COMP90051 Project 1

Team 13: Tianjing Ruan, Jingdan Cui, Jingyu Li

1. Introduction

Social networks can be used to describe the actual network in the real society, which includes the social relationship between people, the predation relationship between species, the cooperative relationship in scientific research, etc. A large number of studies have shown that various social networks in the real world have many common structural features, such as small-world nature, scale-free nature, and community structure. Link prediction in the network refers to how to predict the possibility of a connection between two nodes that have not yet connected in the network through information such as known network structure. This report will introduce data generation about positive and negative samples, then focus on feature engineering and model selection for link prediction. At last, it will analyze the result and expect for the improvement in the future.

2. Dataset

The dataset used in this project is collected from a Twitter social network, which can be represented as an incomplete network graph that contains 4867136 nodes and 24004361 directed edges. The directed edges represent the actual relationship between nodes, which can be regarded as positive sets. For negative sets, we pick nodes from known source nodes to random nodes in the graph to find pairs that do not have an edge in between and try to keep the size of the two sets on the same level. After collecting all the edges, we split them into train and validation dataset to observe the performance of the models. We notice that as the training data size grows, the accuracy of models also increases. The final model we used contains around 500000 real and fake edges.

3. Feature Engineering

This section mainly introduces the features used in the classification model. The feature sets can be divided into two types: network structure features and Twitter account property features. The network structure features are extracted from the topological structure of the graph which was introduced by Liben-Nowell and Kleinberg (Liben-Nowell and Kleinberg, 2004). We also extract the features of Twitter accounts based on Twitter properties, since the features of the network structure alone are not enough to reflect the relationship between Twitter users. Detailed description are explained below.

3.1 Feature Extraction

As mentioned above, the topology structure of the Twitter network can be regarded as a directed graph structure. We refer to the out-degree nodes as “source” and the in-degree node as “sink”. The directed edge information is stored into two forms: two dictionaries storing inward and outward edges for each node and a Networkx graph representing the network. Features are mainly extracted from these two data structures. Dictionaries are easy to understand and compute while Networkx provides many generator functions and facilities to read and write graphs in different formats. In this case, edge measures such as shortest path between two nodes can be easily calculated by pre-defined Networkx functions.

The following candidate features are initially decided to be used to train the model, the topological features are introduced by Bliss, Frank, Danforth, and Dodds(Bliss, Frank, Danforth and Dodds, 2014), and the user account features are indicated by Fu and Chen (Fu and Chen, 2014):

- Topological Structure Features: Common Neighbor, Union Neighbor, Preferential Attachment, Adamic-Adar Coefficient, Jaccard’s Index, Salton Index, Sorensen Index, HPI, HDI, LHN-I, and Shortest Path.

- Twitter Account Features: User Follow, User Fans, and Opposite Direction Friends.

3.2 Feature Selection

After doing several validation tests, we notice some features such as “Source Fans” and “Sink Fans” would reduce the accuracy of the model, which was mainly caused by extreme values. After filtering these features, the following features are used as the final features to train the model.

Features		**Description**
Common Neighbor		The larger the number of people that followed by both sink and source, the higher probability they may know each other. $\|\Gamma(x) \cap \Gamma(y)\|$
Union Neighbor		Measures the sum of follows who are separately followed by source and sink, which indicates the activity level of the user. $\|\Gamma(x) \cup \Gamma(y)\|$
Preferential Attachment		Individuals with more follows and followers are more likely to have the intention of forming relationship between each other. $\|\Gamma(x)\| * \|\Gamma(y)\|$
Node Similarity		Describe the similarity between source and sink. The lower the similarity between the two users, the less probability they know each other.
	Adamic-Adar Coefficient	$\sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\log \|\Gamma(z)\|}$
	Jaccard’s Index	In the following formulas, ‘k’ is the edges of nodes. $\left\|\frac{\Gamma(x) \cap \Gamma(y)}{\Gamma(x) \cup \Gamma(y)}\right\|$
	Salton Index	$\frac{\|\Gamma(x) \cap \Gamma(y)\|}{\sqrt{k_x \times k_y}}$
	Sorenson Index	$\frac{2\|\Gamma(x) \cap \Gamma(y)\|}{k_x + k_y}$
	HPI	$\frac{\|\Gamma(x) \cap \Gamma(y)\|}{min\{k_x, k_y\}}$
	HDI	$\frac{\|\Gamma(x) \cap \Gamma(y)\|}{max\{k_x, k_y\}}$
	LHN-I	$\frac{\|\Gamma(x) \cap \Gamma(y)\|}{k_x * k_y}$
Shortest Path		The fewer edges between source and sink, the more likely they are to know each other.
Opposite Direction Friends		If a user follows you, then you will probably also follow him/her.

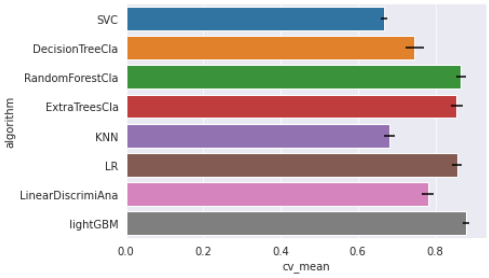
4. Results and Analysis

4.1 Methodology

LightGBM (Light Gradient Boosting Machine) is a distributed gradient boosting framework based on decision tree algorithm. It has good performance in dealing with large data sets and correlated features. On the other hand, logistic regression is sensitive to collinearity. However, the accuracy performance on our validation set is still lower than lightGBM after removing correlated features based on heat map. Compared with other models

such as Random Forest, SVC, Logistic Regression, we find that LightGBM has the lowest mean and standard deviation of AUC score with k-fold validation, which becomes our final choice.

	cv_mean	cv_std	algorithm
0	0.668167	0.008802	SVC
1	0.746020	0.023748	DecisionTreeCla
2	0.865522	0.012560	RandomForestCla
3	0.854920	0.014962	ExtraTreesCla
4	0.680933	0.014313	KNN
5	0.856216	0.012741	LR
6	0.780639	0.014711	LinearDiscrimiAna
7	0.879397	0.009557	lightGBM



4.2 Parameter Tuning

We select LightGBM to go on parameter tuning for the finial performance of this project. For LightGBM, parameter max_depth and num_leaves should be set properly. Large max_depth or num_leaves larger than 2^(max_depth) may lead to overfitting. Learning rate is set to speed up convergence. Feature fraction is used randomly selecting a certain ratio of features into the model to prevent overfitting L1 and L2 regularization can also be used to avoid overfitting. Use a large n_estimators to further optimize the score. GridSearchCV algorithm is used to perform exhaustive search over specified parameter values. Finally, an adjusted LightGBM model has a better performance for link prediction.

4.3 Result and Error Analysis

Validation Set AUC Score	0.956
Test Set AUC Score	0.834

The score obtained from the test data set is about 12% lower than the score from the validation dataset, which indicates that the model might be overfitting. Several reasons may cause the model overfitting. Firstly, half a million training instances may be too little, comparing 20 million pieces of total data. Secondly, the selected hyperparameters may not be the best parameters for the model. Last but not least, the noise data in the training dataset is so large that the model has over-remembered the noise data, but neglected the real relationship.

5. Conclusion

In this report, we use several topological network structure and Twitter users features to generate dataset, use Holdout to split data set, and use LightGBM to learn and predict result. Future work to increase accuracy is concluded as follows: try neural network algorithm to find intermediate layer features; train with more data; improve model hyperparameters.

6. References

Bliss, C., Frank, M., Danforth, C. and Dodds, P. (2014). An evolutionary algorithm approach to link prediction in dynamic social networks. *Journal of Computational Science*, 5(5), pp.750-764.

Fu, Y. and Chen, Y. (2014). *Relationship Analysis Of Microblogging User With Link Prediction*. [ebook] Fujian Provincial Key Laboratory of Networking Computing and Intelligent Information Processing. Available at: <http://gb.oversea.cnki.net/KCMS/detail/detail.aspx? [Accessed 17 September 2020].

Liben-Nowell, D. and Kleinberg, J. (2004). *The Link Prediction Problem For Social Networks*. [ebook] Association for Information Science & Technology. Available at: <https://www.cs.cornell.edu/home/kleinber/link-pred.pdf> [Accessed 17 September 2020].