COMP90051 Statistical Machine Learning

Project 1: Who are my friends?

GooseHead

Summary

Machine learning techniques have been widely used in various fields. In particular, social networking service providers use the methods to recommend potential friends based on their current networks. In this project, we implemented machine learning models to identify existing links from a partial crawled Twitter dataset. K-Nearest Neighbours (KNN), Soft Vector Machine (SVM), Random Forest (RF), and Multi-Layer Perceptron (MLP) algorithms were considered. According to their performance, we selected the best model, MLP, and analysed its performance result with respect to Area under the ROC Curve (AUC) score on unseen test data.

Data and Features

Used data consists of follower lists of 200,000 Twitter users. We generated a network graph and extracted features from random 50,000 existing links as well as 50,000 from non-existing links; users and their relationships are represented as a node and directed edges, respectively. We used 100,000 samples for training and validation purposes. Another 2,000 unseen data was used on the final evaluation. Other than the features in Table 2, we also used the (inner) number of followers and followees.

Table 1. Generated graph description

Number of Nodes	Number of Edges	Avg. In-degree	Avg. Out-degree
4867136	23946602	4.9201	4.9201

Table 2. Applied features

Features	Description
Common Neighbours (CN_{xy})	The number of common neighbours of two nodes (x, y)
Common Neighbours and Distance (CND) [1]	If $\lvert\Gamma(x) \cap \Gamma(y)\rvert = \emptyset, \frac{(CN_{xy}+1)}{2}$. Otherwise, $\frac{1}{d_{xy}}$
Jaccard Index [2]	$\frac{\lvert\Gamma(x) \cap \Gamma(y)\rvert}{\lvert\Gamma(x) \cup \Gamma(y)\rvert}$
Common Neighbour and Centrality based Parameterized Algorithm ($CCPA$) [3]	$\alpha \times \lvert\Gamma(x) \cap \Gamma(y)\rvert + (1 - \alpha) \times \frac{N}{d_{xy}}$ when $\alpha = [0.5, 0.9]$ and N = the total number of nodes
Shortest Path (SP, d_{xy})	The shortest distance between two nodes. If there is no path between two nodes, it returns double the maximum SP

Approaches for Link Prediction

- K-Nearest Neighbours (KNN)

KNN algorithm was our first approach to find the similarities between links since it is the most generally used simple classifier. The algorithm could label the targeted unknown links correctly when there exists known links. However, its prediction result was significantly bad; the AUC of the test-public data was around 55~60 %. Moreover, KNN faced the overfitting issue as well. The model does not selectively choose important features itself, and it is sensitive to noisy data and outliers. According to our analysis of the used features, there were plenty of outliers that range widely (see Table 3). Therefore, KNN could not be comparable.

- Soft Vector Machine (SVM)

To reduce the sensitivity to outliers and overfitting problems, we implemented SVM with optimal hyperparameters. There are two adjustable regularisers on SVM; slack penalty C and γ. They smooth hard-margin on Radial Basis Function (RBF) kernel to get around overfitting. We set them to the best parameters by the greedy searching method. However, regardless of the two values, it always showed over 95 % accuracy on validation (see Figure 2), but poorly predicted unseen test-public data with its best hyperparameters.

The algorithm also could potentially overfit to training data due to its complexity if there are many attributes. Hence, we decided to simplify the model by feature selection. Unlike embedded selection algorithm models such as Decision Tree, SVM requires more sophisticated raw data preprocessing prior to the learning phase. In terms of feature selection, there was a concern about ill-posed problems at the beginning of the model implementation because the majority of the extracted attributes have a similar conceptual background; the relation between common neighbours and how far they are away. Due to the problem, the model could provide multiple solutions which may not be the optimal one. Therefore, L1 regularisation with a lasso penalty was applied to remove less informative features. It returns close to zero for irrelevant attributes by shrinking down its coefficients according to the assumption of linear dependency between input and output values. However, lasso regression supported only *SP* with high regression coefficient vector α and three features with low α. With the 3 attributes, the SVM model could not effectively distinguish differences between linked and non-linked dataset during training. We believe that the critical limitation, Lasso cannot capture nonlinear dependency, which might cause poor performance.

- Random Forest (RF)

RF was applied to tackle previously observed problems. It is one of the ensemble learning algorithms in the field of machine learning. The concept is a combination of multiple Decision Trees (DT). However, its properties are quite different from the DT. RF not only averages the value of multiple Decision Trees but also randomly utilizes different parts of the same training set, while DT can easily suffer from overfitting issues because of bias for highly-branching features. Thus, RF could lower the variance and prevent overfitting. Also, the model structure embedded implicit feature selection. For these reasons, we implemented it as another attempted approach.

However, the results of validation and prediction were not satisfying. The validation result showed that the model is still overfitting the training set even though it is not prone to overfit. We consider that it is because of some significant differences in feature scales for positive and negative links. We will discuss it in the further section of the report. On the other hand, the accuracy of the prediction for the test-public dataset could not increase over around 0.5~0.65 % no matter how we tuned parameters. We found three reasons that decreased accuracy. First, the bias might increase because RF learned from different parts of the same dataset. We can see that the predicted probabilities are distributed mostly in the range from 0.9 to 1.0 (see Figure 1). Yet the labels of the testing set should be 50% true and 50% false. Lastly, the attributes might not be informative enough to predict the potential links. Feature interaction should also be considered. Besides, some features might become potential noises.

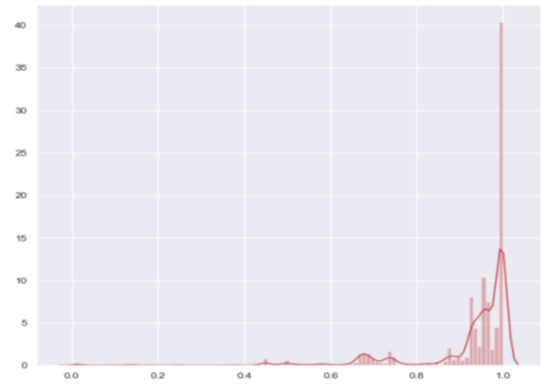


Figure 1. Distribution of predicted probability

- Multi-Layer Perceptron (MLP)

So far, we found that the link could not be directly predicted by examining a few combinations of features applied in other approaches. Interaction between features should be considered more sophisticated. Therefore, our final model, MLP, was implemented to enhance the impact of complex interactions which the other models might overlook. In particular, MLP has considerable merits on automatic data scaling and sensing hidden relationships between features. It may mitigate some human-made errors during manual data pre-processing by applying separate weights depending on the important information each feature has. We hypothesised that MLP could learn the hidden interaction with automatically scaled features by iteratively updating weight through gradient descent. As a result, the experiment result proved the hypothesis was correct; the AUC on the test-public set increased up to 74 %.

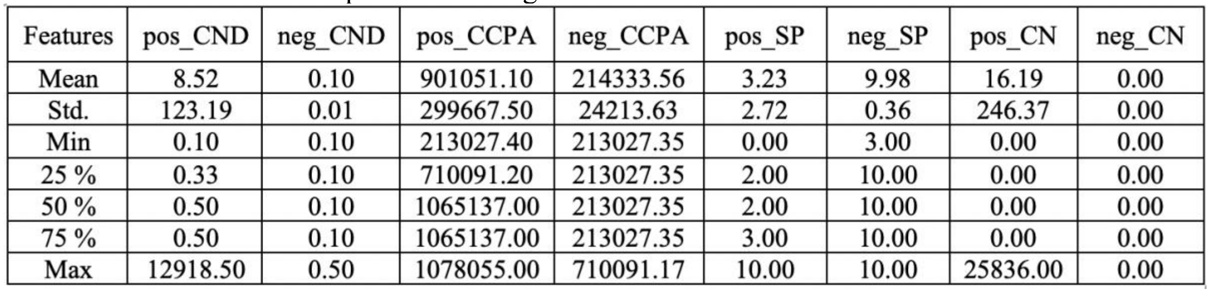
Challenge

During the project, overfitting was the main challenge we faced. Even though there were improvements in prediction performance by adopting the MLP approach, the models were easily overfitting in the first few training stages regardless of the algorithms applied. Our observation is that the main reason behind was the fed features rather than attempted machine learning algorithms. We could find significant differences in feature values for positive and negative links (see Table 3 and Figure 4). In particular, the patterns and density of distribution notably vary in positive and negative links in *CCPA*, *SP*, and *CND*. The models could too easily learn these features only to make predictions. Furthermore, the negative links were randomly generated. These values and distributions of features might be unrealistic to apply in the real network, for example, the test-public set. It could explain the idealistic result from the training and validation phase (see Figure 3). Besides, the sampled data size was too small to use. In general, machine learning models learn and predict better with larger data. Especially, MLP is a data-hungry model. However, our models only learned from a relatively small size of data. Under the shortage of hardware resources, RAM and CPU, we managed to extract only 4% of the links out of the whole dataset from the graph.

Conclusion

In this project, we built several prediction models to identify an existing relationship between two users. The final approach, MLP, formed the best, whereas other algorithms perform relatively poorly. We showed our analysis for each model by comparing its advantages and shortcomings for its properties. According to the overall observation, we concluded that the major problem causing bad performance on the real dataset was the generated feature sets. To be more specific, not only the unrealistic negative links and the significant differences in feature values but also the small sample size.

Table 3. Difference between positive and negative links' features



Features	pos_CND	neg_CND	pos_CCPA	neg_CCPA	pos_SP	neg_SP	pos_CN	neg_CN
Mean	8.52	0.10	901051.10	214333.56	3.23	9.98	16.19	0.00
Std.	123.19	0.01	299667.50	24213.63	2.72	0.36	246.37	0.00
Min	0.10	0.10	213027.40	213027.35	0.00	3.00	0.00	0.00
25 %	0.33	0.10	710091.20	213027.35	2.00	10.00	0.00	0.00
50 %	0.50	0.10	1065137.00	213027.35	2.00	10.00	0.00	0.00
75 %	0.50	0.10	1065137.00	213027.35	3.00	10.00	0.00	0.00
Max	12918.50	0.50	1078055.00	710091.17	10.00	10.00	25836.00	0.00

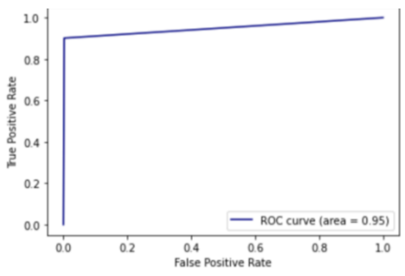


Figure 2. ROC curve result

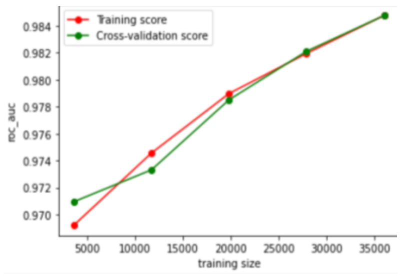


Figure 3. Learning curve result

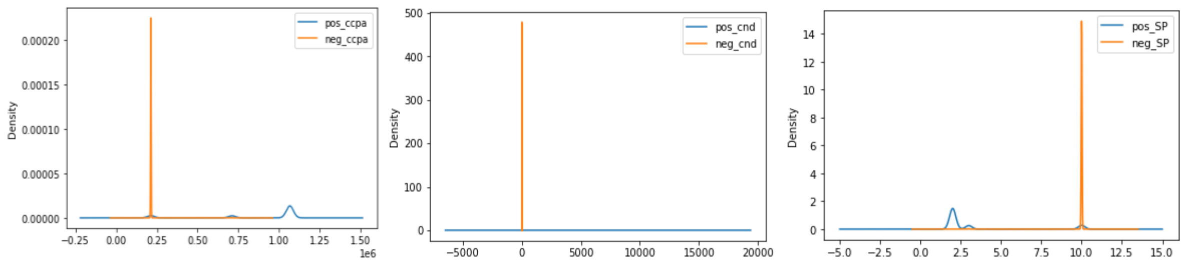


Figure 4. Feature difference between pos and neg links

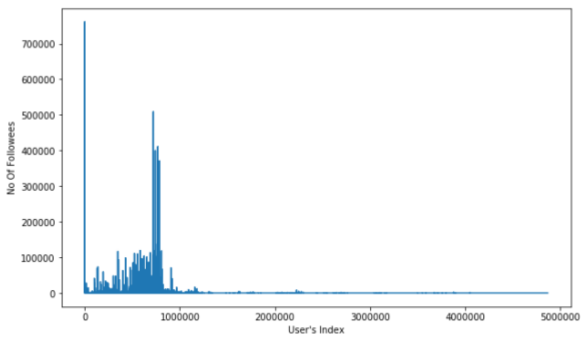


Figure 5. Distribution of the number of followees

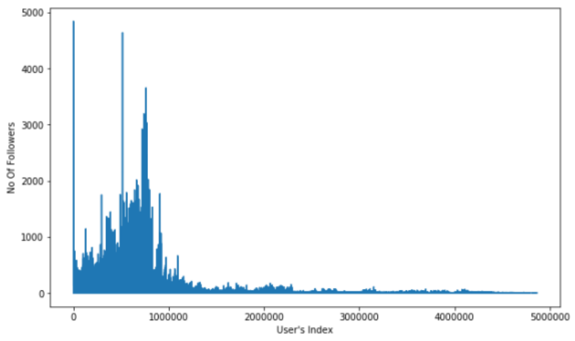


Figure 6. Distribution of the number of followers

Reference

[1] Iftikhar Ahmad, Muhammad Usman Akhtar, Salma Noor, and Ambreen Shahnaz. Missing Link Prediction using Common Neighbor and Centrality based Parameterized Algorithm. *Scientific Reports*, 10(1):364, December 2020.

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

[3] Yang, J. & Zhang, X.-D, Predicting missing links in complex networks based on common neighbors and distance. *Scientific Reports*, 6: 38208, 2016.

[4] C. Xiao, D. M. Freeman and T. Hwa, "Detecting clusters of fake accounts in online social networks", *Proc. 8th ACM Workshop Artif. Intell. Secur.*, pp. 91-101, 2015.