datayo

data mining

Visulization and clustering Facebook Ego Network in R (part 1)

> By giving people the power to share, we're making the world more transparent
> Mark Zuckerberg

Mining social network is an interesting task in the world of facebook, twitter, google plus and other online social networking services.

I this post I will use `NetworkD3` (http://christophergandrud.github.io/networkD3/) to visualize friendship in facebook ego network and use `Markov clustering algorithm` (http://micans.org/mcl/) to cluster these ego-networks. The data set is taken from snap project (https://snap.stanford.edu/data/egonets-Facebook.html).

Note: `ego network` (aka. personal network) is a network of connections between someone's friends. You can find more information about ego network here (http://www.analytictech.com/networks/egonet.htm)

Dataset

So let's take a look at data set. Each ego-network we have `circles`, `edges`, `egofeat`, `feat`, `featnames` file. The detail of these file is described here (http://snap.stanford.edu/data/readme-Ego.txt). In this post, I'm going to cluster people only by their friendship, so we just focus in the edges file

- *edges* : The edges in the ego network for the node 'nodeId'. Each line represent a friendship between two users.

For example, the third line of `0.edges` file

```
24 346
```

means user 24 and user 346 are friends.

Our very first ego-network visualization

To visualiza ego network, I use networkD3, it's very fast, beautiful and importantly, it can be interactable.

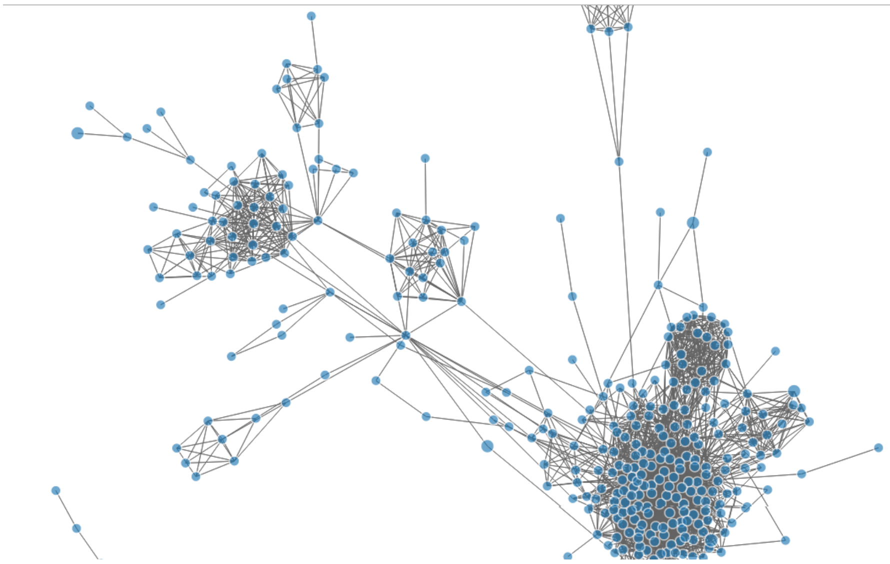
We load data from `0.edges` file

```r
graph.edges <- read.csv(
  file = 'data/facebook/0.edges',
  sep=" ", header=F,
  col.names=c("source", "target"))
graph.edges <- graph.edges - min(graph.edges) +
graph.nodes <- data.frame(
  id=seq(max(graph.edges) - min(graph.edges) + 1
graph.nodes$group <- 1
```

And then we create `forceNetwork` (http://christophergandrud.github.io/networkD3/#force). Notice that index of Links in forceNetwork is start from 0, while index of edges in `0.edges` file is start from 1, so we must subtract 1 from edges

```r
network <- forceNetwork(
  Links = graph.edges - 1, Nodes = graph.nodes,
  Source = "source", Target = "target",
  NodeID = "id",
  Group = "group", opacity = 0.8)
print(network)
```

And this is the result



Looking at this graph, we see some cluster here, and I find an algorithm to do it for me.

Clustering with MCL

We build a function `graph.cluster` to cluster a graph with input are nodes and edges. In this function, we first create an adjacency matrix `adjacency` and run `mcl` algorithm to this matrix to find our cluster

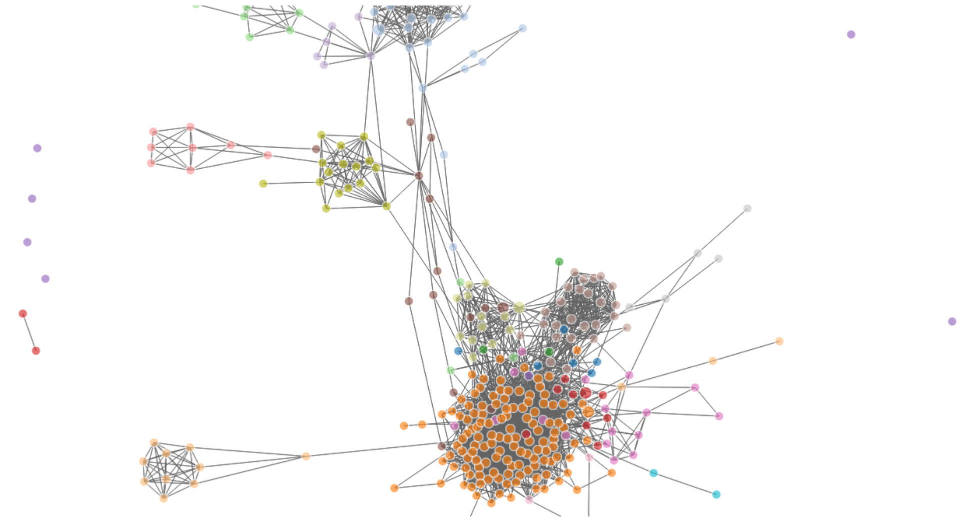
Next we use this function to reassign group to `graph.node` data frame

```
library(MCL)

graph.cluster <- function(nodes, edges){
  n <- nrow(graph.nodes)
  adjacency <- matrix(0, n, n)
  index <- cbind(
    graph.edges$source, graph.edges$target)
  adjacency[index] <- 1
  cluster <- mcl(
    x = adjacency, addLoops=TRUE, ESM = TRUE)
  print(cluster$Cluster)
  cluster$Cluster
}

graph.nodes$group <- graph.cluster(
  graph.nodes, graph.edges)
```

Finally, we use `networkD3` to create graph

```
network.cluster <- forceNetwork(
  Links = graph.edges - 1, Nodes = graph.nodes,
  Source = "source", Target = "target",
  NodeID = "id",
  Group = "group", opacity = 0.8)
print(network.cluster)
```

This code generate a graph like this below image

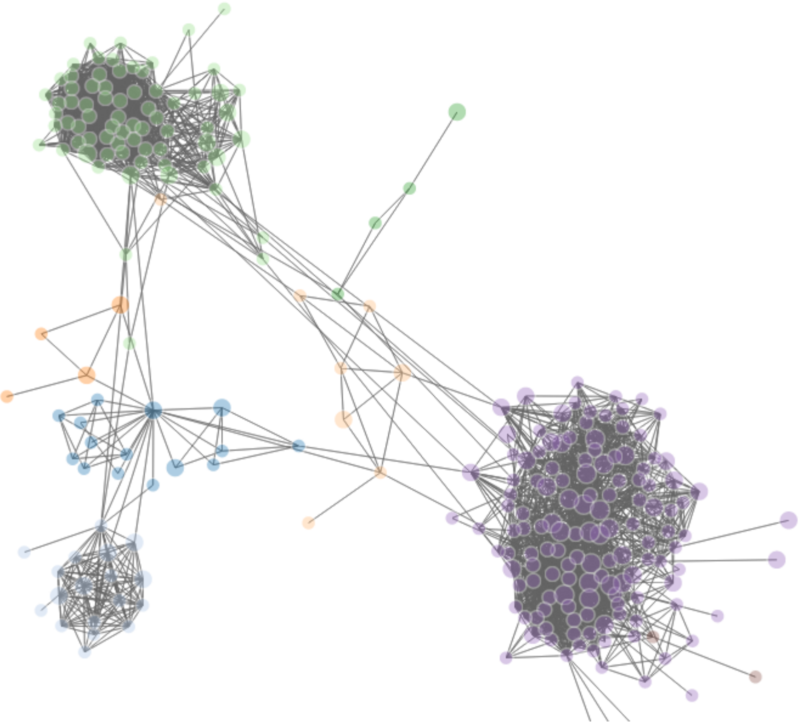


Now we see clusters of this ego network more clearly, there are a big orange cluster in the bottom of graph which contains many nodes, some smaller clusters with only 7-10 node like a light blue cluster in top of graph, some pink cluster, and there are many cluster which contains only one node, which is cluster for people has no friends in this ego network.

Conclusion

It's my very simple approach to cluster ego-network. This method clusters ego-network with only frienship information. MCL works very fast with ego-network with about 1000 nodes. You can try by yourself with larger network (simply change input file to another edges file)

With real data, here is my clustered go-network



The result is very interesting, the purple cluster are my friends from my university, the green one are my friends from my secondary and high schools. the blue one are my lover's friends and so on.

So we have a tool to discover an ego-network quickly in this post. I hope you find fun with this. In the next posts, I will try some more complicated method which cluster ego network with more information of people. See you there.

You can find full code and data of this project in https://github.com/datayo/social-mining (https://github.com/datayo/social-mining)

◎ APRIL 9, 2015 ◎ RAIN1024 ◎ SOCIAL-MINING, VISUALIZATION

2 thoughts on “Visulization and clustering Facebook Ego Network in R (part 1)”

1. amar says:
   Thanks for the post. can you please provide me the link for the next post you talked about.
   (more complicated method which cluster ego network with

more information of people)?

Regards
Amar

- Like

Liked by 1 person

REPLY JULY 23, 2015 AT 3:36 PM

- rain1024 says:

  I'm sorry Amar, I'm afraid that I will post it in 2-3 next months,

  - Like

  Like

  REPLY JULY 28, 2015 AT 2:18 PM

- Follow

Follow "datayo"