Assignment 2 Report

Zhen Zhai
zzhai@eng.ucsd.edu

Qiao Zhang
qiz121@eng.ucsd.edu

Yizhen Wang
yiw248@eng.ucsd.edu

ABSTRACT

Our project builds a binary classifier that predicts the existence of a connection between any pair of nodes in a facebook ego-net graph. We investigate the most representative features to use and compare the performance of Logistic Regression (LR), Support Vector Machine (SVM), and Random Forest (RF). SVM performs well on the specific task. LR performs decently and will potentially play a large role in a friend recommending system across different ego-nets.

1. INTRODUCTION

Friend recommendation is a popular topic in social network. A lot of researches have been conducted and amazing results are yielded. We explore friend recommendation on ego network using data set from facebook. Given a user in one ego network, we recommend people from the same ego network to be friend with this user. It make sense to us that when two people become friends, one would very likely know one's other friends. Therefore, we would like to have a system to recommend friend to a user when this user is new to this ego network. We build our recommendation system based on property features provided by users. We would like to recommend friends who have similar interest, social cycles, and background. We used three well-known models: Logistic Regression, Support Vector Machine, and Random Forest. Random Forest is very popular in industries to use as a classifier. We learn Logistic Regression and SVM from class, knowing that SVM optimizes the classification error rather than the likelihood comparing to logistic regression.

2. DATA DESCRIPTION

The data set we are investigating is the social circles data from facebook [1]. The data set consists of 10 ego networks including approximately 169882 edges and 4037 nodes. Each ego network's number of nodes range from 224 to 1034, and number of edges range from 540 to 60050.

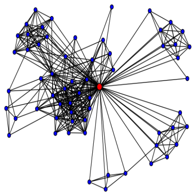

Figure 1: A graphical illustration of user 698's ego-net.

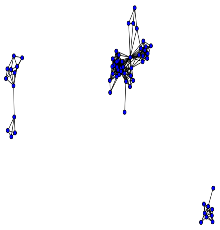

Figure 2: The connected friend nodes, the connected components after removing the ego and edges associated with the ego

On a high level, the data set consists of three important aspects of the facebook social network. First, it gives the graphical structure of the network, in which each node is a user and each edge represents the friendship between two users. Second, it contains some background information about each user, which includes a person's basic demographic information, his/her education, and work histories. Third, it gives the ground-truth communities, or circles on facebook, which represents various interest groups' membership.

2.1 Ego-net Representation

The entire facebook social network is too large to be represented in a compact form. The data set instead gives the *ego-net* perspective. An ego-net graph's node set contains a selected center, the ego, and all of his friends. The edges are induced from the people in the graph according to their friendship status. The nodes in ego-net don't only

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

.

have friendship edge with the ego node, there are also edges between ego's friend nodes.

One interesting way of looking at the ego-net is to remove the ego and its edges. The remaining graph illustrates how the ego's friends are associated with each other. The ego-net we shows in Figure 1 and Figure 2 are of ego with ID 698. There are 64 nodes in this ego-net. Figure 1 shows the ego node in red and all it's friend nodes in blue. Figure 2 is the connected components after we remove the ego and the associated edges. You can see from Figure 2 that there are only a few nodes from Figure 1 are connected with nodes other than the ego. Clusters in this graph potentially suggest some underlying communities.

2.2 Background Information as 0-1 Vectors

Each user's background information is represented as a 0-1 vector. Suppose a user's information contains n properties, and each properties i has m_i possible values. Then the entry of the vector is 1 if the user has property i with value j" for $i \in [n]$ and $j \in [m_i]$, the entry is 0 otherwise. The vector will be $\sum_{i=1}^{n} m_i$ long and is expected to be fairly sparse because each user often have more possible values in each property. For example, one of the properties is birthday. There are 8 possible values for birthday in this ego-net, each user's birthday vector will have seven 0s and one 1. It is also possible that one user doesn't provide birthday information, therefore his/her birthday vector will have eight 0s. The 0-1 vector of each users is concatenated by vectors of different properties. The properties in the data set include birthday, education, name, hometown, language, work, and locations.

2.3 Circles/Communities

The representation for social circles is straightforward: each circle has an ID, and members in the circles form a list under this ID. Each user could be in multiple circles, and the size of the circles vary from 1 to 300.

Overall, we call each ego-net, combining with its corresponding background information file and community membership file, a complete ego-net component, because it contains all the necessary information to investigate the particular ego-net. Our prediction task is binary, which is to predict whether two nodes are connected or not. Therefore, the models we considered are random forest, logistic regression, and support vector machine. We expect random forest to yield the best result because the clustering of the data set indicates that tree structure would be a good fit. We also expect SVM to have better performance than logistic regression, because SVM optimizes the classification error.

3. OUR PREDICTIVE TASK

3.1 Predictor Description

Our predictive task is to predict whether two people are friends in a specific ego-net given the each user's 0-1 information vector and the community membership, i.e. predict the graphical structure given user's personal data. Let X_i be the input vector of the i-th user with length d in the ego-net component. Our predictor f is a binary classifier

$$f : \{0,1\}^d \times \{0,1\}^d \to \{0,1\}$$

which takes two input vector X_i and X_j then output 1 if we predict two people being friends and 0 otherwise.

3.2 Significance of Our Predictor

One natural consequence of this predictive task is to recommend friend in each ego-net. For a user to be in an ego-net, he/she must be friend with the ego user. Therefore, we are predicting friendship between users who are not the ego user. We can view it as a friend recommendation system with auxiliary mutual friend information. For example, user i join the ego-net of user u, in other words, i and u become friends. We would like to recommend friends of u to be friend with i. It is nature to think that when i and u become friends, i would know or would like to know some of u's friends.

Also, compared to treating number of mutual friends as a feature, our predictor can potentially tell how important each mutual friendship is because we can train such predictor for each mutual friend's ego-net and develop an aggregation method.

3.3 Split Train/Validation/Test Data

We are predicting friends between users who are not ego user. Therefore, we excluded the ego node from our ego-net. We then randomly select 70% of the nodes as the training set, 15% to be validation set, and the remaining 15% to be our test set. We label the pair of nodes with friendship to be 1 and the pair with no friendship to be 0.

As we described in 2.1, only a small portion of nodes have friendship with nodes other than the ego. Therefore, there are much more negative labels (no friendship) than positive labels (friendship exists) in training set. We don't want the negative label nodes to be overweight. We solve this issue as following. We first pick a factor α. We include all the pair of nodes with positive labels. We find the size of the positive label pairs set to be s. We then randomly select negatively labeled pairs until the size of negative label pairs reach $\alpha \times s$. In other word, we limit the number of negative label pairs in training set to be no more than α times the size of positive label pairs. In our experiment, we picked α to be 2.

3.4 Evaluating Performance

We use classification error to quantify the performance of our predictor. On a test set X_{test}, the error rate is

$$err = \frac{\text{\# of misclassified pairs}}{\text{\# of pairs in the test set}}$$

Each pair of user in the ego-net component is an input to the predictor, and the label can always be checked from the graph.

Two baselines are used in our evaluation. In the first baseline, we randomly guess the friendship status with 50/50 chance. In the second baseline, we know the mean number of pairs in which friendship exists, and uniformly label the friendship status using majority rule. For example, if we know 60% of the pairs are not friends, then we will guess no friendship for all pairs.

4. MODEL

Our task is to predict whether two nodes are connected or not. Therefore, we considered random forest, logistic regression, and support vector machine as our three models. We first convert two individual feature vectors to one that represent the pair rather than two individuals nodes. Then,

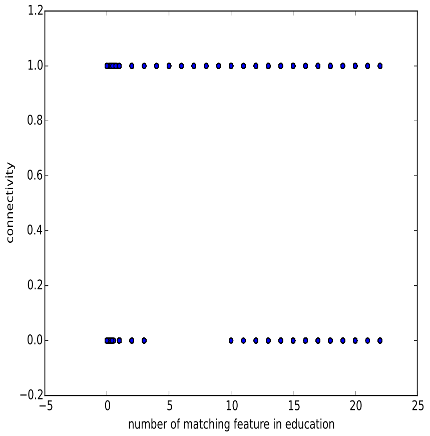

Figure 3: Number of matching education features with connectivity

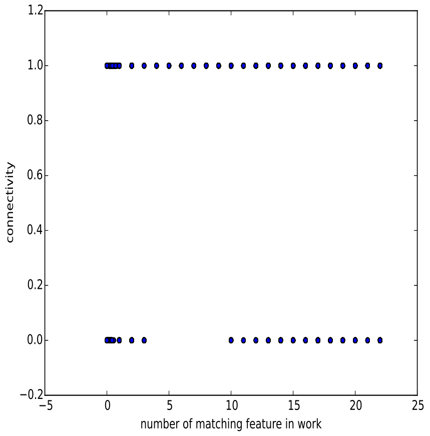

Figure 4: Number of matching work features with connectivity

we tune and test our models. Finally, we compare the results of different models using validation set then pick our final model to run on test set.

4.1 Build Feature Vector

For each input 1-0 vector pair X_i and X_j, we construct a new feature vector $\tilde{X_{ij}}$ using the following features:

1. number of similar features in X_i and X_j, which is just the number of matching feature between the two input vector
2. number of matching education features in X_i and X_j
3. number of matching work features in X_i and X_j
4. number of matching hometown features in X_i and X_j
5. number of matching language features in X_i and X_j
6. difference between number of value 1 features in X_i and X_j

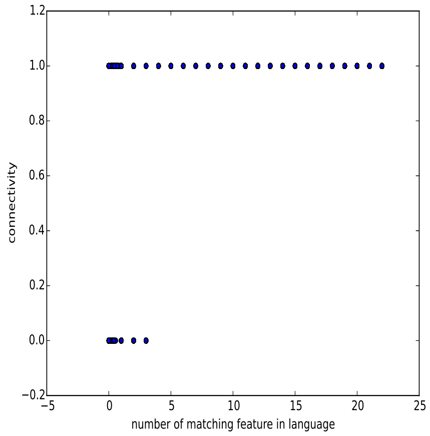

Figure 5: Number of matching language features with connectivity

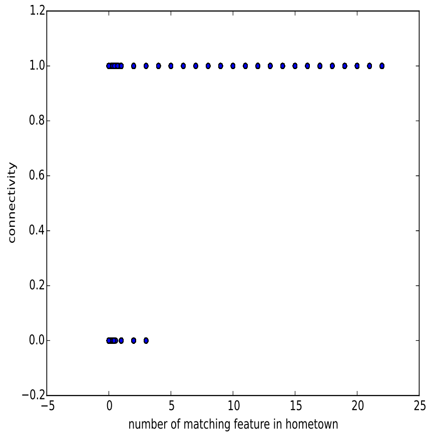

Figure 6: Number of matching hometown feature with connectivity

7. number of common social cycles the two users join
8. difference between number of circles of the two users
9. ratio of number of circles of the two users
10. cosine similarity of input vector X_i and X_j

We plot each features versus connectivity. We attached some of our plots here in Figure 3,4,5,and 6. Figure 3 and 4 show number of matching features in education and work. We can see that when the number of matching features are in range $[4, 10]$, we say with confidence that two nodes are connected. Figure 5 and Figure 6 show the plot of number of matching features of language and hometown. We can see from the figures that when the number of matching features are larger than 4, we say with confidence that two nodes are connected. Therefore, we tuned our feature 4 and 5 to be number of matching features larger than 4. We also tuned our feature 2 and 3 to be number of matching features between 4 and 10. As a result, it yields better result after we tune them.

Table 1: Pros and Cons of Models

Model	Pros	Cons
MR	easy and naive	low accuracy depends on the ratio of positive and negative label nodes
LR	good accuracy	optimize likelihood not actual classification error
SVM	better accuracy than LR	may overfit the train data not probabilistic
RF	high accuracy	slow depends on the size of the forest in some network

4.2 Model Selection

We are using two choices of binary classifier model learned in class: Logistic Regression (LR) and Support Vector Machine (SVM). We also explored a popular model used in industry, Random Forest (RF). We also included a model of majority rule (MR) as our baseline, which always predict unconnected for two nodes. We listed the pros and cons in Table1 of each model we tested.

We used the sklearn package on python to train each model respectively. The SVM object has a built-in cross-validation step when training the classifier, which saves our time to tune the parameters. For random forest, we built the forest with 100 trees and use all the features to do the feature selection.

4.3 Issues

One of our unsuccessful attempt was Naive Bayes. The result is very bad and we quickly shift our model to logistic regression and SVM. We think the reason Naive Bayes failed was because the assumption it made was too much for our data set. For example, it basically assume hometown and language are independent given they are connect or unconnected. We think such assumption is too naive and thus yield such bad result.

We also tried random prediction and as we imaging it give almost 50% accuracy.

Also, some of our features yield overfitting results. For example, we tried using specific feature within education, including school, degree, and even classes. Those information is too specific that it actually overfitted the data.

5. PREVIOUS WORK/LITERATURE

Understanding social network is a popular topic. There have been many previous work reports that overlap with our work. We want to discuss them in the following three categories: work on similar data with different objective, on different data with similar objective and lastly on the exact same data.

5.1 Identify Community in Graph - With Ground Truth Provided

One interesting aspect of the data, compared to other huge volume data crawled from the web, is that our data includes well-defined ground-truth community and the membership is also well classified. A counter-example to the well-defineness is twitter's hashtag. The meaning of a hashtag may vary a lot from different users' persepective and under the context in which it appears, while in our case a social group's focus, for example soccer lovers, is at least agreed among the community members.

It is natural then to detect these underlying communities from graph structure and compare the result to the groundtruth. However, it is not trivial to come up with a sound metric to evaluate the result. Yang and Leskovec [2] attempted to find a good metric. [3] finds that although no metric can guarantee that the detection is close to groundtruth, there does exist a set of metric of which the groundtruth must score high. The metric consists of Separability, Density, Cohesiveness and Clustering Coefficient. A good community detection algorithm, according to [3]. is also expected to decected communities in descending score order.

5.2 Edge Detection with No User Feature Information

There have also being contests on edge detection problem, which is similar to our edge prediction task. Facebook organized a contest [3] using directed graph in which each directed edge represents some following relation. [1], a report written by a prize winner Edwin Chen, illustrates various features one can extract from the raw data, which are eventually passed into a Random Forest model. [1] states that the Personalized PageRank is a key feature; propagation score, after adjusted for nodes with too few degrees, is also good. However, such features is not directly applicable to our problem because in the contest, the global graph structure is given, which is different from our ego-net. Personalized PageRank may not be very representative if some node in the ego-net has a large number of edges truncated from the global graph.

5.3 Generative Model Community Detection on Facebook Ego-net

McAuley's paper [2] proposes a novel approach of community detection. It treats the social circle membership as latent factors, and then tries to generate the ego-net graph as close as to the true graph. User profile features is also incorporated in this generative process. The objective, in short, is to maximize the likelihood of generating exactly the true ego-net. This novel approach not only identifies the circles, but also finds the number of circles without an expert telling the truth ahead.

What [2] encompasses with our work is that [2]'s model can compute the probability of an edge's existence as an intermediate step in the generative model. The model is solving community detection and edge detection simultaneously. The way [2] constructs the feature vector of a node pair could possibly be used in our task. In fact, we adopt some of the ideas including difference vector and common community vector between pair of nodes.

5.4 Similar Data Set Learned in the Past

The result of [2] also applies on ego-net on Google+ and Twitter. On all three data sets, the performance wins the previous state-of-art, which justifies the effectiveness of considering graph structure and user profile at the same time.

[3] uses various large scale graph data obtained from Youtube, LiveJournal and Amazon, with the edge having different meanings in different context. The data sets, although similar in representation, vary a lot in terms of sparsity, diameter and other parameters. The metric's robustness is tested in

[2] http://arxiv.org/pdf/1205.6233v3.pdf

[3] https://www.kaggle.com/c/FacebookRecruiting/

Table 2: Successful Classification of Logistic Regression (LR), SVM, Random Guess(RG) and Majority Rule (MR)

Ego ID	LR	SVM	RF	RG	MR
0	0.7343	0.7543	0.8421	0.4925	0.6667
107	0.7443	0.7620	0.7544	0.5092	0.6667
1684	0.7818	0.8224	0.8372	0.5103	0.6667
1912	0.6824	0.8391	0.8	0.4926	0.6667
3437	0.6701	0.7266	0.7485	0.4971	0.6667
348	0.7377	0.7370	0.7931	0.5277	0.6667
414	0.6950	0.9482	0.7692	0.55	0.6667
686	0.6974	0.7101	0.7333	0.4918	0.6667
698	0.7407	0.6667	0.6667	0.6111	0.6667

Table 3: Successful Classification of RF Model on Test Set

Ego-net ID	RF success rate
0	0.7809
107	0.7842
1684	0.7163
1912	0.8536
3437	0.7196
348	0.9166
414	0.9166
686	0.7222
698	1.000

all situations.

6. RESULTS

For each complete ego-net component, we train a logistic regression classifier and an SVM classifier, then check their successful classification rate on the *validation* set and compare the result to the two baselines mentioned early. The result is collated Table 2.

All three models outperform the baselines. RF gives the best prediction result among all. So we will choose RF as our final candidate.

The RF model result on the test set is shown in Table 3.

6.1 Why Random Forest Performs the Best

Intuitively, Random Forest is non-parametric and does not rely on the existence of a smooth and continuous boundary. As we expected, Random Forest works the best because of the clustering property of the data.

6.2 Why SVM performs better than LR

Intuitively, LR is computing the likelihood of whether a pair of users can be friends, while SVM is looking for a boundary supported by a number of points where the number is regularized. In our case, LR seems to suffer from *under-fitting* as the number of features are not as large as the dimension needed to separate the pairs well, thus it is only giving a general guideline on whether a point is more likely to be positive than negative. The underfitting problem is also reflected by the fact that error on training set is not going down enough. This problem is alleviated as we gradually add selective features to the input vector, which suggests that we are still in the range where expanding feature space/classfier dimension increases the performance.

SVM, on the other hand, can handle the situation because of its non-parametric nature. The number of support vectors can adapt to the geometry of the data. In our case, it is able to outline the cluster boundaries and use that to predict friendship. LR needs more features/more parameters to catch up SVM's performance.

6.3 Why is LR still useful

Recall that our goal is to achieve a general friends recommending system. Although SVM outperforms LR in *one* ego-net, it is not clear how to combine the SVM results across ego-net. LR, however, is a probabilistic representation and can naturally compose across different ego-net.

Consider person A, B and C. A and B have only 1 mutual friend while A and C have many friends. Suppose A and B show a large likelihood of being friends on their mutual friend's ego-net, while A and C show a weak likelihood of being friends on each of their mutual friend's ego-net but have positive result across all such ego-nets. Logistic regression allows us to quantify the likelihood so that we can aggregate the likelihood over all ego-net, while SVM cannot. Therefore we still think LR has potentials.

6.4 Feature selection

We try to evaluate our features by looking at the coefficient of the features in logistic regression model. The following is the coefficient features for ego-network with ego ID 698: $w = [0.433, -0.054, 0, 1.299, -0.738, -0.003, 0.127, 0, 0, 0, -0.873]$. For this ego-net, we can see that some of the features have weight 0, which means the corresponding feature has no significant meaning in our model. One of the corresponding feature of weight 0 is the difference between numbers of features each node has, which does not work as well as other features. Therefore, for future improvement, we will reconsider having the features corresponding to weight 0.

7. CONCLUSION

We compare the performance of three common models, namely Logistic Regression, Support Vector Machine and Random Forest, for an edge detection problem on facebook ego-network. We conclude that RF performs the best given our selection of features, while LR, although outperformed by the other models, can still be kept for cross ego-net usage for its probabilistic nature.

8. REFERENCES

[1] E. Chen. Edge prediction in a social graph: My solution to facebook's user recommendation contest on kaggle, 2012.

[2] J. Leskovec and J. J. Mcauley. Learning to discover social circles in ego networks. In F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 539–547. Curran Associates, Inc., 2012.

[3] J. Yang and J. Leskovec. Defining and evaluating network communities based on ground-truth. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pages 745–754, Dec 2012.