and with $p_{out} < p_{in}$ otherwise. They proved that a cluster can be recovered correctly by a special algorithm with high probability when $p_{out} - p_{in}$ is sufficiently large. Similar bounds were provided by Carson and Impagliazzo [4], and Onsjo *et al.* [21]. Reichardt and Leone [22] considered the sparse networks where these bounds are meaningless since both p_{out} and p_{in} scale with the network size. The common limitation for these studies is they only considered special networks generated by certain models with a special detection algorithm or objective.

In the next, we review other related work in grouping detection, and in randomization test respectively.

5.1 Grouping detection in networks

A multitude of methodologies have been proposed for community detection or graph clustering problem in network analysis (see [24] and [10] for comprehensive surveys). Most of them had consistent assumptions about what constitutes a "good" partition, and some tried to optimize certain partition quality measures, *e.g.*, normalized graph cut [25, 19] and modularity [18, 26] with approximation methods. There has been some comparative analysis of graph clustering techniques on benchmark graphs [8, 14], or real networks [15]. Those studies provide evidence for a universal limit of cluster detectability across a variety of algorithms.

The existence of different types of groupings has received more attentions recently, and a number of recent studies pursued generic methods to discover broad types of groupings [17, 3]. A very recent study by Abrahao *et al.* [1] reported that communities extracted by different algorithms form separable structural classes and are often different from real communities. This makes it necessary to study in what case the real community can be recovered by a certain criterion. Our work can be used as a statistical validation tool when one explores new detection criteria before substantial development of detection algorithms.

5.2 Randomization test

Randomization test has been successfully applied in the network analysis tasks (*e.g.*, [2]). The main benefit is that one is released from the difficult, and sometimes impossible, task of defining an asymptotic distribution for the test statistics. Rosvall and Bergstrom [23] performed permutation test on robustness of community structure in networks. The major difference between their test and ours is that they permuted the positions of the edges and compared the optimal division of the perturbed network to the optimal division of the original network; we keep the original network structure unchanged and compare other possible partitions in the network against the target one.

6 Empirical Evaluation

This section consists of three parts. First, we introduce the fitness measures used in our test. Second, we present the effectiveness of our detectability test in benchmark datasets. Third, we showcase applications of the detectability test in real-world networks.

6.1 Fitness measures

Let d_i be the summation of node degree in $\mathcal{C}_i$; m_{ij} the number of edges with one end in $\mathcal{C}_i$ and the other end in $\mathcal{C}_j$, $m_{ij} = |\{(u,v)|u \in \mathcal{C}_i, v \in \mathcal{C}_j\}|$; $m_{i\setminus j}$ the number of edges with one end in $\mathcal{C}_i$ and the other end not in $\mathcal{C}_j$, $m_{i\setminus j} = |\{(u,v)|u \in \mathcal{C}_i, v \notin \mathcal{C}_j\}|$; and ρ_{ij} the density of links between $\mathcal{C}_i$ and $\mathcal{C}_j$, $\rho_{ij} = \frac{m_{ij}}{|\mathcal{C}_i||\mathcal{C}_j|}$ if $i \neq j$, and $\rho_{ii} = \frac{m_{ii}}{|\mathcal{C}_i|(|\mathcal{C}_i|-1)/2}$.

We study the following fitness measures in this paper since their variations are widely employed in the literature.

- Modularity (+): $\frac{1}{2k|E|}\sum_{i=1}^{k}(m_{ii} - E(m_{ii}))$, where $E(m_{ii}) = \frac{d_i^2}{|E|}$ is the expected number of edges between the nodes in $\mathcal{C}_i$ in a random graph with the same node degree sequence [18].
- Conductance[1] (+): $\frac{1}{k}\sum_{i=1}^{k}\frac{m_{i\setminus i}}{d_i}$.
- Modularity (−): 1− modularity (+).
- Conductance (−): 1− conductance (+).

We use + (resp., −) to indicate if high (resp., low) conductance or modularity corresponds to high fitness.

In addition, we coin a fitness measure which has no known inference algorithm, but has potential usage for grouping detection. It is based on the comparison of link density between different group pairs. We use an ordered quadruple of the group index $q = (q_1, q_2, q_3, q_4)$ to represent a desired comparison $\rho_{q_1,q_2} > \rho_{q_3,q_4}$. A set of such quadruples Q characterizes all the desirable density comparison for a grouping. For a particular grouping g, $Q^+(N,g) = \{q \in Q|\rho_{q_1,q_2} > \rho_{q_3,q_4}\}$ is the set of satisfied quadruples, and $Q^-(N,g) = Q\setminus Q^+(N,g)$ is the rest. And our measure GMoDD is defined as:

(6.4)
$$GMoDD_Q(N,g) = \frac{\prod_{q\in Q^+(N,g)}(\rho_{q_1,q_2} - \rho_{q_3,q_4})^{\frac{1}{|Q^+(N,g)|}}}{e^{|Q^-(N,g)|}}$$

GMoDD is the Geometric Mean of the Density Difference of every density pair in the satisfied quadruple set, penalized by the number of violated quadruples. By varying Q one can gear the fitness measure towards different types of groupings. For example, by making $Q = \{(i,i,i,j)|i \neq j\}$, we require that every group has denser links to itself than to other groups.

6.2 Test on benchmark

The goal of this evaluation is to validate the capability of the proposed methods on the networks with known detectability of node partitions. We use LFR benchmark graphs for the community detection task. The distributions of node degree and community size follow the power law in LFR benchmark [14]. The partition's detectability is controlled by a mixing parameter μ ($0 \leq \mu \leq 1$): each vertex shares a fraction $1-\mu$ of its edges with the other vertices of its community and a fraction μ with the vertices of the other communities.

[1] There are different definitions of conductance. We adopt the one in [15]

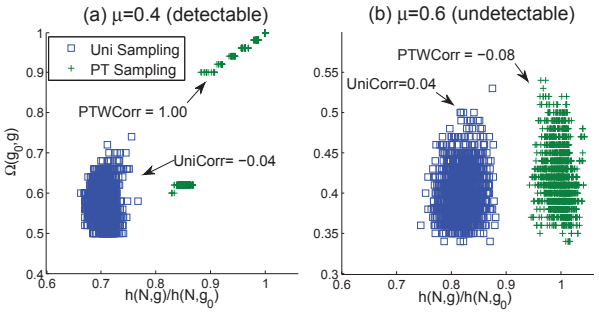

Figure 2: Scatter plot of fitness measure and proximity to target grouping for samples under uniform sampling and PT Sampling. Uniform sampling obtains samples with medium fitness which is not discriminative of detectable and undetectable cases.

We evaluate the four algorithms described in Sections 3 and 4: UniRank (ranking-based test); UniCorr (unweighted correlation via uniform sampling); UniWCorr (weighted correlation via uniform sampling); PTWCorr (weighted correlation via parallel tempering). Although we have several parameters in our testing framework, they can be decided by some principle. If there is no further specification, we use $1 - \alpha\% = 99.9\%, \beta = 0.1$ as the threshold for ranking-based and correlation-based detectability respectively, and the choice will be discussed in Section 6.2.1. For the sampling algorithms, the number of samples $M = 20K$. With confidence analysis in Section 6.2.2, this number of M is sufficient for our experiments. The sample exchange rate $\gamma = 0.15$ in PT Sampling, which is not sensitive to it. We use the following temperature schedule: $\{1, 0.4, 0.2, 0.1, 0.04, 0.02, 0.01, \ldots\}$. The number of temperatures $T = 20$, and the target temperature t_I is selected automatically such that we have sufficient samples with fitness score no worse than the highest score by 10% under t_I. So we do not need to tune t_I manually.

6.2.1 Effectiveness

We generate 100 networks, each with 100 nodes in 2 to 4 clusters, from LFR Benchmark for testing purpose. The node degree follows the power law distribution with expectation 35 and maximum value 50. The mixing parameter μ in LFR is varied from 0.3 to 0.6. Previous theoretical analysis indicated the transition between undetectable and detectable phases begins from somewhere between $\mu = 0.4$ and $\mu = 0.5$ [22]. We perform our detectability test with the fitness measure conductance $(-)$ to see whether it can produce consistent results.

Table 1 summarizes the result of using conductance as the fitness measure. The result with modularity is similar. From Table 1, we find that all the testing methods except UniWCorr roughly identify the transition from detectable to undetectable phase, albeit in different extent.

• UniRank performs well in detectable cases ($\mu \leq 0.4$). However, it is not very informative around the transition point, and misjudges 20% undetectable groupings. The reason is that the target grouping may indeed rank very high, say top 0.1% among all groupings, but the ranking does not tell anything about the accuracy of those top 0.1% groupings. Another disadvantage is that the threshold $1-\alpha\%$ must be set very high (above 0.999) to reduce false "detectable" predictions. It requires a large number of samples for confident estimates of whether the ranking is above threshold.

• UniCorr produces correlation coefficients in a narrow range of mean (-0.09 to 0.36). Though the mean of the correlation efficients for μ from 0.3 to 0.6 roughly has a decreasing trend, the variance is high and they do not separate detectable and undetectable cases well. For example, it produces a negative score -0.0012 for a detectable graph with $\mu = 0.4$, while both UniRank and PTWCorr predict 1. From Figure 2 we can see the reason. The uniform sampling obtains most samples with medium fitness score, whereas the detectable and undetectable networks are mainly distinguished by the samples in the high fitness region.

• UniWCorr produces even worse results, with many negative correlation coefficients even for undetectable networks. The reason is the same as we analyzed above: the uniform sampling misses most samples with high fitness score, and assigns inappropriate high weight to samples with medium or low fitness.

• PTWCorr outputs high correlation (mostly 1.0) when $\mu = 0.3$ and $\mu = 0.4$, low correlation for $\mu = 0.5$ with relatively larger variance, and mostly negative correlation coefficient for $\mu = 0.6$. The large variance for $\mu = 0.5$ is due to the fact that 0.5 is close to the transition point and some networks with $\mu = 0.5$ happen to present weak detectability because of randomness in the graph generation. Therefore, it reveals more information than UniRank when a network is neither strongly detectable nor strongly undetectable. We note that the fitness and proximity may present weak positive correlation in the high fitness region even in undetectable cases. So the threshold β should be set as a small positive value such as 0.1 instead of 0.

In summary, UniRank is good in detectable and strongly undetectable cases, and PTWCorr has strongest capability when it is difficult to determine the detectability. They both outperform the other two methods and should be combined to perform a compre-

Table 1: Test on LFR networks. UniRank ranges from 0 to 1, and the other three range from −1 to 1. Higher value implies more detectable. The left part shows the output mean and deviation, and the right shows the output range (bold means no overlap with the opposite case). UniRank produces "false detectable"; PTWCorr separates the detectable and undetectable cases well; and other two methods confuse the two cases.

μ	0.3 (detectable)	0.4 (detectable)	0.5 (transition)	0.6 (undetectable)	detectable	undetectable
UniRank	1.000 ± 0.000	1.000 ± 0.000	0.963 ± 0.069	0.226 ± 0.400	[1.000,1.000]	[0.000,1.000]
UniCorr	0.175 ± 0.104	0.171 ± 0.097	0.063 ± 0.060	-0.029 ± 0.053	$[-0.001, 0.363]$	$[-0.094, 0.075]$
UniWCorr	0.093 ± 0.821	0.049 ± 0.050	-0.111 ± 0.471	-0.082 ± 0.534	$[-1.000, 1.000]$	$[-0.940, 0.972]$
PTWCorr	1.000 ± 0.000	0.998 ± 0.009	0.235 ± 0.446	-0.013 ± 0.089	**[0.955,1.000]**	$[\mathbf{-0.117}, \mathbf{-0.063}]$

hensive test.

6.2.2 Efficiency

We analyze the running time of three parts in our methods: generating samples, computing the fitness and proximity of each sample, and calculating the detectability. The first part is linear to the number of samples M and network size $|V|$. The second part depends on the fitness measure to test. For conductance and modularity, the complexity is linear to M and the number of edges $|E|$ in the network. The third part is $O(M \log M)$ for correlation and $O(M)$ for ranking. So the overall complexity of UniRank and PTWCorr are $O(M|E|)$ and $O(TM|E| + M \log M)$ respectively. PTWCorr has to generate T replicas of samples though eventually only one replica of samples will be used for correlation computation, so it costs roughly T times of UniRank in order to test with equal number of samples.

The average time for performing UniRank test with 1K samples is 3.15s in LFR networks with 1M edges on our CentOS server running MATLAB R2011a with Intel Xeon X5687 3.6GHz and 96GB RAM. And the confidence interval is below 0.02 for both sampling when the number of samples is greater than 3K. So the cost of our algorithms is acceptable. We need to point out that the fitness measure computation is the dominant cost in our experiments.

Considering both effectiveness and efficiency, we suggest a two-stage detectability test in practice: in the first stage, run UniRank for a quick ranking-based test; and only if it is asserted as detectable, run PTWCorr for a slower but more precise test. Failure in either test suggests undetectability.

6.3 Application

We apply our detectability test in real networks. We show the application of finding suitable fitness measures for the following grouping detection task. A grouping is deemed to be detectable by a fitness measure if it passes both ranking-based test UniRank and correlation-based test PTWCorr.

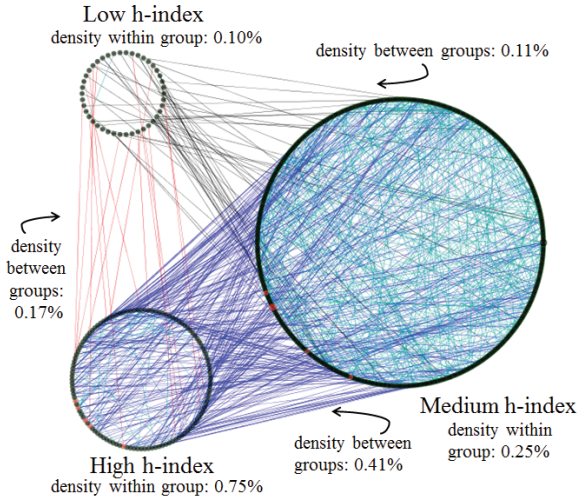

Figure 3: Co-author network of 2000 data mining researchers, grouped by their h-indices as high (> 9), medium (2-8) and low (0-2). In general the links between group (a, b) are denser than (a, c) if group b has higher indices than c.

We use a co-author network of 2000 data mining researchers [11]. Instead of grouping them by research topics, we group them into 3 groups according to their h-indices: high (group 1), medium (group 2) and low (group 3). We test the detectability of this grouping by modularity $(+)/(-)$ and conductance $(+)/(-)$. None of these four fitness measures can be used to detect the author groups by h-indices, according to our test. The reason can be found from the visualization in Figure 3: it is no longer true that most authors collaborate with other authors in the same h-index group, neither the opposite. Instead, authors with lower h-index have a tendency to collaborate with authors with higher h-index. With that intuition, we propose to use a new fitness measure $GMoDD_Q$ (Eq. (6.4)), and specify $Q = \{(i, j, i, j+1) | i = 1, 2, 3, j = 1, 2\}$. We perform detectability test for this measure. As shown in Table 2, the result is promising and encourages an inference algorithm for optimizing $GMoDD_Q$.

Table 2: Test on data mining co-author network. A grouping is predicted to be detectable only if $UniRank > 1 - \alpha\% = 0.999$ and $PTWCorr > \beta = 0.1$.

	UniRank	PTWCorr	Detectability
Modularity (+)	0.931(<)	−0.706(<)	undetectable
Modularity (−)	0.071(<)	−0.006(<)	undetectable
Conductance (+)	**1.000**(>)	−0.092(<)	undetectable
Conductance (−)	0.986(<)	**0.226**(>)	undetectable
$GModD_Q$	**1.000**(>)	**0.504**(>)	detectable

7 Conclusions

In this paper, we study the novel problem of node grouping detectability test. Our recipe is based on two conditions for a grouping to be detectable by a fitness measure: the target grouping must present strong fitness with the network structure than most other groupings; and a random grouping that is more similar to the target grouping should in general have higher fitness score. The experiments show that our approach enables the estimation of achievable detection performance *a priori*. It provides a tool for validation of known criteria on new detection tasks and for exploration of new types of structures when a grouping is undetectable with classical criteria.

8 Acknowledgements

This work was supported in part by the U.S. National Science Foundation grants IIS–0905215 and DMS–1106796, U.S. Army Research Laboratory under Cooperative Agreement No. W911NF–09–2–0053 (NS-CTA). Chi Wang was supported by Microsoft Graduate Fellowship. The authors want to thank Ruoming Jin and Victor E. Lee for sharing the data, and Manish Gupta and Yizhou Sun for their helpful discussion.

References

[1] B. Abrahao, S. Soundarajan, J. Hopcroft, and R. Kleinberg. On the separability of structural classes of communities. In *KDD '12*, 2012.

[2] A. Anagnostopoulos, R. Kumar, and M. Mahdian. Influence and correlation in social networks. In *KDD'08*, pages 7–15, 2008.

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

[4] T. Carson and R. Impagliazzo. Hill-climbing finds random planted bisections. In *Proc. ACM-SIAM Symposium on Discrete Algorithms (SODA '01)*, 2001.

[5] S. Chib and E. Greenberg. Understanding the metropolis-hastings algorithm. *The American Statistician*, 49(4):327–335, 1995.

[6] C. J. Clopper and E. S. Pearson. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):pp. 404–413, 1934.

[7] A. Condon and R. M. Karp. Algorithms for graph partitioning on the planted partition model. *Journal of Random Structures and Algorithms*, 18:116–140, 1999.

[8] L. Danon, J. Duch, A. Diaz-Guilera, and A. Arenas. Comparing community structure identification. *JSTAT*, 2005(09):10.

[9] M. W. Deem. Parallel tempering: Theory, applications, and new perspectives. *Phys. Chem. Chem. Phys.*, 7(23):3910–3916, 2005.

[10] S. Fortunato. Community detection in graphs. *Physics Reports*, 486:3-5, 2010.

[11] R. Jin, V. E. Lee, and H. Hong. Axiomatic ranking of network role similarity. In *KDD '11*, 2011.

[12] R. Kannan, S. Vempala, and A. Vetta. On clusterings: Good, bad and spectral. *J. ACM*, 51(3):497–515, 2004.

[13] D. A. Kofke. On the acceptance probability of replica-exchange Monte Carlo trials. *The Journal of Chemical Physics*, 120:10852–10852, 2004.

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

[15] J. Leskovec, K. J. Lang, and M. Mahoney. Empirical comparison of algorithms for network community detection. WWW '10, 2010.

[16] J. Liu. *Monte Carlo strategies in scientific computing*. Springer Verlag, 2008.

[17] B. Long, X. Xu, Z. Zhang, and P. S. Yu. Community learning by graph approximation. In *ICDM '07*, 2007.

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

[19] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS '01*, 2001.

[20] T. E. Nichols and A. P. Holmes. Nonparametric permutation tests for functional neuroimaging: a primer with examples. *Human Brain Mapping*, 15(1):1–25, 2002.

[21] M. Onsjö and O. Watanabe. A simple message passing algorithm for graph partitioning problems. In *Intl. Symposium on Algorithms and Computation (ISAAC 2006)*, 2006.

[22] J. Reichardt and M. Leone. (Un)detectable cluster structure in sparse networks. *Phys. Rev. Lett.*, 101(7):078701, 2008.

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

[24] S. Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27–64, 2007.

[25] J. Shi and J. Malik. Normalized cuts and image segmentation. In *CVPR '97*, 1997.

[26] S. White and P. Smyth. A spectral clustering approach to finding communities in graphs. In *SDM '05*, 2005.