THE UNIVERSITY OF QUEENSLAND

AUSTRALIA

Social Community Profiling In Social Network

By

Haoze Xu

School of Information Technology and Electrical Engineering,

University of Queensland.

Submitted for the degree of Master of Information Technology

November 7, 2016

Haoze Xu
s4340046@student.uq.edu.au

November 7, 2016

Prof Michael Brünig
Head of School
School of Information Technology and Electrical Engineering
The University of Queensland
St Lucia QLD 4072

Dear Professor Brünig

In accordance with the requirements of the Degree of Master of Information Technology in the School of Information Technology and Electrical Engineering, I submit the following thesis entitled

"Social Communities Profiling in Social Networks"

The thesis was performed under the supervision of Dr. Xue Li. I declare that the work submitted in the thesis is my own, except as acknowledged in the text and footnotes, and that it has not previously been submitted for a degree at the University of Queensland or any other institution.

Yours sincerely

Haoze Xu

Acknowledgments

I would like to express my thanks to my supervisor, Dr. Xue Li, who gave me usefule advices. And I am also grateful to some classmates and friends who help me and support me a lot. And finally, I really appreciate that my parents gave me a lot of advices on my life during this project.

Abstract

Social communities profiling on social network is a challenging task, especially for the social networks with a large amount of users. In this paper, Louvain Algorithm is used for analyzing the links such as following on social networks, which has a decent computation time, while a combination of TF-IDF and agglomerative clustering is applied to the community detection and analysis by social network posts. And this thesis proposed a possible solution for the community detection on a weighted graph, which is a combination of TF-IDF and Louvain Algorithm. The result shows that the algorithms works well and the quality of clustering are satisfying. And this thesis also has discussions on the evaluation and the possible problems and improvements on community detection.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Now it is the Information Age and social network is one of the most important thing in the lives of people who have access to the Internet. There are varieties of social network services, and the number of users of social network services are exploding. People on social networks have frequent communication, and that is a way that a kind of community form. These social network users are often with direct links. There are a lot of this kind of communities existing on social networks, for example, the relationships on Facebook and the groups on Facebook. Furthermore, if there are many people talking about the similar topics, or share the same interest, they should be in a community, even though there are not direct links between each pair of users. Therefore, in general, there are two basic types of community – the relationship networks of people with direct links, and the people with similar features. Since there are communities with a variety of forms, in order to detect them, effective methods should be proposed. Therefore, this project is to apply and develop the feasible and easy-to-use methods to detect these communities. For the first type of community, Louvain Algorithm is

applied. Its running time is linear. And for the second type of community, simple NLP methods, TF-IDF and agglomerative clustering are used. Furthermore, based on the actual cases, this project proposed a method that is a combination of the two solutions of two types of communities, which may be a better solution of community detection.

Actually, nowadays, the data of community detection is really useful. For example, the data will be valuable for marketing and promotions. If promotions are for the group with exact features, for instance, sneakers for those who like playing basketball, and headphones for those who enjoy music. It would also be helpful for building up the recommendation systems in some specific situations. In a large relationship network, if one user is in a community but some people in the same community are not his or her friend, they can be recommended to this user and it is good for building up new friendships or even finding long-lost friends.

Chapter 2

Literature Review

2.1 Modularity-based Methods

There are diverse methods for community detection on networks. M. Girvan and M.E Newman [1] proposed an algorithm that is to recursively remove the edge with the highest edge betweenness to find clusters, which is called G-N Algorithm. It is the first famous modern community detection method, and one of the most important community detection method as well. The edge betweenness of an edge is the number of shortest paths between pairs of vertices that run along it. And the basic community detection steps are: calculate the betweenness; remove the edge with the highest betweenness; recalculate the betweenness; repeat the second step until no edges remain.

And the modularity-based methods are a significant type of methods for community detection. M. E. J. Newman [2] designed a greedy algorithm based on G-N

Algorithm, which is to select the best cut by the maximal value of modularity. The modularity is defined as:

$$Q = \sum_i (e_{ii} - a_i^2)$$

where $a_i = \sum_j e_{ij}$, and e_{ij} indicates the fraction of edges connecting vertices from group i to group j. This algorithm is proved with the running time of $O((m+n)n)$, or or $O(n^2)$ on a sparse graph on a network with m edges and n vertices, which is much better than G-N Algorithm, whose worst-case time $O(m^2n)$, or $O(n^3)$ on a sparse graph. It is also the first proposal of the concept of the modularity, which is a value that measures the density of links inside communities. A. Clauset, M.E.J Newman and C. Moore [3] applied the modularity and designed an algorithm which is called CNM Algorithm. Its running time is $O(md \log n)$ on the network m edges and n vertices, where d represents the depth of the "dendrogram" describing the network's community structure. They believe that in practice, it would be a good indicator of significant community structure if the value of modularity is greater than 0.3. The basic steps of the algorithm are: calculate the initial values of $\triangle Q_{ij}$ and a_i, select the largest $\triangle Q_{ij}$; join the corresponding communities; and repeat the second step until only one community remains. $\triangle Q_{ij}$ and a_i can be calculated by:

$$\Delta Q_{ij} = \begin{cases} \frac{1}{2m} - \frac{k_i k_j}{(2m)^2}, & if\ i,j\ are\ connected \\ 0, & otherwise \end{cases},$$

and

$$a_i = \frac{k_i}{2m}.$$

The modularity was also applied by V.D. Blondel, J.L. Guillaume, R. Lambiotte and E. Lefebvre [4], which is called Louvain Method of Louvain Algorithm. Its time complexity was proved linear on typical and sparse data. The basic process of an iteration is: each node as a community; then assign a node to its neighborhood and calculate the modularity change; and choose the assignment with maximum modularity change as a new community. The algorithm is proved to be appropriate for large scale networks that contains millions of nodes, which is one of the most important community detection method now. Q. Wang and E. Fleury [5] applied Louvain Algorithm and designed an algorithm named Fuzzy Detection Algorithm. The time complexity is proved to be $O(k \cdot n)$, where k is the iteration times of Louvain Method. They proposed the concept of robust cluster, which is and community cores

a group of nodes connected by edges having in-cluster probability no smaller than a threshold. And the concept of community cores is put forward as well, which is robust cluster having the maximum size. The basic steps are: Repeating Louvain Algorithm and compute a co-appearance matrix:

$$\mathrm{P} = \left[p_{ij}\right]_{n \times \mathrm{n}}$$

$$\|P^{k+1} - P^{k}\| = \sqrt{1/m \sum_{(i,j)\in \mathrm{E}} (p_{ij}^{k+1} - p_{ij}^{k})^2} < \varepsilon$$

where k is the time of iterations, and ε is a threshold which is often 0.9; detect the robust clusters; find out the community core; and re-compute the co-appearance matrix.

However, there are some limitations of modularity. A. Lancichinetti [6] made some experiments and believed that there are two major problems: the tendency to merge

small subgraphs when the resolution is low; the tendency to split large subgraphs when the resolution is high. And S. Fortunato [7] thought that a large value of modularity after maximization did not mean that the graph had community structure. For example, in random graphs, it is not supposed to have community structure, since the linking probability between vertices is constant, or depends on a function of the vertex degrees.

2.2 Methods for Detecting Overlapping Communities

In the real world, a member can not only belong to one community. And the community with this kind of nodes is called overlapping community. For this kind of networks, there are some methods to find the overlapping communities. Clique Percolation Method is a traditional one, which is proposed by I. Derényi, G. Palla and T. Vicsek [8]. In this method, k-cliques was analyzed, which is a giant component made of complete subgraphs of k vertices. It is based on the concept that the edges inside a community are likely to form cliques if they are highly dense [9]. And they proposed an important threshold $p_c(k)$ for the forming of k-clique community:

$$\mathrm{p_C(k)} = \frac{1}{(k-1)N^{\frac{1}{k-1}}}$$

where n is the number of vertices of the graph. And there is a simple software package called CFinder that can be applied freely. A. Lancichinetti, S. Fortunato and J. Kertész [10] proposed a method with an idea of expanding a community from a random seed

node to form a natural community, which is known as LFM. And a fitness function is used to identify it:

$$f(G) = \frac{k_{in}^{G}}{(k_{in}^{G} + k_{out}^{G})^{\alpha}}$$

where k_{in}^{G} and k_{out}^{G} are the total internal and external degree of the community c, and α is the resolution parameter controlling the size of the communities. And the basic process of the algorithm is: first pick a random node A; detect the natural community of A; pick a random node not belonging to any community; detect the natural community of B, then exploring all nodes regardless of their possible membership to other groups; repeat from the third step. Its time complexity is $O(n^2)$. [9] Furthermore, there is a method called OSLOM (Order Statistics Local Optimization Method), which is designed by A. Lancichinetti, F. Radicchi, JJ. Ramasco and S. Fortunato [11]. OSLOM includes three basic phases: looking for significant clusters, until convergence; analyzing the resulting set of clusters and trying to detect the internal structure or possible unions of them. It is the first method that detect communities based on the statistical significance, which is expressed by a local optimization of a fitness function. The worse-case running time of OSLOM in general is $O(n^2)$, and its actual running time depends on the community structure of the network in the task [9].

2.3 Community Detection for Social Network Posts

For community detection based on social network posts, which can be regarded as a text clustering problem, there are different solutions. One of the most famous and general one is TF-IDF. And in recent years there are several new methods and models. LSA (Latent Semantic Analysis) is one of the famous methods that comparing texts by a vector-based representation that generates from a corpus, which is proposed by P. Wiemer-Hastings, K. Wiemer-Hastings and A. Graesser [12]. An important technique used in LSA is SVD (Singular Value Decomposition), which is a dimensionality reduction method. And there is an improvement of LSA called pLSA (Probabilistic Latent Semantic Analysis), which is proposed by T. Hofmann [13]. It is key point is latent class analysis, which is based on a statistical model and also a decomposition technique. Assume that the document to be analysis is D and a fixed vocabulary of words is W, then the joint probability distribution of D×W is:

$$\mathrm{p}(d_i, w_j) = \sum_{k=1}^{K} P(z_k)P(d_i|z_k)P(w_j|z_k) = P(d_i)\sum_{k=1}^{K} P(w_j|z_k)P(z_k|d_i)$$

And the parameters can be estimated by EM [14], which is a famous training method. It is to maximize:

$$\mathrm{L} = \sum_{i=1}^{N}\sum_{j=1}^{M} n(d_i, w_j) log P(d_i, w_j)$$

$$= \sum_{i=1}^{N} n(d_i, w_j)[logP(d_i) + \sum_{j=1}^{M} \frac{n(d_i,w_j)}{n(d_i)} log \sum_{k=1}^{K} P(w_j|z_k)P(z_k|d_i)].$$

And LDA (Latent Dirichlet allocation) is proposed by D.M. Blei, A.Y. Ng and M.I. Jordan [15], which is widely used recently. It is based on Dirichlet distribution and in general, Gibbs Sampling and an alternating variational EM procedure are used to train the model. LDA can be regarded as a generalization of pLSA.

There are some other community detection and text clustering methods proposed in recent years. D.M. Blei and J.D. Lafferty [16] developed a correlated topic model (CTM) that can process the articles of the journal *Science* published from 1990–1999 which contains 57M words. It replaces the Dirichlet distribution of per-document topic proportions with a logistic normal. H. Zhang, B. Qiu, C. L. Giles, H. C. Foley, and J. Yen [17] designed an approach based on hierarchical Bayesian network, which is called SSN-LDA (Simple Social Network-LDA), to find probabilistic communities from social networks. And W. Zhou, H. Jin, and Y. Liu [18] develop a generative model called COCOMP (COllaborator COMmunity Profiling) to discover communities by social network posts, which include context in both topics and collaboration groups. They believed that the community can be a group of people that share the same interest. And COCOMP modeled a community as a mixture of topics that is contributed by a corresponding group of users. This model can not only be applied to the social network post, but also the documents like Emails.

2.4 Evaluations

There are some measures for evaluating the result of clustering. Y. Liu, Z. Li, H. Xiong, X. Gao and J. Wu [19] presented a detailed study of a comparison of 11 widely used internal clustering measures, such as R-square, I index, Silhouette index and *S_Dbw* validity index. And they concluded that only *S_Dbw* performed well in all of the experiments. V. Labatut [20] modified some widespread measures for community detection, including F-Measure, Adjusted Rand index, Normalized Mutual Information, to solve the problem that the traditional approaches are not quite appropriate for community detection, which is what they found out in their research.

Chapter 3

Theory and Methodology

3.1 Overview

In this thesis, the main object of research is set to Twitter users. For different objects, the data colleting methods and data formats will be diverse, but the algorithms (modifications may be needed) can be applied to them. The project is implemented by Python 3.5 and Gephi is used for the analysis of the modularity.

3.2 Data Collecting and Data Format

In this project, the data of Twitter users and Tweets are needed. To get data from Twitter, accessing Twitter API is a simplest way. Twitter provides REST APIs, which identifies Twitter applications and users using OAuth. However, it has a rate limit of 15 requests per 15 minutes, which means if a large amount of data is to be collected, it

will spend a lot of time. From Twitter's REST APIs, when making requests, JSON objects will return, which can be used for further use.

For the information of users, there is a "Users" object on Twitter including some significant information like the unique id, the unique name that is called "screen name", nickname and friend counts for a single user. And similarly, for Tweets, the "Tweets" object is also accessible, which contains the attributes such as Tweet id, creating time and text. Furthermore, collecting the data of friends of users is also needed, which is to create links between users for network analysis. Friends of a user on Twitter is the users that a user follows. And by accessing Twitter's API, friends can be fetched and they are shown as a set of JSON objects, which is the "Users" object on Twitter. The Python package "Tweepy" is used to accessing API provided by Twitter.

3.3 Data Storage and Management

Multiple ways are used to store and manage data in this project, because there are different kinds of data needed.

3.3.1 Local Database

In this project, only users' screen name is the critical part and it can be easily extracted from the JSON object returning from Twitter. To store the key data, in this project, MongoDB is used. The schemas of database are like below:

```
Users = {
    screen_name:
    id_str:
}
```

Figure 3.1 The schema of Users

```
Links = {
    user_screen_name:
    friend_screen_name:
}
```

Figure 3.2 The schema of Links

The collection “Users” contains a set of documents that are the key data of users, while the collection “Links” represents the friendships between users. The reason for the use of MongoDB is that it can be easily accessed from a Python program and is easy to reuse. In Python, the packages “pymongo” and “mongoengine” can be used to access local MongoDB. Reading from local files needs I/O cost and format transformation, which is not convenient enough and format errors may occur.

3.3.2 Text Files

And text of a Tweet may look like below:

\u20e32\u20e3 About to start off the pre-show for #HCT @PlayHearthstone https://t.co/a14PL31fDK

So there are some further pre-processing steps: remove emoticons, remove “RT”, remove links and remove “@” and “#”.

Emoticons are shown as Unicode characters in the return data from Twitter. Obviously, Unicode characters will disturb the result, but some Unicode characters are meaningful, for instance, some people may replace a digit with an emoji. This is quite hard to distinguish and it is hard to infer the results for different methods to deal with them. In this project, the emoticons are removed, for the reason that in most of cases they are not meaningful and they do not influence the meaning of Tweets. Besides, the characters that cannot be represented by ASCII codes will also be shown as Unicode characters. But in this project, the range is set to be English Tweets, so this factor can be ignored.

And “RT” is the word that will show if a user retweet a Tweet. In the analysis, this word will appear so many times and obviously it will disturb the result. Therefore, it should be removed.

And for links, they can be a part of words in the process of TF-IDF, and because they cannot be duplicated, there will be an extra dimension, which would affect the running time. And in general they are not meaningful. Therefore, they should be removed.

“@” on Twitter represents that a user mentioned another user, while “#” is a hashtag which can be a popular topic that users can access from searching. On Twitter,

the mentions and hashtags are often meaningful from the aspect of semantic, so in this project, only the symbols “@” and “#” are removed from Tweets.

And after these pre-processing steps, the Tweets will be stored into text files, where each line represents a single Tweet. And each user has a single text file.

3.3.3 CSV Files

In this project, there will be much data generating while executing the algorithms. Some kinds of data require well structured, for example, the data of graphs requires the source and target and possible weight of each edge. Local database is a decent solution for it, but for convenience, CSV (Comma-Separated Values) files are used, which is a type of data representation. The reason is that CSV files can be easily applied to other analysis tools, such as Gephi, a complex network analysis tool, weka and R.

3.4 Algorithms

3.4.1 Community Detection for Social Network Links

Community can be regarded as a group of people with close links.

In order to find community, building a network is the first required step. In this step, the data of links in the database will be used. During the process, duplicate links will be removed, because parallel edges are not allowed in the algorithm.

For community detection on networks, Louvain Algorithm [4] is used in this research, which is based on modularity.

And the basic steps of Louvain Algorithm are: first, regard each point as a community; then try to assign a community to its neighbour communities. For each assignment, the change of modularity will be calculated:

$$\Delta Q = \left[\frac{\sum_{in}+k_{i,in}}{2m} - \left(\frac{\sum_{tot}+k_i}{2m}\right)^2\right] - \left[\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m}\right)^2 - \left(\frac{k_i}{2m}\right)^2\right]$$

where $\sum_{in}$ is the sum of the weights of the links inside a community C and $\sum_{tot}$ is the sum of the weights of the links incident to nodes in C, which include the nodes inside C and linked to C. And k_i stands for the sum of the weights of the links incident to node i, while $k_{i,in}$ indicates the sum of the weights of the links from i to nodes in C. And m is the sum of the weights of all the edges in the network. The community will be assigned to the one with the maximum change of modularity. Repeat this step until the modularity of the whole graph no longer changes. And in an unweighted graph, the weights are all regarded as 1 in Louvain Algorithm.

The reason for applying Louvain Algorithm is that it is one of the most efficient community detection algorithm on networks:

Algorithm	***Running time***
Girven-Newman Algorigthm	$O(m^2 \cdot n)$
Newman Fast Algorithm	$O((m+n) \cdot n)$, or $O(n^2)$ on a sparse graph
CNM Algorithm	$O(n \cdot \log^2 n)$
Louvain Algorithm	Linear

Table 3.1 Comparison of Running Time of Community Detection Methods

In this project, Gephi is used to analyze the network and find communities based on modularity, and the data to construct the graphs in Gephi is computed in Python.

3.4.2 Community Detection for Social Network Posts

Community can be defined as a group of people that talks about similar topics, which is that the similarity of their posts is high.

In this part, Tweets collected from Twitter will be used. Before clustering, the pre-processing of Tweets is needed. The first step is tokenizing, which is to remove the punctuations and divide the sentences into words. Then removing stopwords is needed. This removes the meaningless words like “a”, “the” and “an”, which appear many times but cannot reveal the topics. The next step is to convert the words to lowercases, which is because the NLP is case sensitive. And finally, stemming, which is to get a higher accuracy. For example, in NLP, “cats” and “cat” are different, although in English they

can be regarded as the same word. Stemming can convert both of them to “cat”, which is the same in NLP. It reduces the chance of error. These steps can be implemented by “NLTK”, which is a Python package.

After pre-processing, algorithms can be applied. First, TF-IDF (Term Frequency – Inverse Document Frequency) [21] is to be applied, which is based on Vector Space Model. TF (Term Frequency) is the frequency that each word shows in the document, which is also called document frequency, and IDF can be calculated by:

$$\mathrm{IDF}(word) = \log \frac{total\ documents}{document\ frequency}$$

And the weight of the word can be calculated by:

$$w(word) = TF(word) \cdot IDF(\mathrm{word})$$

Then the weight can be normalized by:

$$w'(word_i) = \frac{w(word_i)}{\sqrt{w^2(word_1) + \cdots + w^2(word_n)}}$$

Then the normalized weights can be used for clustering.

TF-IDF is known as one of the most famous methods that reflect how important a word is to a document. The word with higher weight indicates that it appears many

times in the document it belongs to but not frequently appears in other documents in the corpus. That is why the word can represent the topic of the document well.

After annotated by TF-IDF, the documents will be clustered by agglomerative clustering, which is a type of hierarchical clustering. In this project, the number of expected clusters is not known before the execution of the algorithms. Therefore, some methods such as K-Means is not suitable. The basic step of agglomerative clustering is to merge nodes with the minimum distance. In this research, to calculate this distance, Ward's method [22] is used, which minimizes the sum of squared differences within all clusters. Agglomerative clustering does not provide a single partition, but instead the hierarchy of the clusters that merge with each other is provided, which can be represented by a dendrogram.

Those methods are implemented in Python by the package "sklearn" and some relevant packages like "numpy".

3.4.3 Community Detection by the Combination of Social Network Links and Posts

There is a problem of community detection for social network links. There may be some links between users that are not quite similar. Therefore, errors may occur because the importance of all of the friendships is considered to be equal. To overcome this problem, an effective way to evaluate the importance of each link needs to be found. There could be some factors that can be applied, for example, mutual friends and topics

similarity. In this project, text similarity is used for calculating the weights. The reason is that mutual friends is also based on friendships and the same problem may occur.

In details, first, construct a relationship network. Then using the text pre-processing methods and TF-IDF to generate the vectors for each node in the network. For the node with direct links, calculate the similarities between them as the weights. The weights are represented by the cosine similarities of TF-IDF vectors. And finally, applying Louvain Algorithm to this weighted graph.

It could be inferred that if the two methods above work well, this method will also work well.

3.5 Evaluations

Generally, it is hard to evaluate the results of clustering and community detection, and there are no efficient ways. Therefore, some measures are used depending on the situations and cases. And to test the effect of algorithm, an efficient way is to use the test data with clusters labels.

3.5.1 Evaluations with Ground Truths

And for the measures, in this project, there are 2 situations, which are the situation with and without the ground truth.

For the cases with ground truth, Confusion Matrix [23] and Normalized Mutual Information (NMI) [24] can be applied. Table 3.2 is the confusion matrix used in this project. For clustering, the error occurs only if 2 nodes belonging to the same cluster in the ground truth are assigned to different clusters, or 2 nodes belonging to different clusters in the ground truth are assigned to the same cluster. $C(v_i) = C(v_j)$ means 2 nodes are in the same cluster, while $C(v_i) \neq C(v_j)$ indicates different clusters.

		Ground Truth	
		$C(v_i) = C(v_j)$	$C(v_i) \neq C(v_j)$
Clustering Result	$C(v_i) = C(v_j)$	a	b
	$C(v_i) \neq C(v_j)$	c	d

Table 3.2 Confusion Matrix

The accuracy can be calculated by:

$$accuracy = \frac{a+d}{a+b+c+d} = \frac{a+d}{n(n-1)/2}$$

A higher accuracy indicates a better clustering result.

Another method for evaluation is NMI (Normalized Mutual Information). Mutual information (MI) is a measurement of the mutual dependence between the two variables. It measures the information that two variables share. If two variables are independent and one variable does not give any information about another variable, their mutual information is 0. NMI has a range of [0,1]. In clustering, a high NMI value indicates that the clustering result is close to the expected result.

NMI to evaluate the clustering result with *N* objects can be calculated by:

$$NMI(U,V) = \frac{MI(U,V)}{\sqrt{H(U)H(V)}}$$

Assuming that U and V are 2 label assignments. *H(U)* and *H(V)* are their entropy:

$$H(U) = \sum_{i=1}^{|U|} P(i)\log\left(P(i)\right)$$

$$H(V) = \sum_{j=1}^{|V|} P'(j)\log\left(P'(j)\right)$$

where $P(i) = |U_i|/N$ is the probability of an object picked randomly from U falls into class U_i, which is the same for *V*.

And $MI(U,V)$ represents the mutual information (MI) between U and V, which can be calculated by:

$$\mathrm{MI(U,V)} = \sum_{i=1}^{|U|}\sum_{j=1}^{|V|} P(i,j)\log\left(\frac{P(i,j)}{P(i)P'(j)}\right)$$

where $P(i,j)$ is the probability for a random pick belonging to both U_i and V_j.

3.5.2 Evaluations without Ground Truths

In most of practices, the ground truths are unknown. For links, there are not efficient way to evaluate, but for posts, there are some measures.

For the situation that the ground truth is unknown, the Silhouette Coefficient [25] can be applied:

$$s = \frac{a - b}{\max(a, b)}$$

where a is the mean distance between a sample and all other points in the same class and b represents the mean distance between a sample and all other points in the next nearest cluster. And the Silhouette Value of a clustering result is the average of the Silhouette Value of each point. Silhouette Value is between -1 and 1, where negative values indicate wrong assignment, and values near 0 indicate overlapping clusters. A higher Silhouette Value indicates a better clustering result.

These evaluation measures are available in the Python package “sklearn”.

Chapter 4

Results and Discussions

4.1 Dataset

In this research, the data was mainly collected from Twitter by using Twitter API.

For social network links, 4 lists of users on Twitter were collected first for two experiments. A list named “RV” was used for in-list clustering and the other 3 lists (“gugudan”, “I.O.I” and “DIA”) was used for cross-list clustering. The users in those lists was the supporter of the singer groups, whose names are the list names (except “RV”, which stands for “Red Velvet”). In these two experiments, the ground truths are set. For the experiment of 3 lists, each list is a cluster. And for the list “RV”, 6 clusters are set. There are 5 members in this band, so there are 5 clusters for the supporters of each member. And the remaining cluster is the supporters that focus on the whole band rather than a single member.

For social network posts, the dataset is 100 tweets for each user in a list named "HS", which contains a set of profession HearthStone players, a game produced by Blizzard Entertainment. And the Tweets of 84 users are collected.

For the combination of links and posts, the data of the list "HS" is also used.

4.2 The Experiments on Community Detection for Social Network Links

4.2.1 The Experiment of Lists "gugudan", "I.O.I" and "DIA"

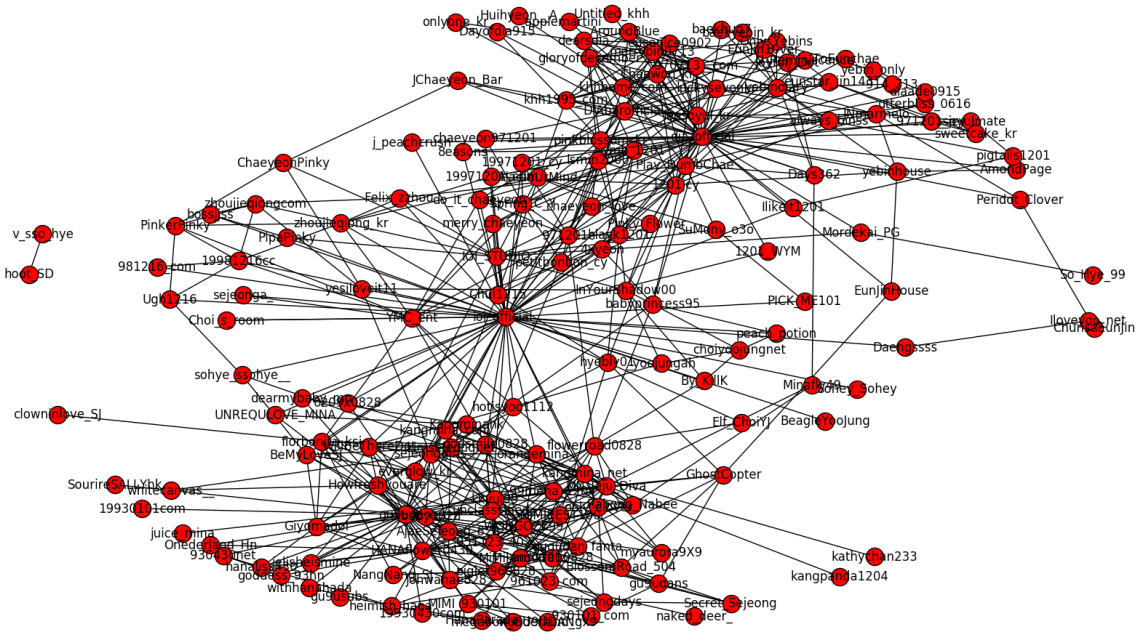


Figure 4.1 The original relationship network of lists "gugudan", "I.O.I" and "DIA"

Firstly, the network is generated and it is shown in *Figure 4.1*. And after applying the algorithm, the result is shown in *Figure 4.2*.

In this case, the number of clusters in the ground truth is 3, but the result is 5. As shown in the *Fig 7*, The cluster with light green indicates the expected cluster "I.O.I" and the one with purple is "DIA". And the cluster with dark green at the bottom has only 2 members, that is because they do not have links to the other nodes. Therefore, this should be a special case and this cluster can be simply ignored. The reason is that the data is collected by human and verification before executing the algorithm is not possible, and some extreme data may exist. And there is another issue that the expected cluster "gugudan" that is expected to contain the blue and orange clusters is divided into two clusters. However, it may still be explicable. The cluster with blue focus on the same member in the band, while the members in the cluster with orange does not have the exactly same focus.

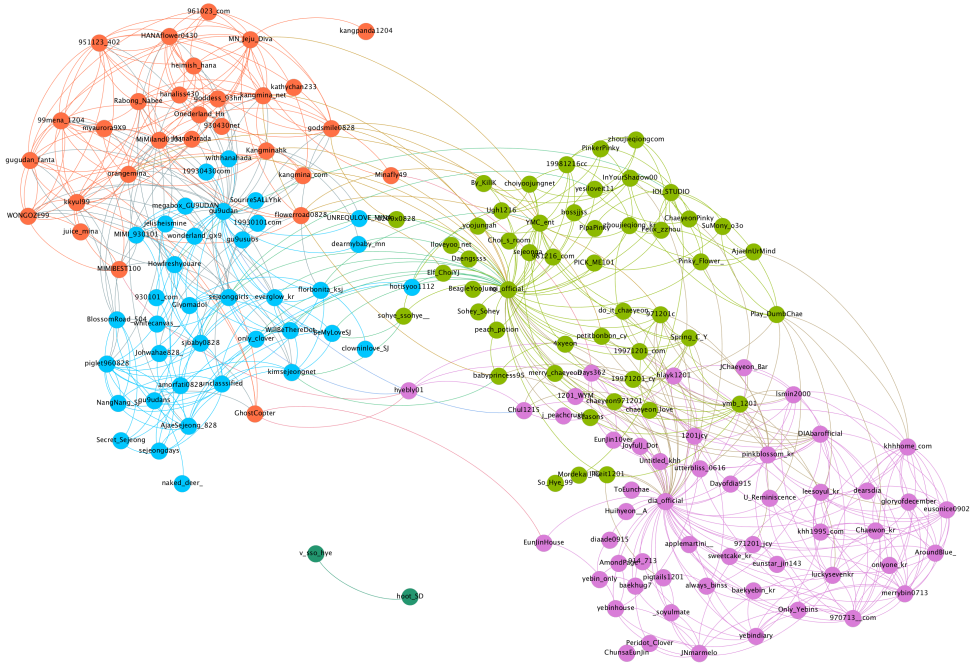


Figure 4.2 The clustering result of lists "gugudan", "I.O.I" and "DIA"

To find out the effect of the algorithm, methods for evaluation are applied. First, the modularity of the graph is 0.522, which is decent. And the accuracy computed by confusion matrix is 0.650887573964497. For NMI, the result is 0.456055367351, which means the result is not quite close to the ground truth.

4.2.2 The Experiment of the List "RV"

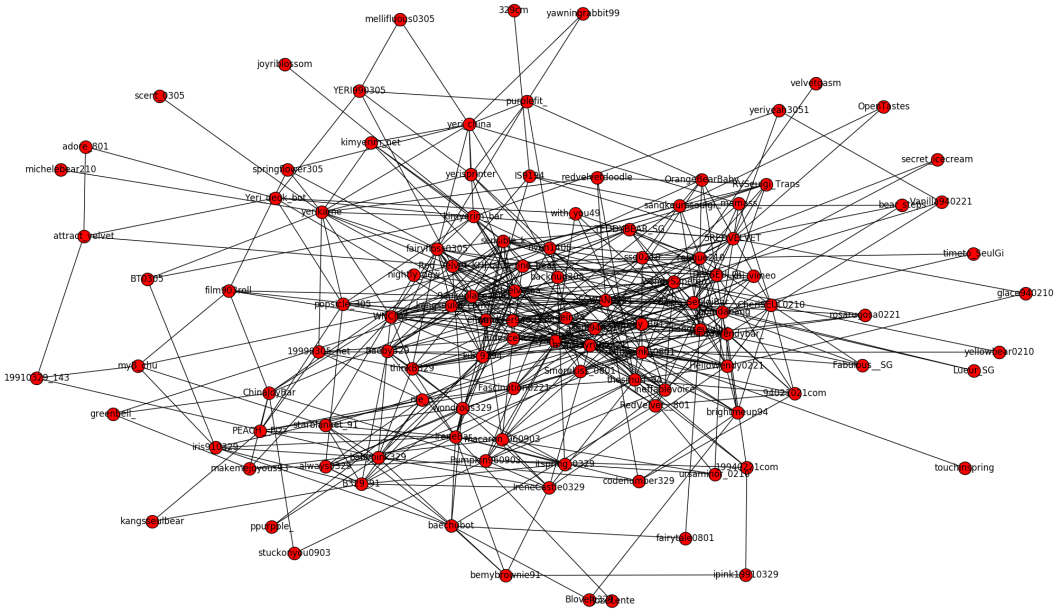


Figure 4.3 The original relationship network of the list "RV"

The network before applying the algorithm is generated and it is shown in *Figure 4.3*. After the algorithm, there are 6 cluster generated, which is exactly the number of clusters in the ground truth. By looking through *Figure 4.4*, the assignments are totally satisfied and the meaning of each cluster seems to match that of the ground truth, although there are several wrong assignments. That is because the expected clusters in the ground truth are labelled by human experience. The links are not the only factor for labelling and there are some other factors like the contents and the profiles. Another reason may be that the whole graph is really dense and the clusters are not quite clear.

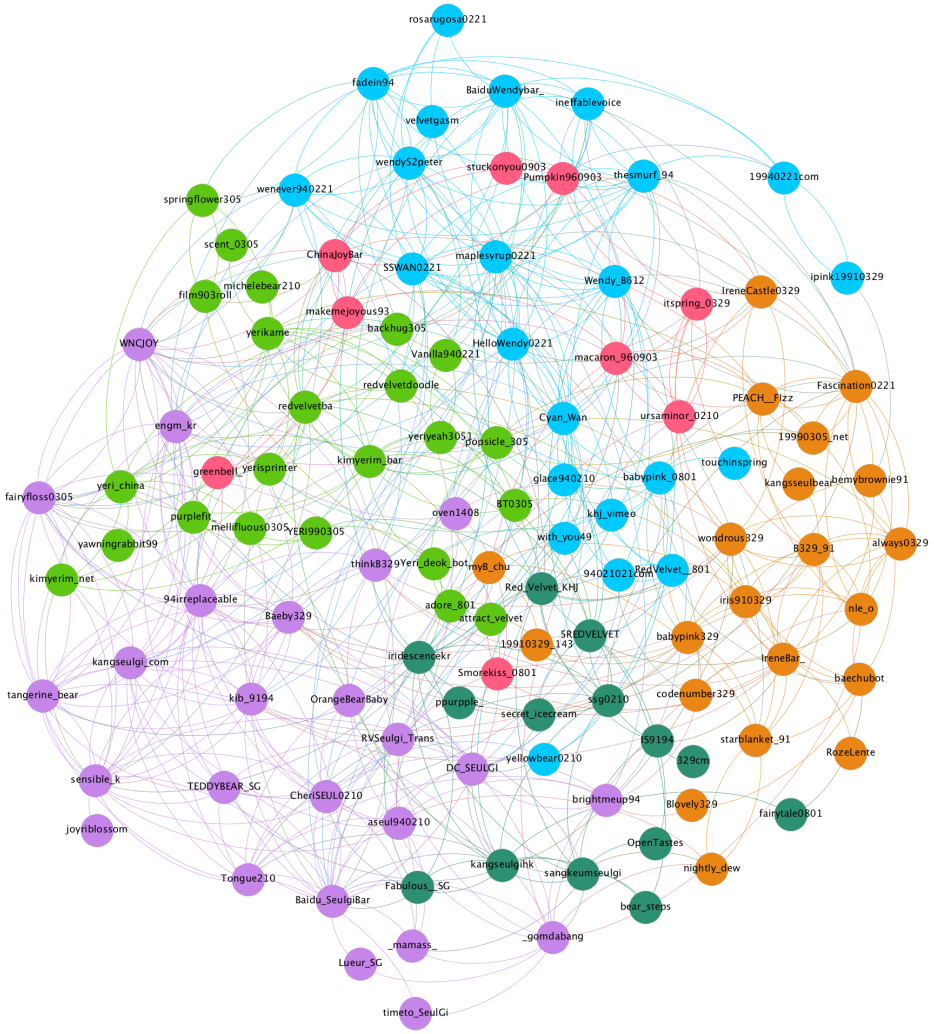


Figure 4.4 The clustering result of the list "RV"

To verify the guesses, the evaluation methods are used. The modularity of the graph is 0.398, which is a good value. And for the accuracy computed by confusion matrix, the result is 0.8096109839816934. It is better than that in the experiment of 3 lists – "gugudan", "I.O.I" and "DIA" because the number of clusters are exactly the same as that in the ground truth. And wrong number of clusters in the experiment of 3 lists - "gugudan", "I.O.I" and "DIA" leads to the lower value of the accuracy. And for NMI,

the result is 0.412397606953, which means the result is not quite close to the ground truth because of the wrong assignments.

4.3 The Experiment on Community Detection for Social Network Posts

In this experiment, agglomerative clustering is used and the dendrogram is shown as *Figure 4.5*.

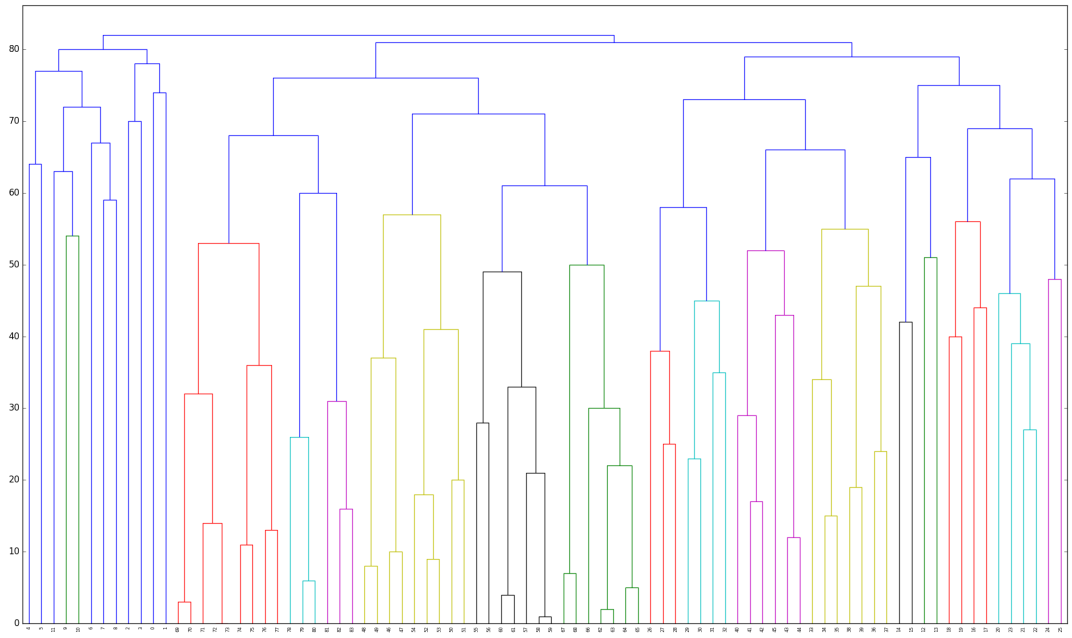


Figure 4.5 The dendrogram of the agglomerative clustering result of the list "HS"

And after setting a threshold, which is shown in the *Figure 4.6*, 3 clusters can be generated:

[69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 48, 49, 46, 47, 54, 52, 53, 50, 51, 55, 56, 60, 61, 57, 58, 59, 67, 68, 66, 62, 63, 64, 65]

[26, 27, 28, 29, 30, 31, 32, 40, 41, 42, 45, 43, 44, 33, 34, 35, 38, 39, 36, 37, 14, 15, 12, 13, 18, 19, 16, 17, 20, 23, 21, 22, 24, 25]

[4, 5, 11, 9, 10, 6, 7, 8, 2, 3, 0, 1]

where the numbers are the order of input sequence, which is also the collecting order of users. By this way, it is easy to explore the details of each user.

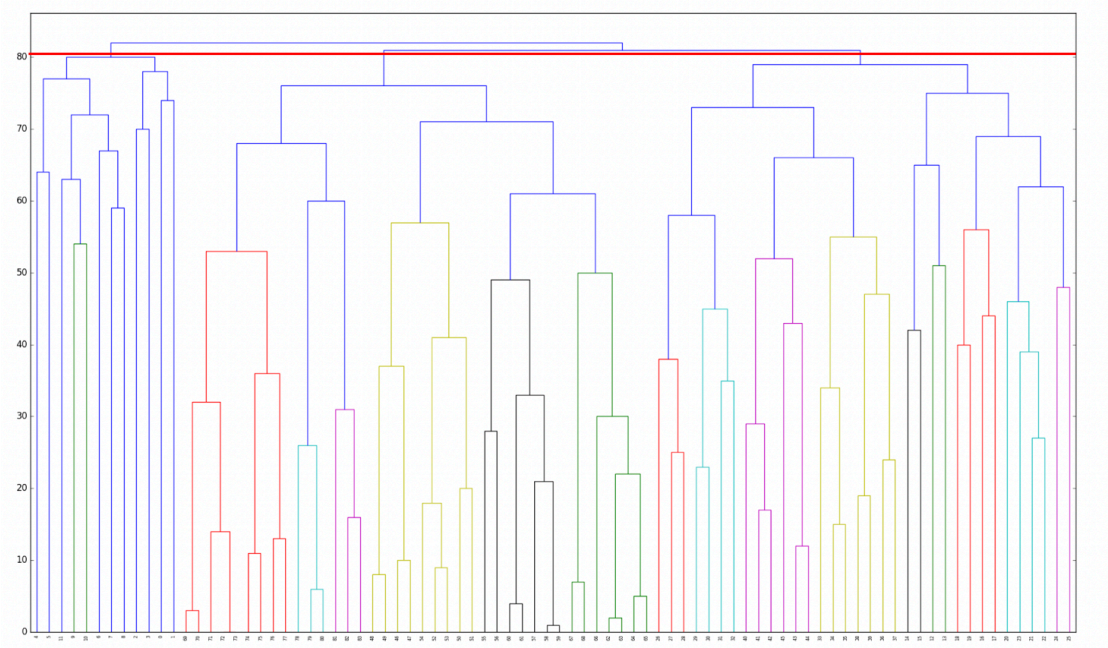


Figure 4.6 The threshold to determine the number of cluster

In this case, the ground truth is unknown, so Silhouette Coefficient is an option to evaluate the clustering result. And the Silhouette value is 0.55279119617, which means the clustering result is satisfied.

4.4 The Experiment on the Combination of Social Network Links and Posts

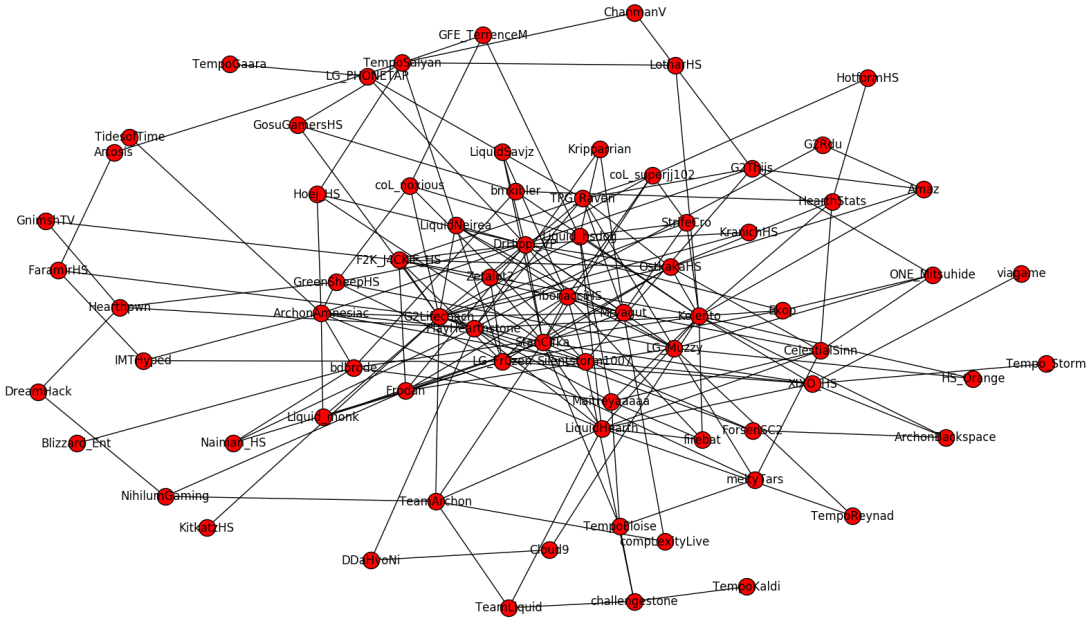


Figure 4.7 The original relationship network of the list "HS"

In this experiment, the Twitter list "HS" is to be analysis and after the first step, the relationship network can be drawn and it is shown in *Figure 4.9*. And based on the network, the weight computed by the cosine similarities of TF-IDF vectors of each user is applied to the edges of the graph. Finally, Louvain Algorithm is applied and the result is shown in the *Figure 4.8*.

The modularity report shows the modularity of this network after applying Louvain Algorithm, which is 0.335. It is an average result of community detection, which indicates a solid outcome. However, since there are not effective ways to evaluate the

result of community detection on network without ground truth, it is not available in this project.

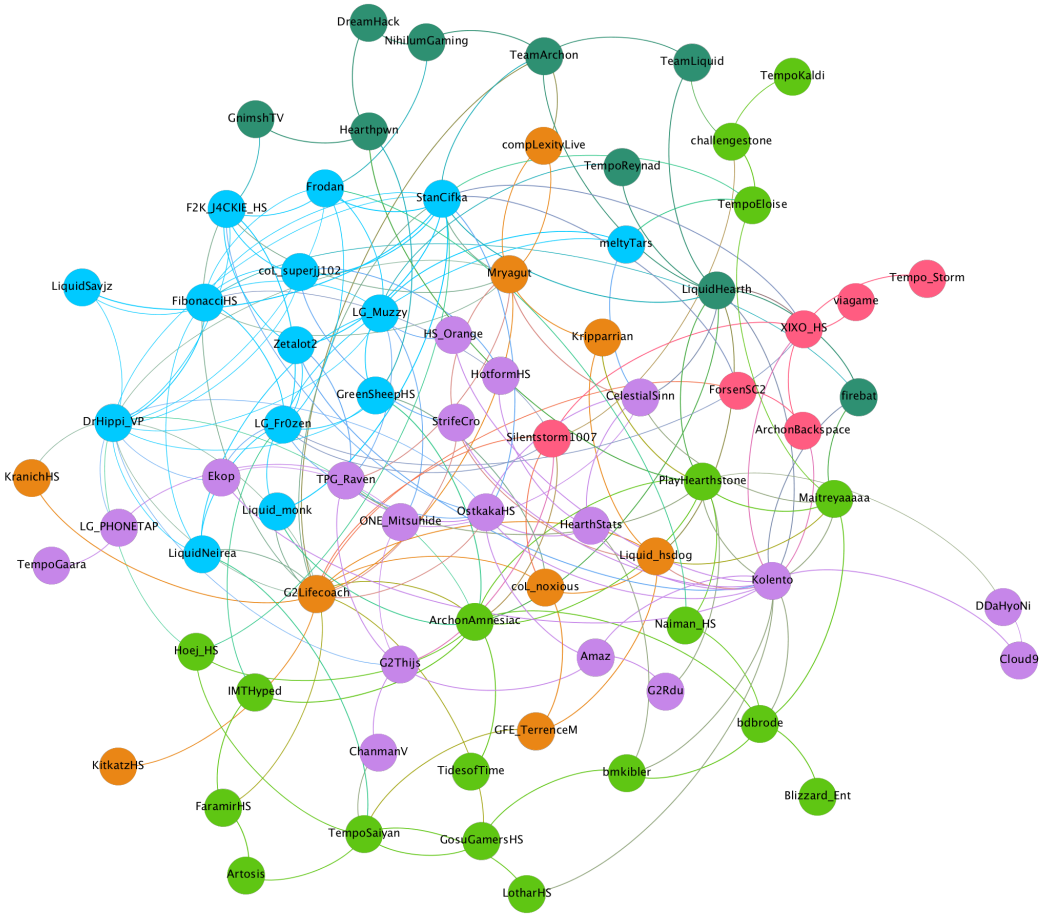


Figure 4.8 The clustering result of the list "HS"

4.5 Discussions

From the results, it is easy to find out that the accuracies of clustering are not quite satisfied. That is because the ground truth is set by human experience, which includes

the factors such as the profiles and the picture in the Tweets that are not considered in the community detection. And for the links, it is impossible to expect that each pair of similar users has directly links. There may be some users following the users not similar to them, which will lead to errors. And in a really dense network, if the links of users to the users inside the expected cluster and other clusters are nearly equal, significant deviations may occur. Although the accuracies are not quite satisfied, if looking into the details of each clusters in the result, the features of them can still be discovered. Therefore, the solutions are feasible for analyzing an unknown group of people and find communities between them. And whether the communities are meaningful depends on the cases. For example, if a group of people has high similarities between each pair, the community detection in this group may be not quite useful because those people should be regarded as a complete community. Sometimes finding sub-communities in a complete community is not meaningful and the results are hard to describe. And that is one of the problems that the last experiment is difficult to evaluate.

And there are several limitations. Firstly, the data is not standard. To test the algorithms, the standardized data is needed, which is the data with clear cluster labels. In this project, the data is collected by human and labeled by experience without verification by algorithms. That could cause the problem that the accuracies are not satisfied, although the results of clustering seem meaningful. However, this type of data is not easy to find and collect. Secondly, for evaluation, in general, there are not effective ways. The evaluation measures could just be references, and the actual effects will depend on the cases. In other words, if the methods are applied to a group of people

without any information, no matter whether the evaluation measures are available, it is still hard to describe the meaning of each cluster, which is also a problem of the last experiment. But if the information is given like the first two experiments, or even the only known information is the rough descriptions of the possible type of people in the target objects, the explanation and evaluation will be much more powerful and meaningful. For example, if the only known information in the second experiment is that there are 5 or 6 types of users, the meaning of each cluster in the result of clustering can still be inferred. And there are disadvantages for simple NLP steps and TF-IDF to deal with texts. One is that Tweets are often too short and some of them are not complete sentences. And there are many cyber words, which is hard to handle by the general NLP packages and software. At last, no multilingual support. If the text pre-processing steps support the language other than English, such as Korean, the experiments will be much more meaningful. That is because the most of users for first two experiments are Korean users and if Korean can be processed, community detection for the combination of social network links and posts can be applied to these users, and there will be a comparison for the effects of links only, posts only and the combination. This requires the support of Unicode decoding and text analysis for a specific language, which are technical problems.

Compared to some other researches, the part using Louvain Algorithm is quite decent, but the detection of overlapping communities is not available in this project. Q. Wang and E. Fleury's Fuzzy Detection Algorithm is also based on Louvain Algorithm and it is also very quick. And for the part of text processing, although TF-IDF is a

general method, it will be slow when the text is long. And the methods based on topic model will be much better, for example, CTM and SSN-LDA, which are be proved to be suitable for a large amount of data.

Although LDA is one of the most popular methods now to deal with texts and extract topics, there are some problems. For example, the texts on Twitter is not like the texts in articles. They are often too short and the form is various. If there are not enough texts that are similar for the model training, the performance will drop significantly, which is also called topic drift.

Chapter 5

Conclusions

This project proposed a feasible solution for community detection in social networks. The community is defined as two types: a group of people with direct links and a group of people with similar features. And multiple algorithms are used in order to solve the problem in different situations. In details, Louvain Algorithm is used for the community detection for social network links, TF-IDF and agglomerative clustering are applied to detect communities for social network posts, and there is a combination of those two methods. Generally, the results of clustering are acceptable, which means the methods worked well. In the first experiments of 3 lists of users, although the number of clusters in the result does not match that in the ground truth, the result can still be explained. Therefore, whether the result is explicable depends on the cases. Whether the results are easily described depends on the information known before applying methods. In some cases, the ground truth is known, but maybe it is not good enough and the results of clustering will have different descriptions. So in this situation, looking into the clusters and trying to find the meanings and explain the results are the

way to overcome this kind of problem. And the evaluation measures are rigid. They are powerful measures and the qualities of clustering can be easily inferred. But it is not reasonable to evaluate the clustering results only by them.

It is admitted that this thesis is not perfect and there could be some future improvements. First, multilingual support and pre-processing improvement for social network posts. It is helpful to deal with much more complicated texts. For example, there may be multiple languages in a Tweet and it will be hard to deal with. As mentioned in the discussions, more effective methods should be applied, such as LDA and even the NLP based on deep neural networks. And I believe that more test data and standard data to test the method will be helpful to evaluate the algorithms. It is hard to collect such amount and types of test data only by myself in a short time. And finally, because of the limitation of technique and time, some complex methods and models are not able to implemented. If they can be implemented, the quality of the result of community detection should be significantly improved. In short, an ideal system for community detection based on the definition of the community in this thesis – users with direct links and users with similar features contains a method based on Louvain Algorithm that can detect overlapping communities and a method based on a currently popular topic model to find the users with similar topics or interests. And both of them should work well for a large amount of data.

Appendix A

A Sample “Users” Object

{"follow_request_sent": false, "profile_image_url": "http://pbs.twimg.com/profile_images/747461123717267456/cIudFjaW_normal.jpg", "profile_text_color": "333333", "utc_offset": -25200, "is_translator": false, "lang": "ko", "profile_link_color": "2B7BB9", "name": "\uc0c1\ud07c\uc2ac\uae30", "default_profile_image": false, "has_extended_profile": false, "notifications": false, "is_translation_enabled": false, "description": "\uc2ac\uae30 \uc88b\uc544\ud568", "listed_count": 32, "favourites_count": 80, "profile_background_image_url": null, "protected": false, "entities": {"url": {"urls": [{"display_url": "sk-seulgi.tistory.com", "url": "https://t.co/KyY9b7A6dX", "expanded_url": "http://sk-seulgi.tistory.com", "indices": [0, 23]}]}, "description": {"urls": []}}, "id": 747428868412211200, "verified": false, "default_profile": true, "screen_name": "sangkeumseulgi", "contributors_enabled": false, "profile_background_tile": false, "profile_banner_url": "https://pbs.twimg.com/profile_banners/747428868412211200/1472908929", "url": "https://t.co/KyY9b7A6dX", "location": "", "id_str": "747428868412211200", "profile_background_image_url_https": null, "time_zone": "Pacific Time (US & Canada)", "statuses_count": 153, "profile_use_background_image": true, "followers_count": 923,

"profile_sidebar_border_color": "C0DEED", "profile_background_color": "F5F8FA", "profile_sidebar_fill_color": "DDEEF6", "profile_image_url_https": "https://pbs.twimg.com/profile_images/747461123717267456/cIudFjaW_normal.jpg", "status": {"text": "RT @JungMal08: The signature in the bottom is a footer in each page of the drawing book gift from @DC_SEULGI \ud83d\ude2d\ud83d\ude2d\n\nhttps://t.co/9Gmu2xiZY4 ht\u2026", "possibly_sensitive": false, "retweeted_status": {"text": "The signature in the bottom is a footer in each page of the drawing book gift from @DC_SEULGI \ud83d\ude2d\ud83d\ude2d\u2026 https://t.co/w7mSQbRoJk", "possibly_sensitive": false, "in_reply_to_screen_name": null, "lang": "en", "in_reply_to_status_id_str": null, "contributors": null, "in_reply_to_status_id": null, "in_reply_to_user_id_str": null, "coordinates": null, "source": "Twitter for Android", "place": null, "id": 784025591578263552, "created_at": "Thu Oct 06 13:40:42 +0000 2016", "favorited": false, "retweet_count": 198, "truncated": true, "id_str": "784025591578263552", "favorite_count": 143, "entities": {"urls": [{"display_url": "twitter.com/i/web/status/7\u2026", "url": "https://t.co/w7mSQbRoJk", "expanded_url": "https://twitter.com/i/web/status/784025591578263552", "indices": [98, 121]}], "hashtags": [], "user_mentions": [{"id": 720529297732075521, "id_str": "720529297732075521", "indices": [83, 93], "name": "DC \uc2ac\uae30 \uac24\ub7ec\ub9ac", "screen_name": "DC_SEULGI"}], "symbols": []}, "retweeted": false, "in_reply_to_user_id": null, "geo": null, "is_quote_status": false}, "in_reply_to_screen_name": null, "lang": "en", "in_reply_to_status_id_str": null, "contributors": null, "in_reply_to_status_id": null, "in_reply_to_user_id_str": null, "coordinates": null, "source": "Twitter for iPhone", "place": null,

"id": 784169381513207808, "retweeted": false, "favorited": false, "retweet_count": 198, "truncated": false, "id_str": "784169381513207808", "favorite_count": 0, "entities": {"urls": [{"display_url": "gall.dcinside.com/board/view/?id\u2026", "url": "https://t.co/9Gmu2xiZY4", "expanded_url": "http://gall.dcinside.com/board/view/?id=seulgi&no=116379", "indices": [113, 136]}], "hashtags": [], "user_mentions": [{"id": 720592019106709504, "id_str": "720592019106709504", "indices": [3, 13], "name": "#\ucda4\ubc14\ubcf4", "screen_name": "JungMal08"},{"id": 720529297732075521, "id_str": "720529297732075521", "indices": [98, 108], "name": "DC \uc2ac\uae30 \uac24\ub7ec\ub9ac", "screen_name": "DC_SEULGI"}], "symbols": []}, "in_reply_to_user_id": null, "geo": null, "created_at": "Thu Oct 06 23:12:04 +0000 2016", "is_quote_status": false}, "friends_count": 32, "geo_enabled": false, "created_at": "Mon Jun 27 13:58:23 +0000 2016", "following": true}

Appendix B

A Sample “Tweets” Object

{"url": "https://t.co/UyRX94ZErn", "favourites_count": 249, "default_profile": false, "profile_image_url": "http://pbs.twimg.com/profile_images/752146853756538880/QIp6sK9D_normal.jpg", "profile_background_image_url_https": "https://abs.twimg.com/images/themes/theme1/bg.png", "screen_name": "DrHippi_VP", "utc_offset": 10800, "notifications": false, "protected": false, "id": 701037689748004866, "is_translator": false, "profile_text_color": "000000", "time_zone": "Kyiv", "is_translation_enabled": false, "entities": {"description": {"urls": []}, "url": {"urls": [{"url": "https://t.co/UyRX94ZErn", "indices": [0, 23], "display_url": "twitch.tv/drhippi", "expanded_url": "http://twitch.tv/drhippi"}]}}, "profile_image_url_https": "https://pbs.twimg.com/profile_images/752146853756538880/QIp6sK9D_normal.jpg", "created_at": "Sat Feb 20 13:36:24 +0000 2016", "profile_background_color": "000000", "profile_use_background_image": false, "profile_banner_url": "https://pbs.twimg.com/profile_banners/701037689748004866/1473703822", "id_str": "701037689748004866", "profile_background_tile": false, "description": "Hearthstone player for

@TeamVirtuspro.", "profile_sidebar_border_color": "000000", "profile_link_color": "FF691F", "follow_request_sent": false, "followers_count": 2265, "profile_background_image_url": "http://abs.twimg.com/images/themes/theme1/bg.png", "geo_enabled": false, "lang": "ru", "verified": false, "statuses_count": 157, "has_extended_profile": true, "listed_count": 27, "following": true, "name": "DrHippi", "friends_count": 141, "profile_sidebar_fill_color": "000000", "default_profile_image": false, "status": {"favorite_count": 3, "id": 787705723564265472, "in_reply_to_user_id": 2853740205, "in_reply_to_user_id_str": "2853740205", "truncated": false, "text": "@FenoHS @LG_Muzzy don't do it. its not ok xd", "entities": {"hashtags": [], "user_mentions": [{"screen_name": "FenoHS", "name": "Chris Tsako", "id_str": "2853740205", "id": 2853740205, "indices": [0, 7]}, {"screen_name": "LG_Muzzy", "name": "Muzzy", "id_str": "3068577192", "id": 3068577192, "indices": [8, 17]}], "urls": [], "symbols": []}, "created_at": "Sun Oct 16 17:24:14 +0000 2016", "contributors": null, "id_str": "787705723564265472", "favorited": false, "in_reply_to_status_id": 787705214547755008, "geo": null, "retweet_count": 0, "source": "Twitter Web Client", "place": null, "in_reply_to_status_id_str": "787705214547755008", "lang": "en", "coordinates": null, "in_reply_to_screen_name": "FenoHS", "retweeted": false, "is_quote_status": false}, "contributors_enabled": false, "translator_type": "none", "location": "\u041b\u0443\u0446\u043a"}

Appendix C

Data for the Weighted Graph of the List “HS”

Source	Target	Type	Weight
DrHippi_VP	Mryagut	undirected	0.610426909
DrHippi_VP	StanCifka	undirected	0.606842063
DrHippi_VP	ArchonAmnesiac	undirected	0.608410059
DrHippi_VP	OstkakaHS	undirected	0.620179567
DrHippi_VP	G2Lifecoach	undirected	0.587230277
FibonacciHS	coL_superjj102	undirected	0.68710707
FibonacciHS	DrHippi_VP	undirected	0.867700925
FibonacciHS	LG_Muzzy	undirected	0.66404704
FibonacciHS	firebat	undirected	0.685044893
FibonacciHS	LiquidSavjz	undirected	0.663076807
FibonacciHS	Silentstorm1007	undirected	0.681632207
FibonacciHS	Frodan	undirected	0.646060508
FibonacciHS	LG_Fr0zen	undirected	0.863187092
FibonacciHS	StanCifka	undirected	0.65912202
LG_Fr0zen	meltyTars	undirected	0.77793842
F2K_J4CKIE_HS	DrHippi_VP	undirected	0.706894959
F2K_J4CKIE_HS	LG_Fr0zen	undirected	0.959021931
DDaHyoNi	Cloud9	undirected	0.647988363
challengestone	Silentstorm1007	undirected	0.789874805

challengestone	TempoEloise	undirected	0.966368264
challengestone	TeamLiquid	undirected	0.7043646
challengestone	TempoKaldi	undirected	0.757012876
Hoej_HS	DrHippi_VP	undirected	0.56936013
LG_Muzzy	TempoReynad	undirected	0.819656232
LG_Muzzy	ONE_Mitsuhide	undirected	0.905146583
LG_Muzzy	meltyTars	undirected	0.943208119
LG_Muzzy	TPG_Raven	undirected	0.81906369
LG_Muzzy	DrHippi_VP	undirected	0.583218634
G2Lifecoach	DrHippi_VP	undirected	0.587230277
G2Lifecoach	KranichHS	undirected	0.935237611
G2Lifecoach	Silentstorm1007	undirected	0.900305926
G2Lifecoach	Liquid_monk	undirected	0.843732762
G2Lifecoach	GosuGamersHS	undirected	0.889092966
G2Lifecoach	F2K_J4CKIE_HS	undirected	0.812373835
G2Lifecoach	FaramirHS	undirected	0.851138246
G2Lifecoach	KitkatzHS	undirected	0.877680713
G2Lifecoach	ArchonAmnesiac	undirected	0.926544964
G2Lifecoach	TeamArchon	undirected	0.929211717
G2Lifecoach	ForsenSC2	undirected	0.846370581
G2Lifecoach	Liquid_hsdog	undirected	0.920320107
G2Lifecoach	coL_noxious	undirected	0.858711338
G2Lifecoach	StanCifka	undirected	0.851109545
LG_PHONETAP	TempoGaara	undirected	0.856360128
LG_PHONETAP	TPG_Raven	undirected	0.848024016
LG_PHONETAP	DrHippi_VP	undirected	0.588549424
meltyTars	LG_Muzzy	undirected	0.943208119
meltyTars	TempoEloise	undirected	0.931544069
meltyTars	LG_Fr0zen	undirected	0.77793842
Naiman_HS	bdbrode	undirected	0.879794661
NihilumGaming	TeamArchon	undirected	0.98942257
NihilumGaming	Frodan	undirected	0.865106669
NihilumGaming	DreamHack	undirected	0.875661161

KranichHS	DrHippi_VP	undirected	0.606157348
TeamArchon	compLexityLive	undirected	0.870542906
TeamArchon	TeamLiquid	undirected	0.890934705
TempoSaiyan	LotharHS	undirected	0.938906352
TempoSaiyan	GFE_TerrenceM	undirected	0.908611622
TempoSaiyan	Artosis	undirected	0.888253544
TempoSaiyan	ChanmanV	undirected	0.87998784
TempoSaiyan	Hoej_HS	undirected	0.890999509
ArchonAmnesiac	IMTHyped	undirected	0.98746907
ArchonAmnesiac	bdbrode	undirected	0.931191117
ArchonAmnesiac	Hoej_HS	undirected	0.885542715
ArchonAmnesiac	TidesofTime	undirected	0.984230978
ArchonAmnesiac	PlayHearthstone	undirected	0.928317584
ArchonAmnesiac	Liquid_monk	undirected	0.916098611
Mryagut	TPG_Raven	undirected	0.909752763
Mryagut	compLexityLive	undirected	0.893587458
Mryagut	StanCifka	undirected	0.927917098
Mryagut	G2Lifecoach	undirected	0.925752096
Mryagut	FibonacciHS	undirected	0.685020795
Mryagut	F2K_J4CKIE_HS	undirected	0.832975696
CelestialSinn	ONE_Mitsuhide	undirected	0.778800904
CelestialSinn	meltyTars	undirected	0.950501606
CelestialSinn	Mryagut	undirected	0.995068283
Liquid_hsdog	Kripparrian	undirected	0.939520469
Liquid_hsdog	StrifeCro	undirected	0.934958051
Liquid_hsdog	GFE_TerrenceM	undirected	0.934915155
Liquid_hsdog	coL_noxious	undirected	0.948703082
coL_superjj102	F2K_J4CKIE_HS	undirected	0.83736571
coL_superjj102	HotformHS	undirected	0.934057581
coL_superjj102	DrHippi_VP	undirected	0.619748834
IMTHyped	LG_Fr0zen	undirected	0.818780741
ArchonBackspace	XIXO_HS	undirected	0.947900977
ArchonBackspace	Kolento	undirected	0.944853335

Silentstorm1007	LG_Fr0zen	undirected	0.812744238
Silentstorm1007	XIXO_HS	undirected	0.95429192
Silentstorm1007	ArchonAmnesiac	undirected	0.974821143
Silentstorm1007	G2Thijs	undirected	0.996484901
GreenSheepHS	DrHippi_VP	undirected	0.619603609
GreenSheepHS	LG_Fr0zen	undirected	0.817029219
GreenSheepHS	coL_noxious	undirected	0.967264482
GreenSheepHS	LG_Muzzy	undirected	0.886724858
GreenSheepHS	ArchonAmnesiac	undirected	0.973355936
G2Thijs	DrHippi_VP	undirected	0.619911976
G2Thijs	ONE_Mitsuhide	undirected	0.736620768
G2Thijs	TPG_Raven	undirected	0.954534645
GosuGamersHS	bmkibler	undirected	0.949264478
GosuGamersHS	TempoSaiyan	undirected	0.957003634
Amaz	OstkakaHS	undirected	0.999396669
Amaz	G2Rdu	undirected	0.993929533
Amaz	Kolento	undirected	0.970775192
Amaz	G2Thijs	undirected	0.988179731
OstkakaHS	DrHippi_VP	undirected	0.620179567
OstkakaHS	G2Lifecoach	undirected	0.879896655
OstkakaHS	F2K_J4CKIE_HS	undirected	0.827059901
OstkakaHS	TPG_Raven	undirected	0.964745981
OstkakaHS	CelestialSinn	undirected	0.966099726
OstkakaHS	LG_Fr0zen	undirected	0.80648778
LiquidHearth	ArchonAmnesiac	undirected	0.950214405
LiquidHearth	Liquid_hsdog	undirected	0.965201818
LiquidHearth	CelestialSinn	undirected	0.962163276
LiquidHearth	TempoReynad	undirected	0.970939448
LiquidHearth	firebat	undirected	0.983048923
LiquidHearth	Kolento	undirected	0.976325641
LiquidHearth	XIXO_HS	undirected	0.978808759
LiquidHearth	TeamLiquid	undirected	0.972631262
LiquidHearth	TeamArchon	undirected	0.935234431

LiquidHearth	StanCifka	undirected	0.984059782
HearthStats	bmkibler	undirected	0.962042463
HearthStats	OstkakaHS	undirected	0.995267542
HearthStats	HotformHS	undirected	0.975014691
HearthStats	CelestialSinn	undirected	0.960879044
GnimshTV	F2K_J4CKIE_HS	undirected	0.822354266
Zetalot2	TPG_Raven	undirected	0.977431687
Zetalot2	F2K_J4CKIE_HS	undirected	0.824631709
Zetalot2	LG_Fr0zen	undirected	0.800406048
Ekop	LG_Fr0zen	undirected	0.798591653
Ekop	DrHippi_VP	undirected	0.612036282
Ekop	ONE_Mitsuhide	undirected	0.694249412
Hearthpwn	GreenSheepHS	undirected	0.969101068
Hearthpwn	DreamHack	undirected	0.984558591
Hearthpwn	GnimshTV	undirected	0.994644992
LiquidSavjz	DrHippi_VP	undirected	0.607292692
coL_noxious	GFE_TerrenceM	undirected	0.990471432
coL_noxious	Silentstorm1007	undirected	0.965172792
bdbrode	bmkibler	undirected	0.974933579
bdbrode	Blizzard_Ent	undirected	0.990122865
PlayHearthstone	Kripparrian	undirected	0.996440644
PlayHearthstone	G2Rdu	undirected	0.989052172
PlayHearthstone	DDaHyoNi	undirected	0.635625314
PlayHearthstone	Naiman_HS	undirected	0.876763345
PlayHearthstone	TPG_Raven	undirected	0.983293277
PlayHearthstone	LiquidHearth	undirected	0.986031709
PlayHearthstone	Hearthpwn	undirected	0.997611138
FaramirHS	IMTHyped	undirected	0.947143815
FaramirHS	Artosis	undirected	0.978464765
StanCifka	Hoej_HS	undirected	0.788451195
StanCifka	TempoEloise	undirected	0.812152088
StanCifka	Zetalot2	undirected	0.990772451
StanCifka	coL_superjj102	undirected	0.944999474

StanCifka	OstkakaHS	undirected	0.97962138
StanCifka	TeamArchon	undirected	0.908337352
StanCifka	Liquid_monk	undirected	0.99363684
StanCifka	Frodan	undirected	0.983641293
StanCifka	LiquidSavjz	undirected	0.996173012
StanCifka	XIXO_HS	undirected	0.994353525
StanCifka	ForsenSC2	undirected	0.995929053
Kripparrian	Mryagut	undirected	0.924825508
ForsenSC2	ArchonBackspace	undirected	0.948798771
viagame	XIXO_HS	undirected	0.999110142
HS_Orange	LG_Muzzy	undirected	0.826390629
XIXO_HS	Tempo_Storm	undirected	0.953180824
XIXO_HS	LG_Fr0zen	undirected	0.783156751
Liquid_monk	LG_Fr0zen	undirected	0.781090376
LiquidNeirea	LG_Muzzy	undirected	0.826413591
LiquidNeirea	FibonacciHS	undirected	0.654803641
LiquidNeirea	G2Lifecoach	undirected	0.844456425
LiquidNeirea	TempoSaiyan	undirected	0.908667878
LiquidNeirea	DrHippi_VP	undirected	0.602099096
LiquidNeirea	LG_Fr0zen	undirected	0.78407141
StrifeCro	coL_superjj102	undirected	0.938744908
StrifeCro	Mryagut	undirected	0.919599703
StrifeCro	G2Lifecoach	undirected	0.844842964
Kolento	Zetalot2	undirected	0.982968783
Kolento	Liquid_hsdog	undirected	0.932966147
Kolento	OstkakaHS	undirected	0.971339717
Kolento	HS_Orange	undirected	0.996101615
Kolento	XIXO_HS	undirected	0.996431272
Kolento	firebat	undirected	0.957197131
Kolento	StrifeCro	undirected	0.99869007
Kolento	Cloud9	undirected	0.984760648
Kolento	bdbrode	undirected	0.988589897
Kolento	PlayHearthstone	undirected	0.988649141

Kolento	Amaz	undirected	0.970775192
Kolento	Ekop	undirected	0.983337383
Kolento	LotharHS	undirected	0.980973184
Kolento	HearthStats	undirected	0.977642177
Kolento	bmkibler	undirected	0.98652062
Maitreyaaaaa	TempoEloise	undirected	0.798085728
Maitreyaaaaa	bdbrode	undirected	0.984041147
Maitreyaaaaa	CelestialSinn	undirected	0.921960299
Maitreyaaaaa	firebat	undirected	0.952352575
Maitreyaaaaa	Liquid_hsdog	undirected	0.927318401
Maitreyaaaaa	PlayHearthstone	undirected	0.983989046
Maitreyaaaaa	Kolento	undirected	0.996096947
Frodan	LG_Muzzy	undirected	0.815584002
Frodan	Naiman_HS	undirected	0.848725596
Frodan	DrHippi_VP	undirected	0.5921503
Frodan	F2K_J4CKIE_HS	undirected	0.810876975
Frodan	LG_Fr0zen	undirected	0.779792041
ChanmanV	G2Thijs	undirected	0.938384525

Appendix D

The modularity report of the experiment on 3 lists

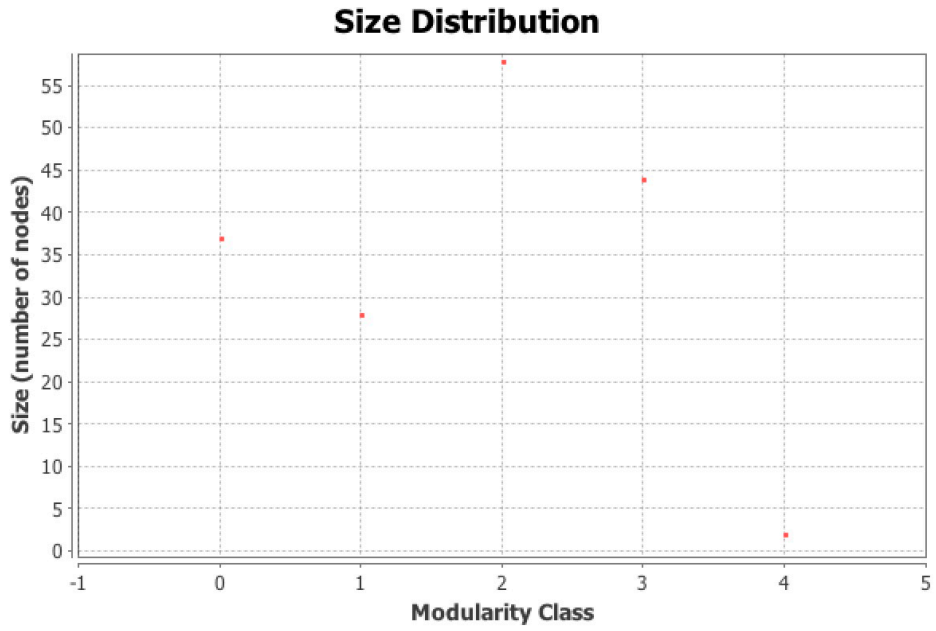
Modularity Report

Parameters:

Randomize: On
Use edge weights: Off
Resolution: 1.0

Results:

Modularity: 0.522
Modularity with resolution: 0.522
Number of Communities: 5



Appendix E

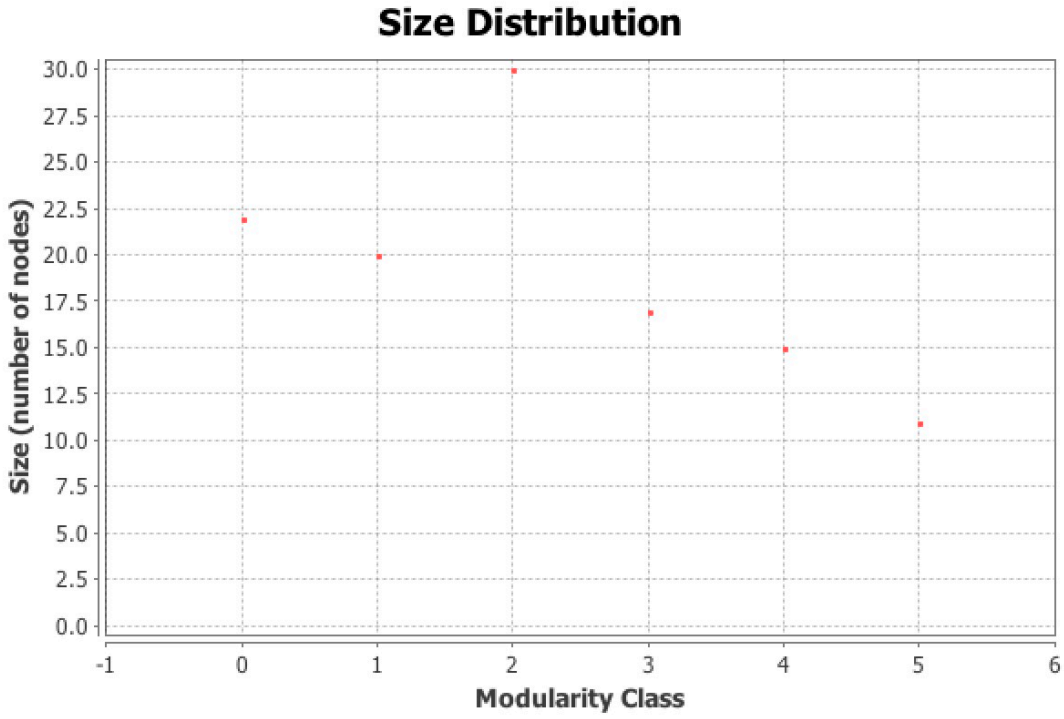
The modularity report of the list “RV”

Parameters:

Randomize: On
Use edge weights: Off
Resolution: 1.0

Results:

Modularity: 0.398
Modularity with resolution: 0.398
Number of Communities: 6



Appendix F

The modularity report of the list “HS”

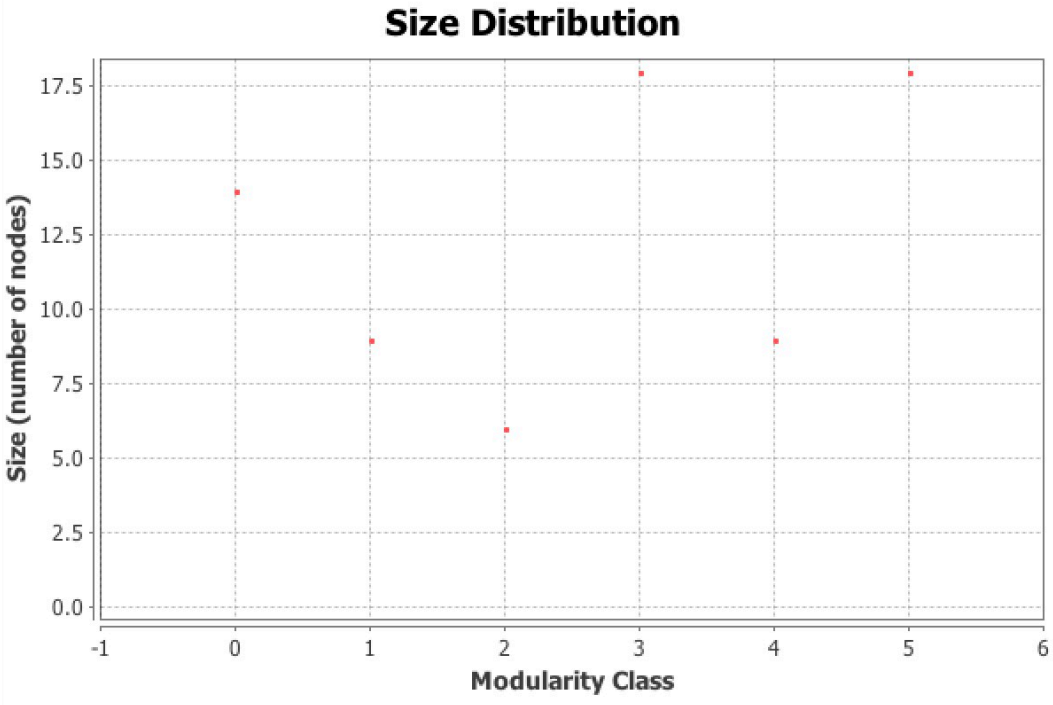
Modularity Report

Parameters:

Randomize: On
Use edge weights: On
Resolution: 1.0

Results:

Modularity: 0.335
Modularity with resolution: 0.335
Number of Communities: 6



Bibliography

[1] M. Girvan and M.E.J Newman, "Community structure in social and biological networks," In *Proceedings of the national academy of sciences*, *99*(12), 2002, pp.7821-7826.

[2] M.E.J Newman, “Fast algorithm for detecting community structure in networks,” *Physical review E*, *69*(6), 2004, p.066133.

[3] A. Clauset, M.E.J Newman and C. Moore, “Finding community structure in very large networks,” *Physical review E*, *70*(6), 2004, p.066111.

[4] V.D. Blondel, J.L. Guillaume, R. Lambiotte, and E. Lefebvre, “Fast unfolding of communities in large networks,” *Journal of statistical mechanics: theory and experiment*, *2008*(10), p.P10008.

[5] Q. Wang and E. Fleury, “Overlapping community structure and modular overlaps in complex networks,” In *Mining Social Networks and Security Informatics*, Springer Netherlands, 2013, pp. 15-40.

[6] A. Lancichinetti, S. Fortunato, “Limits of modularity maximization in community detection,” *Physical review E*, *84*(6), 2011, pp.066122.

[7] S. Fortunato, “Community detection in graphs,” *Physics reports*, *486*(3), 2010, pp.75-174.

[8] I. Derényi , G. Palla and T. Vicsek, “Clique percolation in random networks,” *Physical review letters*, *94*(16), 2005, p.160202.

[9] J. Xie, 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), 2013, p.43.

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

[11] A. Lancichinetti, F. Radicchi, JJ. Ramasco and S. Fortunato, “Finding statistically significant communities in networks,” *PloS one*, *6*(4), 2011, p.e18961.

[12] P. Wiemer-Hastings, K. Wiemer-Hastings and A. Graesser, “Latent semantic analysis,” In *Proceedings of the 16th international joint conference on Artificial intelligence*, 2004, pp. 1-14.

[13] T. Hofmann, “Learning the Similarity of Documents: An Information-Geometric Approach to Document Retrieval and Categorization,” 1999

[14] T. Hofmann, “Unsupervised learning by probabilistic latent semantic analysis,” *Machine learning*, *42*(1-2), 2001, pp.177-196.

[15] D.M. Blei, A.Y. Ng and M.I. Jordan, “Latent dirichlet allocation,” Journal of machine Learning research, 3(Jan), 2003. pp.993-1022.

[16] D.M. Blei and J.D. Lafferty, “A correlated topic model of science,” *The Annals of Applied Statistics*, 2007, pp.17-35.

[17] H. Zhang, B. Qiu, C. L. Giles, H. C. Foley, and J. Yen, "An LDA-based Community Structure Discovery Approach for Large-Scale Social Networks," *ISI,* vol. 200, 2007.

[18] W. Zhou, H. Jin, and Y. Liu, "Community discovery and profiling with social messages," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2012, pp. 388-396.

[19] Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu, "Understanding of internal clustering validation measures," In *Proceedings of 2010 IEEE International Conference on Data Mining* (pp. 911-916). IEEE.

[20] V. Labatut, "Generalised measures for the evaluation of community detection methods," *International Journal of Social Network Mining*, *2*(1), 2015, pp.44-63.

[21] G. Salton and M.J. McGill. "Introduction to modern information retrieval," 1983, pp.24-51.

[22] J.H. Ward Jr, "Hierarchical grouping to optimize an objective function," Journal of the American statistical association, 58(301), 1963, pp.236-244.

[23] L. Tang and H. Liu, "Community detection and mining in social media," Synthesis Lectures on Data Mining and Knowledge Discovery, 2(1), 2010, pp.1-137.

[24] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: is a correction for chance necessary?," In Proceedings of the 26th Annual International Conference on Machine Learning, ACM, 2009, pp. 1073-1080.

[25] P.J. Rousseeuw, “Silhouettes: a graphical aid to the interpretation and validation of cluster analysis,” Journal of computational and applied mathematics, 20, 1987, pp.53-65.