Project 1- Report
Team:Groot - Deshna Jain, Logesh Chinsu Palani, Tarun Dev Thalakunte Rajappa

1. Introduction

Popularity of social networks has increased over the years. These advancements allow us to study the structure and evolution of the network. Network analysis can be used to learn the interaction between people using these networks. One such application of social network analysis is to predict likelihood of link occurring in near future. This can be treated as a machine learning classification problem where a class represents if the link exists or not. The learning is done by adding various similarity features to the classifier. As the graph is large, extracting features is a computationally extensive task. These are then modelled with ensemble methods and are analysed. The notations used in the project are:

G(V,E)	Topological structure of graph G with vertex V and edge E
e(u,v)	Edges between u and v ∈ V
Γout(v)	Vertices adjacent outwards from v
Γin(v)	Vertices inward to vertice v
Γ(v)	Vertices adjacent to v

2. Data Set and Experiment Setup

The two given text files are “train.txt” and “test-public.txt”. “train.txt” is a sub graph crawled from Twitter Social Network, which contains 20000 lines. Each line has a set of nodes with tab-delimited, which represents the first node is following rest of the nodes. In the Graph there are 4867136 nodes and 23946602 edges. Positive edges are the real edges existing in the Graph. Negatives edges are the non-existing edges in the Graph. We randomly selected unique 30K positive edges and unique 30K negative edges to form the train set. “test-public.txt” contains the 2000 samples of negative edges. The below mentioned feature values were extracted for edges in train set and as well as test set.

3. Feature Engineering

The features generated are based on the similarity of the nodes. Similarity is calculated for both real and non-edges in the train and test set. For an edge between the nodes u and v the similarity is calculated based on the following features.

3.1 Adamic-Adar

It is defined as a weighted sum of common neighbors between u and v with z by the total number of neighbors of z. In reality it can be interpreted as, if a node z has acquaintance u and v and high number of edges then it is unlikely that the edge u,v will form than in case the node has lesser edges.

$$AA(u,v) = \sum_{z \in \Gamma(u) \cap z \in \Gamma(v)} \frac{w_{uz} + w_{vz}}{\log\left(1 + \sum_{a \in \Gamma(z)} w_{za}\right)}.$$

The rank of closer neighbors is greater compared to neighbors which have higher degree. It provided a feature importance of 0.111.

3.2 Jaccard's Coefficient

Jaccard's Coefficient is a common feature used in link prediction to calculate the similarity between two sample nodes.

Jaccard’s coefficient(u, v) = |Γ(u) ∩ Γ(v)|/ |Γ(u) ∪ Γ(v)|

It is the ratio between the number of common edges between the two nodes divided by the number of edges the nodes have. It depends on the idea that more the nodes are similar, more are its chances of link formation. This had a feature importance of 0.071.

3.3 Preferential Attachment Score

This is an edge feature which can predict the link based on the concept that if a node has many edges then it creates more edges in future. The link formation is proportional to the degree of the nodes. In twitter the probability of following the user increases with user visibility. This is done by taking the product of source and sink followers (Γin(u)*Γout(v)). The feature importance of this is 0.072.

3.4 Resource Allocation Index

It is based on the nodes connecting two nodes where the degree is not one. In this case there is a resource allocation through the intermittent links. It promotes links with lower strength and punishes neighbors with

higher strength. An unweighted variant of RA is used where the feature importance is 0.237 which is the highest.

$RA(u,v)=z\in\Gamma(u) \cap \Gamma(v)(1/sz)$

4. Methodology and Analysis

Firstly, As the test dataset contains even distribution of positive and negative edges, 2000 positive and negative edges are randomly extracted to the train dataset. This is then used to train the model to predict accurate results for similar test dataset. Using the dataset, 20 features such as node features like source followers, sink followers and subgraph densities along with edge features such as total friends, common friends, transitive friends and shortest path are extracted. It is assumed that the source with more number of followers tend to readily follow other nodes and the node who gets more number of followers tend to get followed by a node. Similarly, the node which follows more nodes and the node with more followers tend to form a link. Also, two friends with common friends tend to follow each other and two nodes with shortest path in the graph tend to form a link. However, with such features, the AUC of the model is estimated as 0.47. On further investigation, It is identified that node features has less impact on the probability of link formation than that the vertex features. Hence, in order to improve the accuracy, 8 features of higher value among the 20 features are identified including common friends, total friends and preferential attachment score. The train data was then extracted which improved the AUC to 0.53. Removing most of the node features and keeping majority of the edge feature provided more feature information on the relational features between given two nodes.

Secondly, Considering the test data has unique nodes in the edge list, the positive and negative edges for train data is sampled in such a way that all the nodes are unique. This provided more feature information that is similar to feature information of test data, as the edges in the test data tend to be random and unique with less repetition of same source or sink node. The number of positive and negative edges sampled were also increased to 19000 each to get better accuracy. It was identified that the increase in data provided more accuracy as when compared to data modelled using random forest classifier. This is due to the variation of trees formed which leads to a more accurate prediction. Hence, this model has increased the AUC to 0.61. Various classifiers, features and data size are explored and implemented but the AUC has remained nearly the same at this point.

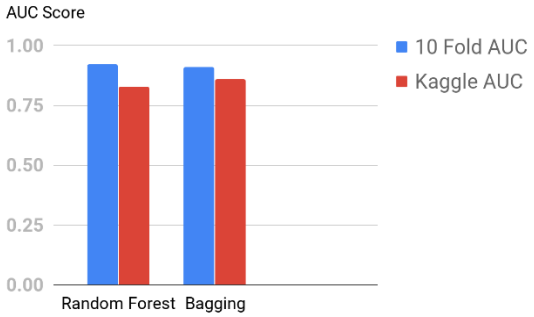
Thirdly, upon noticing that the requirement is to find the probability of an edge formation between those two nodes. Removal of sampled positive edges from the constructed graph will provide more accurate feature information. This feature will help the trained model to predict the probability of a node to be formed between a source and sink node in test data. When the edges are removed, and features are extracted to build the model, the model would contain the feature details of two nodes who certainly will form a link but has not formed yet. These features will help the model predict the nodes in the test data which will eventually form a link when the model determines the test feature information falls under the same category, as the feature information of positive train data edges with the edges removed. With the removal of positive edges from graph, the AUC has increased drastically from 0.61 to 0.74. To increase the AUC further, much higher valued features including cosine similarity, sorsen index, resource allocation index, salmon index and adamic adar index are included and lower valued features like subgraph density, count of bi-directional neighbors and edge features like common friends, total friends and opposite direction friends are removed. This has increased the AUC to 0.79 as the unvalued features provided disturbance in prediction.

Lastly, we increased the train dataset size to 200K samples, which contains even distribution of positive and negative edges. The more samples we have the better the model could learn. We removed positive edges from the graph and then computed the feature values for those features which were found to be the best. We used ensemble techniques such as Random Forest and Bagging to build the model. Random Forest gave AUC 0.84 with parameters n_estimators as 10 and random_state as 42. Bagging gave AUC 0.86 with parameters n_estimators as 100 and Base_estimator as Decision Tree. Clearly, we see an improvement over the previous method, as the features identified to have higher importance is clearly giving us good result. It was observed that Random Forest performed better when all the features were used compared to bagging. It selects random feature subset for each of the tree and overfits in presence of numerous features. But as the number of features were decreased, simultaneously the subset of random features to generate trees also decreased leading to less accurate predictions. Bagging on the other hand gave the highest AUC so far, as it is an aggregation of various predictors that reduces the risk of making a poor prediction, it maintains high accuracy and also reduces the size of the network by using graph sampling. Link prediction problem doesn't

scale well with network growth and therefore it is more efficient to recursively solve smaller problems than a single large problem. Solutions from these multiple models aggregate to provide a more robust process.

5. Result

For the graph we performed 10-fold cross validation approach for two machine learning algorithms such as Random Forest and Bagging. The ensemble algorithm Bagging achieves a high AUC score(0.93) compared to Random Forest AUC score(0.91). However, there was a slight decrease in AUC score of these ensemble algorithms Random Forest and Bagging 0.83 and 0.86 respectively in Kaggle.



6. Conclusion

This report demonstrates how link prediction can be carried out for the given graph, we explored two ensemble algorithms and have seen that feature selection plays an important role in predicting the right probability value. The features that do not add value to the model can negatively affect the AUC score. One of ways to identify these features are through feature importance measure. We evaluated the models by 10 fold cross validation which showed models performed very well in terms of AUC score. However, there was slight drop in AUC scores in Kaggle. Feature selection, selection of samples for model and size of samples are directly proportional to AUC scores irrespective of any algorithms. A possible further research can be carried out in finding new features which will add value to the model and eventually perform better supervised machine learning.

References

[1] M. Fire, L. Tenenboim, O. Lesser, R. Puzis, L. Rokach and Y. Elovici, "Link Prediction in Social Networks using Computationally Efficient Topological Features," *IEEE Computer Journal,* pp. 73- 80, 2011.

[2] C. Ahmed, A. ElKorany and R. Bahgat, "A supervised learning approach to link prediction in Twitter," *Springer,* 2016.