# Midterm Progress Report

Eric L. Lee, Yu-Cheng Weng, Can Jiang, Shu-Hao Chang

November 3, 2019

## 1 Abstract

Our proposal proposes that we want to perform link prediction on network structured data.

In this period, we have collected data and established our evaluation framework. We implemented 6 baselines model in the 8 different data sets. Besides, we perform data analysis and visualize our data set.

In the following three weeks, we will focus on implmenting factorization model and state-of-the-art methods. And try to further improve our result.

## 2 How to Reproduce Our Result

We believe reproducibility is very important when it comes to research. So, we carefully record all of our experiment results in github.

All of our result we presented in the following sections can be easily reproduced using this github repo.

https://github.com/miamiasheep/Purdue_ML_Course_Project

If you want to reproduce all of our result (including data analysis part), you can type the following command:

```
python script.py
```

If you want to reproduce the result of certain data sets, you can type the following command:

```
python main.py --input [file1,file2,...] --goal [auc/f1]
```

If you want to draw the graph of a network, you can type the following command:

```
python main.py --input [file] --draw [N(Integer)]
```

When N is larger, the sample network will be larger.

# 3 Data Analysis

## 3.1 Data Set

We collect 8 different data sets and all the data sets are very unique and have different characteristics. And table 1 shows basic statistics of each data set.

### 3.1.1 Facebook

This is a data set downloaded from SNAP(Staford NEtwork Analysis Project)[3]. The data is an egonet in facebook. Each node represent an account in facebook and each link represents friendship in facebook.

### 3.1.2 Power

Power is an electrical grid of western US[7].

### 3.1.3 NS

NS is a collaboration network of resarchers who publish paper on network sciences [4].

### 3.1.4 PB

PB is a network of US political blogs.[1]

### 3.1.5 Router

Router is a router-level Internet.[5]

### 3.1.6 USAir

USAir is a network of US air lines. [2]

### 3.1.7 Yeast

Yeast is a protein-protein interaction network in yeast.[6]

### 3.1.8 Celegan

Celegan is a neural network of C. elegans.[7]

## 3.2 Visualize Our Data Set

We also implement a tool to visualize our data sets. And all of our visualization graph can be seen in the following link: https://github.com/miamiasheep/Purdue_ML_Course_Project/tree/master/graph

| data set | nodes | edges | average degree | max degree |
|---|---|---|---|---|
| Facebook | 4039 | 88234 | 21.85 | 1045 |
| Power | 4941 | 6595 | 1.33 | 19 |
| NS | 1461 | 2742 | 1.87 | 34 |
| PB | 1222 | 16714 | 13.68 | 351 |
| Router | 5022 | 6258 | 1.26 | 106 |
| USAir | 332 | 2126 | 6.40 | 139 |
| Yeast | 2375 | 11683 | 4.91 | 118 |
| Arxiv | 18772 | 198110 | 10.55 | 504 |
| Celegan | 297 | 2148 | 7.23 | 134 |

Table 1: Basic Information of different data set

Because the network is too big and it is hard to find any clue if we draw all the edges and nodes, we only sample a small portion(a given set of nodes' neighbor and neighbors of the neighbor) of the network on every data set.

By visualizing our data sets. We can find that the characteristic of each data is actually very different. Take facebook(figure1) and Router(figure2) as an example, they are very different. Facebook data set has multiple triangle in the middle of the network. On contrast, Router has no triangle in the data set.

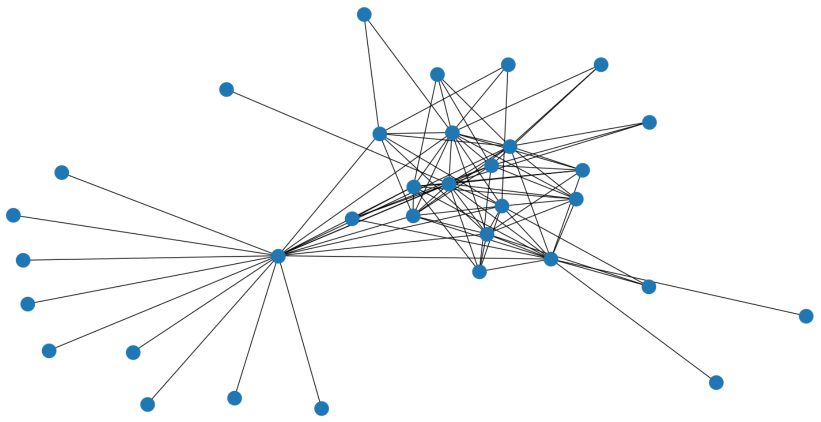

Figure 1: facebook

## 4 Evaluation and Result

### 4.1 Baselines

Given two nodes x and y be the node we want to predict. We let the set of neighbor x be N(x). We implemented six very common heuristics. And we organized the six heuristics in 2.

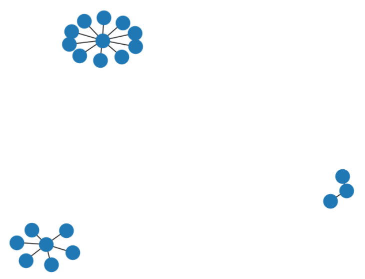

Figure 2: Router

### 4.1.1 Common Neighbors(CN)

It is a very simple heuristic. Two nodes have more common neighbor will be more likely to have a link. The score can represented as $|N(x) \cap N(y)|$.

### 4.1.2 Jaccard(JC)

It is very similar to common neighbors. If x and y are high degree nodes, it is not surprised that they will have many common neighbors. Jaccard wants to consider this effect and put the size of union of the neighbors of two nodes in denominator. It can be represented as $\frac{|N(x) \cap N(y)|}{|N(x) \cup N(y)|}$

### 4.1.3 Adamic Adar(AA)

It is also very similar to common neighbors. Given the assumption that a low degree common neighbor have more contribution to probability that x and y having a link. We divide the weight of each common neighbor by the logarithm of its degree. It is actually very similar to a TFIDF. And we can represented Adamic Adar as $\sum_{z \in N(x) \cap N(y)} \frac{1}{log(|N(z)|)}$

### 4.1.4 Total Neighbors(TN)

It is the most simple baseline which is just to calculate the total size of neighbors of x and y. It can be represented as $|N(x)| + |N(y)|$

### 4.1.5 Preferential Attachment(PA)

Given the assumption that a high degree nodes have more chance to have link have other nodes. Preferential Attachment calculate the product of degree of x and y. It can be represented as $|N(x)| * |N(y)|$

### 4.1.6 Page Rank(PG)

It is very similar to PA. But instead of just using the degree product of nodes, it use the product of the score after performing page rank to our network. Because the product can be very small, we use the sum of logarithm instead. Let $r_x$ be the score of x after performing

| method | description |
|---|---|
| common neighbor(CN) | $\|N(x) \cap N(y)\|$ |
| Jaccard(JC) | $\frac{\|N(x)\cap N(y)\|}{\|N(x)\cup N(y)\|}$ |
| Adamic and Adar(AA) | $\sum_{z\in N(x)\cap N(y)} \frac{1}{log(\|N(z)\|)}$ |
| Total Neighbor(TN) | $\|N(x)\| + \|N(y)\|$ |
| Preferentail Attachment (PA) | $\|N(x)\| * \|N(y)\|$ |
| Page Rank(PR) | $\log r_x + \log r_y$ |

Table 2: link prediction methods

page rank and $r_y$ be the score of y after performing page rank. It can be represented as $\log r_x + \log r_y$

## 4.2 Evaluation

Evaluation is not a trivial problem when it comes to link prediction problem. If we want to do all pair evaluation, it will take O($N^2$) predictions where N is the size of the nodes. And even for a small graph with 1000 nodes, the evaluation time is not trivial.

In fact, make a fair and effective evaluation is also a research topic and Ryan and Nitech has published a paper of how to evaluate link prediction problem. We choose the most common way to evaluate our model which is to do down-sampling on negative samples.

We randomly divide our edges into training, validation and testing according to the ratio 0.8:0.1:0.1. Training set is only for training, validation set is only for grid search of best parameters, and testing set is only for evaluate our model.

For validation and testing we randomly sample negative samples that equal to their size. If it is unlucky enough, some nodes may only exists in validation or testing set and not exist in training set, in this case, we will discard this edges in the evaluation.

To evaluate our result, we mainly use AUC score(area under Receiver Operating Characteristics). Because AUC has a very good characteristic which is the sampling AUC is actually an approximation of actual AUC even if we do down-sampling in negative data set.

However, if our model rank all positive data that higher than 90 percent of negative data but all of the positive data not the top ranked. The AUC score will be great ($> 0.9$) but all of the positive data are not likely to be recommended. Therefore, we also implement f1@k score as an evaluation. For the parameter k, we temporaily set the k equal to the size of positive label. For this k, recall is actually equal to precision. And it has a very intuitive physical meaning which is "the accuracy of finding all of the missing link if we know the size of missing link". In the future, we'll try to set different k and see the influence of the k.

# 5 Result and Discussion

Table 3 shows AUC of six different baselines and table 4 shows F1 score of the six baselines.

According to the result, we can see that no heuristic method can outperform all other heuristics. And we can see that Adamic Adar(AA), Jaccard, and common neighbors perform

| Data Set | CN | JAC | AA | PA | TN | PG |
|---|---|---|---|---|---|---|
| Celegans | 0.8314 | 0.7669 | 0.8443 | 0.7554 | 0.734 | 0.7998 |
| facebook | 0.9882 | 0.9863 | 0.9893 | 0.8324 | 0.7345 | 0.8198 |
| NS | 0.9742 | 0.9747 | 0.9741 | 0.6803 | 0.5204 | 0.7062 |
| PB | 0.9152 | 0.8669 | 0.9176 | 0.908 | 0.8776 | 0.9175 |
| Power | 0.5965 | 0.5964 | 0.5965 | 0.528 | 0.5198 | 0.7757 |
| Router | 0.6109 | 0.6102 | 0.6111 | 0.9298 | 0.9198 | 0.9659 |
| USAir | 0.9509 | 0.9164 | 0.9626 | 0.9021 | 0.868 | 0.9188 |
| Yeast | 0.902 | 0.901 | 0.9029 | 0.8561 | 0.7926 | 0.8864 |

Table 3: AUC of the six baselines

| Data Set | CN | JAC | AA | PA | TN | PG |
|---|---|---|---|---|---|---|
| Celegans | 0.7407 | 0.5926 | 0.713 | 0.5694 | 0.5787 | 0.6296 |
| facebook | 0.9546 | 0.9312 | 0.9544 | 0.6742 | 0.5618 | 0.655 |
| NS | 0.9524 | 0.9524 | 0.9524 | 0.5281 | 0.3939 | 0.4935 |
| PB | 0.8052 | 0.6927 | 0.801 | 0.7762 | 0.7223 | 0.7846 |
| Power | 0.1947 | 0.1947 | 0.1947 | 0.396 | 0.3518 | 0.5199 |
| Router | 0.2279 | 0.2279 | 0.2279 | 0.761 | 0.7574 | 0.8272 |
| USAir | 0.8278 | 0.7608 | 0.8612 | 0.799 | 0.7799 | 0.7608 |
| Yeast | 0.815 | 0.815 | 0.815 | 0.7024 | 0.6184 | 0.7274 |

Table 4: F1 score of the six baselines

pretty well in facebook, NS data set. However, in power and router data set. The result of common neighbor based method are pretty bad. It is not surprised because that in power and router data set, there are few triangles and thus common neighbor might not be a good heuristic.

## 6 Future Work

We want to keep working on what we have proposed in our proposal. Because we already finish our evaluation framework, we can start implement our machine learning models. And the result will be included in our final report.

## References

[1] Ackland, R. Mapping the us political blogsphere: Are conservative bloggers more prominent?

[2] Batagelj, V., and Mrvar, A.

[3] Leskovec, J., and Krevl, A. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data, June 2014.

[4] Newman, M. E. J. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E 74*, 3 (Sep 2006).

[5] Spring, N., Mahajan, R., Wetherall, D., and Anderson, T. Measuring isp topologies with rocketfuel. *IEEE/ACM Trans. Netw. 12*, 1 (Feb. 2004), 2–16.

[6] von Mering, C., Krause, R., Snel, B., Cornell, M., Oliver, S., Fields, S., and Bork, P. Comparative assessment of large-scale data sets of protein-protein interactions. *Nature 417* (06 2002), 399–403.

[7] Watts, D. J. Collective dynamics of small-world networks. *Nature 393*.