COMP90051 Statistical Machine Learning

Project 1 – Link Prediction

Shuai Wang, Jiawei Zheng, Shimei Zhao

1 Problem Definition

The interactions between people can be modelled with social networks, in which a node represents a person and an edge corresponds to some kind of association between persons. In Twitter social network, the nodes in the network are Twitter users, and a directed edge from node A to node B represents that user A follows user B. But there are often missing following edges between Twitter users. This might be caused by errors in data collection process, limited amount of edges or collector's careless. Those missing edges have serious impact on the analysis of the social network. Generally, we call this issue the problem of inferring missing links, which means we need to find out which edges are missing. Furthermore, we also need to predict the likelihood of a future following edge between two Twitter users, which is called link prediction problem. In short, the goal is to predict whether a following edge exists between two Twitter users.

2 Notations

The notations we used in our report are showed in Table 1.

Notation	Explanation
A	One Twitter user
B	One Twitter user
(A, B)	A pair of user A and user B
Γ(A)	The neighbours of user A
Φ(A)	The users that A is following
Π(A)	The users that are following A

Table 1: Notations used in the report

3 Features

This task is trying to predict whether an edge exists between a pair of Twitter users. We can regard it as a binary classification task. For (A, B), if there is an edge between A and B, give it a label 1; Otherwise, give it a label 0 representing an edge does not exist. Therefore, we need to select some features to represent (A, B), and then create a train set to build a classifier used to predict test data. All features we considered as showed in Table 2.

4 Datasets

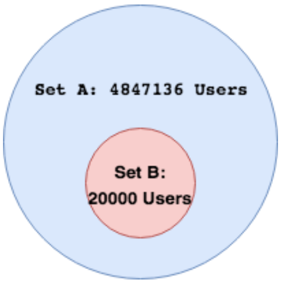
Since the training data include 4867136 users and 24004369 edges, it is almost time consuming and the model might also be overfit, if we put all these data into our model. Therefore, we analyse the test data distribution and the training data set. We find that we have almost all the following information for the users (20000 users) who is in the set B and other 4.8 million are just in the outside circle which only have some followers' information. In the test set, we found that all the sources and 18.8% of sinks come from set B and 81.2% of sinks come from set A. In order to have a better performance and let our model to fit our test set., we generate our training set base on the distribution of the test set.

To build a model to predict test data, a train set is created to train a model and a validation set is used to adjust the hyper-parameters of the model. The details of train set and validation set are showed in Table 3.

Feature	Explanation	Calculation
A_num_of_followings	The number of users that A is following	$\|\Phi(A)\|$
A_num_of_followers	The number of users that are following A	$\|\Pi(A)\|$
A_num_of_neighbours	The number of neighbours of A	$\|\Gamma(A)\|= \|\Phi(A) \cup \Pi(A)\|$

B_num_of_followings	The number of users that B is following	$\lvert\Phi(B)\rvert$
B_num_of_followers	The number of users that are following B	$\lvert\Pi(B)\rvert$
B_num_of_neighbours	The number of neighbours of B	$\lvert\Gamma(B)\rvert= \lvert\Phi(B) \cup \Pi(B)\rvert$
num_of_common_followings	The number of common followings	$\lvert\Phi(A) \cap \Phi(B)\rvert$
num_of_common_followers	The number of common followers	$\lvert\Pi(A) \cap \Pi(B)\rvert$
num_of_common_neighbours	The number of common neighbours	$\lvert\Gamma(A) \cap \Gamma(B)\rvert$
jaccard_coefficient	The Jaccard's Coefficient of A and B	$\frac{\lvert\Gamma(A) \cap \Gamma(B)\rvert}{\lvert\Gamma(A) \cup \Gamma(B)\rvert}$
adar	Adamic/Adar(Frequency-Weighted Common Neighbours) of A and B	$\sum_{z \in \Gamma(A) \cap \Gamma(B)} \frac{1}{\log \lvert\Gamma(z)\rvert}$
preferential_attachment	Preferential attachment score of A and B	$\lvert\Gamma(A)\rvert \bullet \lvert\Gamma(B)\rvert$
resource_allocation_index	Compute the resource allocation index of A and B	$\sum_{z \in \Gamma(A) \cap \Gamma(B)} \frac{1}{\Gamma(z)\rvert}$

Table 2: Features of (A, B)



Dataset	Size	Description
Train Set	284217	142059 labelled 1; 142158 labelled 0
Validation Set	94739	47419 labelled 1; 47320 labelled 0
Test Set	2000	1000 (label 1); 1000 (label 0)

Table 3: Train set and validation set details

Figure 1: Training data analysis

5 Feature Selection

At the beginning of the project, we cannot directly tell which features are better and more relevant to the label. So we tried all of the 13 features. Concretely, build a Logistic Regression model over each feature, and then test their accuracy on the validation set. The results are showed in Figure 2. We found jacard_coefficient, resource_allocation_index and preferential_attachment got higher accuracy than other features. It means that these three features are more helpful to predict the label. We believe that is because for only calculate the number of followings and the followers number of specific user might be unrelated for our problem. If a user follows 10 user in his or her music community and other user is interested in other aspect such as IT. These two user might irrelevant at all, but the features might tell the model that they might have some kind of relation. On the other hand, the graph-based features consider the relation between two users, it computes the distance or the relative factors between two users, which might be the reason that they have a better performance. For example, the Jacard's Coefficient indicate the amount of comment friend and remove the influence of the total number of neighbours. And the preferential attachment demonstrates that users who have a lot of friend have high probability to become a friend later. Therefore, we finally choose jacard_coefficient, resource_allocation_index and preferential_attachment features.

6 Approaches

We tried some of popular machine learning techniques. Their AUC scores of each model on test data are showed in Figure 3. First of all, we try some traditional machine learning algorithm. The best model is decision tree and de random forest. In our opinion, the random forest algorithm can maintain accuracy even if a large proportion of the date are missing and it can also provide less bias than the decision tree algorithm. Fast to compute on a large dataset is also another features we are looking for. The other reason is that base on W J. Cukierski et. al. research, random forest achieved a good result for link prediction on Flickr datasets. However, the random forest still gets about 85% AUC, which is not quite good. We then try to use ensemble learning to get a better prediction.

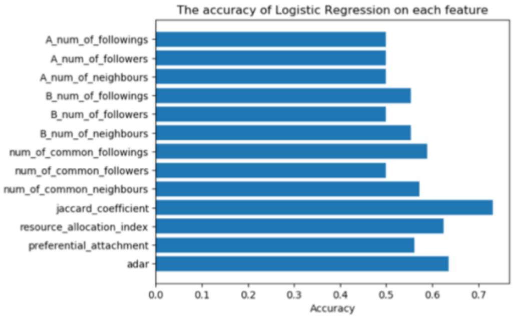


Figure 2: The accuracy of each feature

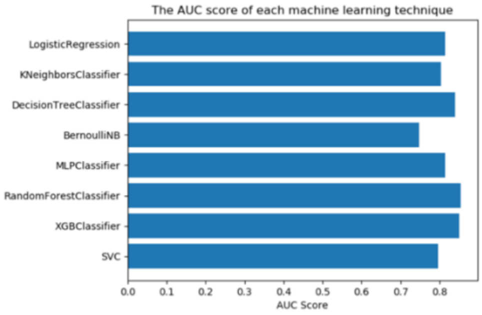


Figure 3: The AUC scores of each ML techniques

7 Ensemble Learning

Ensemble learning is the integration of several weak classifiers into one strong classifier. In this project, we considered the problem as a supervised learning problem (classification problem) and then implemented one of the most important ensemble learning methods, the XGBoost classifier, to solve it. Here, we used the sigmoid transformation to get the probability predictions and the main steps we took to build this model is listed as follows:

- Firstly, we pre-processed the training data by randomly selecting the same number of positive instances and negative instances for each head node in the training set.
- Then, we obtained three main features for each instance, including the Jaccard's Coefficient, the Resource Allocation Index and the Preferential-Attachment-Score and labelled them with 0 or 1.
- After tuning the parameters, we finally built the XGBoost Classifier model with 400 estimators (i.e. 400 decision trees), 5 as the maximum depth of trees, 1 as the minimum child weight, 80% of features of all trees used when training each tree.
- Followed by that, we input the features and their labels in the format of vectors.
- We fit 80% of the training data into the XGBoost and use the left 20% of data as the validation data.
- Finally, we used the trained model to predict the probability of one given instance belonging to class 1, that is how likely the former node followed the latter.

The result showed that we achieved an accuracy of more than 85%, which proved that XGBoost model can work well in solving social network link prediction problem.

8 Conclusions

In this report we present methods for our features extraction, features selection and the performance for each machine learning model. We find out that the graph based features have a good representation for the users link features. If we have other information, such as username, twist they post and the users group information, we might extraction more useful features and have a better prediction. We demonstrated that we also try to use the ensemble model to get a better result, but the AUC score seem to be there is no much improvement than the traditional machine learning algorithm.

Reference

- Al Hasan, M., Chaoji, V., Salem, S., & Zaki, M. (2006, April). Link prediction using supervised learning. In *SDM06: workshop on link analysis, counter-terrorism and security*.
- Benchettara, N., Kanawati, R., & Rouveirol, C. (2010, August). Supervised machine learning applied to link prediction in bipartite social networks. In *Advances in Social Networks Analysis and Mining (ASONAM), 2010 International Conference on* (pp. 326-330). IEEE.
- Cukierski, W., Hamner, B., & Yang, B. (2011, July). Graph-based features for supervised link prediction. In *Neural Networks (IJCNN), The 2011 International Joint Conference on* (pp. 1237-1244). IEEE.
- Zhang, M., & Chen, Y. (2018). Link Prediction Based on Graph Neural Networks. *arXiv preprint arXiv:1802.09691*.