Toward Small Community Discovery in Social Networks

by

Ran Wang

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved April 2015 by the
Graduate Supervisory Committee:

Huan Liu, Chair
Charles Colbourn
Arunabha Sen

ARIZONA STATE UNIVERSITY

May 2015

ABSTRACT

A community in a social network can be viewed as a structure formed by individuals who share similar interests. Not all communities are explicit; some may be hidden in a large network. Therefore, discovering these hidden communities becomes an interesting problem. Researchers from a number of fields have developed algorithms to tackle this problem.

Besides the common feature above, communities within a social network have two unique characteristics: communities are mostly small and overlapping. Unfortunately, many traditional algorithms have difficulty recognizing these small communities (often called the resolution limit problem) as well as overlapping communities.

In this work, two enhanced community detection techniques are proposed for reworking existing community detection algorithms to find small communities in social networks. One method is to modify the modularity measure within the framework of the traditional Newman-Girvan algorithm so that more small communities can be detected. The second method is to incorporate a preprocessing step into existing algorithms by changing edge weights inside communities. Both methods help improve community detection performance while maintaining or improving computational efficiency.

To my beloved family and friends.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Huan Liu, for his guidance, encouragement, and support during my dissertation research. He is an outstanding mentor, an easygoing friend, and the most dedicated researcher I have ever known. The experiences with him are my lifelong assets. I would like to thank my dissertation committee members, Charles Colbourn, Arunahba Sen for their valuable interactions and feedback.

Members of our Data Mining and Machine Learning Lab inspired me a lot through discussions, seminars, and project collaborations. Particularly, I would like to thank Isaac Jones for his valuable suggestions in both research and writing.

This material is based upon work supported by the U.S. Army Research Office (ARO) under contract/grant number W911NF-12-R-0012: Modeling of Complex Systems: Mining Suspicious Tiny Sub-Networks in a Massive Social Network.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Chapter 1

INTRODUCTION

1.1 Social Networks and Communities

Everyone has their own social network consisting of their friends, family and colleagues. A *social network* is a network that consists of people and their interpersonal relationships, such as kinship, friendship, classmates, colleagues, etc.. People also gain new friends from time to time, social networks are not static. Connections are generally built one at a time. Online social networks record pre-existing interpersonal relationships and are updated to show new relationships. The explosion of online social networks, such as Facebook, LinkedIn, Flickr show the importance of keeping track of social relationships in our daily life.

A simple abstraction or visualization for a social network is the graph structure. By viewing each person as a node and their interpersonal relationship (friends) as links, we can visualize social networks as graphs. Different from other network types, social networks are nearly always sparse. Another differentiating factor is that the node degrees in social networks are power-law distributed. Moreover, different from random networks with the same degree distribution, social networks have a distinct and important structure – communities.

Communities are very important in social networks. In addition to links, communities may be held together by common interests, common goals or geographical location. This means that links alone do not define community membership, other factors, like common interests defines community membership, though members of the same community are more likely to be friends.

Reseachers have used many different ways of defining communities. In this work, I adopt the following definition based on modularity:

Definition 1 (Community) *Let $G = (V, E)$ be a graph and $|E| = m$. Let $\mathcal{C} = \{C_1, C_2, ..., C_N\}$ is a cover of V. A community $C \in \mathcal{C}$ is defined as a set of nodes such that the modularity Q is maximum, where Q is defined as:*

$$Q = \frac{1}{N} \sum_{i} \sum_{u,v \in C_i} [A_{uv} - \frac{k_u k_v}{2m}]$$

where $A = (A_{uv})$ is the adjacency matrix and k_v is node v's degree.

A thorough explanation of modularity is given in Section 3.1.

In the context of a social network, a real-world community is a group of nodes that share the same interest, property, or location. For example, the communities within the Amazon Co-purchasing Network are the connected components of the 'people who buy this also buy' feature, while the communities within the Youtube Network are just user-defined groups. We then can evaluate the quality of communities from Definition 1 against these real-world communities.

1.2 Detecting Small Communities

Communities are not always explicit in a social network. Thus, how to detect implicit communities has been an important problem for us to better understand social networks. For this task, we assume that no information is provided except the nodes and the links. Adding more features from a social network will generate better results, but adds complexity and is not discussed here.

Many algorithms have been proposed recently, including many not specifically designed for social networks. These methods have been successful in fields not limited to discovering communities in social networks. See Chapter 2 for details.

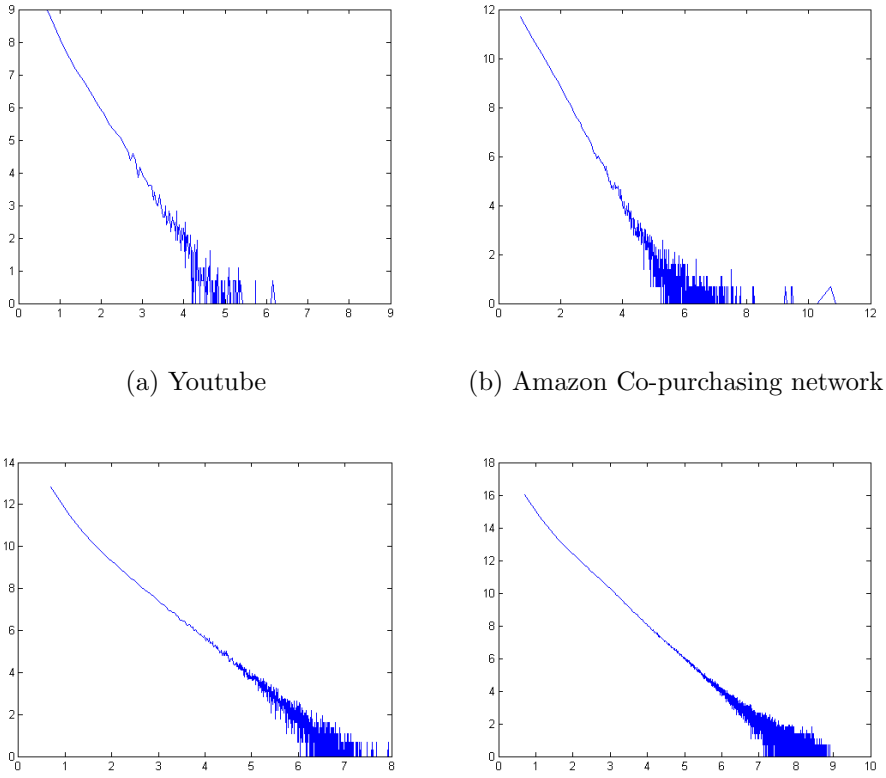
However, in social networks, communities show different characteristics from those of other subjects, such as biology and physics. For example, in some subjects, community sizes are more likely to be evenly distributed. That is, they are all roughly the same size in a network. Within a social network, not all of these groups are the same size. A small group may consist of only a few several people while a large group consists of dozens of thousands of people. Agarwal *et al.* (2007) was among the first to explore this relationship. Tang and Liu (2010) later provided further supporting evidence. We also confirm this analysis in Figure 1.1. The power-law distribution of real community sizes reveals the fact that most of the communities in a social network are small.

This presents a challenge: although detecting these groups while they are large in size is a well-studied problem as we discuss in Chapter 2, detecting small communities is not well-studied. Some evidence is shown in Figure 1.2. The algorithms of Infomap and FastModu both detect a number of large communities. However, in this dissertation, we aim to search for small communities. That is, we want to detect more small communities that consist of 5 or more nodes.

1.3 Basic Notation and Problem Formulation

We discuss undirected social networks in this dissertation, since the symmetric adjacency matrix can be easily handled as shown in Section 3.3.1. Notice that a real social network can be asymmetric (Twitter, Weibo, etc.). That is, when one is following the other, it is not necessarily true that the other person also follows back. We symmetrize a directed network simply by making the directed links undirected, though we may change the network by making followers as followees.

Some basic notation is as follows. Let $G = (V, E)$ be the graph associated with the network, with the node set V and the edge set E. G is a sparse matrix since it



(a) Youtube

(b) Amazon Co-purchasing network

(c) LiveJournal

(d) Orkut

Figure 1.1: x-axis: natural log of community size. y-axis: natural log of occurrence of community size. Data is from snap.stanford.com (Yang and Leskovec (2012))

represents a real-world social network. n and m are the numbers of nodes and edges in G, respectively. A is represented as the adjacency matrix of G, where $A = (A_{ij})$:

$$A_{ij} = \begin{cases} 1 & \text{if } ij \in E \\ 0 & \text{if } ij \notin E \end{cases}$$

Let $N(v)$ denote the neighborhood of a vertex $v \in V$, which is the set of vertices adjacent to v. Let k_v denote the degree of the vertex v, where $k_v = |N(v)|$. A *cover* of a set S is a collection of subsets $\mathcal{P} = \{P_1, P_2, \cdots, P_n\}$ so that $\bigcup_i P_i = S$. A *partition* of a set S is a collection of disjoint subsets $\mathcal{P} = \{P_1, P_2, \cdots, P_n\}$ such

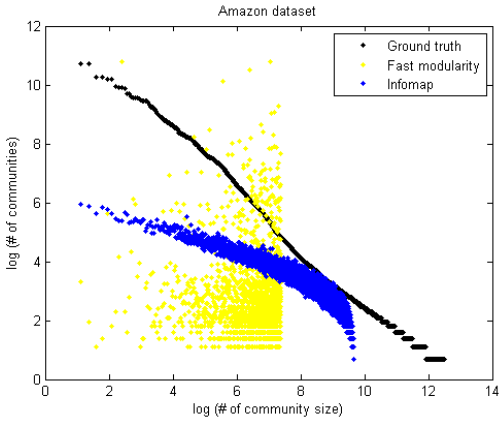


Figure 1.2: The Distribution of Community Sizes of the Ground-Truth v.s. the Results of Some Existing Algorithms

that $\bigcup_i P_i = S$. A *community assignment* is a vertex cover for the network. Here the condition $\bigcup_i P_i = S$ is required as each vertex should belong to at least one community, which is viewed as a convention in the subject of community detection.

The base community detection problem is therefore to partition the vertices of the graph G into a set of communities $\mathcal{C} = \{C_1, ..., C_k\}$ such that the partitioning is representational of the hidden community assignments that underly the network. In our unique variant of community detection, we seek to emphasize the small communities. Thus, we want to minimize $|C_i|$ for each community C_i while maintaining the quality of detected communities. One additional requirement is that $|C_i| \geq 5$. Our objective is enable the finding of small communities with varied sizes.

Two methods are proposed for solving the problem. One method is to introduce a modified measure of modularity, discussed in Chapter 3. The other is to use a reweighting method as a preprocessing step for existing community detection algorithms, discussed in Chapter 4. Both methods are successful in finding small communities while maintaining the quality of detected communities.

Chapter 2

RELATED WORK

2.1 Introduction to Community Detection Algorithms

Multiple overlapping community detection algorithms have been developed in the recent years. A survey (Xie *et al.* (2013)) was written recently, with almost all the proposed overlapping community detection algorithms before 2011 involved. Here, we mainly focus on some important and useful algorithms that can be helpful or have potential to generate small communities. Besides, we do not limit our sight in overlapping detection methods, but also non-overlapping methods.

2.1.1 Fuzzy Detection (Probabilistic Model)

Fuzzy detection is one kind of community detection algorithms that we do not decide if one node belongs to one specific community. Instead, we use a vector of numbers to represent the probability for each single node. We call the vector as *belonging vector*. More specifically, we can write a $N \times C$ matrix A where $\sum_{j=1}^{C} A_{ij} = 1$ for all $j \in \{1, 2, \cdots, N\}$. One obvious problem for such algorithms is that the community number C is hard to determine.

A lot of algorithms have been proposed using this basic idea. As one of the earliest examples, fuzzy c-means (Pedrycz (1990)) modifies the method of original K-means by adding a probability to show that one node can probably belong to multi communities. But it also inherits the problem from the original K-means that determining C is hard.

As a model-based community detection algorithm, MOSES (McDaid and Hurley

(2010)) shows that probability can play an important role in community detection. The probabilistic model assumes that when two nodes are connected by one link, there is a greater chance that these two nodes belong to the same communities. Therefore, we have a latent factor of probability for the links which determines the community structure. Then we want to maximize this probability by Bayes theory. The solution for this model is to add/delete edges between network to maximize the likelihood function value. EM algorithm will give an approximate optimization. The author also claimed that the approximate solution can be trapped at a local minimum which can be far away from the global maximum. However, no solution has been given in the work.

Yang and Leskovec (2012) better describes the theoretical basis for the probability model. It uses 6 real-world networks to show that there are more edges for the overlapping part of communities. From this fact, the idea of the algorithm is very similar to MOSES, except that it introduces affiliation graph instead of community-node vector. And the affiliation graph is updated from Metropolis-Hastings algorithm by randomly deleting/adding/switching one edge. The same author introduces another improved method (Yang and Leskovec (2013)) based on this paper. They consider affiliation network as a latent factor that can generate the edges of the original graph.

2.1.2 Label Propagation

Compared to the other kinds of community detection algorithms, label propagation algorithm has a natural advantage that by its quasi-linear time complexity. Thus, the algorithm is able to complete the task in more complex networks. Thus, even if the method didn't have a very good performance at first, it still attracted vast attention of researchers.

COPRA (Gregory (2010)) can be considered as one of the earliest such algorithms.

From the view of belonging vector, each node updates its belonging coefficients by averaging the coefficient from all its neighbors at each step symmetrically. It limits its maximum number of communities v to save time and space.

DEMON (Coscia *et al.* (2012)) is another such algorithm. In the algorithm, the author uses Ego network, which is a vertex's neighborhood along with the node itself. For each step, we propagate the label for this Ego network and view this set as a community. When we cannot update any of these communities, we stop and delete the communities of extreme high overlap. Since we do propagate the label from each vertex, we can reach a set of different overlapping communities finally.

Similar to DEMON, SLPA (Xie *et al.* (2011)) artificially provides the overlapping communities results. I leave the detail of the algorithm to Section 4.3.

2.1.3 Information Theory

As a recent popular method, Infomap has proven its success on the community detection problems (Fortunato (2010)). Again, I leave the detail of the algorithm to Section 4.3.

2.1.4 Modularity-based algorithm

Modularity (Newman and Girvan (2004)) was originally developed as a measure to determine the quality of a graph partition. In this work, we definition community with modularity. A thorough introduction and an improvement of the measure is written in Chapter 3 .

2.1.5 Local-based expansion

The idea of local-based expansion is that, by fixing a seed node and adding it into an empty set A, we expand the set from some specific measure. We continue

when the measure is going to optimal, otherwise we will stop. The advantage is that by exploring the local structure, the algorithm will always be computationally cheap and thus scalable to a large network. Also, there are several disadvantages: (1) The seeds are always hard to find. Therefore, people proposed the seeding strategy to solve the problem. (2) Quite opposite to the global strategy, the local method can always trap communities to extreme small sizes. Therefore, each local method need to add different strategies to avoid the problem.

Here we list some of the local methods. Notice that for all the local methods, we can naturally assume it generates a set of overlapping communities, since start from two different seeds from the same communities can generate two communities which are different with each other.

Andersen *et al.* (2006) proposed a PageRank based method. There is a random walker starting from a seed vertex. At each time it moves towards its neighbors by probability α or restart from the starting seed by $(1-\alpha)$. Then the author concludes that they can find a nice community by its Pagerank vector. This idea is further adapted by several papers such as Gleich and Seshadhri (2012), Whang *et al.* (2013).

Gleich and Seshadhri (2012) analyzes the quality of Ego networks. They prove that there is at least one high-quality community in a social network. Also, they suggest the nodes of low conductance are the seeds that can be expanded into some high quality communities.

Whang *et al.* (2013) is a recent work that implements the paper (Andersen *et al.* (2006)). As a complete community detection algorithm, it consists of four phases: filtering, seeding, expansion and propagation. And for seed expansion, the algorithm implements the above PageRank based method. For filtering phase, the algorithm deletes the 'tail' part to find the biconnected component. For propagation phases, the existing communities grow the whole communities by recovering the whole com-

munities from the biconnected core. The algorithm improves the work in (Gleich and Seshadhri (2012)) by classifying more nodes into communities.

Another idea is to generate communities from seed communities. In this way problem (2) is likely to be avoided. However, finding seed communities always involve seeking for small density cores, which is considered as an expensive preprocessing step. EAGLE (Shen *et al.* (2009)) is a such agglomerative method. First, all maximal cliques are found to be the initial communities. Then, the communities with highest similarity merge with each other. The algorithm stops when the modified modularity with overlap reaches the maximum.

Similar to EAGLE, GCE (Lee *et al.* (2010)) identifies the maximum cliques as the seed communities. Then it expands these seeds by greedily optimizing a local fitness function. GCE also deletes the communities that are highly similar to each other afterwards.

2.1.6 Other Algorithms

The clique percolation method (CPM) (Palla *et al.* (2005)) is based on the assumption that a community consists of a set of adjacent cliques. The method starts from identifying all the cliques size k (typically 3 to 6). Then the algorithm treats all the clique as nodes. Two nodes are connected to each other when the cliques share $k-1$ nodes. After this construction, all the connected components are recognized as the communities. Since one node can be in multiple cliques which do not necessarily connect to each other, the algorithm can find overlapping communities. The algorithm works pretty well with a high density core. But it cannot terminate for a real social network with large size. It is suspicious that the algorithm will provide satisfying results for the social network because the social network is sparse as well as scale-free. CPM can be viewed as an expansion of seed communities with high time

complexity.

Girvan-Newman Algorithm (Newman and Girvan (2004)) is another algorithm that have been widely applied. I leave the detail to Section 3.2.2, as techniques in Chapter4 are mostly based on this algorithm.

Gregory (2007) developed an algorithm CONGO that can detect overlapping communities from Girvan-Newman algorithm. The idea is to split vertices with high betweenness. The algorithm still derives the complexity of the original Girvan-Newman algorithm. Furthermore, (Gregory (2008)) developed another improved algorithm by using local information only. More specifically, he calculated the betweenness measure only through the paths $\leq t$ where t is a constant. Although this is only an approximation, he shows from experiments that this method can provide good results.

Link partition method (Ahn *et al.* (2010)) is explored as another version of Girvan-Newman algorithm. The difference is that instead of considering the original graph, we look at its line graph, that is, view the edges as the vertices. In this way, we allow one link in one community, but also one node can be in multiple communities. At last we build a link dendrogram to find communities. The algorithm stops when we reach the maximum modularity. The algorithm detects overlapping communities, as two different links incident to the same edge can belong to different detected communities and thus the node incident to the link can belong to different communities.

Spectral method (Shi and Malik (2000)) is another very large class in community detection area. A representative spectral method is METIS (Karypis and Kumar (1995)). The philosophy of the method is that we can subdivide a cluster to smaller clusters by the technique of eigenvectors. Then we can incorporate different modules to some bigger clusters to reach optimum. However, it seems hard to generalize spectral methods to overlapping community detection. Only a few algorithms implement

the idea. Zhang *et al.* (2007) proposed an algorithm. Given the community number k, the top $k-1$ eigenvectors are calculated. Then the graph nodes are projected to a vector space of $d \leq k-1$ dimension. Then we use fuzzy c-means to obtain a soft assignment. The accuracy is heavily dependent on the value k, which we cannot pre-determine.

Assortativity is the property of social network that high degree edges are more likely to be connected to each other. The paper (Ciglan *et al.* (2013)) discusses the problem of how assortativity can affect the precision of community detection methods. The conclusion is that community detection methods can better partition a graph with higher assortativity. And if we reweight the graph by its assortativity measure for these graphs, it will further increase the precision of these algorithms.

2.2 Scalability

There is no paper aiming at the scalability analysis in particular. Therefore, we search all the papers for each algorithm's complexity. Our propose is to distinguish the scalable algorithms from those of high complexity, since the scalable algorithms are more applicable to large networks. We summarize the complexity of the above algorithms as follows:

Algorithm	From	Complexity
CFinder (CPM)	Palla *et al.* (2005)	Polynomial
Infomap	Rosvall and Bergstrom (2008)	$O(tm \log n)$
SLPA	Xie *et al.* (2011)	$\tilde{O}(tm)$
Link	Ahn *et al.* (2010)	$O(nk_{max}^2)$
Spetral	Shi and Malik (2000)	$O(kn^2)$
METIS	Karypis and Kumar (1995)	$O(n^2 \log n)$
GCE	Lee *et al.* (2010)	$O(mh)$
CIS	Kelley (2009)	$O(n^2)$
CONGO	Gregory (2007)	$O(m^2 n)$
CONGA	Gregory (2008)	$O(m^2 \log(n))$
PageRank	Andersen *et al.* (2006)	$\tilde{O}(mn)$
CNM	Clauset *et al.* (2004)	$O(dm \log(n)$
Louvain	Blondel *et al.* (2008)	$O(m \log(n))$ *
GN	Newman and Girvan (2004)	$O(m^2 n)$
DEMON	Coscia *et al.* (2012)	nK
MOSES	McDaid and Hurley (2010)	*
Seed Expansion (SE)	Whang *et al.* (2013)	$O(km)$

The notation is as follows: m is the number of edges. n is the number of nodes. k is pre-determined number of communities. K is the number of seeds. d is the depth of dendrogram. t is the number of iterations. k_{max} is the maximum degree.

(*) means the original paper does not imply anything about the complexity. For Louvain algorithm, only an approximate complexity is given since the authors claimed that it is hard to evaluate its real complexity. Fortunately the two algorithms marked as (*), Louvain and MOSES, can run a network at least as large as Amazon Co-

purchasing network (Yang and Leskovec (2012)) within a reasonable time.

From the table, we can see that not many algorithms are scalable. Infomap, SLPA, GCE, Louvain, CNM, DEMON and SE are of linear complexity. Only GCE, SLPA, DEMON and SE are generating overlapping communities. These are all local methods.

Among all these scalable algorithms, SLPA is not so efficient with its time and space complexity. It cannot perform on LiveJournal dataset from SNAP networks (Yang and Leskovec (2012)), which is of about 4 million nodes and 34 million edges. According to the personal communication, the author claimed that the implementation is not efficient enough. Also from the above, we can see that the dynamic algorithms (label propagation, random walk, seed expansion) have its advantage over the others since it can usually discover overlapping communities in linear time.

2.3 Evaluation Measures

For detecting small and overlap communities, one measure seems not to be enough to reveal if one algorithm is good or not based on our current experiments. What is more difficult is for small communities, we cannot find any evaluation measure that can specially apply to small communities. A combination of different measures from below may be a good choice for our purpose. Here we collect a set of popular measures with different characteristics:

2.3.1 Normalized mutual information

NMI is a standard measure that is used to compare the similarity of two partitions of a network. Lancichinetti *et al.* (2009) proposed a generalized NMI in their work so that it can compare different overlapping communities partitions. Suppose we have two different partitions $\mathbf{X} = \{X_1, X_2, \cdots X_N\}$ and $\mathbf{Y} = \{Y_1, Y_2, \cdots Y_M\}$. Then NMI

is defined as:

$$NMI(\mathbf{X}|\mathbf{Y}) = 1 - (H(\mathbf{X}|\mathbf{Y})_{norm} + H(\mathbf{Y}|\mathbf{X})_{norm})/2.$$

$$H(\mathbf{X}|\mathbf{Y})_{norm} = \frac{1}{N}\sum_{k}\frac{\min_l H(X_k|Y_l)}{H(X_k)}.$$

$$H(\mathbf{Y}|\mathbf{X})_{norm} = \frac{1}{M}\sum_{k}\frac{\min_l H(Y_k|X_l)}{H(Y_k)}.$$

where $H(X|Y)$ and $H(Y|X)$ are conditional entropy.

When we experiment on real world datasets, we found the fact that for networks with highly overlap and small communities, NMI may not be a good measure . CNM (Clauset *et al.* (2004)) has a very high NMI measure by detecting the right large communities. However, as the large communities takes a really small portion of the overall network with also many small communities contained inside and we do not care about huge communities so much, this may be biased. Except this, NMI is a highly reliable measure.

2.3.2 Omega Index

Omega index (Gregory (2011)) calculates the agreement of two covers on the same pairs. The agreement is defined as the same number of occurrences on the pairs for both sets. Thus, omega index determines how many pairs are clustered right for all occurrences in communities. And we only care about the pairs, but not the communities itself. Here is the definition:

Let K_1 and K_2 be the number of the communities in covers C_1 and C_2. Then the omega index is defined as

$$w(C_1, C_2) = \frac{w_u(C_1, C_2) - w_e(C_1 - C_2)}{1 - w_e(C_1, C_2)}$$

where

$$w_u(C_1, C_2) = \frac{1}{M}\sum_{j=0}^{\max(K_1,K_2)} |t_j(C_1) \cap t_j(C_2)|$$

$$w_e(C_1, C_2) = \frac{1}{M^2} \sum_{j=0}^{\max(K_1,K_2)} |t_j(C_1)| \cdot |t_j(C_2)|$$

where $M = n(n-1)/2$ and $t_j(C)$ is the number of node pairs that appear exactly j times in cover C.

2.3.3 F-measure

F-measure accounts for the balance between the quantity and quality for the overlapping nodes for detected communities. More specifically, it is defined as:

$$F = \frac{2 \cdot precision \cdot recall}{precision + recall}$$

where recall is the number of correctly detected overlapping nodes divided by the true number of overlapping nodes, and precision is the number of correctly detecting overlapping nodes divided by the total number of the detected overlapping nodes. The measure reaches its best and worst value at 1 and 0, respectively.

F-score is a good measure when we consider if a single node is successfully marked as the overlapping nodes or not by community detection algorithm. For its deficiency, it considers only for the single node, not the whole community.

2.3.4 Jaccard Index

Jaccard Index (Ball *et al.* (2011)) is defined as

$$S(e_{ik}, e_{jk}) = \frac{|S \cap V|}{|S \cup V|}$$

where S is the set of vertices in the true overlap and V is the set of vertices the algorithm identifies as being in the overlap. Thus the range of Jaccard Index is [0,1]. And when the index is 1, all the overlapping nodes are identified. When the index is 0, none of the overlapping nodes are identified. The measure is simple, and it only measures if the overlapping nodes are detected or not.

2.3.5 Comsim

Comsim (Ciglan *et al.* (2013)) is a generalization of Jaccard Index. A simple explanation is that for partition $\mathcal{P} = \{P_1, ..., P_k\}$ and $\mathcal{Q} = \{Q_1, Q_2, ..., Q_l\}$, we calculate for each part the most similar one using the Jaccard Index and then integrate all this information together. It seems that there are similar properties between Comsim and NMI. However, the paper (Ciglan *et al.* (2013)) show their difference with respect to the empirical results. The formal definition is as follows:

$$cossim(\mathcal{P}, \mathcal{Q}) = \frac{\sum_{P_i \in \mathcal{P}} sim(P_i, \mathcal{Q})}{|\mathcal{P}|}$$

where

$$o(T, \mathcal{Q}) = \{P : P \in \mathcal{P} \wedge \forall P_i \in \mathcal{P} | P_i \cap T| \leq |P \cap T|\}$$

$$b(T, \mathcal{Q}) = \{P : P \in o(T, \mathcal{Q}) \wedge \forall P \in o(T, \mathcal{Q}) | P_i| \geq |P|\}$$

$$sim(P, \mathcal{Q}) = J(P, Q) : Q \in b(P_i, \mathcal{Q})$$

Comsim is a better measure than the Jaccard Index itself since it incorporates community information. The measure is not widely used yet.

2.3.6 Overlap Rate

This is introduced in the same paper of link partition (Ahn *et al.* (2010)). The paper introduces 4 measures for evaluating the quality of the overlapping communities. (1) Overlap coverage: how much overlap was discovered. (2) Community coverage: how much of the network was classified by each algorithm. (3) Overlap quality and (4) community quality: similarity of the nodes they contain. By adding these four measures, we can get a total score that reflects the result well. The measure is adopted by Ahn *et al.* (2010), Yang and Leskovec (2013), Yang and Leskovec (2012).

2.4 Reweighting Methods

In this part, I list the literature on the reweighting techinque, which I explore in Chapter 4. In social media mining, people are usually researching the reweighting methods for two main purposes: assortativity mixing and to avoid the modularity resolution limit.

Assortativity mixing (Newman (2002)) is a preference for the network's nodes to attach to others that are similar in some way. In social networks, highly connected nodes tend to be connected with other highly connected nodes. For the relationship between the concept of our work, Ciglan *et al.* (2013) has shown that degree assortativity reflects, to some extent, the precision of community detection algorithms. If the graph is more assortative, the detected communities show higher precision if we replace the adjacency matrix with assortativity.

People also have been trying some reweighting methods to solve the resolution limit problem. Berry *et al.* (2011) developed a reweighting measure by applying the node's neighbor information. Khadivi *et al.* (2011) implemented a weighting scheme for intra-cluster and inter-cluster weight by multiplying with edge betweenness. Lai *et al.* (2010) proposed another preprocessing step using random walks. Their idea is to calculate similarity between nodes based on random walk pattern similarity.

Chapter 3

MODULARITY AND COMMUNITY DETECTION

3.1 Modularity

In this section, I write a thorough explanation of modularity, which is used for defining community. Modularity was proposed in Newman and Girvan (2004) and, as Chapter 2 already showed, went on to be used by many algorithms.

One of the ways to formally define modularity is as follows:

$$Q = \sum_{ij \in E} [\frac{A_{ij}}{2m} - \frac{k_i k_j}{(2m)^2}] \delta(c_i, c_j) \tag{3.1}$$

where the δ function is defined as:

$$\delta\left(c_i, c_j\right) = \begin{cases} 1 & c_i = c_j \\ 0 & c_i \neq c_j \end{cases} \tag{3.2}$$

Modularity is the fraction of the edges that fall within the given groups minus the expected such fraction if edges were distributed at random. The value of the modularity lies in the range [-1/2,1). It is positive if the number of edges within groups exceeds the number expected on the basis of chance. For a given division of the network's vertices into some community assignment, modularity reflects the concentration of edges within communities compared with the random distribution of links between all nodes.

3.1.1 Resolution Problem

As mentioned previously, modularity has an inherent limit to the size of communities it is capable of distinguishing. This limit indicates that modularity may not be a good evaluation measure for detecting small communities in the large social networks. An example is that with the presence of huge well-defined communities (cliques), small communities (also cliques) can be misclassified and thus very few impacts to the modularity will be made. The proof of this resolution limit is given in Fortunato and Barthelemy (2007).

The crux of the proof is that for a network of sufficiently large size and given two communities, C_1 and C_2, and a community assignment on the rest of the network U, there exists a set of conditions for which modularity is reduced by merging C_1 and C_2. Generally speaking, the first of these conditions is that the size of the network must be very large compared to the size of C_1 and C_2. In addition, the density of connections, represented by the clustering coefficient, in the network is important in determining the modularity limit. If the network is sparse, as social networks tend to be, optimizing for modularity may incorrectly determine that the edges between C_1 and C_2 are internal edges and thus merge the two communities together. This results in a better total modularity value, since the edge probabilities (represented by the $\frac{k_i k_j}{(2m)^2}$ term) are so low.

This problem of the resolution limit becomes especially clear when the distribution of community sizes of social networks is examined. Figure 1.1 makes this particularly obvious. In the social networks of YouTube [1] , LiveJournal [2] , Orkut [3] , and the

[1] www.youtube.com

[2] www.livejournal.com

[3] www.orkut.com

network formed by Amazon [4] customers' purchasing patterns [5], community sizes follow a power-law relationship expected from analysis of related work.

3.1.2 Modularity Biases to Smaller Communities

Because of the resolution limit problem, we may discard better community assignment. As a consequence, it will decrease the quality of the detected communities. An improvement is to introduce a new modularity metric, which we define as:

$$Q = \frac{1}{N} \sum_{i} \sum_{u,v \in C_i} \frac{1}{|M_i|^{\alpha}} [A_{uv} - \frac{k_u k_v}{2m}] \tag{3.3}$$

This formulation is more able to detect small communities because of $\frac{1}{|M_i|^{\alpha}}$. Intuitively, this term penalizes the communities that grow large. Therefore we can keep more small communities than usual. Figure 3.1 shows a result of comparing the original modularity scores with the modification.

3.2 Introduction to Girvan-Newman Algorithm

In Section 3.1, we have already proposed an improved measure of modularity to evaluate the quality of existing community assignments. In this section, we introduce several algorithms that make such a community assignment. We start from the original classical Girvan-Newman algorithm and increase the efficiency by modifying it.

3.2.1 Betweenness Measure

Communities in graphs, even small communities, are marked by the density of their connections. Areas of high density indicate the presence of communities. This

[4] www.amazon.com

[5] Links between nodes represent items purchased together.

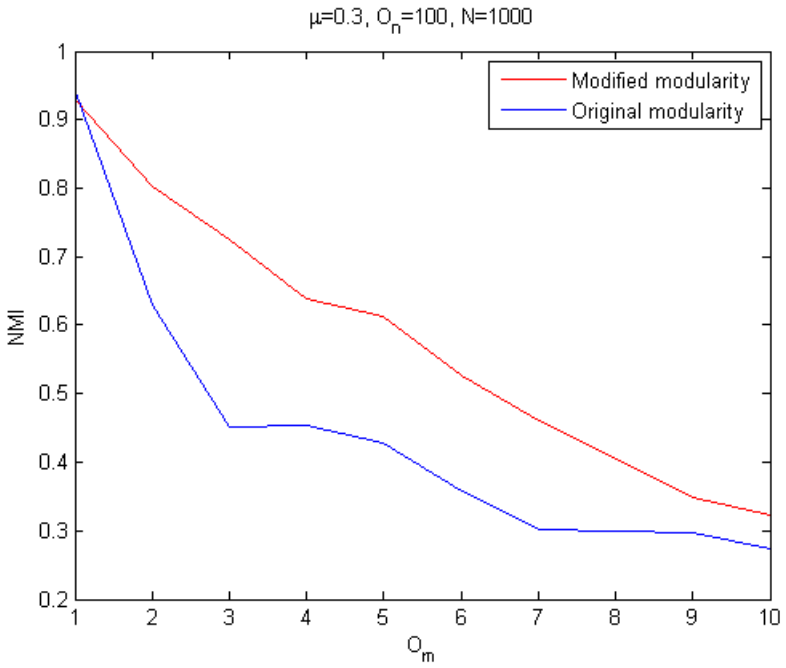


Figure 3.1: Comparison of Community Sizes between Original Modularity and Modified Modularity in Synthetic Dataset

feature has a synergy with one of the common measures of network centrality, betweenness. Betweenness measures how central a node or edge is to the network by analyzing the paths incident to that node. Intuitively, it would seem that areas of the network where communities exist would have low betweenness since the density of edges is higher. This feature implies that communities can be detected by computing betweenness and looking for areas of low betweenness.

Freeman (1977) proposed a classical type of betweenness, which is defined as the number of shortest paths from all vertices that pass through that node. Newman and Girvan (2004) proposed the *edge betweenness* by substituting 'node' with 'edge'. From this definition, it is not that efficient to calculate the exact edge betweenness. The complexity is $O(n^2m)$ and later improved to $O(nm)$ by Brandes (2001). The speed of calculating this betweenness measure for a social network with millions nodes is

thus slow and unacceptable.

Another possible way to calculate betweenness on the edge of a network is called *Current-Flow Betweenness*. In this betweenness formulation, the network is re-imagined from a social network to a network of resistors linking nodes in an electrical circuit. This allows the network to be solved for betweenness like a system of linear equations, since Kirchhoff's laws allow for solving such a network as a system of linear equations. The formula can be written as follows:

$$Lv^{(st)} = b_{st} \tag{3.4}$$

Where L is the *impedance network* formed by the edges of the network, b_{st} is the objective current flow vector, and $v^{(st)}$ is the applied voltage vector. Kirchhoff's laws presuppose the existence of two special nodes in the network, a voltage source and a voltage sink, represented by the applied voltage vector. Since Kirchhoff's laws were intended for resistor networks, they assume that the network has an attached power source. This assumption is a problem for social network analysis, as no native source or sink nodes exist. To remedy this issue, Current Flow Betweenness assumes that all combinations of two nodes are taken as source/sink pairs. Integrating all of these instances together gives the following final formula for Current Flow Betweenness:

$$c_{i,j} = \frac{2}{(n)(n-1)} \sum_{s \neq t, s, t \in V(G)} |v_i^{(st)} - v_j^{(st)}| \tag{3.5}$$

In Section 3.3, we will show that we can compute current-flow betweenness efficiently.

3.2.2 Girvan-Newman Algorithm

In Newman and Girvan (2004), an algorithm is proposed that allows community detection on any network using any measure of betweenness. The outline is as follows:

1. Compute betweenness for the entire network.

2. Find the edge with the greatest betweenness score.

3. Calculate Modularity for the entire network, assuming that the connected components represent the communities.

4. Repeat from 1 if the Modularity increases.

Along with its high accuracy for discovering the communities, the Girvan-Newman Algorithm is, unfortunately, not scalable to large social networks. Many measures of edge betweenness require a significant investment of computation time. Naive edge betweenness takes $O(n^2m)$ time to compute for a network and with the expansion in computation time required by the Girvan-Newman algorithm's iterative process this makes naive edge betweenness, and many other methods for calculating betweenness, unsuitable for community detection in large-scale networks like social networks.

3.3 Approximation Algorithm for Girvan-Newman

Though the computational complexity of the Girvan-Newman algorithm is far too large to analyze a large network, there exists the possibility of an approximation algorithm that receives the same quality of results as the original algorithm but is not subject to the same computational complexity problems.

To create an approximation algorithm, we can first break down the problem into two sections. First, we must find a method for computing betweenness that works much more quickly than the naive betweenness discussed previously. To that end, we look back to the Current-Flow betweenness discussed in Section 3.2.1. In this discussion, we mentioned that Kirkhoff's laws can be reduced to a system of linear equations.

3.3.1 Koutis Theorem

A system of linear equations is a common representation for a large variety of problems. Thus, there exists a substantial literature on solving systems of linear equations. One such piece of work is the following theorem:

Theorem 1 (Koutis *et al.* (2010)) *The linear system $Ax = b$ can be solved with computational complexity $\tilde{O}(m \log n)$, where A is a symmetric, sparse, semi-definite matrix.*

Since A is the adjacency matrix of our network, applying this to our Current-Flow Betweenness problem allows us to compute the betweenness metric much more quickly. This addresses one part of the complexity problem with the Girvan-Newman algorithm, but it leaves the problem of edge removal iterations.

3.3.2 Hoeffding's Inequality

The original Girvan-Newman algorithm calls for the computation of betweenness, which, despite the speed-up afforded by Theorem 1, would take far too long to compute the complete Current-Flow Betweenness since this method requires computing the result of Kirkhoff's Laws using every pair of nodes as a source and sink. Normally, this would make the computation of Current Flow Betweenness infeasible due to the time required for large networks.

In order to address this, we refer to the literature to reduce the computation time for betweenness. Here, we apply Hoeffding's Inequality from Hoeffding (1963):

Theorem 2 (Hoeffding (1963)) *If $x_1, x_2, \cdots, x_k$ are independent random variables, $a_i \leq x_i \leq b_i$, and $\mu = E(\sum x_i/k)$ is the expected mean, then for any $\epsilon > 0$,*

$$Pr\{|\frac{\sum x_i}{k} - \mu| \geq \epsilon\} \leq 2e^{-2k^2\epsilon^2/\sum_{i=1}^{k}(b_i - a_i)^2}.$$

This inequality allows for a significant reduction in the amount of computation required to obtain an approximation for current flow betweenness. Given an ϵ, this inequality can be used to determine how many random trials (in this case, pairs of source/sink nodes) are required to approximate the value of current flow betweenness.

3.3.3 Sampling Theorem

Using the results of the Koutis Theorem and Hoeffding's Inequality, we propose the following, which we call the Sampling Theorem:

Theorem 3 (Sampling Theorem) *We can approximate the value of current-flow betweenness in the entire network within an absolute error of* ϵ *with a high probability* $p = 1 - \frac{2}{\sqrt{n}}$ *using* $k = \lfloor \log \frac{n}{\epsilon^2} \rfloor$ *samples.*

This theorem is an intuitive consequence of the previous two. However, we give a proof as follows:

Proof: From Section 3.2.1, we know that the Current Flow Betweenness for some u is $c^{(u)} := \sum |v_i^{(u)} - v_j^{(u)}|/(n(n-1)) = E[\sum x^{(u)}/k]$ Let $x^{(u)} = |v_i^{(u)} - v_j^{(u)}|$. Then, $x^{(u)}$ is independent for different u. In addition, $0 \leq x^{(u)} \leq 2$, since we use a voltage of 1 for the source node and -1 for the sink node. Thus, Hoeffding's Inequality applies, and we can conclude that when $k \geq \log n/\epsilon^2$:

$$\Pr(|\frac{\sum x^{(u)}}{k} - c| \geq \epsilon) \leq 2e^{-2k^2\epsilon^2/(4k)} = 2e^{-k\epsilon^2/2} \leq 2/\sqrt{n} \tag{3.6}$$

Given this result, we can estimate the number of samples required to reach a given p with a given ϵ. For example, selecting $p = 0.9$ and $\epsilon = 0.1$ results in the requirement to have a sample size (k) of only 600 source/sink pairs.

3.3.4 *Efficient Current-Flow Betweenness Algorithm*

Using these results, we can reduce the complexity of the basic Girvan-Newman Algorithm with some modifications. The algorithm is given in Algorithm 1.

Input: Adjacency Matrix A

Output: Community Assignment Matrix C

Define modified modularity $Q_m = 0$ and $Q'_m = 0$

while $Q'_m \geq Q_m$ **do**

  $Q_m \leftarrow Q'_m$

  Define d as the degree vector for all nodes

  Set the Laplacian L to be $A - \text{diag}(d)$.

  Randomly select a set of pairs of nodes T of size k

  foreach $s \in T$ **do**

    Solve for v in $Lv = d_s$ per the Koutis Theorem

    Update the Current Flow Betweenness for all edges.

  end

  Delete the edge with maximum betweenness

  if *a new component is formed by edge deletion* **then**

    Recompute Q'_m.

  end

end

Algorithm 1: Algorithm for Efficient Current-Flow Betweenness

Using this algorithm as a replacement for the standard Girvan-Newman algorithm provides a substantial speed-up. This reduces the computational complexity of a Girvan-Newman-style algorithm from $O(m^3n^2)$ to $O(m^2 \log n)$, a substantial improvement. However, even this improvement is not enough to allow community detection in a reasonable time frame.

3.3.5 *Approximate Current-Flow Betweenness*

Though the improvement made by utilizing the Sampling Theorem is substantial, it does not go far enough to ensure that communities can be detected from networks of any size. Thus, we must further reduce the runtime of the algorithm, and we do so by modifying the edge deletion method. Instead of deleting only one edge per repetition, we use Algorithm 2 to delete multiple edges.

Input: Current-Flow Betweenness for a graph $G = \langle V, E \rangle$

Output: An edge set E_0 to be deleted.

Sort Current Flow Betweenness descending by magnitude, call this array C.

Compute 1.5IQR; $i = 1$

Unlabel all vertices

while $C(i) > t$ **do**

  if *Both endpoints of* E_{C_i} *are unlabeled* **then**

    Label the endpoints.

    $E_0 \leftarrow E_0 \cup \{e\}$

    $i \leftarrow i + 1$

  end

end

Algorithm 2: Algorithm for Removing Multiples Edges per Iteration

In the above algorithm, interquartile range (IQR) is defined as the 3rd quartile subtracts the 1st quartile. Though this method does not reduce the theoretical computational complexity of the Efficient Current Flow Betweenness (ECFB) method described in Algorithm 1, it does reduce the empirical runtime, as demonstrated in Figure 3.2.

From the figure, we can see that, empirically, the runtime scales linearly with the size of the network with the multiple edge deletion modification. Algorithm 2 is

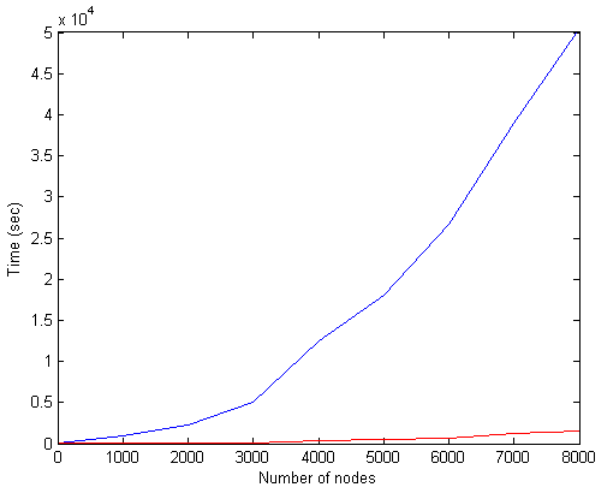


Figure 3.2: Network Size vs. Runtime of Efficient Current Flow Betweenness Algorithm with (Red) and without (Blue) Multiple Edge Deletion.

somewhat more complex than necessary to simply remove multiple edges.

3.4 Results

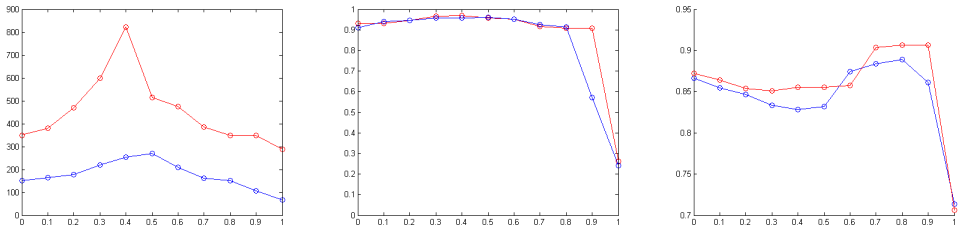
Using our novel algorithm with its optimizations, we seek to demonstrate that our method can outperform other methods when detecting communities. In addition, we wish to show that our modification to modularity allows the metric to detect small communities better than the original metric.

3.4.1 Evaluation Metrics

As a convention, we adopt NMI (Section 2.3.1) as the metric for comparison. We compute NMI between detected communities and real-world communities.

3.4.2 Tuning

First, our modification to modularity introduces a new parameter, α. In order to find the value for α that maximizes the detection of small communities without



(a) (b) (c)

Figure 3.3: α Balues vs. (a) Number of Detected Communities, (b) Original Modularity Score for Partitions, and (c) NMI Score in Networks of 1000 (Red) and 5000 (Blue) Nodes.

over-penalizing the large communities that are still significant, we perform a series of experiments varying the α parameter. The results of these experiments can be found in Figure 3.3. These experiments show that there is a "sweet spot" for the value of α in the upper range between 0 and 1.

To confirm this sweet spot, we perform a similar experiment but use real-world networks. Figure 3.4 shows the results of testing modularity on these real networks. Since ground truth is available for these networks as it was for the synthetic networks of Figure 3.3, we can perform a similar analysis.

This second test confirms the "sweet spot" for Modified Modularity between ≈ 0.6 and ≈ 0.9, since the ACF network evidences a sharp rise after ≈ 0.6 and the NMI of the synthetic communities drops off sharply after 0.9. In order to capitalize on this "sweet spot" as much as possible, we use $\alpha = 0.75$.

3.4.3 Sample Sizing

In Section 3.3.3, we claimed that 600 samples were sufficient to yield a reasonable approximation of Current Flow Betweenness for a network. Though this result

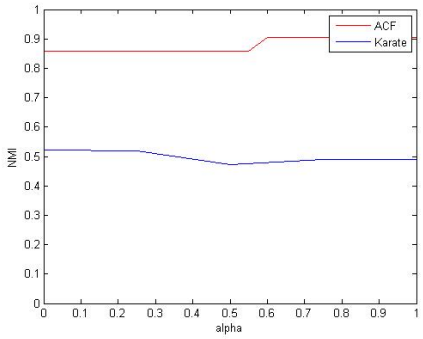
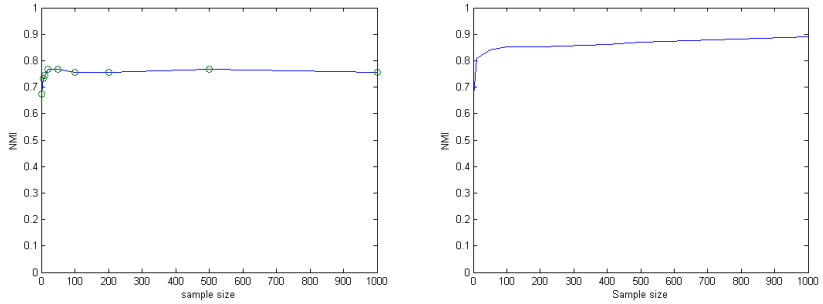


Figure 3.4: α values vs. NMI in Two Real-world Datasets.

is theoretically sound, we would like to verify this result. Figure 3.5 shows a comparison between the NMI of a detected community assignment and the number of positive/negative pole pairs used to estimate Current Flow Betweenness.



(a) American College Football (b) Synthetic network $n = 1000$

Figure 3.5: Node Pair Sample Size vs. NMI in Two Network Datasets.

The results of this analysis demonstrate that our estimate is, in fact, conservative for networks of these sizes. Clearly, there is significant benefit in increasing the sample size above ≈ 10, but for the ACF network the results become stable after approximately 100 samples. The synthetic network shows similar results. The results on the synthetic network continue to improve with the sample size, but these improvements quickly become marginal. In fact, increasing from 150 to 1000 samples provides less

total benefit than increasing from 50 to 150.

3.4.4 Baseline Comparisons

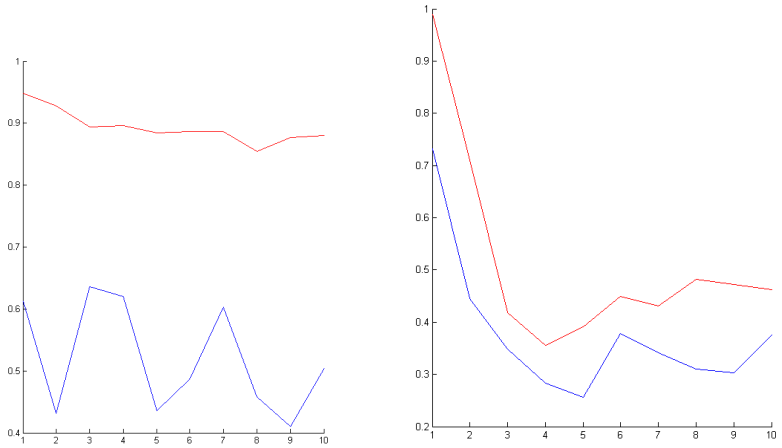
In order to verify that our novel algorithm, ECBA, provides value to community detection as a field, we would like to verify that our method gives community detection performance comparable to or exceeding the performance of algorithms with similar scalability features. The two exemplar methods we chose to compare against were the ones that provided the best performance in the existing literature. These two algorithms are InfoMap (Rosvall and Bergstrom (2008)) and SLPA (Xie *et al.* (2011)). More information on these two methods can be found in Chapter 2.

In addition, we would also like to compare the difference in performance between ECBA with and without the multiple edge deletion variant to show that the multiple edge deletion does not affect the performance of the algorithm. Figure 3.6 shows that the performance for the multiple edge deletion variant (red) actually often outperforms the single edge deletion variant (blue).

This result is somewhat surprising. From simply the description of the algorithms, it would be natural to expect that the multiple edge deletion variant of ECBA would be less accurate. However, if we consider that both variants have approximation as an incontrovertible part of the algorithm, it may be reasonable to conclude that the multiple edge deletion variant provides better results because the single edge deletion variant changes its approximation so quickly that the quality of the detected communities is compromised.

With the confirmation that our ECBA with multiple edge deletion is a good candidate for comparison against similarly complex algorithms, we compare our algorithm against the two previously mentioned baselines in Figure 3.7.

Figure 3.7 shows that the performance of our algorithm is comparable against the



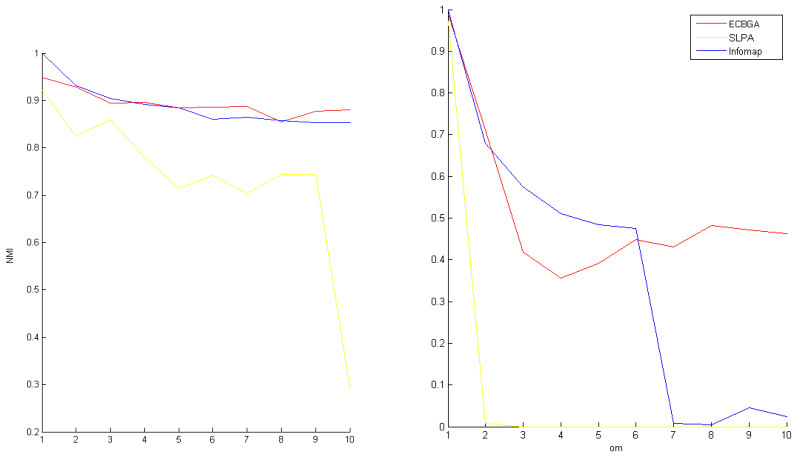
(a) (b)

Figure 3.6: Number of Overlapping Nodes per Community vs. NMI of Community Assignment for ECBA with (Red) and without (Blue) Multiple Edge Deletion.

baseline algorithms, but both our algorithm and the baselines have drastic falloffs in performance if the overlap of community memberships gets too high.

In Section 3.1 and Figure 1.1, we demonstrated that community sizes tend to follow a power-law distribution and that modularity cannot detect these small communities that make up the majority of the communities. We tested the size distribution of the communities we detected with our modified modularity in order to verify that the distribution matches the one we expect to see from real-world networks. Figure 3.8 shows this result, and shows that our method detects communities in the size distribution we expect, while Infomap does not. SLPA was omitted from this comparison since it did not detect communities with comparable accuracy to ECBA and Infomap.

Next, we compare a similar performance metric on a real-world data set that is discussed in Section 4.4.1, the Amazon dataset. The Amazon dataset contains an extremely large number of communities, approximating a real-world data set, but



(a) (b)

Figure 3.7: LFR Network Graphs with 1000 nodes, of which 100 (Left) and 500 (Right) Nodes in Multiple Community Memberships.

retains ground-truth communities. Figure 3.9 shows that ECBA's community size distribution closely matches the size distribution of the ground truth.

Unlike InfoMap and SLPA, the ECBA's detected community distribution closely matches the community distribution of the ground truth. By comparing the vectors of community sizes together, we can obtain an objective measurement of the similarity. Using cosine distance, we can objectively measure how close the distributions actually are. As we suspect, ECBA is the closest, with a cosine similarity of 0.45. In close second is SCAN, scoring 0.42. Infomap and FastComm trail with scores of 0.21 and 0.29, respectively. Knowing that the community distributions match, we then compare the NMI for community detection on our real-world data sets in Table 3.1.

Of particular note in Table 3.1 is our poor performance on the Karate Club data set. The ECBA algorithm is optimized toward finding communities where the communities are very small compared to the size of the network. In the Karate Club dataset,

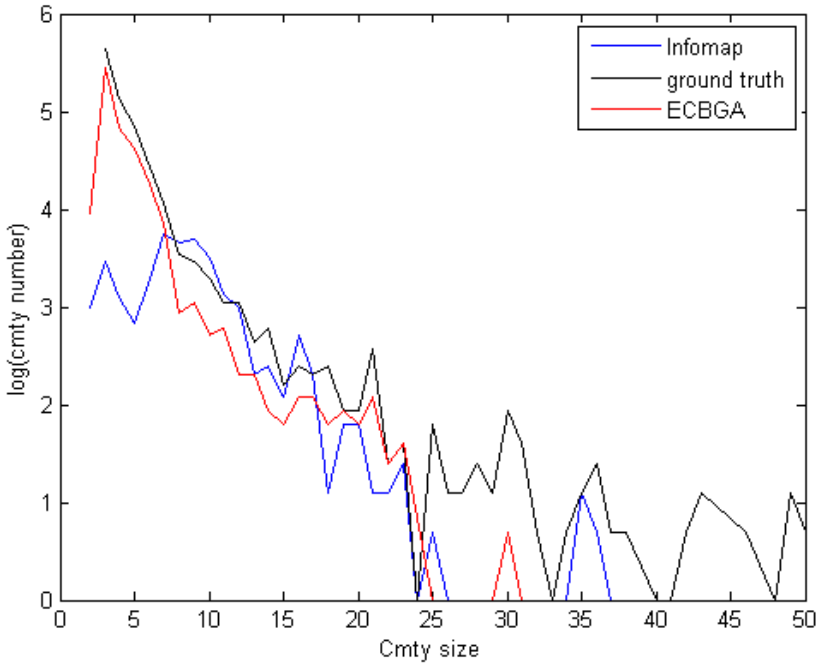


Figure 3.8: Community Size Distribution in a Synthetic Network.

NMI Score	ECBA	Infomap	SLPA	SNAP
Karate Club	0.1595	0.4465	0.4465	0.2110
ACF	0.7513	0.8087	0.5182	0.7220
Amazon	0.1487	0.0207	–	0.1863

Table 3.1: NMI for Real-world Datasets.

the two communities split the network relatively evenly, which makes detecting these communities a challenge for ECBA.

3.5 Summary

In this chapter, I have described and documented a novel algorithm, ECBA, for detecting small communities in real-world networks that is scalable to arbitrary network

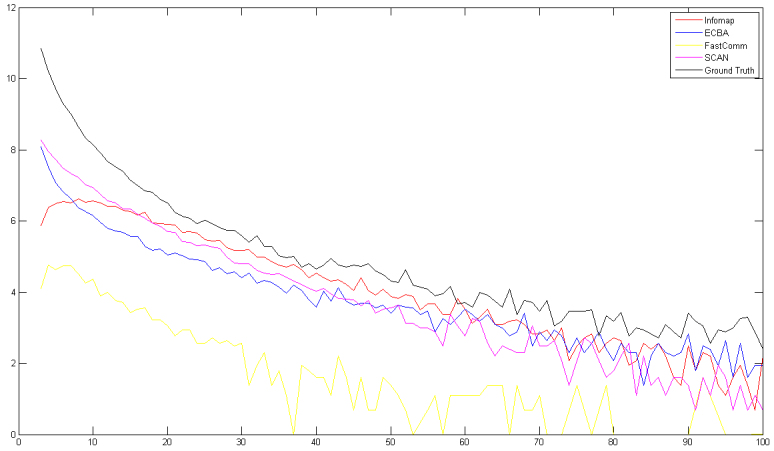


Figure 3.9: Community Size Distribution in the Amazon Co-Purchasing Network

sizes, shows comparable performance with competing methods, and outperforms competing methods when small communities are the search object. This method, based on the method proposed by Newman and Girvan (2004) remains highly scalable to large network sizes thanks to work done by Koutis *et al.* (2010), Hoeffding (1963), and our combination of the two in Section 3.3.3.

In addition to this, we are the first, to the best of our knowledge, to document the power-law distribution of communities in social networks' impact on methods based on maximizing the Modularity (Newman and Girvan (2004)) metric. In order to circumvent this metric's resolution limit, we propose a Modified Modularity metric in Section 3.1 that uses a penalty term scaling with the size of the partition to avoid the pitfalls of combining small partitions described by Fortunato and Barthelemy (2007). As we show in Section 3.1, the modified modularity metric can be used to evaluate the community assignment that includes more small communities without the aforementioned pitfall.

One possible concern with re-biasing community detection algorithms towards

small communities is the risk that this will result in an overwhelming number of small communities. Part of the idea of algorithmic community detection is that the amount of data that human analysts are required to actually manually analyze is significantly reduced. However, detecting an extremely large number of small communities threatens to undermine that effort by presenting a number of communities that is unreasonable for manual inspection. To remedy this issue, some future work in the area of small communities is to leverage the significant work done with community outlier detection and minimum description length to narrow down the number of the small communities necessary for manual analysis. Intuitively, these methods can find the 'interesting' communities from a large set of uninteresting communities by finding those communities who structure is substantially different from the average.

Chapter 4

REWEIGHTING PROCESS AND COMMUNITY DETECTION

4.1 Introduction to Edge Weighting in Community Detection

4.1.1 Intuition

In the social network of Facebook, there are huge university communities with over ten thousand of people that each consists of a number of students and professors. It also contains the small communities of swimming clubs and research interest groups with only around ten people.

Besides, different communities that share the similar interest also have more connection with each other. The members of two karate clubs that come from different universities may become friends. Because of the tight connection between the two communities, the number of inter community edges increases, and thus the algorithms may detect one community instead of two.

From another angle, individual always belongs to more than one community, since he/she has different family, work and entertainment circles. This results in communities of high overlap, which further make the community detection task much more difficult.

Under the above circumstances, useful communities can be completely "hidden" in the bigger communities, which makes the community detection task extremely difficult. Community detection algorithms tend to detect the super communities instead of the small ones, since more links between communities increase the opportunity for detecting large community. Several toy examples are introduced in Figure 4.1 In these figures, popular algorithms (Infomap, SLPA) misclassify the communities by

recognizing their supersets.

One can observe that a large class of existing algorithms exploit and depend on edge information heavily: Infomap and Personalized PageRank algorithm rely on the fact that the random walker stay inside the communities for a short period; SLPA propagates label from one node to its adjacent neighbors by the same probability.

Therefore, the reweighting technique is purposed. Whenever there is a denser structure, the links inside the structure should receive higher weights. In this way, random walks and labels will not leave the denser communities easily. These weighted links play an important role in the performance of the algorithm.

4.1.2 Introduction to Intimacy

To describe the idea clearly, let us consider an example of random walk in a network. Figure 4.2 shows the choice of the choice of a random walker at one specific node. It has 3 neighbors. The red edge is inter-community edge, while the blue edges are intra-community edges. Originally, the network is unweighted. The walker originally travels to its adjacent nodes with the same probability. Thus, in Figure 4.2, the chance of the walk going outside the community is 1/3. However, after applying intimacy network, this chance is reduced (1/4 for this specific example) since we have a much higher chance that the intra-community intimacies are higher than the inter-community ones. Thus the random walker will stay in the community much longer, and the community is easier to recognize in this network. The label propagation based method is similar.

As we claim above, a weighted network is beneficial to community detection algorithms. Unfortunately, little work has been done in this field. We need to address two problems in the following section: (1) How do we find a set of suspected inter-community edges? (2) How do we assign the weight to make it more efficient? In

Section 4.1.3, we introduce a new measure – intimacy to address these problems.

From above, our goal in this chapter to assign weights to the edges $ij \in E(G)$, such that the inter-community edges receive relatively higher weights, while the intra-community edges receive lower weights. Furthermore, we expect that these weights can be applied to improve the precision of current community detection algorithms without affecting the original algorithm's complexity.

4.1.3 Intimacy Formulation

In a network, *intimacy* is a weighting function $w : E \to \mathbb{R}$ that indicates if two individuals are likely to be in a same community or not.

Intimacy measure is a general measure that increases the weight of intra-community edges while decreases the weight of inter-community edges. Different definitions can make intimacy work. In the next section, two different candidate measures for intimacy are defined for improving community detection algorithms such that more small communities can be detected.

4.2 Implementing Intimacy

In this section, we propose two ways of defining intimacy. One is related to the betweenness. Another is about the number of triangles that the edge is in.

4.2.1 The Idea of Betweenness Intimacy

Definition 2 (Betweenness Intimacy) *In a network, the intimacy measure for each edge $ij \in E(G)$ is defined as a real number I_{ij} that is inversely proportional to its betweenness measure c_{ij}:*

$$I_{ij} := \max_{i,j} c_{ij} + \min_{i,j} c_{ij} - c_{ij}$$

To calculate intimacy, we need to first calculate the betweenness measure over the network. Since betweenness is usually computationally expensive, we apply the approximation from Section 3.3.1 and 3.3.3 to estimate it. From Theorem 1 and Lemma 3, we can see the approximation is accurate and efficient. The same method will be incorporated in the algorithm in below.

From the description above, Algorithm 3 formalizes the procedure for calculating intimacy. We add ϵ for each edge $e \in G$ to keep the final intimacy network still connected.

Input: Original network adjacency matrix A

Output: Intimacy

Choose a set T of k samples of positive pole and negative poles

for *each* $s \in T$ **do**

Solve v in the linear equation $(\mathbf{D} - \mathbf{A})\mathbf{v}^{(st)} = \mathbf{b}^{(st)}$.

Calculate the current-flow betweenness c_{ij} for all edges ij

end

Find $c_{\min} = \min c_{ij},\ c_{\max} = \max c_{ij}$.

Calculate $S = c_{\max} + c_{\min}$.

for *each edge* $ij \in E$ **do**

Compute $I_{ij} = S - c_{ij}$

end

for *each pair* (i, j) *such that* $A_{ij} = 1$ **do**

$I_{ij} = I_{ij} + \epsilon$

end

Algorithm 3: Intimacy Calculation

Since intimacy is calculated from current-flow betweenness, it derives the following properties from current-flow betweenness:

1. Similar to random walks, betweenness is a global measure. Betweenness is calculated for each pair of source and sink, while a random walk spreads to adjacent nodes by same probability. Therefore betweenness is complementary to random walk. It is therefore extremely suitable for the reweighting technique.

2. High betweenness edges result in low intimacy edges, while low betweenness edges result in high intimacy edges with high probability. This fact is community-based. The betweenness measure provides a group-based perspective to the network. Therefore, higher intimacy means the two nodes at the endpoints of the edge are more likely to be in the same community and vice versa. (This is not an explicit result in Newman and Girvan (2004), we verify it in Section 4.2.2.)

Figure 4.3 is an example illustrating how intimacy works. We can see closeness in the figure through color-coding. Bluer edges represent low intimacy between the two connected nodes. Redder edges, therefore, represent high intimacy between the two nodes. As we can see, from the right side, there is an obvious community consisting of Node 5, 6, 7, 11, 17, where the edges are very red. In the lower left corner, there are lots of triangles, which indicated intimacy also performs nicely. Nodes 1 and 33 are the two instructors from literature Fortunato (2010), and we can see any path between them are blue/green, which indicates that they are not close to each other, a reasonable conclusion for this dataset.

4.2.2 Statistical Soundness for Betweenness Intimacy

In this section, we show empirically that intimacy is a good measure for distinguishing inter-community edges from intra-community edges. More specifically, we want to show that inter-community edges have larger intimacy values than intra-

community edges with high probability.

To verify the claim above, we perform two experiments. (1) We perform the Mann-Whitney-Wilcoxon (MWW) test (Mann *et al.* (1947)) to clarify the significant distribution difference for inter-community edges' and intra-community edges' intimacies. (2) With the synthetic datasets with ground truth under different mixing parameter, we list all pairs of intra-community edge and inter-community edges. Then we verify that there is a great probability that inter-community intimacy is greater than intra-community intimacy.

We use the LFR benchmark synthetic networks described in Section 4.4.1 with ground truth non-overlapping communities (Set #5 and #6) as the datasets for this test. The links are divided into two groups: inter-community edges and intra-community edges.

Now we perform experiment (1). The Mann-Whitney-Wilcoxon test is a non-parametric test of the null hypothesis that two populations are the same against an alternative hypothesis, especially that a particular population tends to have larger values than the other. Since we cannot have any parametric assumption for the distribution of intimacy, this is the best statistical test we can find. In this test, we have the null hypothesis:

H_0*: The distributions of inter-community intimacies and intra-community intimacies are equal.*

H_1*: The distributions of inter-community intimacies and intra-community intimacies are not equal.*

The result is that the maximum p-value for all realizations is 3.64×10^{-8}, and thus the MWW tests all reject the null hypothesis.

Since we have such a low significance level, it does not reveal a lot of information for us. Here we do experiment (2) to further explore the intimacy's properties: for all

realizations, we estimate the probability $P := \Pr(I_{intra} > I_{inter})$. Then we can plot Figure 4.4. We can see that it shows the similar patterns for $N = 1000$ and $N = 5000$. As μ increases P decreases, which also validates the conclusion from experiment (1). However, even when $\mu = 0.8$ we still have $P \approx 0.58 > 0.5$ which means we still have greater intimacy of intra-community edges than that of inter-community edges.

From experiments above, we conclude that the new measure intimacy does distinguish the inter-community edges from intra-community edges. This also answers our question: the edges of low betweenness are more likely to be intra community edges.

4.2.3 The idea of Traid Intimacy

Triads in social networks are triangle in graphs. From the social science angle, a triad is a person's friend's is also his friend. Triad plays an important role in social theory. Research has showed that the transitive rule – if person A is the friend of person B, who is the friend of person C, then person A is also the friend of person C holds with a high probability.

Compared with links, which reveal the friendship, triad makes higher possibility for us to believe these three people are from a same community. Then we should assign these three links in the community with higher weight.

Using this idea, we introduce the following definition:

Definition 3 (Triad intimacy) *Assign each edge $e \in E$ with a weight w_e that is defined as the number of triangles that is in. Then triad intimacy is defined as $I_e = w_e(w_e + 1)/2 + 1$.*

Here I apply a quadratic function to w_e since more triangles it is in, the much higher chance this link is in at least one community. Also $I_e \geq 1$ is guaranteed to maintain the connectivity of the network.

It is very possible that the measure defined above can be generalized to a measure that involves short cycles not only triangles. However, experiments have been done and no better result has been shown until now.

4.2.4 Algorithm

It is not trivial to develop an algorithm that can calculate the above measure sufficiently. As social network is sparse, a trivial algorithm is as follows:

Input: Original network G

Output: Triad Intimacy network I

for *each vertex* v *in* G **do**
| Record the neighbor list $N(v)$

end

for *each edge* $e = xy$ *in* G **do**
| $w(e) = |N(x) \cap N(y)|$
| $I(e) = w(e)(w(e) + 1)/2$

end

Algorithm 4: Triad intimacy: Naive method

The algorithm takes $O(m\Delta)$ where Δ is the maximum degree within the graph. As the social network is power-law distributed, the algorithm does not provide a satisfying complexity with a high maximum degree. There is another algorithm modified from Tsourakakis (2008). By listing the triangles, the algorithm gives the best possible complexity. Besides, the paper proves that Algorithm 4 can perform at the worst complexity with a power-law graph.

The basic idea of the above algorithm is to limit the number of neighbors each vertex will visit. The maximum neighbor one vertex can visit is $O(\sqrt{m})$ and thus the algorithm only takes $O(m^{3/2})$. It has a higher computational complexity. However,

Input: Original network G

Output: Triad Intimacy network I

Sort the vertices by the degrees from largest to smallest, denote the order function f

Let A be an array of n arrays initially empty.

for *each vertex* v *taken in increasing* $f(v)$ **do**

  for *each* $u \in N(v)$ *with* $f(u) > f(v)$ **do**

    for *each* $w \in A[u] \cap A[v]$ **do**

      $w(uv) = w(uv) + 1$

      $w(uw) = w(uw) + 1$

      $w(vw) = w(vw) + 1$

    end

    $A[u] := A[u] + v$

  end

end

for *each edge* e *in* G **do**

  $I(e) = w(e)(w(e) + 1)/2$

end

Algorithm 5: Triad intimacy: efficient method

with 100 thousand of nodes, the algorithm needs similar time empirically because it has a lower constant for calculation. For incorporating the measure into the original community detection algorithm, I refer to Section 4.3.

4.3 Incorporate intimacy into existing algorithms

Now we integrate the intimacy measure into a few popular community detection algorithms. To get the best performance, we require the algorithm to be scalable

and based on exploiting local information. We introduce some algorithms to discuss which part of the algorithm can most benefit from intimacy information.

4.3.1 Existing algorithms

1. Infomap (Rosvall and Bergstrom (2008)). As a recent popular method, Infomap has proven its success on community detection problems. As a tool to illustrate the random walk property, by optimizing the map equation, community discovery is converted to a minimum description length problem, which can be solved theoretically. The algorithm generates a number of modules from its random walks and stopping rules. The algorithm then optimizes communities by combining and separating these modules to minimize the map equation. This method outperforms most methods for non-overlapping communities. Although its performance is reduced on a complicated network with large numbers of overlapping communities, it still outperforms most overlapping community detection algorithms according to NMI (Xie *et al.* (2013)).

2. Speaker-listener Label Propagation Algorithm (SLPA) (Xie *et al.* (2011)). As a Label Propagation method, SLPA artificially provides an overlapping communities result. At the very beginning, each node receives a distinct label. At each step, a node receives all its neighbor's labels asymmetrically. After one step, the nodes are marked with several labels, each with different weights. The algorithm then keeps any labels above some threshold. Thus, it can detect overlapping communities by retaining multiple labels. It is, however, tricky to reach a stable state for all labels and requires tuning many different parameters. By propagating labels to reach the global optimum, the idea does not use any global information immediately.

3. Louvain's algorithm (Blondel *et al.* (2008)). As a very popular modularity optimization method, the Louvain's algorithm is an agglomerative heuristic algorithm. Starting from a singleton, it basically incorporates and divides modules to find the best partition by computing the modularity. Similar to Infomap, this algorithm uses local information, in neighboring modules, and some global information, the modularity.

By utilizing local information, the methods listed above show success in detecting both non-overlapping and overlapping communities. To incorporate our method into these algorithms, we simply write down Algorithm 6. In Section 4.4, we show our results for this modification.

Input: Original network N

Output: Detected communities

Apply Algorithm 1 to calculate intimacy for each edge in N

Write out intimacy as a matrix M.

Replace the original adjacency matrix A by M.

Perform an original community detection algorithm

Output the detected communities.

Algorithm 6: Hybrid community detection scheme

4.3.2 Exploring the Idea behind the Reweighting Technique

From Section 4.2.2, intimacy shows very good performance and distinguishes inter-community edges from intra-community edges. Therefore, we can consider replacing the old unweighted network with a new weighted one using intimacy to weight the edges. Then, we can implement community detection algorithms on the new intimacy networks. We expect there is a better performance.

Due to the vague definition of community – a subset of nodes of graph that there are more links between instead of going outside, it is hard to write down proofs by formal mathematical language. It is difficult to even define ground-truth communities in a graph. Therefore, it is almost impossible to provide a rigorous proof why the reweighting method performs better. Here we just provide a case study to show that the reweighting method can detect smaller community compared to the original method.

Figure 4.5 is an induced subgraph of a large unweighted network. It consists of 12 nodes with links shown in the figure. The optimize partition is marked with two different colors. When implement SLPA algorithm for the network, a possible order is propagate nodes is $(1, 2, \cdots, 12)$. We may be in trouble with node 8, since it has 2 neighbors in the blue community and 2 neighbors in the red community. Thus, we may label it in the wrong community for 50% chance. Then we will misclassify node 9 to the same community since it is a neighbor of node 8 and so on. Finally, we may recognize the whole 12 nodes as one community.

However, when we apply the reweighting technique first by using intimacy, we can find the close nodes first. It can be viewed in the picture that nodes 1-7, 8-9, 10-12 are actually much closer to each other. When we apply SLPA, it will be more stable to classify node 8 to be a different member from nodes 1-7.

4.4 Experimental setup and Results

4.4.1 Dataset

In this section, we introduce the dataset we use to perform our experiments.

- LFR Benchmark network (Lancichinetti *et al.* (2008)). This benchmark generates random scale-free networks based on the planted l-partition model. It can

control N – the size of the graph, d – the average degree of the nodes. u_1, u_2 – the degree and community size power-law distribution constants, overlapping nodes O_n, overlapping nodes in O_m communities, and μ – topological mixing parameter. The default setting for this paper is $u_1 = u_2 = 2, d = 25$. Since it can generate different networks by changing different parameters, it has been a popular benchmark in recent years.

- Zachary's Karate Club (Zachary (1977)). Zachary recorded the friendship between 34 members in a club at a university during three years. The ground truth consists of two communities: One is around Node 34 (president), the other is around Node 1 (instructor).

- American College Football (ACF) (Girvan and Newman (2002)). This network contains the network of American football games between different divisions during the regular season in Fall 2000. The ground truth is teams' division.

- SNAP datasets. These datasets can be retrieved from snap.stanford.com. It involves several datasets: Amazon, YouTube, and DBLP. The benefit of these datasets is that ground truth is provided. Therefore, we can easily compare our detected communities with the ground truth communities. For the ground-truth communities of these networks, Amazon Co-purchasing network is based on the 'Customers Who Bought This Item Also Bought' feature of the Amazon website. If a product i is frequently purchased with product j, the graph contains an undirected edge between i and j. YouTube is based on user-defined groups. The DBLP dataset, is based on authors who published in the same journal or conference.

A summary of all datasets is listed in Table 4.1:

Network	Nodes	Edges	Communities
Karate club	34	78	2
ACF	115	613	12
Amazon	334,863	925,872	271,270
Youtube	1,134,890	2,987,624	8,385
DBLP	317,080	1,049,866	13,477

Table 4.1: Real-world Dataset Statistics

4.4.2 Setup and Results

In this section, we firstly give the results of incorporating the intimacy network with popular community detection methods including Infomap, SLPA, and Louvain on the synthetic networks. Then we apply our method to the various real-world datasets. Comparative analysis is constructed by running the algorithms with 3 network setup: (1) the original network (blue) (2) the betweenness intimacy network (red) and (3) the triad intimacy network, (4) random weighted network (black). We include (3) for strong evidence that our reweighting method is also better than randomized reweighting. We use Normalized Mutual Information (NMI) (Section 2.3.1) as our measure. However, standard NMI cannot be used for overlapping communities. Therefore, we use generalized NMI Lancichinetti *et al.* (2008) in this paper. The implementation can be found [1] here.

To exclude possible random factors, each point is shown as an average NMI of 30 network realizations. We run three algorithms described in 4.3 in 6 generated sets of LFR benchmarks to evaluate our results. The detailed parameters for the bench-

[1] https://sites.google.com/site/andrealancichinetti/mutual

marks are as follows (this is a popular set-up used in Xie *et al.* (2013) and Khadivi *et al.* (2011) for testing synthetic datasets except community size constant. Here, we expect more small communities to increase authenticity): $u_1 = u_2 = 2, d = 25$ for all benchmarks. Set #1: nodes N=1000, overlapping nodes O_n take up to 10% of the overall network, $\mu = 0.3$, each overlapping node is in 1 to 10 communities. Set #2: nodes N=1000, overlapping nodes O_n take up to 50% of the overall network, $\mu = 0.1$, each overlapping node is in 1 to 10 communities. Set #3 and Set #4 are similar to Set #1 and Set #2, except that $N = 5000$. Set #5: N =1000, no overlapping nodes, μ is from 0.1 to 0.8. Set #6 is similar to Set #5 except that $N = 5000$.

Randomly weighted networks are constructed as follows: for each original network, instead of using default weight, we randomly select the weight i.i.d from the uniform distribution of $[1, 5]$.

For the running parameters, we run Infomap for 10 times with best two-level structure. We run SLPA by setting $r = 0.45$.

From Figures 4.6, 4.8, and 4.10 we can see the synthetic results. Red curves show the results using intimacy networks, black curves show results from original networks, and blue curves show results from random networks. Though we adjust the parameters used to generate the random networks, we can observe that all results are improved by using intimacy networks. Infomap shows the smallest improvement while SLPA shows largest improvement. This can be explained by Infomap's incorporation and subdivision steps which utilize the global information between different modules to some extent. However, SLPA only uses local information, so it is intuitive that it improves the most. One strange pattern is that randomize weighted network also shows improvement in SLPA and Louvain's algorithm which surprises us. We have a possible explanation raised up at the end of this section.

From another angle, we can see the community sizes dramatically decrease when

we apply the reweighting technique for each of three cases. It can be observed that while the original methods discover the larger communities, the smaller communities are recognized by the reweighting method.

For more detailed analysis of synthetic dataset results, we observe that when we increase the number of nodes in the network, we have a small performance improvement. This is because we don't increase the average degree of the networks, which make the graphs more sparser and thus easier for the original algorithms to detect communities. In addition, we can see that there are huge NMI drops when the number of overlapping nodes becomes very large, where our methods have larger improvements. The third row of each figure, which demonstrates the condition where no overlapping communities exist, our method's performance improves greatly under high mixing parameters.

The original SLPA does not provide good and stable results from Figure 4.8. The reason is that the algorithm decides the label randomly when there is a tie Xie *et al.* (2011). However, after reweighting, the probability of a tie dramatically decreases. Thus, its performance increase to be comparable to Louvain and Infomap. We can suggest our method as another solution the author searches for in Xie *et al.* (2013).

For the different form of intimacy. We can see when the network has lower overlapping parts, betweenness performs better than triad. Otherwise, triad intimacy is more stable.

Now we evaluate our method's performance on real-world datasets. Again, each method is run on the original network (ori), the randomly weighted network (rwn), and the intimacy network (in). We use Infomap (Im), SLPA, and Louvain (Lv).

In Table 4.2, we can see our reweighting method improves most when combining with SLPA. For each different dataset, SLPA shows approximately 10% improvement in the NMI score except on Youtube dataset. Among the large datasets, the Amazon

Co-purchasing Network's results show the most improvement, from 0.242 to 0.273. Louvain's algorithm also shows improvement for large dataset. Although it does not have improvement on DBLP dataset, it shows the highest precision on Amazon dataset.

However, for these large datasets, Infomap shows improvements only after fourth decimal place. In addition, even if we replace the edge weight to a random weight, Infomap still performs approximately the same. This result, coupled with the results from synthetic datasets, show that Infomap is a stable method that can provide very stable results even with a differently weighted graph.

The results from the Karate Club dataset are also very interesting. Surprisingly, the randomized network shows the best performance. One possible explanation is as follows. In a network, there are two kinds of edges – inter-community edges and intra-community edges. While we randomize the weights of a network, we will make some inter-community edges high weights and some with low weights. For those edges of low weights, it will appear more like an inter-community edge. However, inter-community edges appears much more likely an intra-community edges while they are assigned high weight. Therefore, these algorithms will detect fewer communities by recognizing bigger modules with high weight inter-community edges as intra-community edges. Therefore, the algorithms detect fewer communities than intended. But, the Karate Club dataset has only two communities in ground truth, which benefits the randomized network the most. Better results for randomized synthetic dataset can also be explained by a similar argument.

To validate our guess, we calculate the following statistics to show the average community size difference for big network datasets in Table 4.3. It can be observed that, except for Infomap, while there are not dramatically changing size from original method to intimacy method, the random weight method does demonstrate a change.

Community sizes are dramatically larger than before. It benefits more from the aspect of NMI that allows finding supersets of overlapping ground truth communities to count positively in the score. Thus, the result of synthetic datasets increases continuously. This random reweighting, however, becomes actively harmful when we want to find small communities in a network.

4.5 Summary

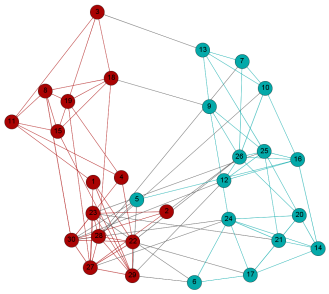
This chapter develops a new measure of network reweighting to improve the performance of existing community detection methods without compromising the computational complexity of original methods.

When combining the reweighting measure with exclusively local information-based algorithms, the scheme will perform very well. An example illustrating this point is SLPA.

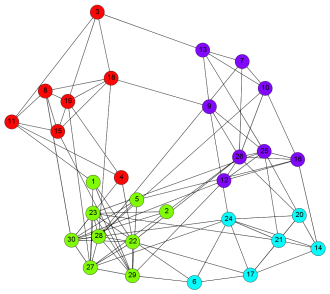
As for the drawbacks, the algorithm can only apply to undirected networks. Currently, generalizing to directed networks requires an increase in complexity to subquadratic. Besides, as a preconditioning method from numerical analysis, the algorithm is also memory-intensive. At least, the method improves lesser when the size of the network becomes larger.

As we mention that intimacy can be a measure that discovers whether two nodes are 'close' or not, it provides a brand new prospective for discovering link information while we are lack further details. Intimacy can be used to better describe a social network from a community detection standpoint.

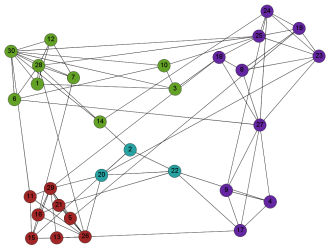
The last valuable point is that we are not able to establish a theoretical basis for explaining the reason that intimacy network performs better. It is emergent for us to systematically state the definition and theory on community structures for exploring further on intimacy measure.



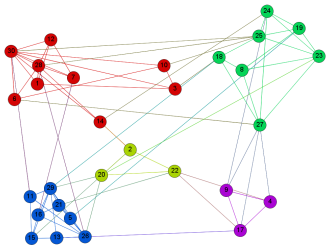
(a) Toy example #1: SLPA



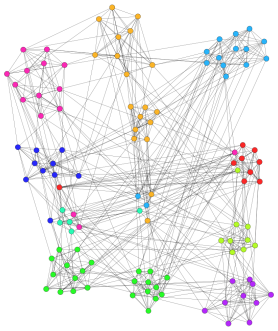
(b) Toy example #1: SLPA + reweight



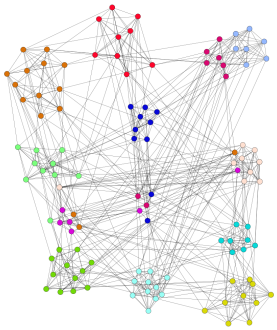
(c) Toy example #2: Infomap



(d) Toy example #2: Infomap + reweight

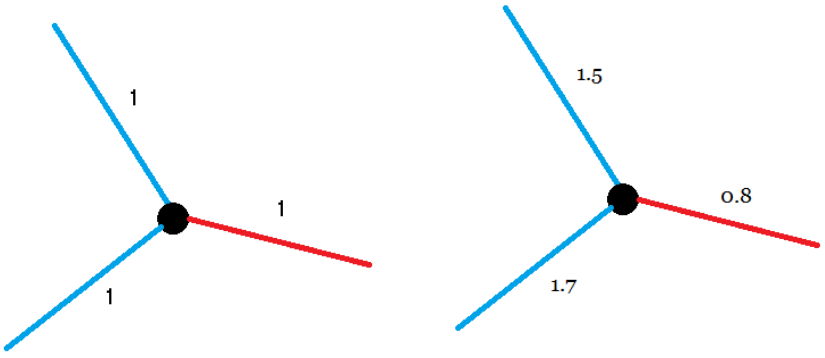


(e) Americian College Football: SLPA



(f) Americian College Football: SLPA + reweight

Figure 4.1: Examples for Showing the Original Method Recognizing Super-communities while Reweight Method Shows Better Small Communities



(a) Original Network (b) Intimacy Network

Figure 4.2: The Choice of a Random Walker at One Node: Red Lines are Inter-community Edges. Blue Lines are Intra-community Edges.

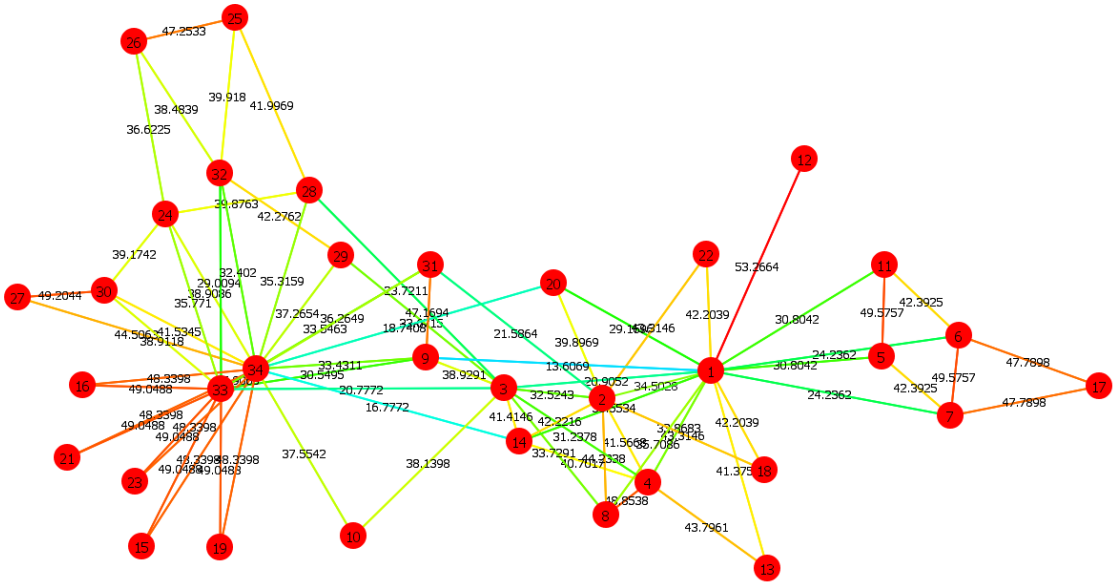


Figure 4.3: Karate Club: An Example Showing the Intimacy Idea: The numbers of links are calculated intimacies.

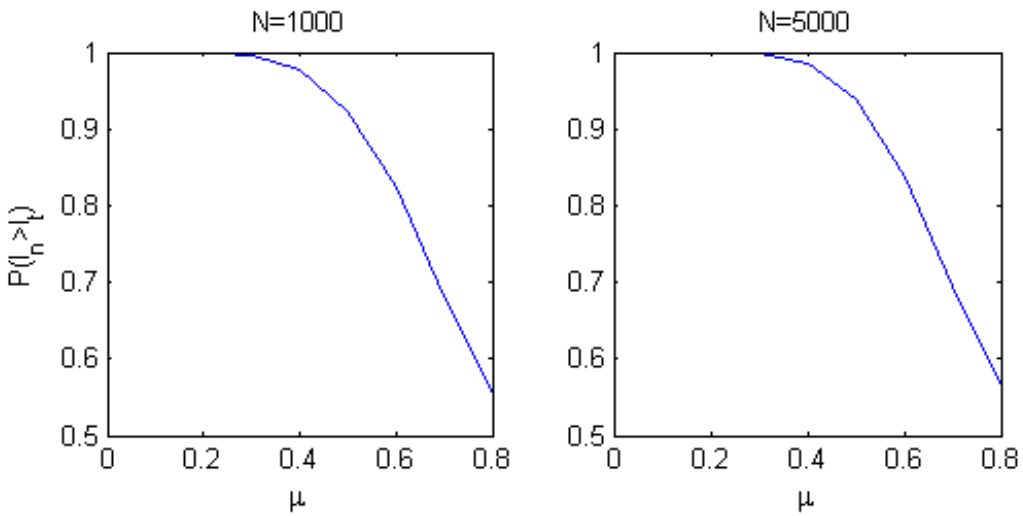


Figure 4.4: Probability for Pair Comparisons

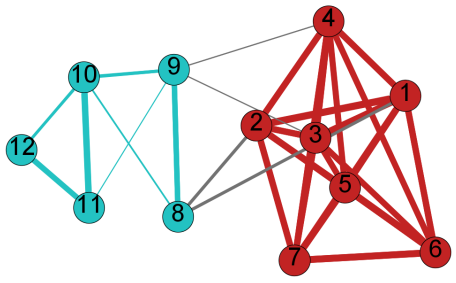


Figure 4.5: A Case Study for Exploring Reweighting Technique

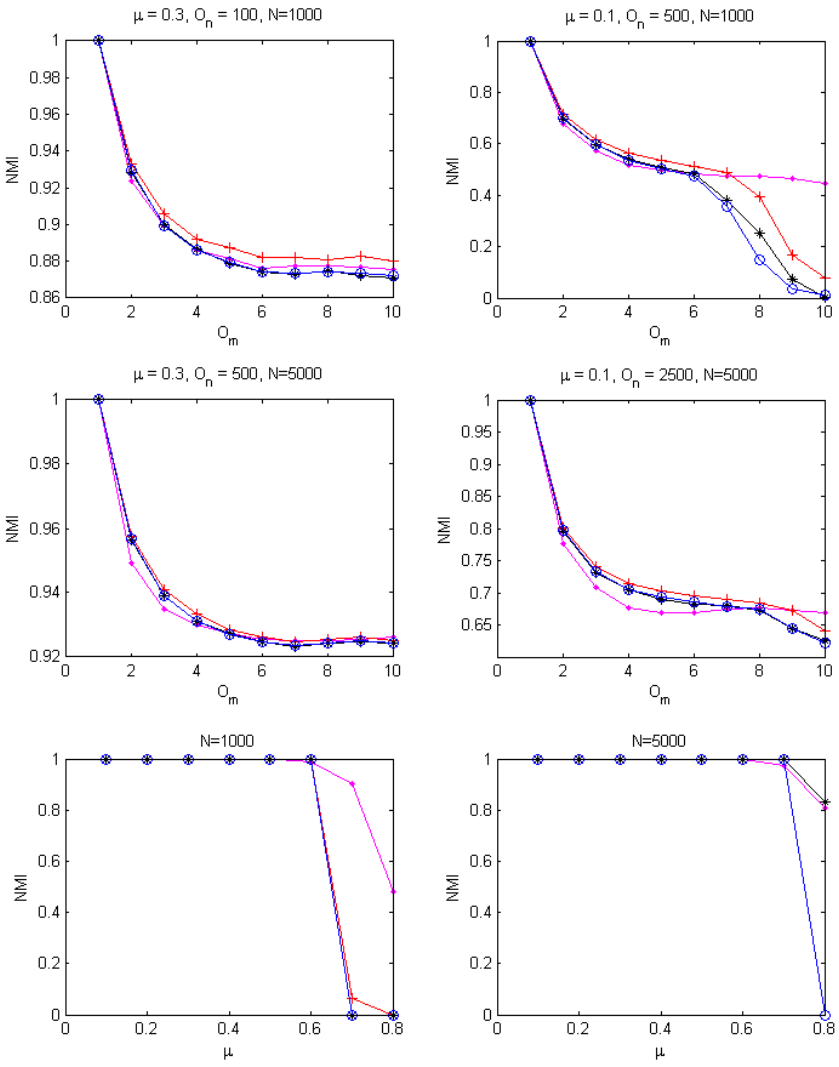


Figure 4.6: Synthetic Results: Infomap NMI

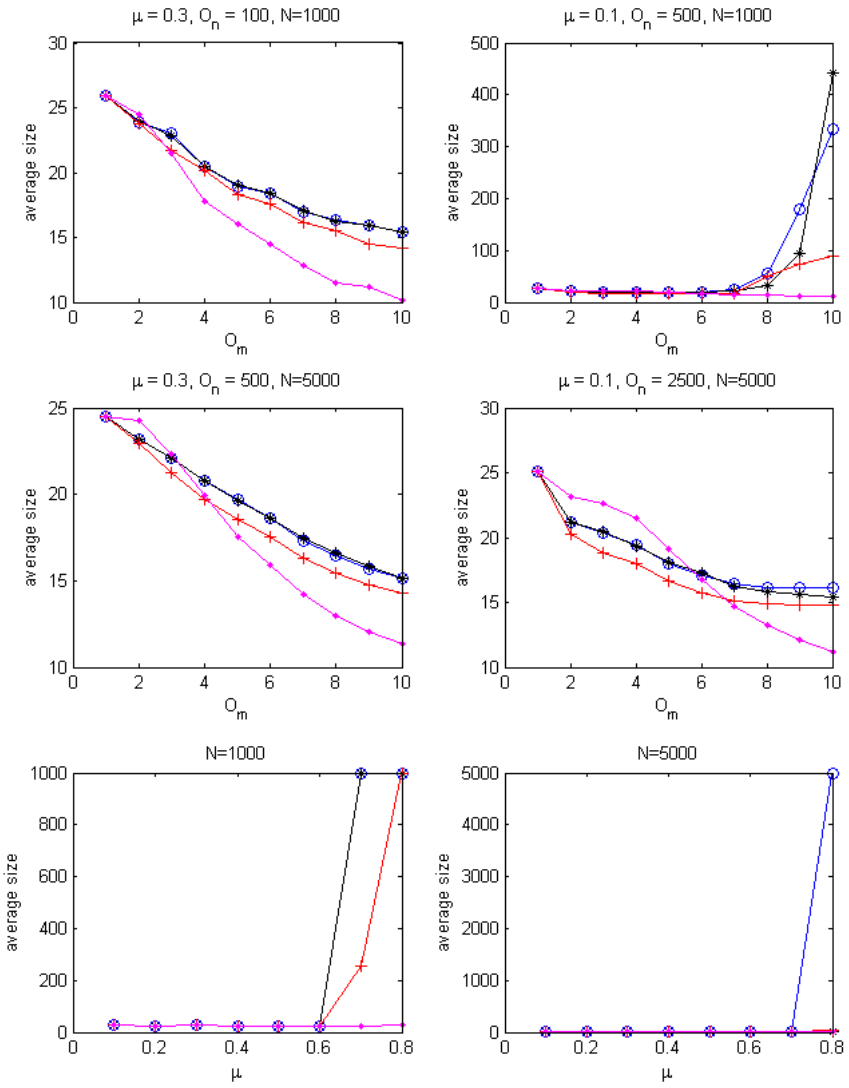


Figure 4.7: Synthetic Results: Infomap Average Community Size

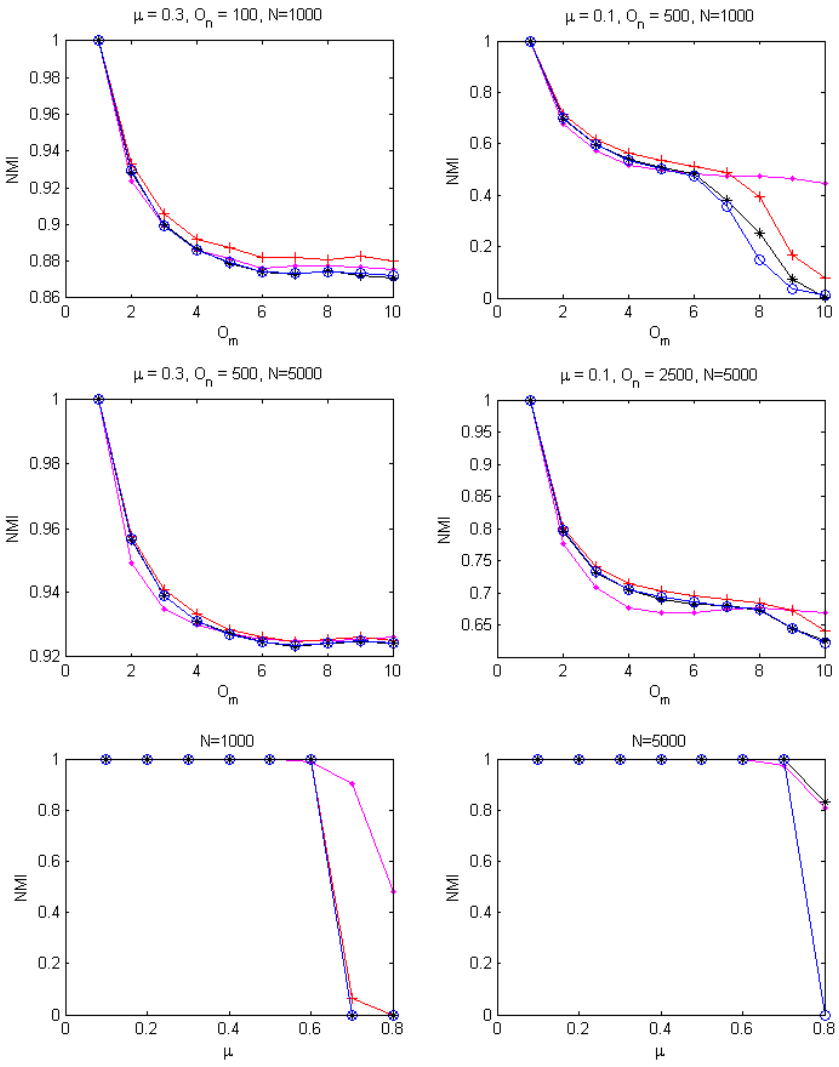


Figure 4.8: Synthetic Results: SLPA NMI

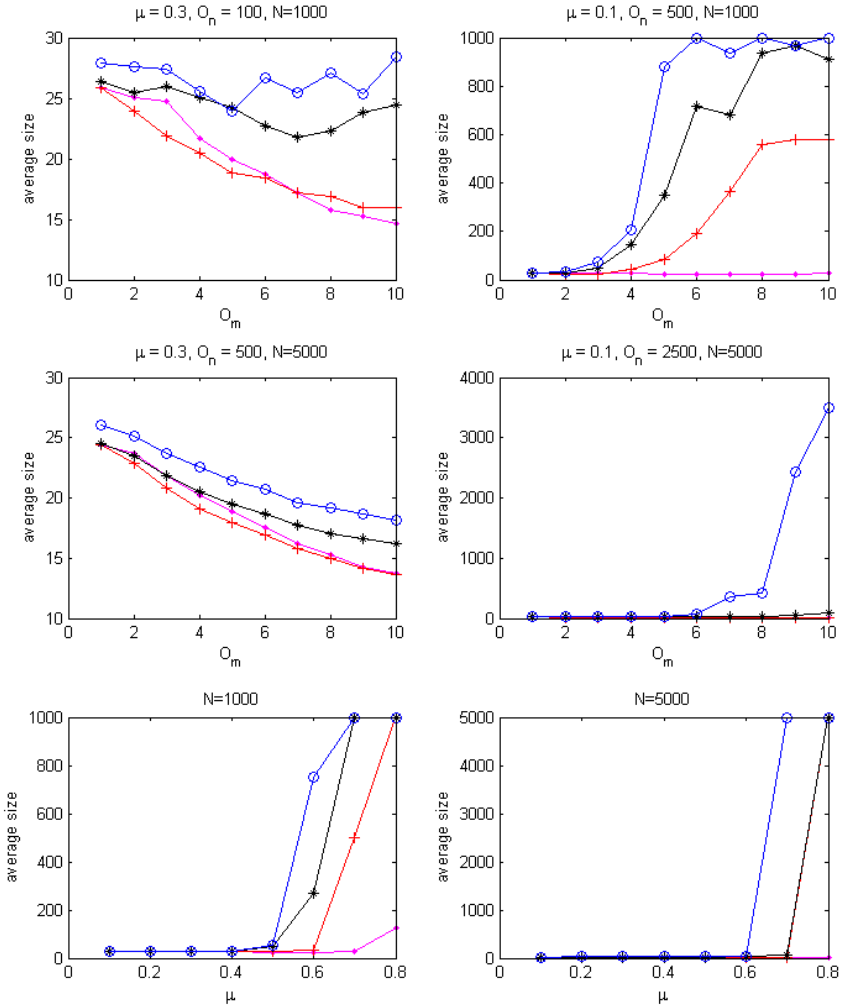


Figure 4.9: Synthetic Results: SLPA Average Community Size

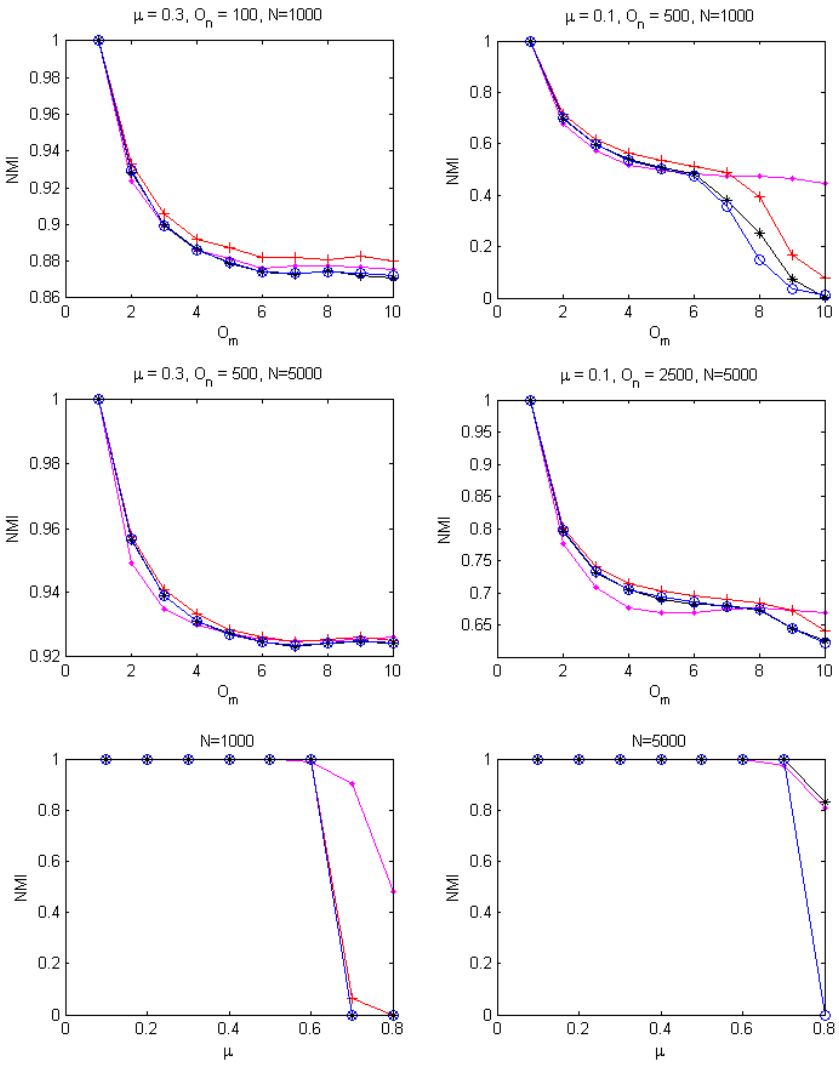


Figure 4.10: Synthetic Results: Louvain NMI

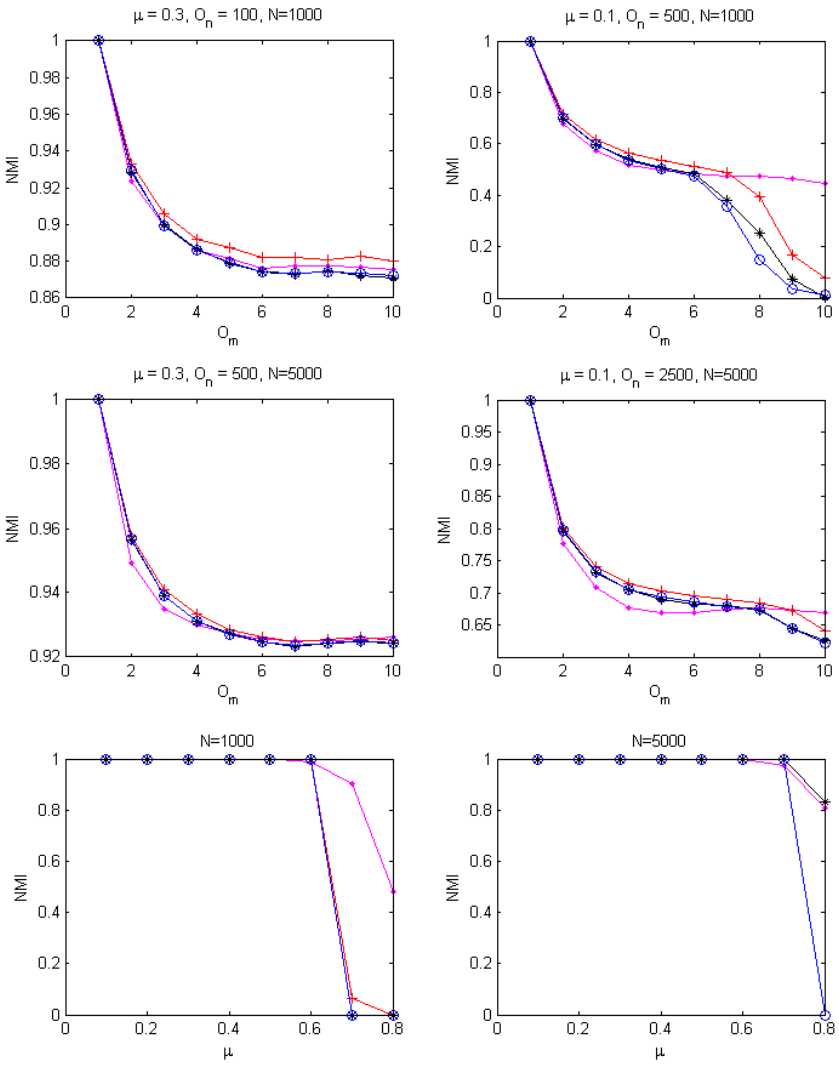


Figure 4.11: Synthetic Results: Louvain Average Community Size

Network	SLPA(ori)	SLPA(in)	SLPA(rwn)	Im(ori)	Im(in)	Im(rwn)	Lv(ori)	Lv(in))	Lv(rwn)
Karate club	0	0.46	0	0.59	0.46	0.80	0.33	0.31	0.41
ACF	0.78	0.89	0.87	0.92	0.92	0.93	0.86	0.92	0.88
Amazon	0.242	0.273	0.250	0.226	0.226	0.224	0.280	0.285	0.263
Youtube	0.024	0.018	0.013	0.004	0.006	0.005	0.027	0.026	0.018
DBLP	0.130	0.143	0.132	0.096	0.096	0.091	0.150	0.149	0.136

Table 4.2: Comparison for Different Methods on Real-world Networks

Network	ground truth	SLPA(ori)	SLPA(in)	SLPA(rwn)	Im(ori)	Im(in)	Im(rwn)	Lv(ori)	Lv(in))	Lv(rwn)
Amazon	18.38	6.44	7.65	10.48	15.82	20.58	20.51	7.08	7.80	9.40
Youtube	12.49	8.63	13.74	16.30	18.98	18.79	18.79	9.20	9.45	12.29
DBLP	52.41	11.90	7.65	10.38	20.63	20.41	19.75	5.51	5.81	8.31

Table 4.3: Comparison on Average Community Sizes for Different Methods on Real-world Networks

Chapter 5

CONCLUSION

5.1 Summary of Contribution

In this thesis, I applied a tool from numerical analysis to solve sparse linear systems in approximately linear time. By applying this tool, the betweenness measure can be easily approximated in linear time, which is an improvement over the state-of-the-art. In addition, it maintains the high accuracy.

Next, I introduced two techniques for community detection. The first technique is related to modularity. Due to the well-known problem of the resolution limit problem for modularity, it cannot detect all small real-world communities. In this thesis, I proposed a way that smaller communities should can be weighted to affect modularity with a higher proportional weight, while the large communities do the opposite. In this way, we detect smaller communities. In addition, the resolution problem is partially solved.

The second technique circumvents the resolution problem for modularity. Instead, it suggests a new technique that can be applied to nearly every community detection method. Using the reweighting technique that assigns intra-community edge higher weight, we have successfully improved all tested methods' precision. In addition, we retain the complexity of original methods since the reweighting process takes sub-quadratic time. The betweenness and triad measures show success in the reweighting process.

5.2 Further Work

Detecting small communities is a very important problem, since we have a special interest with the possible small groups one person is in. By applying reweighting techniques we are able to find more of these small communities. At the same time, there are still several problems that need to be explored further:

- Since we have already found many small communities, how can we distinguish those of high quality from those of low quality?
- Are there any recent developed techniques, other than betweenness and triad, that can be applied to computing intimacy? Among these measures, which measures are better? Is there a way that these measures can be combined together to optimize performance?
- We used the NMI measure in the thesis to measure community quality. However, a deficiency for the measure is that it punish algorithms that detect small communities embedded in larger communities. Can we develop a measure that can help to detect both large and small communities at the same time?

The first and second are easier tasks: a plan may include involving F-measure and conductance to find high quality communities among the results. We can rank the communities by lowest conductance to the highest conductance and choose the top ones.

For the third point, smaller communities, if correctly discovered, should be given more credit than it is in NMI. It is reasonable since we focus more on smaller communities. However, how to assign the new weights to make it suitable for social networks? The field requires a more thorough knowledge of statistics.

REFERENCES

Agarwal, N., H. Liu, J. Salerno and P. S. Yu, "Searching for familiar strangers on blogosphere: problems and challenges", in "NSF Symposium on Next-Generation Data Mining and Cyber-enabled Discovery and Innovation (NGDM)", (2007).

Ahn, Y.-Y., J. P. Bagrow and S. Lehmann, "Link communities reveal multiscale complexity in networks", Nature **466**, 7307, 761–764 (2010).

Andersen, R., F. Chung and K. Lang, "Local graph partitioning using pagerank vectors", in "Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on", pp. 475–486 (IEEE, 2006).

Andersen, R. and K. J. Lang, "Communities from seed sets", in "Proceedings of the 15th international conference on World Wide Web", pp. 223–232 (ACM, 2006).

Ball, B., B. Karrer and M. Newman, "Efficient and principled method for detecting communities in networks", Physical Review E **84**, 3, 036103 (2011).

Berry, J. W., B. Hendrickson, R. A. LaViolette and C. A. Phillips, "Tolerating the community detection resolution limit with edge weighting", Physical Review E **83**, 5, 056119 (2011).

Blondel, V. D., 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).

Brandes, U., "A faster algorithm for betweenness centrality*", Journal of Mathematical Sociology **25**, 2, 163–177 (2001).

Chen, W., L. V. Lakshmanan and C. Castillo, "Information and influence propagation in social networks", Synthesis Lectures on Data Management **5**, 4, 1–177 (2013).

Chen, W., Y. Wang and S. Yang, "Efficient influence maximization in social networks", in "Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining", pp. 199–208 (ACM, 2009).

Ciglan, M., M. Laclavík and K. Nørvåg, "On community detection in real-world networks and the importance of degree assortativity", in "Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining", pp. 1007–1015 (ACM, 2013).

Clauset, A., M. E. Newman and C. Moore, "Finding community structure in very large networks", Physical review E **70**, 6, 066111 (2004).

Coscia, M., G. Rossetti, F. Giannotti and D. Pedreschi, "Demon: a local-first discovery method for overlapping communities", in "Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining", pp. 615–623 (ACM, 2012).

De Meo, P., E. Ferrara, G. Fiumara and A. Provetti, "Enhancing community detection using a network weighting strategy", Information Sciences **222**, 648–668 (2013).

Dhillon, I., Y. Guan and B. Kulis, "A fast kernel-based multilevel algorithm for graph clustering", in "Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining", pp. 629–634 (ACM, 2005).

Domingos, P. and M. Richardson, "Mining the network value of customers", in "Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining", pp. 57–66 (ACM, 2001).

Fortunato, S., "Community detection in graphs", Physics Reports **486**, 3, 75–174 (2010).

Fortunato, S. and M. Barthelemy, "Resolution limit in community detection", Proceedings of the National Academy of Sciences **104**, 1, 36–41 (2007).

Fouss, F., A. Pirotte, J.-M. Renders and M. Saerens, "Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation", Knowledge and Data Engineering, IEEE Transactions on **19**, 3, 355–369 (2007).

Freeman, L. C., "A set of measures of centrality based on betweenness", Sociometry pp. 35–41 (1977).

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

Gleich, D. F. and C. Seshadhri, "Vertex neighborhoods, low conductance cuts, and good seeds for local community methods", in "Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining", pp. 597–605 (ACM, 2012).

Gregory, S., "An algorithm to find overlapping community structure in networks", in "Knowledge discovery in databases: PKDD 2007", pp. 91–102 (Springer, 2007).

Gregory, S., "A fast algorithm to find overlapping communities in networks", in "Machine Learning and Knowledge Discovery in Databases", pp. 408–423 (Springer, 2008).

Gregory, S., "Finding overlapping communities in networks by label propagation", New Journal of Physics **12**, 10, 103018 (2010).

Gregory, S., "Fuzzy overlapping communities in networks", Journal of Statistical Mechanics: Theory and Experiment **2011**, 02, P02017 (2011).

Hoeffding, W., "Probability inequalities for sums of bounded random variables", Journal of the American statistical association **58**, 301, 13–30 (1963).

Karypis, G. and V. Kumar, "Metis-unstructured graph partitioning and sparse matrix ordering system, version 2.0", (1995).

Kelley, S., *The existence and discovery of overlapping communities in large-scale networks* (RENSSELAER POLYTECHNIC INSTITUTE, 2009).

Kempe, D., J. Kleinberg and É. Tardos, "Maximizing the spread of influence through a social network", in "Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining", pp. 137–146 (ACM, 2003).

Khadivi, A., A. A. Rad and M. Hasler, "Network community-detection enhancement by proper weighting", Physical Review E **83**, 4, 046104 (2011).

Koutis, I., G. L. Miller and R. Peng, "Approaching optimality for solving sdd linear systems", in "Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on", pp. 235–244 (IEEE, 2010).

Kumpula, J. M., J. Saramäki, K. Kaski and J. Kertész, "Limited resolution in complex network community detection with potts model approach", The European Physical Journal B-Condensed Matter and Complex Systems **56**, 1, 41–45 (2007).

Lai, D., H. Lu and C. Nardini, "Enhanced modularity-based community detection by random walk network preprocessing", Physical Review E **81**, 6, 066118 (2010).

Lancichinetti, A. and S. Fortunato, "Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities", Physical Review E **80**, 1, 016118 (2009).

Lancichinetti, A., S. Fortunato and J. Kertész, "Detecting the overlapping and hierarchical community structure in complex networks", New Journal of Physics **11**, 3, 033015 (2009).

Lancichinetti, A., S. Fortunato and F. Radicchi, "Benchmark graphs for testing community detection algorithms", Physical Review E **78**, 4, 046110 (2008).

Lee, C., F. Reid, A. McDaid and N. Hurley, "Detecting highly overlapping community structure by greedy clique expansion", arXiv preprint arXiv:1002.1827 (2010).

Leskovec, J., A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen and N. Glance, "Cost-effective outbreak detection in networks", in "Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining", pp. 420–429 (ACM, 2007).

Leskovec, J., K. J. Lang, A. Dasgupta and M. W. Mahoney, "Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters", Internet Mathematics **6**, 1, 29–123 (2009).

Mann, H. B., D. R. Whitney *et al.*, "On a test of whether one of two random variables is stochastically larger than the other", The annals of mathematical statistics **18**, 1, 50–60 (1947).

McDaid, A. and N. Hurley, "Detecting highly overlapping communities with model-based overlapping seed expansion", in "Advances in Social Networks Analysis and Mining (ASONAM), 2010 International Conference on", pp. 112–119 (IEEE, 2010).

Newman, M. E., "Assortative mixing in networks", Physical review letters **89**, 20, 208701 (2002).

Newman, M. E. and M. Girvan, "Finding and evaluating community structure in networks", Physical review E **69**, 2, 026113 (2004).

Palla, G., I. Derényi, I. Farkas and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society", Nature **435**, 7043, 814–818 (2005).

Pedrycz, W., "Fuzzy sets in pattern recognition: methodology and methods", Pattern recognition **23**, 1, 121–146 (1990).

Richardson, M. and P. Domingos, "Mining knowledge-sharing sites for viral marketing", in "Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining", pp. 61–70 (ACM, 2002).

Rosvall, M. and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure", Proceedings of the National Academy of Sciences **105**, 4, 1118–1123 (2008).

Shen, H., X. Cheng, K. Cai and M.-B. Hu, "Detect overlapping and hierarchical community structure in networks", Physica A: Statistical Mechanics and its Applications **388**, 8, 1706–1712 (2009).

Shi, J. and J. Malik, "Normalized cuts and image segmentation", Pattern Analysis and Machine Intelligence, IEEE Transactions on **22**, 8, 888–905 (2000).

Tang, L. and H. Liu, "Community detection and mining in social media", Synthesis Lectures on Data Mining and Knowledge Discovery **2**, 1, 1–137 (2010).

Tsourakakis, C. E., "Fast counting of triangles in large real networks without counting: Algorithms and laws", in "Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on", pp. 608–617 (IEEE, 2008).

Wang, C., W. Chen and Y. Wang, "Scalable influence maximization for independent cascade model in large-scale social networks", Data Mining and Knowledge Discovery **25**, 3, 545–576 (2012).

Whang, J. J., D. F. Gleich and I. S. Dhillon, "Overlapping community detection using seed set expansion", in "Proceedings of the 22nd ACM international conference on Conference on information & knowledge management", pp. 2099–2108 (ACM, 2013).

Xie, J., S. Kelley and B. K. Szymanski, "Overlapping community detection in networks: The state-of-the-art and comparative study", ACM Computing Surveys (CSUR) **45**, 4, 43 (2013).

Xie, J., B. K. Szymanski and X. Liu, "Slpa: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process", in "Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on", pp. 344–349 (IEEE, 2011).

Yang, J. and J. Leskovec, "Defining and evaluating network communities based on ground-truth", in "Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics", p. 3 (ACM, 2012).

Yang, J. and J. Leskovec, "Overlapping community detection at scale: a nonnegative matrix factorization approach", in "Proceedings of the sixth ACM international conference on Web search and data mining", pp. 587–596 (ACM, 2013).

Zachary, W., "An information flow model for conflict and fission in small groups", Journal of anthropological research **33**, 4, 452–473 (1977).

Zafarani, R., M. A. Abbasi and H. Liu, *Social Media Mining: An Introduction* (Cambridge University Press, 2014).

Zhang, S., R.-S. Wang and X.-S. Zhang, "Identification of overlapping community structure in complex networks using fuzzy c-means clustering", Physica A: Statistical Mechanics and its Applications **374**, 1, 483–490 (2007).