ISyE 6416 Project Proposal

Team Member Names: Junying He, Xi He

Project Title: An Application of Clustering in Weighted Social Network

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 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 result of our method with the graphical representation of the social network, to see if our result is reasonable.

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, design a reasonable criterion to build communities, and develop an efficient and accurate algorithm 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 distribution 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 distribution is larger than some parameter.

The algorithm will then be constructed based on density and distribution. The basic steps of the algorithm are:

1. Decompose a graph *G* into communities using edges with heavy weights
2. Grow the communities if there are nodes with large distributions
3. Merge any two communities that are too similar
4. Contract each community to a node and store the new graph as *G*
5. Repeat the procedure from step 1 until *E(G)*=∅

Expected Results

The final results that we are expecting to get include sets of nodes that represent communities in this social network, a graphical representation of the network, and a hierarchical tree that tracks the movement of nodes throughout the algorithm. Ideally, the communities detected will be consistent with the graph of the network.