An Application of Clustering in Weighted Social Network

ISyE 6416 Project Report

Team Member Names: Junying He, Xi He

Problem Statement

Social networks are ubiquitous in today's world. Thanks to Facebook, Twitter, Linkedin, etc., people are now connected with each other in more ways than ever before. Such connections among people are usually modelled by *networks*, where each person is represented as a *node*, and connection between two people as *edge* between nodes. In a lot of real-world networks, there are also *weights* associated with edges, in order to show how strong the connections are. For example, in the network of email contacts, if person A and B have more email communications than A and C, then we can assign more weight to the edge between A and B than that between A and C, to differentiate these two edges.

One common question on a social network is: are there any communities in this social network? Or in other words, are there groups of people such that people communicate very frequently with those within the same group, but rarely with those from the other group? It is intuitive to see that detecting communities in a social network is just solving a clustering problem. However, finding network's communities is more challenging than ordinary clustering problems in the sense that the criterion to assign a node to a group is hard to define. Unlike clustering data points with specific numbers, where we can assign a point to the group whose center has shortest distance to the point, there is no obvious distance and no centers of groups in the case of social network. Fortunately, the general idea of clustering still applies: a node should be assigned to its closest community. To define the closeness between a node and a community, one natural thought is to introduce *density*, which describes how intense the communications are among a set of nodes. More details about clustering in social network will be shown in later sections and final report.

In this project, we want to apply the idea of clustering to the Enron email dataset. Our goal is to detect communities within the company, based on the email communications among the employees. We will also compare the graphical representation of the social network based on three different models to see which one is more reasonable for our study.

Data Source

The data we are using in this project is from Enron email dataset. This dataset was originally collected and prepared by the CALO Project, and made public by the Federal Energy Regulatory Commission. Later, the email dataset was purchased by Leslie Kaelbling at MIT,

and corrected by folks at SRI. The dataset is currently available on the website of CMU.

Methodology

First of all, we adopt some basic concepts (node, edge, weight, subgraph, etc.) from graph theory to model the Enron email dataset. To detect the communities in the Enron email network, we apply the general idea of clustering with some modifications to adapt to the situation of weighted social network. Specifically, we need to redefine the concept of closeness in social network, and then select efficient and accurate algorithms to analyze the data.

Let $G = (V, E)$ be a graph with node set *V* and edge set *E* with weight *w(e)* on every edge *e*. For a subgraph *C* such that $|V(C)|>1$, we define the density of *C* by

$$d(C) = 2\sum_{e\in E(C)} w(e)\,/|V(C)||V(C)-1|$$

According to its definition, the density is able to describe how close the nodes within a set are.

For a node v not in *V(C),* define the contribution of *v* to *C* by

$$c(v, C) = \sum_{u\in V(C)} w(uv)\,/|V(C)|$$

The concept of contribution is used in the algorithm as the criterion to decide whether to add a node to an already dense community. The node will be added to the community if its density is larger than a specified threshold value.

There are several algorithms that can facilitate our work. In this report, we will use three different algorithms to detect community structure of the Enron email data:

1. Greedy Optimization of Modularity (GOM)

2. Multi-level Optimization of Modularity (MOM)

3. Overlapping Cluster Generator (OCG)

Results and Evaluation

Using the *igraph* package in R, we can visualize the social network as in Figure 1.

Figure 1. Visualization of Enron email network

The Greedy Optimization of Modularity (GOM) algorithm gives the following results of community structure:

```
Community 1
 [1] "5"   "28"  "34"  "41"  "45"  "48"  "65"  "77"  "78"  "79"  "92"
[12] "100" "103" "131" "44"  "120" "32"  "101"
Community 2
 [1] "6"   "36"  "52"  "58"  "62"  "72"  "82"  "86"  "132" "27"  "67"
[12] "64"
Community 3
 [1] "1"   "2"   "11"  "13"  "37"  "54"  "55"  "63"  "66"  "74"  "88"
[12] "94"  "106" "111" "115" "121" "124" "127" "122" "116"
Community 4
 [1] "21"  "73"  "75"  "83"  "102" "108" "110" "129" "95"  "17"  "98"
Community 5
 [1] "3"   "19"  "20"  "30"  "35"  "53"  "84"  "105" "112" "126" "118"
[12] "93"  "8"
Community 6
[1] "18" "71" "76" "70" "10"
Community 7
[1] "14" "50" "38" "23"
Community 8
[1] "22"  "91"  "113"
Community 9
 [1] "9"   "16"  "24"  "33"  "39"  "40"  "46"  "47"  "57"  "60"  "61"
[12] "68"  "69"  "87"  "99"  "123"
Community 10
```

[1] "26" "133" "89"
Community 11
[1] "49"

The visualization of the community structure is show in Figure 2.

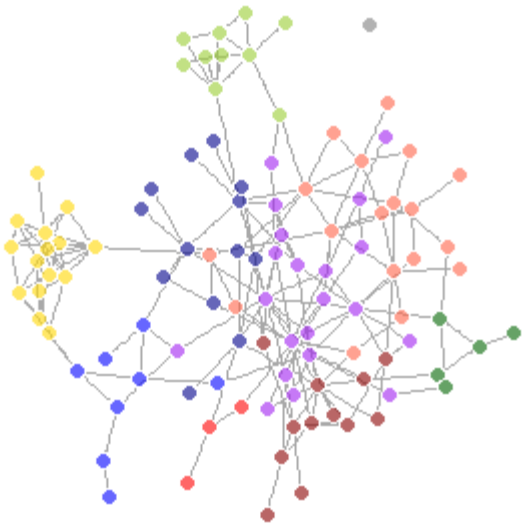

Figure 2. Visualization of communities by Greedy

The hierarchical structure of the network can also be shown using a dendrogram, which is in Figure 3.

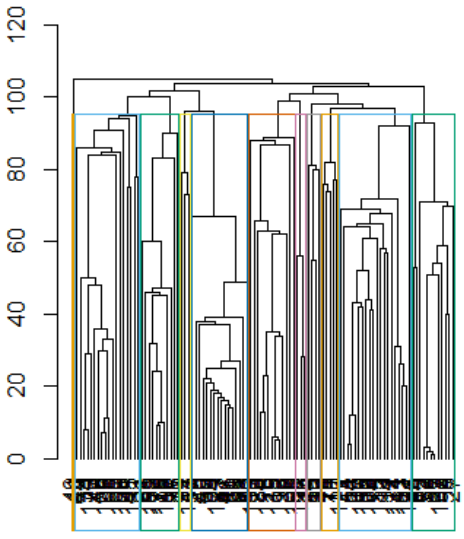

Figure 3. Dendrogram of Greedy

Since there are too many nodes in the data set, it is quite difficult to distinguish which node goes through which path. However, a general view of how the algorithm detect communities can still be obtained here.

The Multilevel Optimization of Modularity (MOM) returns similar results as the Greedy Optimization of Modularity. And the results are as the following:

```
Community 1`
[1] "49"
Community 2
 [1] "5"   "28"  "34"  "41"  "45"  "48"  "65"  "77"  "78"  "79"  "92"
[12] "100" "103" "131" "44"  "120" "32"  "101"
Community 3
 [1] "9"   "16"  "24"  "33"  "39"  "40"  "46"  "47"  "60"  "61"  "68"
[12] "69"  "87"  "99"  "123"
Community 4
 [1] "21"  "73"  "83"  "102" "108" "110" "129" "95"  "17"  "98"
Community 5
[1] "26"  "133" "89"
Community 6
[1] "14"  "22"  "50"  "57"  "91"  "113" "38"  "23"
Community 7
 [1] "1"   "2"   "11"  "13"  "37"  "54"  "55"  "63"  "66"  "74"  "75"
[12] "88"  "94"  "106" "111" "115" "121" "124" "127" "122" "116"
Community 8
[1] "18" "71" "76" "70" "10"
Community 9
 [1] "3"   "19"  "20"  "30"  "35"  "53"  "84"  "105" "112" "126" "118"
[12] "93"  "8"
Community 10
 [1] "6"   "36"  "52"  "58"  "62"  "72"  "82"  "86"  "132" "27"  "67"
```

And its visualization is shown in Figure 4.

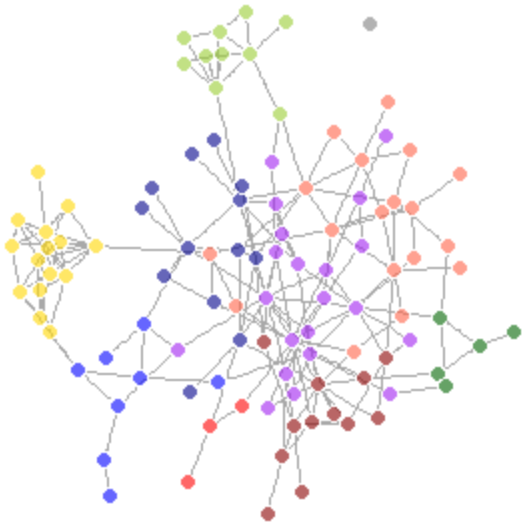

Figure 4. Visualization of Communities by Multi-level

If one carefully compares the members of each communities coming from the two algorithms, he/she will see that the communities are about the same, even though there are 11 communities detected in the first algorithm and 10 in the second one. In fact, these two algorithm basically use the same criteria to detect communities. The major difference is that the GOM starts by assuming all the nodes are in one community and then separates them, while the MOM starts by assuming each node is one community and then merges them.

Note that both GOM and MOM can only return communities without overlapping. In the case of Enron data set, it makes more sense to have overlapping communities in the company. For example, an HR manager might have frequent communications with two separated departments. In this situation, it is intuitive to put the HR manger inside both communities of the departments.

The algorithm of Overlapping Cluster Generator (OCG) was recently developed to allow for the existence of overlapping clusters. The results given by this algorithm is shown below:

Number of nodes = 106
Number of edges = 593
Number of communities = 44
Number of nodes in largest cluster = 12
Modularity = 69088
Q = 0.7998

Since the number of communities is quite large, the details of members inside each community are not shown here. The number of communities is larger than the previous results as expected, because one node can be in more than one community. The visualization of the communities and the dendrogram are shown in Figure 5 and 6 respectively.

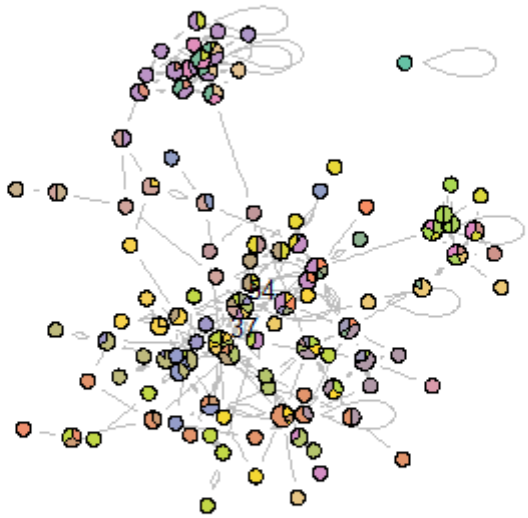

Figure 5. Visualization of Community by OCG

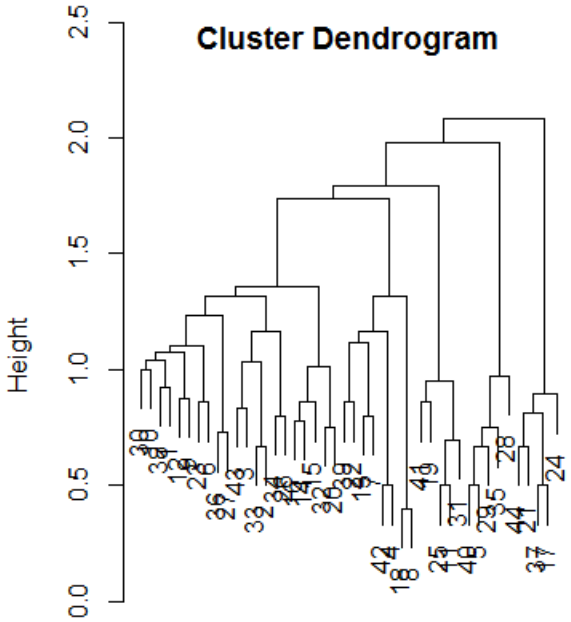

Figure 6. Dendrogram by OCG

It is very difficult to judge which algorithm is better over the other, because each of them has its own strength and weakness. However, in our case, we prefer the OCG algorithm, mainly because it allows overlapping clusters and is a more accurate depiction of the Enron email dataset. For different datasets and purposes, the choice is likely to be different.

Reference

1. Becker, E., B. Robisson, C. E. Chapple, A. Guenoche, and C. Brun. "Multifunctional Proteins Revealed by Overlapping Clustering in Protein Interaction Network." *Bioinformatics* 28.1 (2011): 84-90.

2. Blondel, Vincent D., Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. "Fast Unfolding of Communities in Large Networks." *J. Stat. Mech. Journal of Statistical Mechanics: Theory and Experiment* (2008.10).

3. Clauset, Aaron, M. E. J. Newman, and Cristopher Moore. "Finding Community Structure in Very Large Networks." *Physical Review E Phys. Rev. E* 70.6 (2004).

Distribution of work

Junying and Xi worked together on the methodology and Junying contributed the majority of the final report with modification done by Xi.