COMP90051 Statistical Machine Learning

Project 1 report

Zihang Su 710118 **Tianyi Ou 872565** **Yongkang Liu 849892**

Introduction:

The problem for this project is to predict whether an edge is real or fake. The given information are real edges between two twitter IDs (node). There are three main parts we did to solve this problem: training set selection, features extraction and learning model construction.

Notations: if A follow B, B is **successor** for A, A is **predecessor** for B.

Training set selection:

There are over 20 million edges in the training data TXT file (very large). Since our goal is to estimate the 2000 edges in test set, therefore, we want to find the node pairs which are more relevant to the test set as our training set.

To select the pairs, we loop over all the source nodes in the test set and for each source node we randomly select 20 nodes has positive labels (means there is a real edge between the source node and this random node) and also randomly select 20 nodes has negative labels (there is no real edge in between). Therefore, each source in test set will have several randomly selected nodes as node pairs. The labels between those pairs are half positive and half negative.

The reason for selecting half positive and half negative is to avoid the training model to over predict on one class and under predict on the other. For example, if we select many positive labels but few negative labels, the training model would tend to give more positive predictions.

Similarly, for each sink node in the test set, we randomly select 20 nodes with positive label and 20 nodes with negative label to form the rest part of training set. Thus, both source nodes and sink nodes from test set have half positive and half negative node pairs as our training set.

After selecting the training set, we randomly hold out 10% of the training set for validation. The actual training data are 90% of the training set. We use the validation set for verifying whether the model perform good or not for making further adjustment.

Features extraction:

We included 5 features in our final approach:

1. Intuitively, if person A follows person B and B follows C, it is likely that A also follows C. Hence, our feature 1 is about whether a source node follows another node who follows the sink node. This is a binary feature with ‘0’ and ‘1’.
2. Like feature 1, if A follows many people (set B) who follow C, the larger the set B is, the more likely that A would follow C. Hence, our feature 2 is about the number of people in set B.

3. In practice, if A follows B, there is also a chance for B to follow A (friends follows each other). Hence, our feature 3 is about whether the sink node reversely follows the source node (Binary feature).
4. If A and B have intersections between them, the larger the intersection the higher chance for them to follow each other. Therefore, we use Jaccard similarity between the source node and sink node as our feature 4. The formula shown as below.

$$Jaccard\ similarity = \frac{\Gamma(A) \cap \Gamma(B)}{\Gamma(A) \cup \Gamma(B)}$$

   where $\Gamma(X)$ stands for the predecessors of X. The higher Jaccard similarity means A and B shares more similar interests which means they are likely to follow each other. And our empirical result shows Jaccard similarity make a significant improvement for prediction accuracy.
5. In practice, each person would follow a different number of people, if A follows B and A also follows lots of other people, then B might not be a very important person for A. However, if B is the only person that A follows, it implies that B is very important to A. Hence, we use resource allocation as our feature 5, formula shown as below.

$$resource\ allocation = \sum_{W \in \Gamma(A) \cap \Gamma(B)} \frac{1}{|\Gamma(W)|}$$

   where $\Gamma(X)$ stands for the predecessors of X, therefore, W is the intersection of both A's and B's predecessors, and $|\Gamma(X)|$ stands for the number of predecessors for X.

For feature 4 and feature 5, we only use predecessors but not successors, because in test set, there are only about 300 out of 2000 sink nodes which have the information of successors. From the dataset be given, we can't find any successor information about the other 1700 sink node in test set to calculate feature 4 and feature 5. Hence, we didn't use successors.

Learning model construction:

Some traditional models for training a binary classifier are logistic regression, decision tree, support vector machine.

Logistic regression is not performed well for this task because the training data and our selected features is not linear separable, it might be better if we could convert the logits formula to a non-linear model. As for decision tree, it trend to overfit the model. For example, if we only use a certain feature A as input data, the AUC output is 0.49 (this feature A not perform well), but if we add feature A to a group of other features the AUC will improve compared to the previous AUC even though feature A seems not to be a good feature. Therefore, as we add more and more features, the AUC from decision tree getting better and better, not getting worse, which could cause overfit problem. SVM is performed slightly better, but the training time is much longer.

Our final approach is a deep learning approach with feed forward neural networks. The network includes 3 layers and each layer has 200 units, we use sigmoid as our activation function. The optimizer we used is stochastic gradient descent with learning rate of 0.01. We randomly iterate

the whole training data set with a batch size of 100 for each epoch (500 epochs in total).

We tried logistic regression, decision tree and support vector machine by using the built-in function in scikit-learn. We build a 3-layers neural network model with TensorFlow. The AUC from our validation set are shown as below (the AUC calculated by built-in function in scikit-learn as well).

Logistic regression	Decision tree	Support vector machine	Neural network
67.9%	77.6%	78.3%	80.5%

Discussion:

The advantage of our final approach is that the training data set we selected is relatively large and apply neural network on a large dataset will get better performance. In addition, the multilayer neural network could convert the model to a non-linear model which is good for solving this not linear separable problems (better than logistic regression). As for features extraction, we tested lots of features, but we didn't include as much feature as possible to avoid overfitting, all the features we included have a significant contribution to prediction (by testing each feature alone). However, neural network has lots of parameters and tends to overfitting which is a disadvantage.

The prediction result from Kaggle competition is not as good as what we expected might for several reasons.

1. The training set selection have some bias. In order to calculate the features, we need to select the source nodes which have the information of follow some person (follow information) and have the information of be followed by some person (be followed information). If a node doesn't have follow or be followed information according to the given dataset, we exclude it from the training set. Therefore, the training set is not exactly randomly selected which might bring in some bias.
2. There might be some other powerful features which could improve the performance of prediction. We could explore more features to make improvement.
3. Features and training model might not be suitable to each other. When we select features, we also need to consider whether a certain feature is suitable for a certain training model. For example, the feature 2 are numbers, we found our neural network is not performed well if the feature contains the very large numbers. After taking the log value of feature 2, the performance improved. Hence, we suspect that there might be some other tricks for making features more suitable for our training model to make an improvement.
4. The deep learning model is not appropriately built in terms of number of units and layers, activation functions, optimization method etc. Therefore, we could improve it by building a better architecture. However, we are new for neural network and TensorFlow, we would like to learn it more in further study.

Finally, what we learned from this project is that training set selection, feature extraction and learning model construction are three very important components for this task. If any part is inappropriately implemented, there would be a big loss on your prediction accuracy.