Pairwise Relationships Prediction System

Hanzhong Wang[*], Xinnan SHEN[†], and Ziyue WANG[‡]

Team Name: group8

School of Computing and Information Systems, University of Melbourne

{***hanzhongw,xinnan.shen, ziyue2***}**@student.unimelb.edu.au**

1 INTRODUCTION

1.1 Background

Nowadays, pairwise relationships are used to store many forms of data. However, when we build the pairwise relationships, some edges between nodes are inevitably absent in the graph, which might mislead others a lot. Therefore, finding a way to predict the edges between nodes is an important step. In this project, we have implemented a system to predict the pairwise relationships between nodes by using machine learning tools. Given the nodes and some edges in the graph, the system can predict whether there is a relationship between them. The results have shown that the machine learning model works well in predicting the edges between nodes. Although there are some errors, the model still has great performance.

1.2 Related Works

Many types of research related to pairwise link prediction have been conducted. For example, Liben-Nowell has created a relationship prediction model that can be used in social networks (Liben-Nowell and Kleinberg, 2007). Also, Zhou et al. have mentioned that they have adopted machine learning tools to predict the relationships between people (Lü and Zhou, 2011). In our project, we will also use machine learning methods to make predictions. Meanwhile, some researchers believe that deep learning is a strong tool that can be used to predict relationships (Wang et al., 2019), which has also been adopted in our model.

2 DATA SAMPLING AND FEATURE ENGINEERING

2.1 Data Sampling

The original training data “train.txt” contains 20,000 records of user-follow from Twitter, with a total of 4,867,136 nodes(users). Due to oversized experimental data, in this experiment, we only randomly selected 50,000 edges (source node, target node) from the original data as the true edges (positive sample) of the experimental training data, and randomly selected 50,000 edges (source node, target node) that do not exist in the original training data as the false edges (negative sample) of the experimental training data.

2.2 Feature Engineering

We first created an undirected graph using all of the given data and then generated multiple features of train data and test data based on the graph: resource allocation Index, Jaccard coefficient, and Adamic-Adar index, and preferential attachment score. In addition, according to the follower-followee relationship of the data nodes, we built a directed graph in which the source node points to the sink node. We predicted that if user M follows A, B, and C, and N is very similar to A, B, and C, we assume M may follow N. Thus, we generate two features based on the directed graph: cosine similarity and Jaccard similarity coefficient of the sink node and sinks list of the source node, which will help increase the accuracy of prediction (Jaccard Similarity of followees, Cosine Similarity of followees).

[*] 1029740

[†] 1051380

[‡] 1014037

3 APPROACHES AND RESULTS

3.1 Logistic Regression

Logistic Regression is suitable for binary classification problems, which we have treated as a baseline for this system (Menard, 2010). We have divided the processed data into training set and validation test (F. et al., 2011). To prevent the model from overfitting, we have used validation set not only to tune hyperparameters, but also to see the performance of the model in a very different dataset. Based on the performance of the model in the validation set, we have chosen the hyperparameters with the greatest accuracy to train the data and make predictions. The performance of Logistic Regression model can be seen in table 1.

From table 1, we can see that the logistic regression does not have a very good performance, but it is suitable to act as a baseline for this system.

Table 1. Logistic Regression Result

Precision	Recall	F-1 Score
0.99	0.26	0.41

Table 2. KNN Result

Precision	Recall	F-1 Score
0.62	0.64	0.63

3.2 K-Nearest Neighbour

We have also used KNN classifier to try to improve the performance of the model (Rebala et al., 2019). Similar to what we have done in Logistic Regression, we have also divided the data into training set and validation set. By using scikit-learn tools (F. et al., 2011), we can train the model and predict the results more easily. Table 2 has shown the results of K-Nearest Neighbour model. It is evident that KNN performs a little bit better than Logistic Regression, as the F-1 score is much higher. However, we still need to use some more powerful models to enhance the performance even further.

3.3 Deep Learning

In this project, we have also used deep learning to generate predictions. We have tried to use fully connected neural networks, and the activation function of output layer is sigmoid. The loss function is binary cross-entropy, which is suitable for binary classification problem (Martín Abadi et al., 2015). In particular, we have used the validation dataset to tune hyper-parameters so that we can select the model with the best performance to predict the final results.

For DNN, we have tried models with 2, 3 and 4 hidden layers. The results are shown in figure 1. From the graph, we can see that the performance of DNN has improved a lot, much better than Logistic Regression and KNN. In addition, the model with 2 hidden layers has the greatest F1-score in the validation set (82%). Therefore, it performs best among all the models, and we have chosen this model to produce the final predictions.

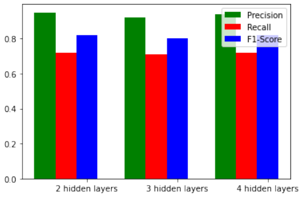

Figure 1. Deep Neural Networks Result

4 CRITICAL ANALYSIS

The factors resulting in the accuracy of prediction include the problems within the dataset, improper feature input, and the choice of algorithms. The problems within the dataset are mainly due to the fact that the dataset is too large and the sink list of each source from the training set is not balanced. In order to solve this issue, we have sampled and analyzed the data from the training set, and select as much data as possible. When selecting experimental samples, the source node is first randomly selected. For positive samples, the sink node is randomly selected in the sink list of this node; for negative samples, the node which is not in the sink list of this node is randomly selected as the sink node, so that the training data is more representative. However, there are still some limitations. For feature selection, we used the experimental results to analyze the effect of each feature on the training accuracy and AUC of the model, and thus chose Jaccard Similarity of followees which is based on directed graphs, and Resource Allocation Index as well as Adamic-Adar Index which is based on undirected graphs, as features for the data training. This is because by comparison of the accuracy of the results, Jaccard similarity is more appropriate for the data than Cosine similarity. Moreover, the accuracy of the positive data prediction greatly depends on the sampling data, which may lose the information of the remaining data. Meanwhile, the method of generating negative data is to randomly choose the source node in the training set and the node in the sink list that does not exist in selected source node as the new sink, which might lead to improper data selection. Finally, due to improper data selection, our model may have overfitting problems, which will affect the accuracy of prediction results.

Logistic regression is a binary classification method. Compared with the naive Bayes method, logistic regression does not need to consider whether the features are related. Compared with the decision tree and SVM, logistic regression easily be updated with the generated data by using online gradient descent. However, the dataset is too large and the generated feature space is very large in this project, which results in poor performance of logistic regression. The model is likely to suffer from the underfitting problem and cannot handle a large number of multi-type features or variables well.

In this project, the performance of the KNN model is not very good, as this training set is large and each sample is not balanced. For example, some source nodes have very few nodes in the sink list, while others have quite many nodes in the sink list, which results in imbalance distribution of data samples and a relatively large deviation. We have achieved better accuracy by using deep neural networks, as this method performs better than LR and KNN in feature space. By using deep neural networks, we try to reduce the complexity of the neural network and focus on training the model with positive and negative characteristics generated by data sampling. Parameters such as the number of hidden layers can be adjusted to better capture the nature of input features and avoid overfitting.

5 CONCLUSION AND FUTURE DIRECTIONS

5.1 Conclusion

Pairwise relationship is a useful relationship to store a number of data in real life. In this project, we have implemented a system to predict pairwise relationships, and the system can predict satisfactory results.

5.2 Future Directions

Although the system can generate satisfactory results, with high F1 score, its performance can be further improved by considering some other aspects. Firstly, feature engineering. It is worth trying to generate more complex features from the training data. Secondly, it is also a good idea to consider Convolutional Neural Networks, as it might have better performance. Finally, other ML tools might help. We could try some other machine learning methods, like SVM.

REFERENCES

F., P., Varoquaux G., G. A., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Liben-Nowell, D. and Kleinberg, J. (2007). The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7):1019–1031.

Lü, L. and Zhou, T. (2011). Link prediction in complex networks: A survey. *Physica A: statistical mechanics and its applications*, 390(6):1150–1170.

Martín Abadi, Ashish Agarwal, P. B. et al. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Menard, S. W. (2010). *Logistic regression : from introductory to advanced concepts and applications.* SAGE.

Rebala, G., Ravi, A., and Churiwala, S. (2019). *An introduction to machine learning.* Springer Engineering eBooks 2019 English+International. Springer.

Wang, W., Wu, L., Huang, Y., Wang, H., and Zhu, R. (2019). Link prediction based on deep convolutional neural network. *Information*, 10:172.