Link Prediction in Graph Dataset

Saurabh Vidyarthi

Problem Statement:

We have been provided with training data (Training network), The training network is partial crawl of the Twitter social network collected several years ago. In this training data a directed edge from node A to node B signifies that user A follows user B. For test data we have 2000 edges, task is to learn training data and determine whether each of these test edges genuinely exists in Twitter network or if they are fabricated so we have posed this problem as classification problem where we return 1 if the edge is genuine and 0 if it is fabricated edge. In Kaggle we have submitted probabilities of edge being genuine edge.

True edge: genuine edge that exist in Twitter social network

False edge: fabricated edge that does not exist in Twitter social network

Sampling Method:

Random sampling method is used where we have sampled 20000 True edges and 20000 false edges (from new train data), method used for generating False edges was, we pick 2 source nodes randomly from given dataset, let's say two users are A and B we perform set difference of followings of user A with user B and select any random edge from user A to resultant set.

Final approach:

We have used Random Forest with bootstrap as our final model with following features Indegree of source node, Outdegree of source node Jaccard coefficient of source node and destination node. We have reached 80.755 accuracy with this model and mentioned features accuracy measure used is AUC (Area Under Curve)

Feature Engineering:

We have experimented with 10 different types of features each of them is explained briefly, our sampled dataset is in edge list format where directed edge between user A and user B indicates user A follows user Here, we are saying user A as source node and user B as destination node if there is directed edge from user A to user B in our sampled data

(i) **Indegree of Source Node**: Number of users, who follows source node in given training data

(ii) **Outdegree of Source Node**: Number of users, source node follows in given training data

(iii) **Indegree of Destination Node**: Number of users, who follows destination node in given training data

(iv) **Outdegree of Destination Node**: Number of users, destination node follows in given training data

(v) **Common Followers**: It calculates number of common followers between source node and destination node, in training dataset we check who all are following source node and who all are following destination node we take cardinality of intersection of both sets.

(vi) **Common Following**: It calculates the number of common followings between source node and destination node, in training dataset we check all user which source node follows and all users which destination set follows and take cardinality of intersection of both sets.

(vii) **Transitive Friends**: It calculates the number of transitive friends, it takes cardinality of set A, set A takes all user which source node follows intersection with all users who follows destination node.

(viii) **Jaccard coefficient**: It is a measure of similarity or overlap between two nodes, it is calculated by taking ratio of number of common neighbours between source node and destination node to number of neighbours after taking union of both source node and destination node neighbours.

(ix) **Opposite direction friend**: It basically finds if there is a directed edge between destination to source, that means does user B follows back user A

(x) **Shortest Path**: Shortest Path between source node and destination node

Useful features which achieved AUC of .80755 are:

- ➔ Indegree of Source Node
- ➔ Outdegree of Source Node
- ➔ Jaccard coefficient
- ➔ Transitive Friends

Features which did not worked out and reason behind that:

- ➔ **Shortest Path** : This features was giving very high accuracy in training data but it performed poor on test data, this feature introduced overfitting, reason for that was, we have been provided with a subset of twitter network data and while we sample data from provided data, all true edges has shortest path as 1 and all false edge has shortest path >1 in our sampled data ,so for model training it becomes very easy to overfit and classify it as false edge if shortest path is >1 but in reality we don't have whole twitter data so even if shortest path is >1 in our sample data, it has chances of being true edge.
- ➔ **Outdegree of Destination Node :** This feature did not provided variation for edges in sampled data, most of the values in this feature for destination node were zero, reason for that is in the provided data we have 20,000 rows that means we have complete information about 20,000 users only but in data we have lot more unique id's than 20,000 which means when we sample data, in our destination nodes there are many nodes whose information we don't have ,by information we mean all users id which this id follows so in most of the cells this feature returns 0.
- ➔ **Indegree of Destination Node:** Same problem appeared with this feature also but here some variation was there as compared to outdegree of destination node.

Alternative Models used and their results:

Feature sets:

Set A = {Indegree of source node, Outdegree of source node, Jaccard coefficient, Transitive Friends, Opposite direction friend}

Set B = {Indegree of source node, Outdegree of source node, Jaccard coefficient, Transitive Friends}

Set C = {Indegree of source node, Outdegree of source node, Jaccard coefficient, Transitive Friends, Opposite direction friend, Shortest Path, Indegree of Destination Node}

MODEL	FEATURE SET	ACCURACY (MEASURE AUC)
Logistic Regression	Set C	0.50852
Logistic Regression	Set B	0.62905
Logistic Regression	Set A	0.69222
Decision Tree	Set C	0.65905
Decision Tree	Set B	0.70816
Decision Tree	Set A	0.73484
Random Forest Classifier	Set B	0.80654
Random Forest Classifier	Set A	0.80755

Set of features which worked well was Set A because we have eliminated the feature which were overfitting or did not had variation in that particular feature and which feature is eliminated and reason why that feature did not work well has been mentioned above in section “Features which did not worked out and reason behind that”

So, we finally went with this set, Set A = {Indegree of source node, Outdegree of source node, Jaccard coefficient, Transitive Friends, Opposite direction friend}

Model which did not worked well and reason for that:

- ➔ **Logistic regression**: Reason why Logistic regression is not working well is because of dominance of some features over other, Indegree of source node and Outdegree of source node has median value in sampled data 132.0 and 2951.0 respectively, now if we observe values in Jaccard coefficient it is very small values as compared to other feature’s value so other feature tends to dominate. One solution is to standardize other features but after standardization also other features dominates as Jaccard coefficient values are very small in nature.

Model which worked well and reason for that:

- ➔ **Decision Tree**: The reason Decision Tree performed better than Logistic Regression is that decision Tree is not affected by the feature which contains large value, that is why Decision Tree does not demand feature standardization, it works on principle of Information Gain.
- ➔ **Random Forest Classifier**: Random Forest classifier with bootstrap max dept 5 along with Set A features achieved AUC of 0.80755 or an accuracy of 80.755 percent.

Problems Faced while working with data:

(i) The modified train data which was provided was not supporting Pandas data frame hence we were not able to load or read data in pandas data frame, reason for that was Pandas has to maintain equal number of elements in all rows and in our train data number of elements in rows were different.

(ii) To Solve this problem, we have used csv module of python, using csv we read the data and pushed that data to 2-D List to perform further operations.

(iii) Due to limitation of hardware resource we sticked to computationally efficient features to calculate, there were more features to explore Page Rank for directed graph, Kartz measure

References:

[1] Fire, Michael & Tenenboim, Lena & Lesser, Ofrit & Puzis, Rami & Rokach, Lior & Elovici, Yuval. (2011). Link Prediction in Social Networks Using Computationally Efficient Topological Features. 73-80. 10.1109/PASSAT/SocialCom.2011.20

[2] W. Cukierski, B. Hamner and B. Yang, "Graph-based features for supervised link prediction," *The 2011 International Joint Conference on Neural Networks*, San Jose, CA, USA, 2011, pp. 1237-1244, doi: 10.1109/IJCNN.2011.6033365.

[3] NetworkX documentation : https://networkx.org/documentation/latest/

[4] csv documentation: https://docs.python.org/3/library/csv.html

[5] ChatGPT-3.5 : https://chat.openai.com/

[6] Bard (From Google)

[7] https://snap.stanford.edu/data/twitter-2010.html