Social Network Link Prediction using Machine Learning Algorithms

AKASH GHOSH* and SUTHRAM VINAY KUMAR*, Indian Institute of Technology, Ropar, India

Abstract: Ever since the existence of Social Networks, one of the most fundamental problems to be solved by social networking companies is how to provide appropriate friend recommendation suggestions to a particular user on their respective platform. In this work, we have studied how this problem could be boiled down to the link-prediction problem. We have studied different features that could be used to solve this problem. Features are from a simple neighborhood based, features based on similarity measures to complex network algorithms like PageRank, hubs and authorities etc. We have used these different features and fed them to well-known Machine Learning algorithms like Random-Forests and Support Vector Machines to solve this as a binary classification problem. We used techniques like Grid-Search CV to get the best hyperparameters for the Machine Learning models.

ACM Reference Format:
Akash Ghosh and Suthram Vinay Kumar. 2022. Social Network Link Prediction using Machine Learning Algorithms. 1, 1 (April 2022), 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

A social network is a social structure made up of a set of social actors (such as individuals), sets of dyadic ties, and other social interactions between actors. The social network perspective provides a set of methods for analyzing the structure of whole social entities as well as a variety of theories explaining the patterns observed in these structures. The study of these structures uses social network analysis to identify local and global patterns, locate influential entities, and examine network dynamics [1]

In this work we are mainly focused in link-prediction task in social-networks. In the binary classification formulation of the link prediction task the potential links are classified as either true links or false links. Link prediction approaches for this setting learn a classifier M_b that maps links in E' to positive and negative labels i.e. $M_b : E' \rightarrow \{0, 1\}$. In the probability estimation formulation, potential links are associated with existence probabilities[2]. Link prediction approaches for this setting learn a model M_p that maps links in E' to a probability i.e. $M_p : E' \rightarrow [0, 1]$

In the present work, we try to solve this problem using various graph mining feature extraction techniques. We use two powerful binary classification algorithms on top of these features. We choose F1-score as the metric for evaluation. F1-score is the harmonic mean of both precision and recall. It penalizes both false positives and false negatives so it's a better metric than simple accuracy. We try to tune various hyperparameters for the model to get the best performance on our dataset.

*Both authors contributed equally to this research.

Authors' address: Akash Ghosh, 2020aim1002@iitrpr.ac.in; Suthram Vinay Kumar, 2020csm1019@iitrpr.ac.in, Indian Institute of Technology, Ropar, IIT Ropar, Ropar, Punjab, India, 140001.

XXXX-XXXX/2022/4-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 MOTIVATION

The main motivation of this project is to explore various graph theory feature extraction techniques which are developed over the years and how they can be employed to perform various predictions on graph data using Machine Learning algorithms.

3 LITERATURE REVIEW/RELATED WORK

The paper [4] by Jon kleinberg discusses many meaningful inferences that are derived from observed network data and understand the relative effectiveness of network proximity measures adapted from techniques in graph theory,computer science and social sciences like graph distance,common neighbours,jaccard coefficient,adar,katz etc. to name a few.

The paper [5] by Aditya Krishna Menon and Charles Elkan uses Matrix Factorization techniques which can be used as a feature for link prediction.

The paper [3] by Jon kleinberg discusses an algorithm named HITS score which can also be used as a feature for any graph theory network for any prediction task.

Our work is to apply Machine Learning algorithms on top of these feature engineering techniques based on graph theory for better prediction on data which can be represented as a graph.

4 METHODOLOGY

The following steps are involved to solve this problem.

4.1 Data collection and Overview

The data collected to solve this problem is taken from the contest Facebook Recruiting competition. Every data point is a pair of numbers which are actually a pair of vertices which are directly connected in the social graph structure. The data set has 9437519 rows and 2 columns. There are 1862220 unique nodes and 9435519 edges approximately in the graph data.

4.2 Data Analysis

We deep dive to analyze the data before going ahead. Visualizing the first 100 rows of the dataset shows that it's a dense graph (Refer: figure 1).

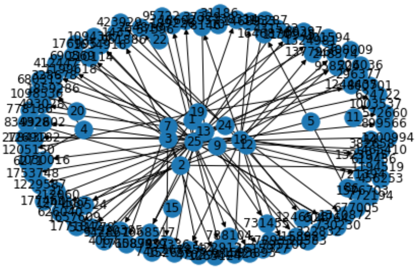

Fig. 1. Network Graph

Then we analyze the followers and followee information in the graph data. We get the insight that the data is highly skewed with respect to the followers and followee parameters. The analysis shows that 99 percentile of people in the network has around 40 followers and the top 1 percentile has around 552 followers (Refer: figure 2). The same case is with the followee (Refer: figure 3), 99 percentile of people follows around 40 people while the top 1 percentile follows around 1566 people.

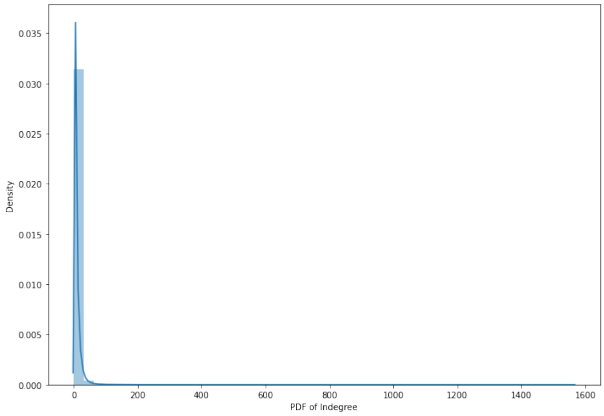

Fig. 2. follower

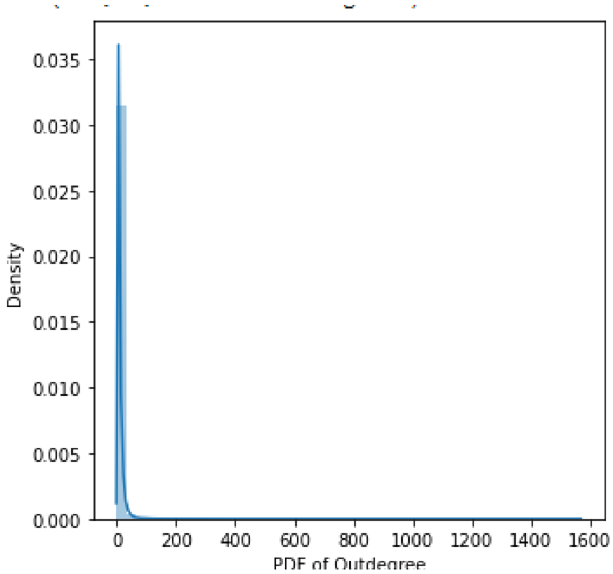

Fig. 3. followee

4.3 Mapping to a Machine Learning Problem

We took around 5 lakh random pairs nodes from the graph and if they have a direct edge between them then the output label is 1. If there is no edge between them then it's 0. So, the problem is now formulated as a binary classification problem and so given the structure of the graph data, the model will learn to predict whether 2 nodes are friends or not(ie there is a direct edge between them or not). 80% of the data is used for training the model and

4.4 Feature Extraction

We studied various features ranging from simple observation based to complex algorithms like PageRank which could be leveraged for this problem. First, we employ simple features which are based on the neighbourhood of the node like the number of followers, number of followees, number of common followers and followees and whether the other person is following or not. Then we explore features which are based on similarity measures like Jaccard Distance. With respect to social network analysis, Jaccard distance is a number which is the ratio of the number of common elements in both sets to the total number of unique elements in the set $J(A,B) = \frac{|A \cap B|}{|A \cup B|}$. Cosine Distance{Otsuka Ochiai Coefficient}: It is represented as $K = \frac{|A \cap B|}{\sqrt{|A| \times |B|}}$ Here, A and B are sets, and $|A|$ is the number of elements in A. If sets are represented as bit vectors, the Otsuka–Ochiai coefficient can be seen to be the same as the cosine similarity. Then we used graph-based features like katz centrality which is defined as the centrality of the node based on the centrality of the neighbour. It is the generalization of the eigenvalue centrality, the Adamic–Adar index which is defined as the sum of the inverse logarithmic degree centrality of the neighbours shared by the two nodes. $A(x,y) = \sum_{u \in N(x) \cap N(y)} \frac{1}{\log |N(u)|}$. We use features like the shortest path, connected components which contain vital info about the position of the nodes in the graph. The intuition is that if the distance between them is less and if they lie in the same connected component then the probability that they will be friends in near future is pretty decent.And finally we use few famous complex network algorithms like the page-rank algorithm and HITS algorithm which computes two numbers for every node namely. Authorities estimate the node value based on incoming links. Hubs estimate the node value based on outgoing links.

5 RESULTS

We use F1-score for evaluation which is the harmonic mean of precision and recall, to make sure it takes into consideration both false positives and negatives into consideration. We use two well-known machine learning models namely Support Vector Machines and Random Forests algorithms for the task. We use the grid-search CV technique for hyperparameter tuning. For Support Vector machines we use C and gamma as hyperparameters. The value of C determines how hard or soft we want the SVM margin to be. Gamma is a parameter which balances bias and variance. A small gamma will give you low bias and high variance while a large gamma will give you higher bias and low variance. Based on hyperparameter tuning , we got the best results with C=100 and gammma=0.001 For Random Forests, we use max_depth and n_estimators as hyperparameters.Max_depth is the parameter which controls the overfitting of the model. High depth tree fits complex hyperplanes but performs poorly in the test dataset. Similarly, n_estimators is the This is the number of trees you want to build before taking the maximum voting or averages of predictions.A higher number of trees give you better performance but makes your code slower. So, we need to tune these parameters for the best possible performance of the model. We use 5-fold cross-validation for each of the cases to check there is no overfitting. The F1-score for both train and test set using the above-mentioned algorithms is given in the table below:

Model	Train	Test
Support Vector Machine	0.90	0.89
Random Forest	0.93	0.91

Table 1: Results

6 CONCLUSION AND FUTURE WORK

So, we study different graph-mining techniques which can be used as features for Machine-Learning algorithms for link prediction problems. Though we got good results there is always the possibility to improve the performance of the models if we can play with other hyperparameters of the algorithms. We want to extend the project and check if we can get better results after playing with more hyperparameters. Also, we want to apply cutting-edge graph neural networks for this problem and compare their performance relative to our existing models.

REFERENCES

[1] https://en.wikipedia.org/wiki/Social_network.

[2] https://en.wikipedia.org/wiki/Link_prediction.

[3] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, sep 1999.

[4] David Liben-nowell and Jon Kleinberg. The link prediction problem for social networks. *Journal of the American Society for Information Science and Technology*, 58, 01 2003.

[5] Aditya Menon and Charles Elkan. Link prediction via matrix factorization. pages 437–452, 09 2011.