1. Introduction：

The main task of this experiment is to do the link prediction in social networks by providing abstract twitter data. Use the appropriate model to determine whether the link between two points is real or faked. The report will firstly introduce the dataset provided in this project. Then, the report will demonstrate the methodology of doing this project. Next, the report will compare different models and demonstrate the reason for choosing models. Finally, the report will discuss the limitation of this project and illustrate the future work.

1. Data set

The data sets provided in this project are highly abstract. Each row of data only shows the user and his list of followed users Data is provided as an adjacency list. Therefore, the data set only provide the topology relation between nodes. Based on the data, we can construct a directed graph to represent the concern relationship between nodes. The number of source node is 20000. The total number of following nodes is nearly 4.86 million. The source node's collection is completely included in the following node's collection. The data provided by the test data set is also included in the 4.86 million sets.

1. **Feature selection**

**n Features**

To solve a link prediction problem, a lot of generic link prediction metrics can be taken into use. Most of them are usually based on information of nodes or topology [].

**n.1 Node-based Metrics**

It is believed that the probability of two people to be friends grows with the increase of their similarity since people tend to communicate with those ones who are in similar age, have similar background or similar hobbies. But, in this problem, there is not any textual information about each node which can be treated as node attribute because the dataset is highly abstract. Therefore, it is not suitable to do feature extraction based on information of nodes.

**n.2 Topology-based Metrics**

Given that there is not any direct information of nodes, an alternative way to find out whether two people will be connected is to focus on the topological information. The prediction metrics can be built on the graph structure features. Nowadays, a lot of proven topology-based metrics have been proposed which are based on the idea that people tend to build relationship with the ones who are close to them. Most of them directly focus on the neighbors which are most close to a given node. However, this cannot be directly taken in to use since it is not a complete graph that can represent the whole social network in the training dataset. Instead, there are a huge number of nodes which only drift at the edge of the graph which means only the people who follow them are given whereas the nodes they point to are unknown. Based on this information, for every two nodes (x, y) in the graph between which there is a potential relationship that the node x follows the node y, the following twenty-two features are used in the model.

* Followers of *x*:
* Nodes Followed by x:
* Followers of *y*:
* Common Neighbors:

This can be divided into two situations, one is the common nodes of followers of *x* and the followers of *y*: . Another one is the common nodes of nodes followed by *x* and followers of *y*: .

* Normalized Neighbor-based Metrics:

Based on the common neighbors metric, there are several methods to normalize the size. For example, Jaccard Coefficient, which is defined as and , measures the ratio of common neighbors to all the neighbors they have. Salton Cosine Similarity, which is defined as and , shows the cosine similarity of *x* and *y*. Besides, metrics like Sorensen Index, Hub Promoted, Hub Depressed and Leicht-Holme-Nerman are also used in the model.

* Preferential Attachment:

The metric is based on an idea that people tend to follow the one who has a great number of neighbors and usually is a famous person. This is defined as: and

* Weighted Resource Allocation:

This metric is originally proposed by Zhou et al. [2], the idea is to give a punishment to high-degree nodes to suppress their contribution. However, in this model, the metric is slightly adjusted to fit the condition. For each feature value of a neighbor *z*, it is punished more heavily in four situations: 1) *x* follows too many nodes; 2) *z* has too many followers; 3) *z* follows too many nodes; 4) *y* has too many followers;

* Intermediate Nodes between x and y:

This feature focuses on the situation that there is not a path which only takes a single node as step stone to connect x and y which means x and y do not have any common neighbors. For this feature, if the value is greater than 1 and is less than 5, it represents the number of intermediate nodes. If the value equals 6, it means the path between x and y is not found in 4 steps but there is potentially a longer path connecting them. If the value equals 9, it means that it is confirmed that there is not a path between x and y after breadth first search in 4 steps.

1. Model selection

LightGBM:

LightGBM is a gradient boosting framework, using decision tree based on learning algorithm. LightGBM is a distributed model that can improve training efficiency. Other tree-based algorithms grow trees horizontally, while LightGBM grows trees vertically, which means that LightGBM grows in the leaf-wise. It will choose the leaf with max delta loss to grow. When growing the same leaf, Leaf-wise algorithm can reduce more loss than a level-wise algorithm. LightGBM is a decision tree algorithm based on Histogram. It can be optimized based on the sparse feature of histogram. The basic idea of histogram algorithm is to discretize the continuous floating point eigenvalues into k integers and construct a histogram with width k. Discrete values are used as indexes to accumulate statistics in histogram while traversing data. After traversing the data once, the histogram accumulates the required statistics. According to the discrete value of histogram, the optimal segmentation point is searched through. In this project, we called the python LightGBM package. The boosting type is selected as Gradient Boosting Decision Tree. Because the number of feature selected is small. Therefore, the depth of the tree is limited to and the number of leaves should be small, which is in order to prevent overfitting.

other alternative method

1.Exploration of complex topological features

Besides neighbor-based metrics we also considered some complex topological features. For example, the average length of all paths between source node and target node. We are also interested in some unsupervised methods in feature extraction like clustering. A community clustering can be used to get“communities” which are connected-components with low out-degree. This can be achieved by greedily add the node which has enough connections from nodes inside current cluster. This algorithm will eventually converge when there is no longer such a node that have connections to this cluster over threshold. However, both ideas are time-consuming and we may get a bias result due to given dataset

Reference:

[1] Data set, “Kaggle In-class link prediction https://www.kaggle.com/c/iclp18/data

[1] WANG Peng, XU BaoWen, WU YuRong, ZHOU XiaoYu. Link Prediction in Social Networks: the State-of-the-Art. Sci China Inf Sci, 2015, 58: 011101(38), doi: 0.1007/s11432-014-5237-y

[2] Zhou T, Lv L, Zhang Y C. Predicting missing links via local information. The European Physical Journal B, 2009, 71: 623-630