**Title**
Social Network Analysis: Statistical Model, Community Detection and Friend Recommendation



**Author**
Yu, Xiaolu

**Publication Date**
2017

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
Los Angeles

# Scoial Network Analysis: Statistical Model, Community Detection and Friend Recommendation

A thesis submitted in partial satisfaction
of the requirements for the degree
Master of Science in Statistics

by

Xiaolu Yu

2017

ABSTRACT OF THE THESIS

Scoial Network Analysis:
Statistical Model, Community Detection and
Friend Recommendation

by

Xiaolu Yu
Master of Science in Statistics
University of California, Los Angeles, 2017
Professor Qing Zhou, Chair

In recent years, Social Network Service (SNS) is a novel, popular way to make friends and convey information online. Therefore, the analysis of network data has attracted a lot of attention. It is an area that is rapidly growing, both with Statistics and Computer Science. This paper first provides a summary of statistical methods used in network data analysis, including basic definitions, measurements, and descriptive statistics. We then introduce the Exponential Random Graph Model to fit network data. Secondly, we dig into a more specific area of network analysis: Community Detection. We discuss two different methods to explore the community stucture, one is Louvain algorithm and the other is Mixed Membership Stochastic Blockmodels. After that, we combine the community identification with a two-stage user similarity algorithm to build a friend recommendation method. In the empirical study section, we apply this method to a real-world dataset and evaluate its performace through specific measurements.

The thesis of Xiaolu Yu is approved.

Arash Ali Amini

Hongquan Xu

Qing Zhou, Committee Chair

University of California, Los Angeles

2017

*To my mother . . .*

*who—among so many other things—*

*saw to it that I learned to touch-type*

*while I was still in elementary school*

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

# CHAPTER 1

# Introduction

Internet applications have gone through the development of Web 1.0 era, ushered in a more humane Web 2.0 era. The main differences between the two stages are the performance of system and user interaction. During Web 1.0 peroid, the user can only browse information provided by the system manufacturer. Then Web 2.0 mode website focuses on the user's interactive participation, the user can not only be the reader, but also be the information provider.

Social Network Services (SNS) is a typical application of Web2.0. After a rapid developing period, there appears a lot of typical SNS websites such as Facebook, Twitter, Instagram. Social network makes full use of convenience of the network instant messaging and brings the friends circles in real life onto the online world. With the rapid expansion of network size, social network creates a huge amount of information which has high analytical value. Research based on the social network data has attracted a lot of attention in different aspects. For example, how to recommend users with their potential friends, how to help expand thier social circles, how to increase the stickness between users and keeping the stickiness between the users and the social network.

In this thesis, we start from a concise introduction to the definition and concepts of graph theory which are commonly used in network analysis. Chapter 3 discusses the classical probablistic model fitted in network data, and mainly focuses on the Exponential Random Graph Model as well as the Goodness-of-fit method. In Chapter 4 we go through two methods of community detection. The two methods can represent the two main ideas used in exploring community structures nowadays. Louvain algorithm based on the modularity optimization to divide network while Mixed Membership Stochastic Model induced from

the idea of LDA model. In Chapter5, we come up with a friend recommendation method combining community detection mentioned in the Chapter4 and a two-stage user similarity algorithm. This method aimed to recommend latent friends for the users in the net in a more efficient way. The empirical study are done in Chapter 6, in which we apply both the ERGM model and the new latent friend recommendation method to a facebook network dataset. Chapter 7 outlines the whole conclusion about the thesis together with remaining problems and future work.

For simplicity, we restrict our study to the models for static binary network data. That is to say, the relationship between nodes is a link (edge) either present or not, with relationships only considered at one occasion.

# CHAPTER 2

# Graph Theory and Summary Statistics

In this chapter, we mainly focus on the basic definitions and concepts about graph theory and summary statistics that will be mentioned or used in this thesis.

## 2.1 Basic Definitions

Network is typically described using graph theory. A graph $G(V, E)$ consists of a set of N vertices (nodes) $N = \{n_1, n_2, ..., n_N\}$. The set of $E$ edges $E = \{e_1, e_2, ..., e_E\}$ denotes the links between nodes. When applied to the social network data, nodes represent the users in the net and edges linking two different nodes show there are connections between the two users. Examples of network graph are shown in Figure 2.1. Zachary Karate Club Network Dataset is a very typical dataset used to represent network structure. Each node is a club member and the links between them show their friendship with each other.

**Adjacency Matrix** Adjacency matrix is a square matrix used to represent a finite graph $G$ of dimension $N \times N$. The elements in the matrix indicate whether the pairs of nodes are adjacent or not in the graph.

$$\begin{cases} y_{ij} = 1 & \text{links exists from node } n_i \text{ to node } n_j \\ y_{ij} = 0 & \text{links doesn't exist between node } n_i \text{ to node } n_j \end{cases} \tag{2.1}$$

Note that this equation only applies to binary network, valued networks are using non-negative integer values for the entries in matrix.

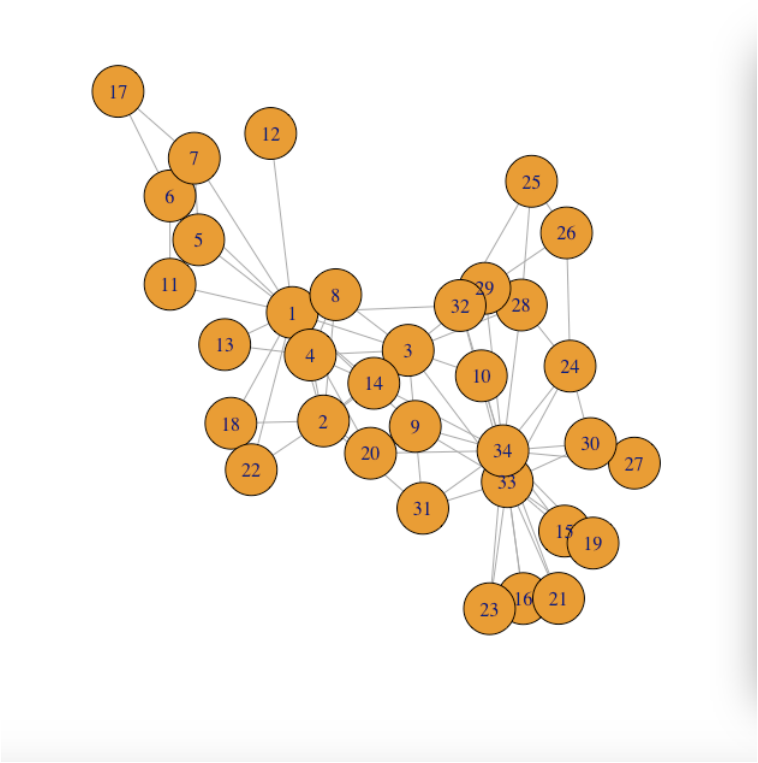


Figure 2.1: Examples of graph theory, present an undirected graph derived from Zachary Karate Club network dataset

**Symmetry** Links in a network may be symmetric or asymmetric, in another word, undirected or directed. For example, the links in facebook are undirected cause the ties of friendship will be returned and the links in twitter are directed as your friend may not add you into his or her friend list. From adjacency matrix perspective, an undirected graph's matrix is symmetric ($y_{ij} = y_{ji}$).

**Geodesic Distance** The geodesic distance $d(i,j) = min_k y_{ij}^{[k]} > 0$ is a tool to measure the connectivity structure of network in graph theory. It shows the length of the shortest path between vertices $n_i$ and $n_j$. For instance, $y_{ij}^{[k]} = 1$ indicates there is a path of length $k$ between nodes $i$ and $j$. The graph is connected if all the geodesic distances are finite for nodes in the network. Otherwise, it is unconnected. The diameter of a graph is said to be the largest geodesic distance between two nodes in the network.

## 2.2 Summary Statistics

The behavior in the graph can be summarized using various statistics, these statistics not only show general information about network but are also relevant to some statistical models like ERGM.

**Degree** The degree of a node is the number of edges incident on it. It is the simplest indicator which directly shows how a node is connected within a graph and how important it is. In a directed graph, in-degree is the number of incoming links and out-degree is the number of outgoing edges.

**Density** The density of a network graph is the number of existing edges divided by the number of possible ones. It is easy to conclude that a network with higher density is more strongly connected and as a result usually can resist link failures better.

**Betweenness Centrality** This is a measure of the degree to which a given node lies on the shortest paths (geodesics) between other nodes in the graph. For node $v$ in graph $G$, the betweenness centrality $C_b$ is defined as:

$$C_b(v) = \sum_{s,t \neq v} \frac{\Omega_v(s,t)}{\Omega(s,t)}$$

$\Omega(s,t)$ is the number of distinct geodesics from $s$ to $t$ and $\Omega_v(s,t)$ is the number of geodesics from $s$ to $t$ that pass through $v$.
A vertix has high betweenness if the shortest paths between many pairs of other nodes in the graph pass through it.

**Closeness Centrality** CLC measures the centrality of a node by its closeness (distance) to ohter nodes. CLC of a node $v$ is defined as:

$$CLC(v) = \frac{|V| - 1}{\sum_{i, v \neq v_i} d(v, v_i)}$$

$|V|$ indicates the number of nodes in the specific graph and $v_i$ is the node $i$ of the graph. The closeness centrality decreases if the amout of vertices reachable from the given vertix decreases, or the distances between the nodes increase.

**K-cliques and K-stars** These two are particular types of sub-graph. K-cliques is a sub-graph of $k$ nodes where all nodes are connected to each other. K-stars is a sub-graph of $k+1$ nodes in which $k$ of the nodes are connected through a single node. Examples of the sub-graph structures are shown in Figure 2.2.

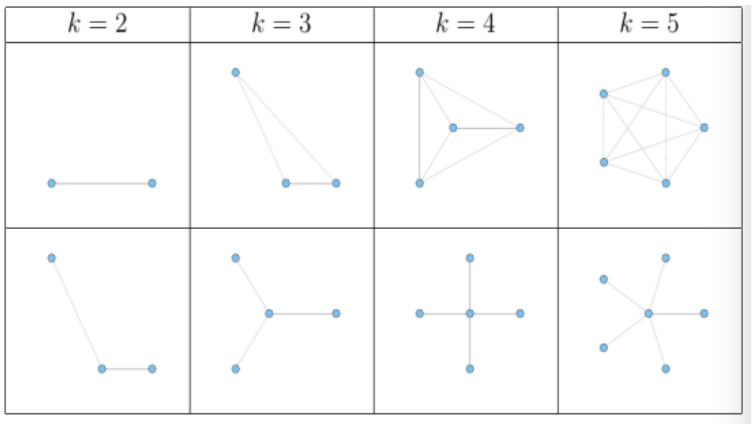


Figure 2.2: Sub-graph structures: top row is k-cliques and bottom row is k-stars

# CHAPTER 3

# Statistical Network Modeling

Exponential-framily random graph model (ERGMs)[FS86] is a kind of classical probablistic models which assumes a likelihood function for the network data given some underlying parameters. We then can obtain maximum-likelihood estimates for the parameters in the model from a given dataset, evaluate specific models for goodness-of-fit, perform model comparisons and do network simulation with the underlying distribution of the model.

## 3.1 Exponential(family) Random Graph Model

Whenever the density of a random variable being written as:

$$f(x) \propto \exp\{\theta^t s(x)\}$$

the family of all such random variables is called an exponential family. And since the random graphs in our model form an exponential family, we call the model Exponential (family) Random Graph Model. The form for an ERGM model can be expressed as:

$$P_\theta(X = x) = \frac{\exp(\theta^t s(x))}{c(\theta)}$$

$X$ is the random network with $n$ nodes (a matrix of 0's and 1's), $\theta$ is a vector of parameters, that is, the vector of coefficients for those statistics. $s(x)$ is a known vector of model statistics on network $x$, $\theta^t$ is the transpose of $\theta$. Typical choices for statistics include the number of edges, the number of triangles and the number of k-stars for different k values.

$$c(\theta) = \sum_{\text{all possible graph y}} \exp\{\theta^t s(y)\}$$

$c(\theta)$ is a normalizing "constant" represents the quantity in the numerator summed over all possible networks (typically constrained to be all networks with the same node set as $x$). Replacing $s(x)$ by $s(x) - s(x^{obs})$ leaves $P_\theta(X = x)$ unchanged, thus we recenter $s(x)$ so that $s(x^{obs}) = 0$

Then the loglikelihood function is:

$$\ell(\theta) = -\log\ c(\theta) = \log \sum_{\text{all possible graph y}} \exp\{\theta^t s(y)\}$$

Fitting the ERGM then involves finding the estimates of the parameters for all the network statistical terms in the model. However, merely evaluating (maximizing) $\ell$ is actually computationally berdensome. As it is intractable to compute the summation item $c(\theta)$. For example, a network containing only 10 nodes have $3.52\times10^{13}$ possible network configurations. Thus there are several approaches to fit ERGM models without summing over all possible networks, including maximum pseudolikelihood estimation[SI90], Monte Carlo MLE[VGH09] and MCMC[CF11].

Here we choose the Monte-Carlo based inference to do approximation. Frequentist MCMC methods for ERGMs (MCMCMLE) approximate parameter estimates by comparing the observed network with a set of simulated networks given a parameter configuration. Note that a computationally feasible sample of networks consistent with the observed summary statistics is used in the set of all consistent networks. We first initialize an arbitrary estimate $\theta^{(0)}$, then use MCMC to sample M networks from an ERGM model with the parameters. We define these sampled networks by their adjacency matrices $X^{(1)}, ..., X^{(M)}$ and derive the MCMC log-likelihood equation:

$$\log P_M(Y|\theta) = \theta^t S(X) - \log(c(\theta^{(0)})) - \log(\frac{1}{M}\sum_{m=1}^{M} \exp(\theta^t S(X^{(m)}) - (\theta^{(0)})^t S(X^{(m)})))$$

As $M \to \infty$, the limit of this MCMC log-likelihood is equal to the log-likelihood of the ERGM. The argmax of $\theta$ is referred to as the MCMCMLE of the network.

## 3.2 Goodness-of-Fit

The Goodness of Fit diagnostics maybe the most sophisticated and developed methods for model validation. This method is developed at first specifically for the ERGM class of models, but now can also been applied to other models like latent space models. As a matter of fact, this method can be applicable to any fitted models which have already simulated networks. ERGMs can be seen as generative models when they show the process that covers the global patterns of links prevalence from a local perspective. And this method is motivated by the generative properties.

To test whether the local model fits the data we now focus on how well it reproduces the observed global network properties which are not in the model. "When ERGM parameters are estimated and a large number of networks are simulated from the resulting model, these networks frequently bear little resemblance at all to the observed network"[HRS03]. We do this by choosing a network statistic that is not in the model, for example, the observed degree distribution, and compare the value of this statistic observed in the original network to the distribution of values we get in simulated networks from our model.

# CHAPTER 4

# Community Detection

In this chapter, we dig into a very popular field in social network analysis: Community Detection. Exploring community structure can help us get deeper comprehension of the whole network and its properties. In the real-world social network study, community detection can help with the analysis of the user behavior in the specific community, provide more effecient support of commercial promotions.

## 4.1 Definition of Community

A precise definition of what a "Community" really is does not exist yet. One of the most widely accpted and uesd definition is given by Newman and Gievan: A community is a subgraph containing nodes which are more densely linked to each other than to the rest of the graph or equivalently, a graph has a community structure if the number of links into any subgraph is higher than the number of links between those subgraphs[New04].
In community detection, what we do is to confirm $nc(>1)$ communities in the Graph $G(V, E)$.

$$C = \{C_1, C_2, ..., C_{nc}\}$$

and the union of vertices in all communities can cover $V$ in graph $G$.
If the intersections of every two communities' vertices are all null sets, we define $C$ as disjoint community, otherwise it is called overlapping community. Through this concept, we can also divide the algorithms of community detection into disjoint community detection (Figure 4.1) and overlapping community detection (Figure 4.2).

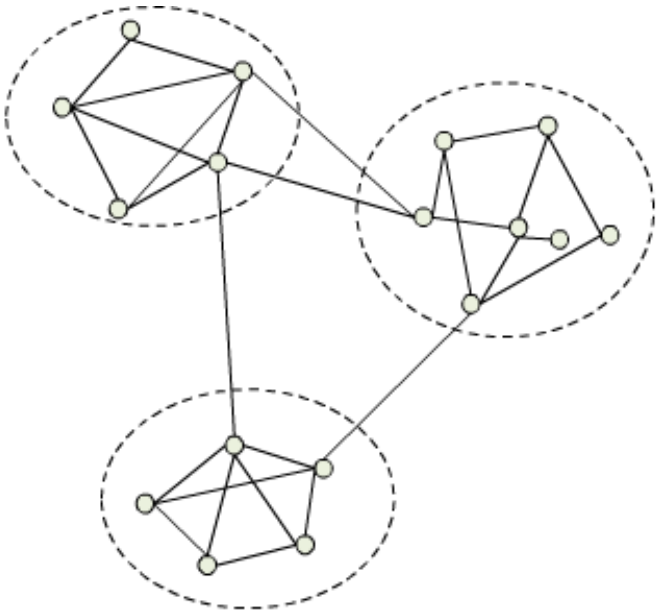

Figure 4.1: Disjoint community detection

## 4.2 Overview of Community Detection Algorithm

After Newman and Girvan came up with a classical community detection algorithm: GN algorithm, the study of social network community ushered into a rapid develop period. Until now, there are huge amount of algorithms and many more are being created. Generally speaking, all the algorithms can be divided into two different types.

**Classical Community Detection Algorithm** The traditional community detection algorithms mostly work on undirected, unweighted network and create disjoint communities. But now some of them have been improved to discover overlapping communities like Spealer-listener Label Propagation Algorithm (SLPA). The main idea of these methods is to get use of the information brought by vertices or edges, like betweenness, modularity and so on. Use the messages to classify or cluster nodes within the net to achieve community detection. For instance, hierarchical clustering uses Euclidean Distance as a measure of similarity between two targeted nodes and take it as the standard to do clustering(Figure 4.3).

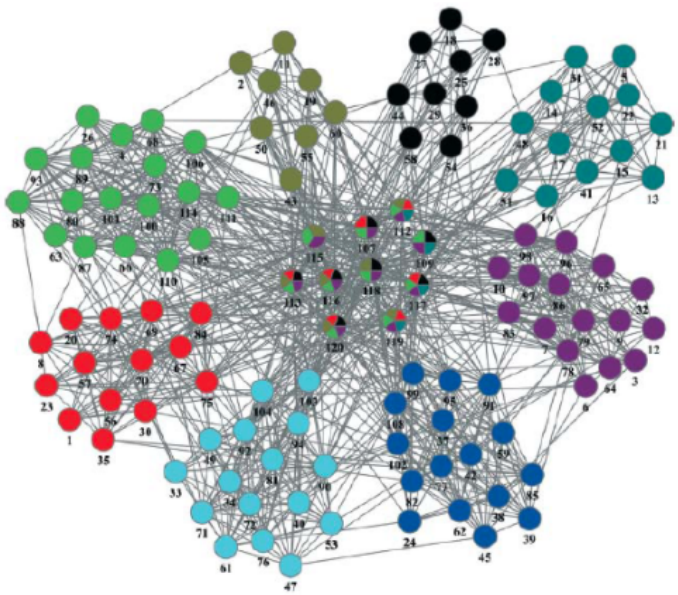

Figure 4.2: Overlapping community detection

**Community Detection based on Statistical Model** In recent years some algorithms has been created of detecting community structure in social networks through statistical modeling. The core idea of this method is to establish a statistical model based on the actual network data and simulate the observed network according to the model. Use the observed network data and statistical methods to transform the community detection problem into bayesian inference problem. And then use statistical theory and observed data to obtain the characteristics of the data set, so as to divide the network. This method of statistical modeling has a reliable theoretical basis of probability, and can be a well performed community detection algorithm of social networks. This method has become a hot spot in recent years' studying. For example, Planted Partition Model and Latent Position Cluster Model (Figure 4.4).

## 4.3 Louvain Algorithm

The Louvain method [DFF11]is a simple, easy-to-implement method for identifying communities in large networks. It is a disjoint community detection, but as it can implement network data in high efficiency, the method is quite popular and has been applied success-

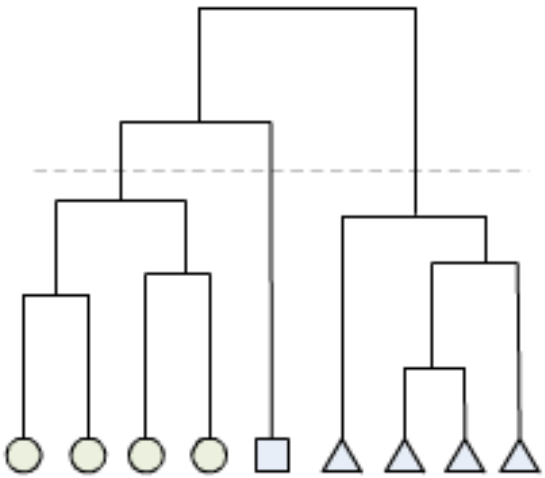

Figure 4.3: Hierarchical clustering

fully onto many types of massive network datasets of sizes up to 100 million nodes and billions of links. The method reveals hierarchies of communities and allows to zoom within communities to discover sub-communities, sub-sub-communities, etc. Until now it is one of the most widely used methods for detecting communities in large networks.

The Louvian method is a greedy optimization method that attempts to optimize the modularity of a partition of the network.

**Modularity** Modularity is a measure of the structure of networks or graphs. It was designed to measure the strength of division of a network into modules (Communities). Networks with high modularity have dense connections between the nodes within communities but sparse connections between nodes in different communities. Modularity is widely used in optimization for exploring community structure in networks. The modularity $Q$ usually be defined as:

$$Q = \frac{1}{(2m)} \sum_{vw} \left[ A_{vw} - \frac{k_v k_w}{(2m)} \right] \delta(c_v, c_w) = \sum_{i=1}^{c} (e_{ii} - a_i^2)$$

$m$ is the number of links in the graph, $A$ represents the adjacency matrix, $k_w$ and $k_v$ indicate the degree of the specfic vertices, $c_u$ is the number of the community which vertix $u$ belongs to.

$e_{ij}$ is the fraction of edges with one end vertices in community $i$ and the other in community

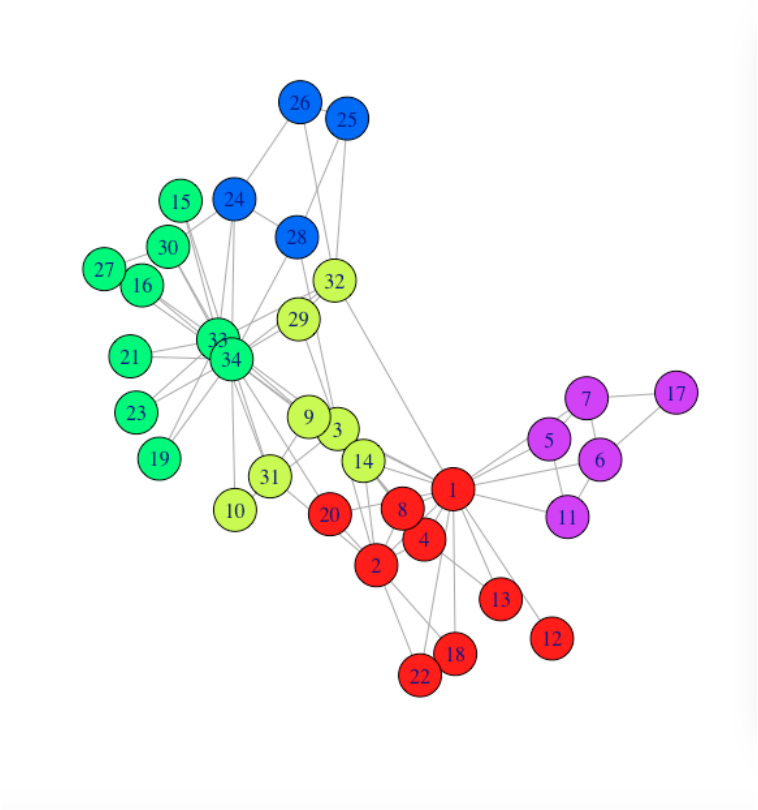


Figure 4.4: Latent Position Cluster Model applied on Zachary Karate Club network dataset

$j$:

$$e_{ij} = \sum_{vw} \frac{A_{vw}}{2m} 1_{v \in c_i} 1_{w \in c_j} e_{ij} = \sum_{vw} \frac{A_{vw}}{2m} 1_{v \in c_i} 1_{w \in c_j}$$

and $a_i$ is the fraction of ends of edges that are attached to vertices in community $i$:

$$a_i = \frac{k_i}{2m} = \sum_j e_{ij} a_i = \frac{k_i}{2m} = \sum_j e_{ij}$$

**Algorithm**

1. Treat each node in the graph as a separate independent community. At this point the number of communities is the same as the number of nodes.

2. For each node $i$, try to allocate it to the communities which its neighbor nodes belong to respectively. Then calculate the difference of modularity before and after allocation as $\Delta Q$. Record the neighbor node with the largest $\Delta Q$. If $max \Delta Q > 0$, assign node

$i$ to the community where the neighbor node of $\Delta Q$ is the largest, otherwise remain unchanged.

3. Repeat Step 2, until the community of all nodes does not change.

4. Compress the graph, compress all the nodes in the same community into a new node, the weight of the edges between the nodes in the community are then transformed into the new node weighted sides.

5. Repeat Step 1, until the modularity of the entire graph no longer changes.

The method seems to run in time complexity $O(nlogn)$ with most of the computational effort being spent on the optimization at the first level.

## 4.4 Mixed-Membership Stochastic Blockmodel

Mixed Membership Stochastic Blockmodel[ABF08] is a Bayesian model for overlapping communities detection. In this model, the community memberships are treated as hidden random variables. Given an observed network, such as a social network of friendship ties, we can discover the hidden community structure by estimating its conditional distribution.

The classical community membership models, for example, the stochastic blockmodel, assume that each vertix belongs to just one community. This kind of models doesn't realize that a specific node's link can be expressed by its memberships in several overlapping communities. And this situation is very common in real-world networks. On the contrary, MMSB is a type of "mixed-membership" model, that means each vertix can exhibit multiple communities.

MMSB assumes there are $K$ communities and each vertix $i$ is related with a vector of community memberships $\theta_i$. This vector is a distribution over the communities.

$$\sum_i \theta_i = 1 \qquad \theta_i > 0$$

For example, in a real-world social network, a user has half of her friends from school and

the other half from her neighborhood. For this vertix, $\theta$ would place one-half of its mass on the school community and the other half onto the neighborhood community.

To construct a network, the model thinks of each pair of vertices. For each pair $i, j$, it chooses a community indicator $z_{i\to j}$ from the $i$th vertixs community memberships $\theta_i$ points to one of the communities that its corresponding node belongs to, and then chooses another indicator $z_{j\to i}$ from $\theta_j$. If these indicators point to the same community, then it means vertices $i$ and $j$ has high probability to be connected, otherwise, they are likely to be disconnected.

These assumptions show that ties between vertices can be explained by their memberships in multiple communities, even without the information of where those communities are. Now we compute the probability that the model connects for a single pair of vertices$(i, j)$, conditional on their community memberships.

$$p(y_{ij} = 1|\theta_i, \theta_j) = \sum_{k=1}^{K} \theta_{ik}\theta_{jk}\beta_k$$

$\beta_k$ is the probability that two vertices are connected given that their community indicators are both equal to $k$. Make an assumption that the two vertices have zero probability of being connected if the indicators point to different communities. $\theta_{ik}$ and $\theta_{jk}$ represent the probabilities that both vertices draw an indicator for the $k$th community from their memberships. Then we can see the probability of connection will be high if $\theta_i$ and $\theta_j$ share high weight for at least one community, such as the two users work in a same company. It will be low if there is little overlap in their communites. The main idea of this model is that: Vertices with similar memberships will be more likely connected with each other.

Now for the full network, the model assumes the following generative process:

1. For each vertix, draw community memberships from Dirichlet distribution: $\theta_i \sim Dirichlet(\alpha)$
2. For each pair of vertices $i$ and $j$ $(i < j)$:
   (a) Draw community indicator $z_{i\to j} \sim Multinomial(\theta_i)$
   (b) Draw community indicator $z_{j\to i} \sim Multinomial(\theta_j)$

(c) Draw the conncetion between them from Bernoulli distribution:

$$p(y_{ij} = 1 | z_{i \to j, z_{j \to i}}) = \begin{cases} \beta_{z_{i \to j}} & z_{i \to j} = z_{j \to i} \\ \epsilon & z_{i \to j} \neq z_{j \to i} \end{cases}$$

Now given the observed network, the model fits a posterior distribution which gives a division of the nodes into $K$ overlapping communites, that is, the conditional distribution of the hidden community structure.

$$p(\theta, z|y) = p(\theta, z, y)/p(y)$$

$\theta$ is the joint probability distribution over the $N$ per-node community memberships, $z$ is the per-pair community indicator and $y$ is the observed network. The posterior will place high probability on configurations of the memberships that describe densely connected communities. With this posterior, we can investigate the community structure. However, the posterior inference in this bayesian model is difficult. The numerator is easy to compute as mentioned above, it is just a joint distribution defined by modeling assumption. But the denominator is the marginal probability of the network data, which sums over all possible hidden community structures.

$$p(y) = \int_{\theta} \sum_{z} p(\theta, z, y)$$

Just as the normalizing "constant" $c(\theta)$ in ERGM, compute this marginal over $N$ variables will require a summation over $K^{N^2}$ configurations of community indicators.

To approximate the posterior and parameters, this model applies VEM algorithm instead of MCMC (like Gibbs Sampling). VEM is an improved algorithm of EM algorithm and this novel action largely enhances MMSB's efficiency, makes it possible to apply model onto massive network data containing hundreds, thousands, millions vertices. The complexity of this method is $O(KN^2)$.

# CHAPTER 5

# Friend Recommendation Algorithm

Social network is designed to help users build and expand their "social circles". It gradually changes the way of people's information sharing and communication. Nowadays, social networking platforms are attracting millions of users from different ages, nationalities. Research shows that in social networks, users not only want to connect with friends that have alraady been recognized in real life, but also hope to get to know some of the new friends who share the same interests. But with the explosive growth of social network size, it is becoming increasingly difficult for users to obtain new friends. As a result, latent friend recommendation algorithms are created to help with this issue. In this chapter, we first introduce a two-stage friend recommendation algorithm based on user similarity and then combine it with community detection to build a more efficient friend recommendation algorithm.

Currently, collaborative filtering recommendations and content-based filtering recommendations are the most widely used recommendation algorithms. Collaborative filtering based on the basic idea: use the user's project score to find neighbors who have similar interests with the target users. And then use the neighbor user's score of the targeted project to recommend. Content-based recommendation algorithm based on the users' description of their interests and their behavior to establish interests models, and then conduct similarity analysis with the targeted projects. These two recommendation methods are based on the network of users and items. For the potential friends recommendation in the social network, the recommended object itself is also the social network's user, so the algorithms for recommending items and recommending friends are different.

Similarity is a significant standard in recommendation algorithms based on social networks. In the recommendation algorithms, the measurements of similarity between users are mainly

divided into cosine similarity, correlation similarity and so on. In addition, it is typical to calculate the user similarity based on the link information between the users.

## 5.1 Two-Stage Recommendation Model

The recommendation algorithm based on user similarity here uses a two-stage friend recommendation model[Zho11]. The first stage of the model is the use of cosine similarity to evaluate user similarity. It based on the adjacency matrix of the user's chain relation network $G(V, E)$. From the adjacency matrix $C$, we obtain the eigenvector representation of each user, and use the method of cosine similarity to calculate the function of the two users' similarity. The second stage of the model is the spread of similarity which takes the impaction of the user's friends into consideration. For example, if many of user A's friends like user C, then we can assume that user A may be interested in user C.

Now represent user A and B's eigenvector as $(c_{a1}, c_{a2}, ..., c_{an})$ , $(c_{b1}, c_{b2}, ..., c_{bn})$, where $c_{ij}$ is the entry in the adjacency matrix.

For the first stage, we calculate user A and B's cosine similarity as:

$$Sim(a, b) = \frac{\sum_{i=1}^{n} c_{ai} c_{bi}}{\sqrt{\sum_{i=1}^{n} c_{ai}^2} \sqrt{\sum_{i=1}^{n} c_{bi}^2}}$$

For the second stage, the similarity between two users spread along user's friend circles. Assume user A and C, besed on the stage 1, we improve the user similarity equation as:

$$Sim^{'}(a, c) = (1 - \lambda) \times Sim(a, c) + \lambda \times \sum_{c_{uv}=1} \frac{Sim(v, c)}{\sum_u c_{uv}}$$

$\lambda$ is a parameter shows the influence by user's friends, to simplify calculation, set $\lambda$ as a constant and $\lambda \in [0, 1]$. Through empirical study, the author of this model set the $\lambda$ as 0.9, $v$ is the set of friends of A and $u$ is the set of all the users in the graph.

## 5.2 Combine Community Detection with Friend Recommendation Algorithm

For the application of the recommended algorithms, most of the friends recommendation algorithms need to depend on certain information such as user connection information, friend similarity to build the recommended model. This kind of models often use matrix to achieve their goal and produce recommendations. In the real-world social network, however, with size over millions or even tens of millions of large data sets, the similarity calculation will cost huge time and space. In turns the user may be waiting for too long to get the recommendation they want. As a result, it will cause poor user experience and some other bad issues. But if we apply the algorithms after community detection, that is to say, we apply our recommendation algorithms onto a smaller sub-graph, the problems above maybe solved.

Through the observation and analysis of social networking site, within the huge amount of network users, most of them will gradually form a small group structure, this small group structure is just the community we talked in chapter 4. In general, users in social networks are most likely to be friends with people from the same social circles, and the probability of making friends in other circles is relatively low.

What's more, after the community detection for the network, we can target different users and recommend different communities to meet their various needs. And since the division of the community is reusable in a period of time, we can effectively improve the recommendation efficiency.

The whole process of the improved algorithm is as below:

1. Apply Community Detection Algorithm (In this thesis, I apply Louvain algorithm and MMSB) to the input social network dataset and output the divided communities corresponding with vertices in the network.

2. Input the user $u$ who needs recommendation, search for the community he belongs to, output the name of communities where he in.

3. Input user $u$'s community and construct its adjacency matrix $C$, use two-stage sim-

ilarity algorithm to calculate the similarity $Sim^{'}$ between $u$ and other users in the community. Output the list of similarity values of similarities.

4. Input the number of recommended friends $K$, take the $k$ users in the community which have the largest $k$ similarity values with user $u$. Output the No. of these users.

Note that if user $u$ is in multiple communities after overlapping community detection, then merge the overlapped communities into one and calculate the similarity based on the merged adjacency matrix. If $K$ is larger than the number of nodes in $u's$ community, calculate the similarity over the whole network dataset.

# CHAPTER 6

# Experiments

## 6.1 Data Selection

I collect the social network dataset from SNAP: Stanford Network Analysis Project Website (https://snap.stanford.edu/index.html). The network dataset is grabbed from facebook. Table 6.1 provides the basic summary statistics of this dataset. Figure 6.1 shows the degree distribution of this dataset. Since the number of nodes in the network is over 1000, visualization of this network comes up with the problem of "hairball" phenomenon: the structure is too complex to project onto two or even three dimensions and the edges overlap heavily. Even using the layout algorithms like Fruchterman-Reingold[FR91] can not provide a good performance. So here, I create an adjacency matrix visualization of the facebook network (Figure 6.2). The visualization of the adjacency matrix may be the simplest way to visualize a network. The rows and columns of the adjacency matrix are reordered such that the vertices are gathered into highly connected clusters. Here we can see that a local tie structure is apparent from the blocks along the diagonal.

| Nodes | Edges | Average degree | Density |
|---|---|---|---|
| 4039 | 88234 | 43.6901 | 0.0144 |
| Triangles | kstar 1 | kstar 2 | kstar 3 |
| 1612010 | 176468 | 9314849 | 727318426 |

Table 6.1: Summary Statistics of Facebook Social Network Dataset

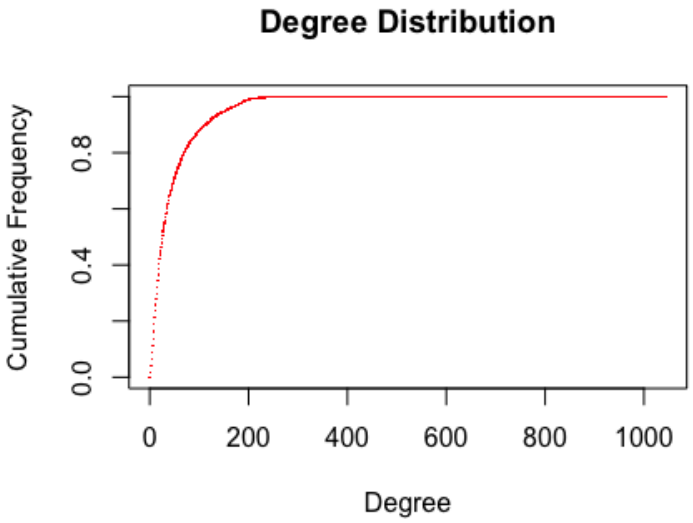


Figure 6.1: Facebook network data degree distribution

| Edges | Triangles | Gwdegree |
|---|---|---|
| -5.168 | 0.123 | -0.454 |

Table 6.2: Coefficients of terms used in ERGM fitting

## 6.2 ERGM Fitting

We fit an Exponential Random Graph Model to the facebook network dataset using the statistical terms: number of edges, triangles and geometrically weighted degree. These are among the most common choices for ERGM terms. After the iterations of MCMCMLE methods, we got the coefficients as shown in Table 6.2.

Then the conditional log-odds of three actors having a tie is:

$$-5.168 \times \text{change in the number of ties} + 0.123 \times \text{change in number of triangles}$$
$$-0.454 \times \text{change in number of geometrically weighted dgree}$$

For the ERGM, we know that better performance could be achieved using a more carefully chosen set of network summary statistics. The gooodness of fit results for this fiting is the three plots: Figure 6.3, Figure 6.4 and Figure 6.5. The three plots depict the fit for ERGM in terms of the distribution of nodal degree, edgewise shared-partners and geodesic

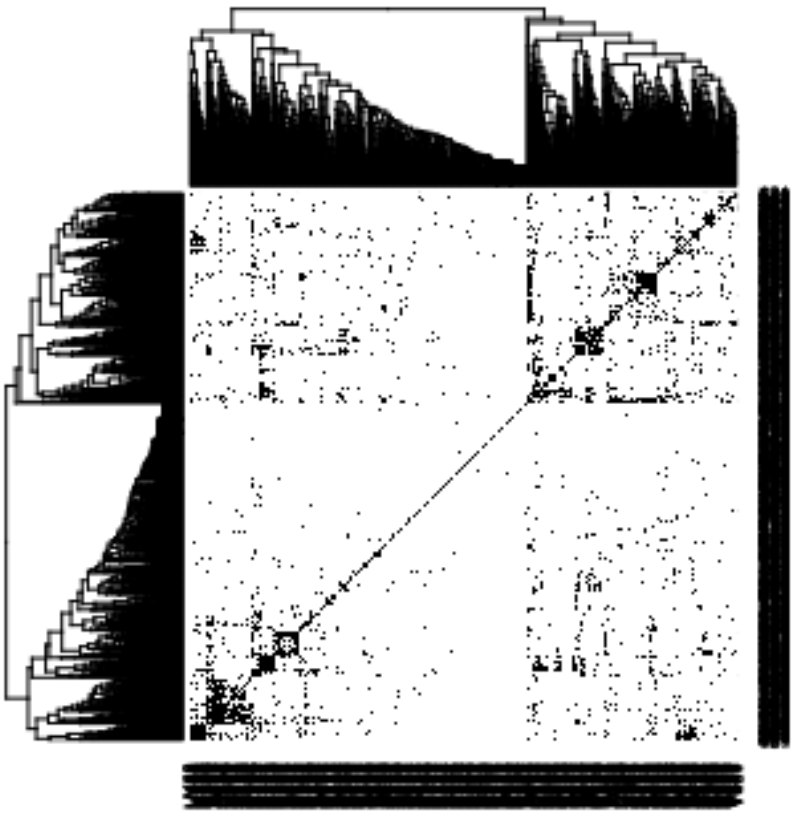

Figure 6.2: A hierachical clustering adjacency matrix visualization

distance between dyads respectively. We can see that the model performs well at capturing geodesic distance distribution, while just so so on simulating networks with the correct degree distribution. It performs quite poor at capturing the edgewise shared-partner distribution.

## 6.3 Friend Recommendation Algorithm Application

Randomly divide the social network dataset into 60% of the training set and 40% of the test set, that is, for each user, randomly selected 60% of all his or her friends. The relationship of this 60% friends is used as training set, the rest as test set.

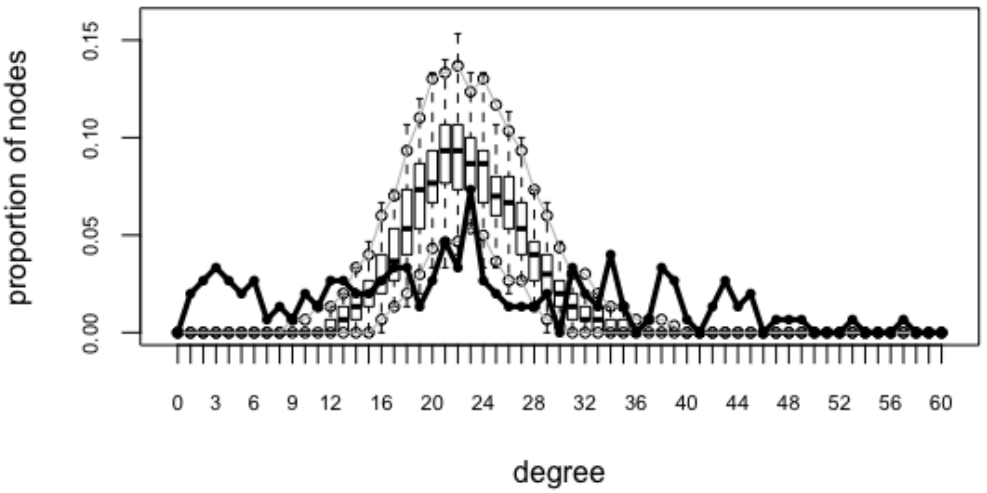


Figure 6.3: Degree goodness-of-fit plot for the Exponential Random Graph Model using edges triangles and geometrically weighted degree terms

**Measurements** Currently the typical indicators to evaluate the recommendation algorithm are: precision, recall and F1rate. Precision and recall rate are two important concepts in the field of information retrieval. They also reflect the performance of searching results. The precision refers to the amount of information related to the total amount of information retrieved. It is a measure of signal-to-noise ratio of the search system. The recall rate refers to the percentage of the amount of retrieved information related to the total amout of information involved. It is an indicator of the successness for the search system. F1 rate is based on the above two evaluations.

$$\text{Precision Definition :} \qquad precision = \frac{K_{correct}}{K_{total}}$$

$$\text{Recall Definition :} \qquad recall = \frac{K_{correct}}{K}$$

$$\text{F1 Definition:} \qquad F1 = \frac{2 \times \text{recall} \times \text{precision}}{\text{recall} + \text{precision}}$$

**Community Detection** We apply both Louvain algorithm and Mixed Membership Stochastic Model to the facebook dataset. As these two methods not only represent the disjoint

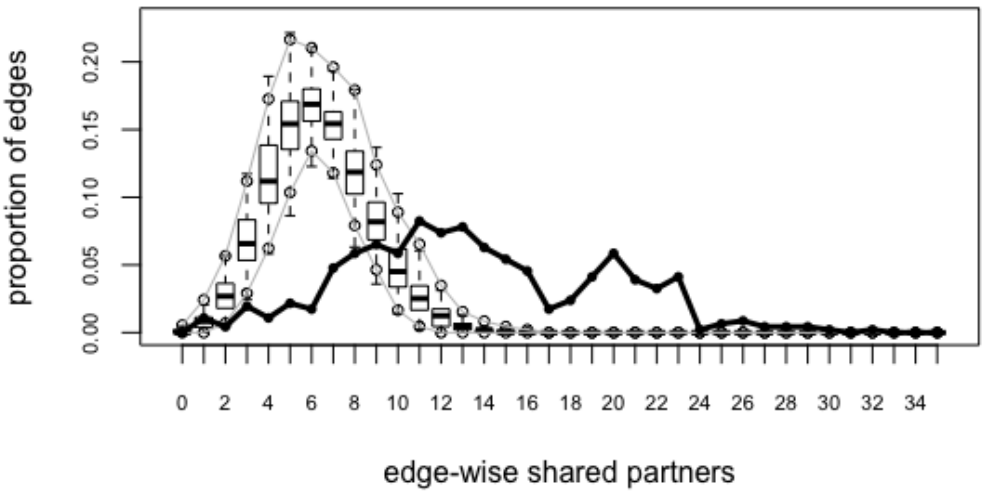


Figure 6.4: Edgewise-Shared-Partners goodness-of-fit plot for the Exponential Random Graph Model using edges triangles and geometrically weighted degree terms

detection and overlapping detection respectively, but also represent classical methods and statistical methods for community structure exploring. Apply Louvain Algorithm to the dataset and we obtain a disjoint community structure with 41 communities and the largest one has 503 nodes within it while the smallest ones have only 3. Fit MMSB to the dataset, set the number of communities to 30 and the output provides quite good performance with $AUC = 0.794$. Figure 6.6 is the AUC curve of the MMSB model. There are 206 nodes belong to multiple communties.

**Results** After applying the two-stage user similarity algorithm and Top K recommendation algorithm, we obtain the result from training data. We compare three different types of friend recommendation methods. Figure 6.7 and Figure 6.8 shows the performance of these methods with the change of number K (The number of friends recommended). One is to use MMSB as community detection algorithm, one is using Louvain Algorithm to do community detection and the ordinary line represents directly applying user similarity algorithm to the whole dataset without doing any community structure exploring. We can see that no matter

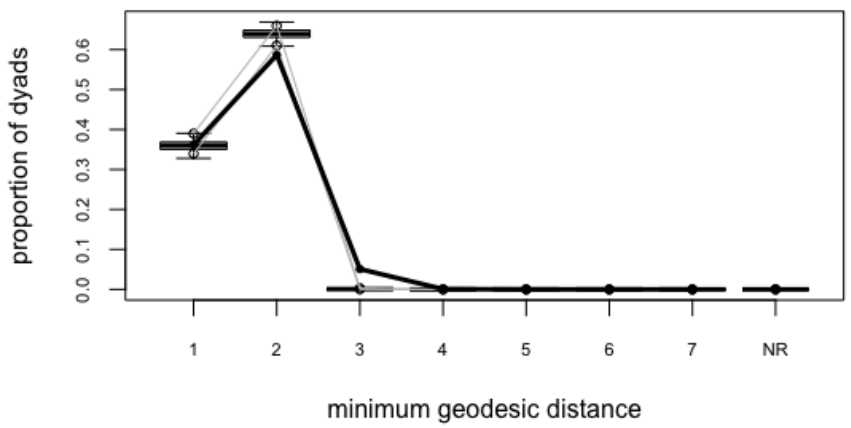


Figure 6.5: Geodesic distance goodness-of-fit plot for the Exponential Random Graph Model using edges triangles and geometrically weighted degree terms

with precision measure or F1 rate measure, the MMSB always provides the best result, Louvain method does not perform well. It's easy to understand that MMSB can discover the hidden structure of the social network from statistical perspective, which is quite reliable. The method using modularity has the limit in dividing communities, especially it will always divide communities into very large ones and quite small ones.

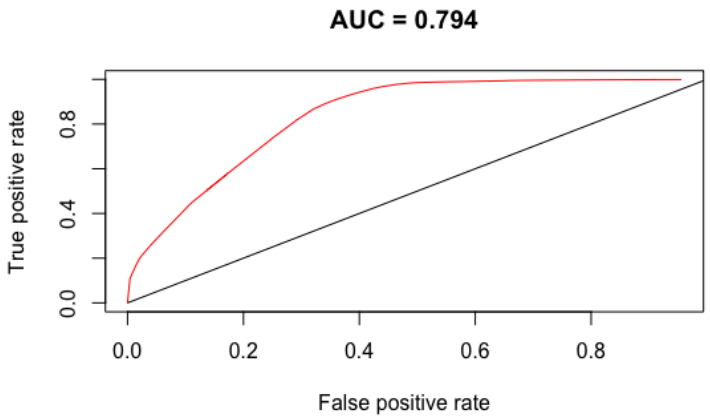


Figure 6.6: AUC curve of MMSB applied on facebook network data

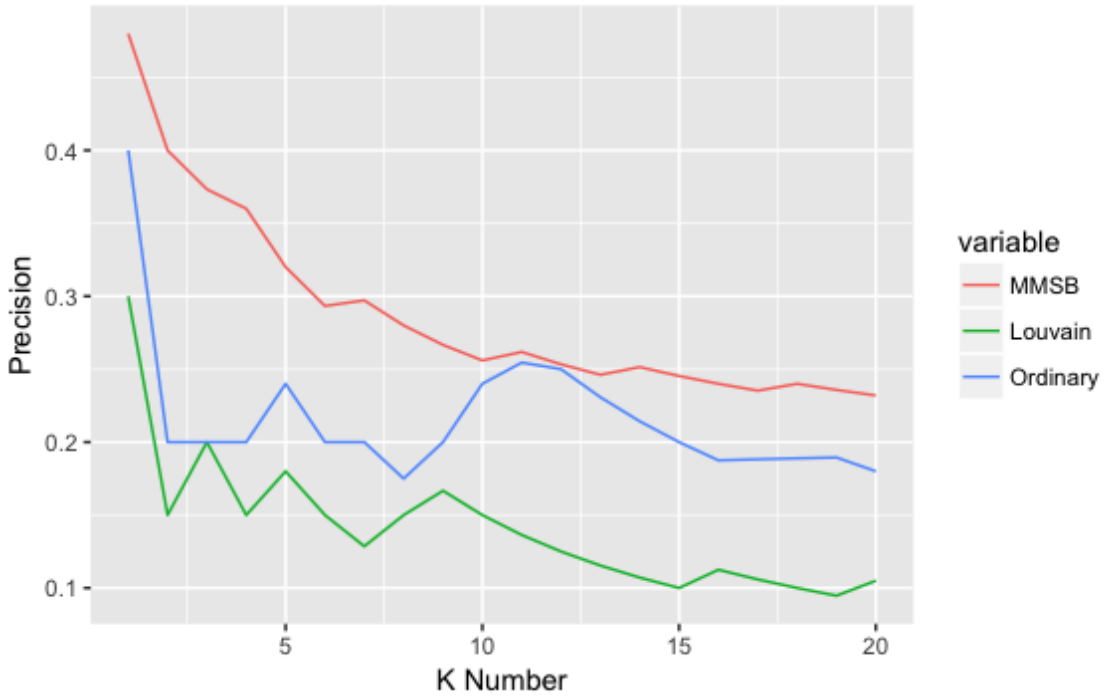


Figure 6.7: Precision of the Friend Recommendation Algorithm

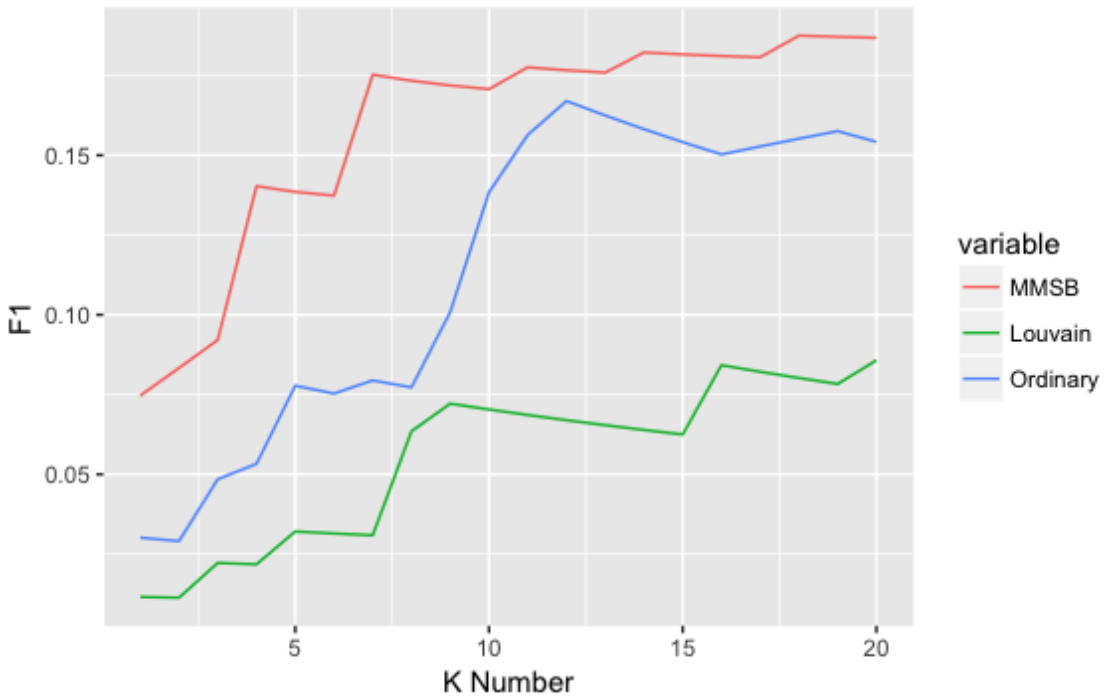


Figure 6.8: F1 curve of the Friend Recommendation Algorithm

# CHAPTER 7

# Discussion

In this thesis, we first introduce the background of current social network and the characteristics of it. Then we provide an overview with graph theory, summary statistics and basic concepts of social network. After that, we fit the statistical model: ERGM to the social network data. This reveals the main idea of statisticians working on network and can help to obtain a deeper understanding of social network analysis. Finally, combining with the community detection algorithm, we come up with an improved friend recommendation algorithm by which we can enhance the efficiency and performace especially on large social network.

Although this thesis discusses some methods about social networking friend recommendation, there are still many problems need to be solved in the future:

1. In spite of the good fitting performance MMSB provides, the time spend to apply this model is much longer than to apply the louvain algorithm. Actually, there are many improved MMSB model like aMMSB, WMMSB which make improvement in this aspect.

2. We need to consider the dynamic network as this is the type appeared in the real-world. When there are new users participate into the network, we need to reconstruct the whole net structure with the information they bring in. Then we need to redo community detection. This will waste a lot of time. Therefore, how to add new users into the divided communities with only slight adjustment is a problem.

3. In this thesis, we only consider the binary static network, which is the simplest one. In real-world network there exists large amount of other information, how to apply model

to the complicated network is a problem waiting to study.

REFERENCES


[ABF08] Edoardo M Airoldi, David M Blei, Stephen E Fienberg, and Eric P Xing. "Mixed membership stochastic blockmodels." *Journal of Machine Learning Research*, **9**(Sep):1981–2014, 2008.

[CF11] Alberto Caimo and Nial Friel. "Bayesian inference for exponential random graph models." *Social Networks*, **33**(1):41–55, 2011.

[DFF11] Pasquale De Meo, Emilio Ferrara, Giacomo Fiumara, and Alessandro Provetti. "Generalized louvain method for community detection in large networks." In *Intelligent Systems Design and Applications (ISDA), 2011 11th International Conference on*, pp. 88–93. IEEE, 2011.

[FR91] Thomas MJ Fruchterman and Edward M Reingold. "Graph drawing by force-directed placement." *Software: Practice and experience*, **21**(11):1129–1164, 1991.

[FS86] Ove Frank and David Strauss. "Markov graphs." *Journal of the american Statistical association*, **81**(395):832–842, 1986.

[GZF10] Anna Goldenberg, Alice X Zheng, Stephen E Fienberg, Edoardo M Airoldi, et al. "A survey of statistical network models." *Foundations and Trends® in Machine Learning*, **2**(2):129–233, 2010.

[HMM00] Ivan Herman, Guy Melançon, and M Scott Marshall. "Graph visualization and navigation in information visualization: A survey." *IEEE Transactions on visualization and computer graphics*, **6**(1):24–43, 2000.

[HRS03] Mark S Handcock, Garry Robins, Tom AB Snijders, Jim Moody, and Julian Besag. "Assessing degeneracy in statistical models of social networks." Technical report, Citeseer, 2003.

[KS14] Younghoon Kim and Kyuseok Shim. "TWILITE: A recommendation system for Twitter using a probabilistic model based on latent Dirichlet allocation." *Information Systems*, **42**:59–77, 2014.

[MYL08] Hao Ma, Haixuan Yang, Michael R Lyu, and Irwin King. "Sorec: social recommendation using probabilistic matrix factorization." In *Proceedings of the 17th ACM conference on Information and knowledge management*, pp. 931–940. ACM, 2008.

[New04] Mark EJ Newman. "Detecting community structure in networks." *The European Physical Journal B-Condensed Matter and Complex Systems*, **38**(2):321–330, 2004.

[SI90] David Strauss and Michael Ikeda. "Pseudolikelihood estimation for social networks." *Journal of the American Statistical Association*, **85**(409):204–212, 1990.

[VGH09] Marijtje AJ Van Duijn, Krista J Gile, and Mark S Handcock. "A framework for the comparison of maximum pseudo-likelihood and maximum likelihood estimation of exponential family random graph models." *Social Networks*, **31**(1):52–62, 2009.

[Wan13] Binghui Wang. "Latent Friend Recommendation Algorithm in Social Network." 2013.

[Zho11] Qiudan Li Zhongfeng Zhang. "Lateent Friend Recommendation in Social Network Services." *Journal of China Society for Scientific and Techinical Information*, **30**(12):1319–1325, 2011.