Report (Project Two)

Student Name:Yansong Li
Student ID: s4722166
Student Email:s4722166@student.uq.edu.au

This report will describe the link prediction algorithms implemented like neighbor based algorithms Jaccard and cosine similarity and 'preferential attachment' or tried to but not successfully implemented such as spectral clustering algorithm and their related knowledge, corresponding outcomes.

Task

First implemented algorithm is 'Jaccard similarity', one of neighbor based algorithms which estimates how similar two nodes are based on the size of common neighbors, each pair of nodes can get a similarity score and the score will tell the possibility that they may connect to each other in the future. In the Jaccard algorithm, for each pair of nodes in the existing links, for every node, we find and store all their connected nodes. To calculate the similarity between two nodes, we count their shared contacts n and total contacts from both m, jaccard similarity is calculated by n/m which is simple to implement. The prediction accuracy tested by the validation dataset is 73%.

After the Jaccard algorithm, cosine similarity is also implemented which is quite similar as Jaccard in both implementation and result, the accuracy obtained from cosine similarity is also 73%, the predicted lines and score ranks are almost the same after comparison. The performance for preferential attachment is very poor with accuracy less than 30%. So we may conclude that 73% accuracy is the best performance that neighbor based algorithms can achieve.

Later on, I tried to implement the spectral clustering algorithm because I believe measuring node similarities based on their communities is very promising. Node feature matrix with shape (5240,80) was created from 80 eigenvectors corresponding to smallest 80 eigenvalues of laplacian matrix, then similarity score for each pair of nodes in the validation data is computed, but something may be wrong, 100 nodes with largest scores do not contain any potential link.

At last, I explored node2vec using existing libraries with extra negative samples added to train the classifier, and embedding is created based on random walk. I found that the way to calculate node similarity from embedding(decoder) may greatly affect the outcome. The best decoder here is l2_norm (71%) and the worst is based on average (20+%). And some hyperparameters can also affect the accuracy result such as window size and max walk length,, the best accuracy 71% is achieved by L2 norm decoder, window size 10, p=1, q=0.5, n=10, walk length=80. And also in node2vec, different runs may produce slightly different results with the same settings.

Result

Algorithm	Accuracy(%)
Cosine similarity/Jaccard similarity	73
Preferential Attachment	24

Node2Vec	71
Spectral Clustering	0% not successfully implemented

:

Summary

As a result, Predicting links based on common neighbors of nodes is powerful and very easy to implement, and it also costs much less computation compared with clustering or network embedding but can still produce good results. But node2vec or GCN may be able to discover deeper information/relationships among nodes especially in more complex or larger graphs, thus producing better prediction, in this condition, neighbor based algorithms may have limited performance.

Reference

1. https://stellargraph.readthedocs.io/en/stable/demos/link-prediction/node2vec-link-prediction.html
2. https://towardsdatascience.com/unsupervised-machine-learning-spectral-clustering-algorithm-implemented-from-scratch-in-python-205c87271045
3. https://realpython.com/gradient-descent-algorithm-python/#:~:text=Stochastic%20gradient%20descent%20is%20an,an%20inexact%20but%20powerful%20technique.