Large Scale Social and Complex Networks: Design and Algorithms
ECE 232E Summer 2024
Prof. Vwani Roychowdhury
UCLA, Department of ECE

Project 2
Social Network Mining

Due on Monday, July 22, 2024 by 11:59 PM PDT

In this project, we will study the various properties of social networks. In the first part of the project, we will study an undirected social network (Facebook). In the second part of the project, we will study a directed social network (Google +). **You can complete the Project using R or Python.**

1. Facebook network

In this project, we will be using the dataset given below:

http://snap.stanford.edu/data/egonets-Facebook.html

The Facebook network can be created from the edgelist file (`facebook_combined.txt`)

1. Structural properties of the Facebook network

Having created the Facebook network, we will study some of the structural properties of the network. To be specific, we will study

- Connectivity
- Degree distribution

QUESTION 1: A first look at the network:

QUESTION 1.1: Report the number of nodes and number of edges of the Facebook network.

QUESTION 1.2: Is the Facebook network connected? If not, find the giant connected component (GCC) of the network and report the size of the GCC.

QUESTION 2: Find the diameter of the network. If the network is not connected, then find the diameter of the GCC.

QUESTION 3: Plot the degree distribution of the facebook network and report the average degree.

QUESTION 4: Plot the degree distribution of Question 3 in a log-log scale. Try to fit a line to the plot and estimate the slope of the line.

2. Personalized network

A personalized network of an user v_i is defined as the subgraph induced by v_i and it's neighbors. In this part, we will study some of the structural properties of the personalized network of the user whose graph node ID is 1 (node ID in edgelist is 0). From this point onwards, whenever we are refering to a node ID we mean the graph node ID which is 1 + node ID in edgelist.

QUESTION 5: Create a personalized network of the user whose ID is 1. How many nodes and edges does this personalized network have?

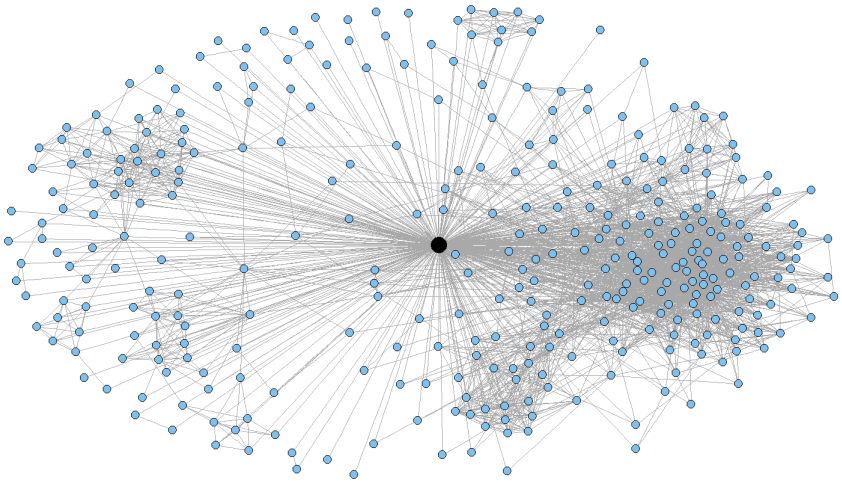
Hint Useful function(s): `makeegograph`

QUESTION 6: What is the diameter of the personalized network? Please state a trivial upper and lower bound for the diameter of the personalized network.

QUESTION 7: In the context of the personalized network, what is the meaning of the diameter of the personalized network to be equal to the upper bound you derived in Question 6. What is the meaning of the diameter of the personalized network to be equal to the lower bound you derived in Question 6 (assuming there are more than 3 nodes in the personalized network)?

3. Core node's personalized network

A core node is defined as the nodes that have more than 200 neighbors. For visualization purpose, we have displayed the personalized network of a core node below.



An example of a personal network. The core node is shown in black.

In this part, we will study various properties of the personalized network of the core nodes.

QUESTION 8: How many core nodes are there in the Facebook network. What is the average degree of the core nodes?

3.1. Community structure of core node's personalized network

In this part, we study the community structure of the core node's personalized network. To be specific, we will study the community structure of the personalized network of the following core nodes:

- Node ID 1
- Node ID 108
- Node ID 349
- Node ID 484
- Node ID 1087

QUESTION 9: For each of the above core node's personalized network, find the community structure using Fast-Greedy, Edge-Betweenness, and Infomap community detection algorithms. Compare the modularity scores of the algorithms. For visualization purpose, display the community structure of the core node's personalized networks using colors. Nodes belonging to the same community should have the same color and nodes belonging to different communities should have different color. In this question, you should have 15 plots in total.

Hint Useful function(s): `clusterfastgreedy`, `clusteredgebetweenness`, `clusterinfomap`

3.2. Community structure with the core node removed

In this part, we will explore the effect on the community structure of a core node's personalized network when the core node itself is removed from the personalized network.

QUESTION 10: For each of the core node's personalized network (use same core nodes as Question 9), remove the core node from the personalized network and find the community structure of the modified personalized network. Use the same community detection algorithm as Question 9. Compare the modularity score of the community structure of the modified personalized network with the modularity score of the community structure of the personalized network of Question 9. For visualization purpose, display the community structure of the modified personalized network using colors. In this question, you should have 15 plots in total.

3.3. Characteristic of nodes in the personalized network

In this part, we will explore characteristics of nodes in the personalized network using two measures. These two measures are stated and defined below:

- **Embeddedness** of a node is defined as the number of mutual friends a node shares with the core node.
- **Dispersion** of a node is defined as the sum of distances between every pair of the mutual friends the node shares with the core node. The distances should be calculated in a modified graph where the node (whose dispersion is being computed) and the core node are removed.

For further details on the above characteristics, you can read the paper below:

http://arxiv.org/abs/1310.6753

QUESTION 11: Write an expression relating the **Embeddedness** between the core node and a non-core node to the degree of the non-core node in the personalized network of the core node.

QUESTION 12: For each of the core node's personalized network (use the same core nodes as Question 9), plot the distribution histogram of embeddedness and dispersion. In this question, you will have 10 plots.

Hint Useful function(s): `neighbors`, `intersection`, `distances`

QUESTION 13: For each of the core node's personalized network, plot the community structure of the personalized network using colors and highlight the node with maximum dispersion. Also, highlight the edges incident to this node. To detect the community structure, use Fast-Greedy algorithm. In this question, you will have 5 plots.

QUESTION 14: Repeat Question 13, but now highlight the node with maximum embeddedness and the node with maximum $\frac{\text{dispersion}}{\text{embeddedness}}$ (excluding the nodes having zero embeddedness if there are any). Also, highlight the edges incident to these nodes. Report the id of those nodes.

QUESTION 15: Use the plots from Question 13 and 14 to explain the characteristics of a node revealed by each of this measure.

4. Friend recommendation in personalized networks

In many social networks, it is desirable to predict future links between pairs of nodes in the network. In the context of this Facebook network it is equivalent to recommending friends to users. In this part of the project, we will explore some neighborhood-based measures for friend recommendation. The network that we will be using for this part is the personalized network of node with ID 415.

4.1. Neighborhood based measure

In this project, we will be exploring three different neighborhood-based measures. Before we define these measures, let's introduce some notation:

- S_i is the neighbor set of node i in the network
- S_j is the neighbor set of node j in the network

Then, with the above notation we define the three measures below:

- Common neighbor measure between node i and node j is defined as

$$\text{Common Neighbors}(i,j) = |S_i \cap S_j|$$

- Jaccard measure between node i and node j is defined as

$$\text{Jaccard}(i,j) = \frac{|S_i \cap S_j|}{|S_i \cup S_j|}$$

- Adamic-Adar measure between node i and node j is defined as

$$\text{Adamic Adar}(i,j) = \sum_{k \in S_i \cap S_j} \frac{1}{\log(|S_k|)}$$

4.2. Friend recommendation using neighborhood based measures

We can use the neighborhood based measures defined in the previous section to recommend new friends to users in the network. Suppose we want to recommend t new friends to some user i in the network using Jaccard measure. We follow the steps listed below:

1. For each node in the network that is not a neighbor of i, compute the jaccard measure between the node i and the node not in the neighborhood of i

$$\text{Compute } \mathsf{Jaccard}(i,j) \ \forall j \in S_i^C$$

2. Then pick t nodes that have the highest Jaccard measure with node i and recommend these nodes as friends to node i

4.3. Creating the list of users

Having defined the friend recommendation procedure, we can now apply it to the personalized network of node ID 415. Before we apply the algorithm, we need to create the list of users who we want to recommend new friends to. We create this list by picking all nodes with degree 24. We will denote this list as N_r.

QUESTION 16: What is $|N_r|$, i.e. the length of the list N_r?

4.4. Average accuracy of friend recommendation algorithm

In this part, we will apply the 3 different types of friend recommendation algorithms to recommend friends to the users in the list N_r. We will define an average accuracy measure to compare the performances of the friend recommendation algorithms.

Suppose we want to compute the average accuracy of the friend recommendation algorithm. This task is completed in two steps:

1. Compute the average accuracy for each user in the list N_r
2. Compute the average accuracy of the algorithm by averaging across the accuracies of the users in the list N_r

Let's describe the procedure for accomplishing the step 1 of the task. Suppose we want to compute the average accuracy for user i in the list N_r. We can compute it by iterating over the following steps 10 times and then taking the average:

1. Remove each edge of node i at random with probability 0.25. In this context, it is equivalent to deleting some friends of node i. Let's denote the list of friends deleted as R_i
2. Use one of the three neighborhood based measures to recommend $|R_i|$ new friends to the user i. Let's denote the list of friends recommended as P_i
3. The accuracy for the user i for this iteration is given by $\frac{|P_i \cap R_i|}{|R_i|}$

By iterating over the above steps for 10 times and then taking the average gives us the average accuracy of user i. In this manner, we compute the average accuracy for each user in the list

N_r. Once we have computed them, then we can take the mean of the average accuracies of the users in the list N_r. The mean value will be the average accuracy of the friend recommendation algorithm.

QUESTION 17: Compute the average accuracy of the friend recommendation algorithm that uses:

- Common Neighbors measure
- Jaccard measure
- Adamic Adar measure

Based on the average accuracy values, which friend recommendation algorithm is the best?

Hint Useful function(s): `similarity`

2. Google+ network

In this part, we will explore the structure of the Google+ network. The dataset for creating the network can be found in the link below:

`http://snap.stanford.edu/data/egonets-Gplus.html`

Create directed personal networks for users who have more than 2 circles. The data required to create such personal networks can be found in the file named **gplus.tar.gz**.

QUESTION 18: How many personal networks are there?

QUESTION 19: For the 3 personal networks (node ID given below), plot the in-degree and out-degree distribution of these personal networks. Do the personal networks have a similar in and out degree distribution? In this question, you should have 6 plots.

- 109327480479767108490
- 115625564993990145546
- 101373961279443806744

1. Community structure of personal networks

In this part of the project, we will explore the community structure of the personal networks that we created and explore the connections between communities and user circles.

QUESTION 20: For the 3 personal networks picked in Question 19, extract the community structure of each personal network using Walktrap community detection algorithm. Report the modularity scores and plot the communities using colors. Are the modularity scores similar? In this question, you should have 3 plots.

Having found the communities, now we will explore the relationship between circles and communities. In order to explore the relationship, we define two measures:

- Homogeneity
- Completeness

Before, we state the expression for homogeneity and completeness, let's introduce some notation:

- C is the set of circles, $C = \{C_1, C_2, C_3, \cdots\}$
- K is the set of communities, $K = \{K_1, K_2, K_3, \cdots\}$
- a_i is the number of people in circle C_i
- b_i is the number of people in community K_i with circle information
- N is the total number of people with circle information
- A_{ji} is the number of people belonging to community j and circle i

Then, with the above notation, we have the following expressions for the entropy

$$H(C) = -\sum_{i=1}^{|C|} \frac{a_i}{N} \log(\frac{a_i}{N}) \tag{1}$$

$$H(K) = -\sum_{i=1}^{|K|} \frac{b_i}{N} \log(\frac{b_i}{N}) \tag{2}$$

and conditional entropy

$$H(C|K) = -\sum_{j=1}^{|K|} \sum_{i=1}^{|C|} \frac{A_{ji}}{N} \log(\frac{A_{ji}}{b_j}) \tag{3}$$

$$H(K|C) = -\sum_{i=1}^{|C|} \sum_{j=1}^{|K|} \frac{A_{ji}}{N} \log(\frac{A_{ji}}{a_i}) \tag{4}$$

Now we can state the expression for homogeneity, h as

$$h = 1 - \frac{H(C|K)}{H(C)} \tag{5}$$

and the expression for completeness, c as

$$c = 1 - \frac{H(K|C)}{H(K)} \tag{6}$$

QUESTION 21: Based on the expression for h and c, explain the meaning of homogeneity and completeness in words.

QUESTION 22: Compute the h and c values for the community structures of the 3 personal network (same nodes as Question 19). Interpret the values and provide a detailed explanation. Are there negative values? Why?

3. Cora dataset

One of the well-known categories of machine learning problems is "supervised learning". In supervised learning, we are given some information called "input" features about certain objects. For each object, we are also given an "output" or target variable that we are trying to predict about. Our goal is to learn the mapping between the features and the target variable. Typically, there is a portion of data where both input features and target variables are available. This portion of the dataset is called the training set. There is also typically another portion of the dataset where the target variable is missing and we want to predict it. This portion is called the "test set". When the target variable can take on a finite number of discrete values, we call the problem at hand a "classification" problem.

In this project, we are trying to solve a classification problem in settings where some additional information is provided in the form of "graph structure". In this project we work with "Cora" dataset. Cora consists of a set of 2708 documents that are Machine Learning related papers. Each documents is labeled with one of the following seven classes: `Case_Based`, `Genetic_Algorithms`, `Neural_Networks`, `Probabilistic_Methods`, `Reinforcement_Learning`, `Rule_Learning`, `Theory`. For each class, only 20 documents are labeled (a total of 140 for the seven classes). We refer to them as "seed" documents. Each document comes with a set of features about its text content. These features are occurrences of a selection of 1433 words in the vocabulary. We are also given an undirected graph where each node is a document and each edge represents a citation. There are a total of 5429 edges. Our goal is to use the hints from text features as well as from graph connections to classify (assign labels to) these documents.

To solve this problem for Cora dataset, we pursue three parallel ideas. Implement each idea and compare.

QUESTION 23: Idea 1

Use Graph Convolutional Networks [1]. What hyperparameters do you choose to get the optimal performance? How many layers did you choose?

QUESTION 24: Idea 2

Extract structure-based node features using Node2Vec [2]. Briefly describe how Node2Vec finds node features. Choose your desired classifier (one of SVM, Neural Network, or Random Forest) and classify the documents using only Node2Vec (graph structure) features. Now classify the documents using only the 1433-dimensional text features. Which one outperforms? Why do you think this is the case? Combine the Node2Vec and text features and train your classifier on the combined features. What is the best classification accuracy you get (in terms of the percentage of test documents correctly classified)?

QUESTION 25: Idea 3

We can find the personalized PageRank of each document in seven different runs, one per class. In each run, select one of the classes and take the 20 seed documents of that class. Then, perform a random walk with the following customized properties: (a) teleportation takes the random walker to one of the seed documents of that class (with a uniform probability of $1/20$ per seed document). Vary the teleportation probability in $\{0, 0.1, 0.2\}$. (b) the probability of transitioning to neighbors is not uniform among the neighbors. Rather, it is proportional to the cosine similarity between the text features of the current node and the next neighboring node. Particularly, assume we are currently visiting a document x_0 which has neighbors x_1, x_2, x_3.

Then the probability of transitioning to each neighbor is:

$$p_i = \frac{\exp(x_0 \cdot x_i)}{\exp(x_0 \cdot x_1) + \exp(x_0 \cdot x_2) + \exp(x_0 \cdot x_3)}; \text{ for } i = 1, 2, 3. \tag{7}$$

Repeat part b for every teleportation probability in part a.

Run the PageRank only on the GCC. for each seed node, do 1000 random walks. Maintain a class-wise visited frequency count for every unlabeled node. The predicted class for that unlabeled node is the class which lead to maximum visits to that node. Report accuracy and f1 scores.

For example if node 'n' was visited by 7 random walks from class A, 6 random walks from class B... 1 random walk from class G, then the predicted label of node of 'n' is class A.

Submission

Please submit the report to Gradescope. Meanwhile, please submit a zip file containing your codes and report to BruinLearn. The zip file should be named as "Project2_UID1_..._UIDn.zip" where UIDx are student ID numbers of team members. If you have any questions please feel free to ask them over email or piazza.

References

[1] Kipf, Thomas N., and Max Welling. "Semi-supervised classification with graph convolutional networks." arXiv preprint arXiv:1609.02907 (2016).