CS498 Fall 2016
Social & Information Network

Final Project

Bangqi Wang (bwang34)
Kedan Li (kedanli2)
12/07/2016

Introduction

The project focuses on the Learning Social Circles in Network, which helps the user to organize their social networks. The goal of this project is to cluster the ego network of the central user according to the features of the users in that network. With the developing of online social network, organizing the social circles becomes more and more important. People need an effective way to manage their social network, such as, Google+, Facebook, and Twitter. However, managing the social circle is sophisticated and challenging because the data sets is extremely huge and there is no good way to automatically organize the social circles. We use the KMean algorithm to cluster the users and get a submission output.

Related Work

1. McAuly, J., & Leskovec, J. Learning to Discover Social Circles in Ego Network:
This paper defines a novel machine learning task for social circles. This paper considers the social circle problem as the clustering problem with overlapping and hierarchy. What's more, the paper creates the evaluation metrics that could work on unsupervised environment.
The strength of this paper is that the algorithm form the circle based on some common aspects and user can pinpoint the aspects that caused the circle to form. What's more, this paper uses a real, large data sets from Google and Facebook, and the result is credible. The paper test multiple clustering methods only on profiles, only on network, and on the both.
However, the weak of this paper is obvious. The final algorithm used is Multi-Assignment Clustering which focuses on the profile information and ignore the network information.

2. Yang, J., Mcauley, J., & Leskovec, J. (2013). Community Detection in Networks with Node Attributes. 2013 IEEE 13th International Conference on Data Mining. doi:10.1109/icdm.2013.167: This paper introduces the technical method that clusters the users with the combination of the node attribute and the edge structure, CESNA. The method uses logistic model to calculate the features weight based on the weight of node in communities. Then calculating the gradient to update the weight. The strength is that the CESNA method is the first overlapping community detection method that model the node and the dependency and the runtime is linear. The weak is that the method cannot handle more general types of attributes and cannot cluster the attributes into "topics". The method should rely on other sources except node attributes.

Implementation

We implemented KMeans with python without using any third party library. We divide our code into the following modules: Network, People, Person, KMean, Similarity, Evaluation, Weight, Egonet.

When computing the social circle, the data structure will be initialized and load the egonets data from all the files under /egonets folder. It will also load the data from feature.txt, featureList.txt and all the training data under /testing folder. The data structure will be passed in to Kmean object.

We consider both features similarity and friendship similarity in when computing social circle. The Similarity class calculates the similarity between two friends based on their existing features and their frienship. It also manages the weight.

For each person, kmean will compute cluster of his friends. We approximate the number of center to be the square root of the number of friends the person has plus one. We run the kmean algorithm with the approximated number of center and assign the clustering result as the social circle for each person.

The result is then parsed into Evaluation class to run the Jaccard algorithm. The results are also written into folder /results. For testing purpose, we comment our evaluation functions.

By dividing different parts of the algorithm into different modules, we achieve a high readability and scalability. One can add a new algorithm easily.

Interpretation:

We compare the result of our cluster with the result in the training set, and found that most clusters are largely consistent with a few elements that are mismatched. We presume that the method didn't fully exploit the feature data but haven't fully the graph connectivity.
Interpretation

Final Algorithm:

1. Using logistic model to calculate weight for features in each node.
2. Calculating the features weight matrix for each pair of nodes in each egonet.
3. Combining the 57 features matrix and adjacent matrix to single weight matrix
4. Weight Matrix = W_0*Adjacent + W_1*$Feature_0$ + W_2*$Feature_1$ + … + W_{n+1}*$Feature_n$
5. Using K-means for clustering.

Algorithm Distribution:

Kedan Li: implementing K-means from scratch and uses K-means for features clustering.
Bangqi Wang**:** clustering with weight matrix calculated from features and egonets.

Reference:

1. Learning Social Circles in Networks, https://github.com/j-a-c/LearningSocialCircles, 8 Dec 2014, Joshua A. Campbell
2. Evaluation class is referenced to Zhenchao Liu (zliu80)