CS529

Link Prediction

GROUP NAME : group_name

Arpit Gupta(160101015)

Rajas Bhadke(160101021)

Sweety Dhale(194101052)

Introduction

The objective of this assignment is to understand various link prediction methods (also known as network completion methods) and its application on various real world problems. The assignment is broadly divided into two classes of approaches i.e., supervised and unsupervised. The task is to understand the working principle of these approaches and compare their performances.

Two network datasets were studied for analysis of various link prediction approaches.

- Foursquare Restaurant Review Dataset
- Blog catalog data

Topological Methods (Unsupervised Approaches)

The topological based methods can also be broadly divided into two classes i.e., node proximity and edge proximity. Node proximity based methods exploit node neighbouring characteristics, whereas edge based methods exploit edge neighbouring characteristics. As neighbouring edges are inherently captured by node neighbours, majority of the topological based link prediction methods consider node proximity. The node proximity based methods are further grouped into two: local and global.

- ## Local measures
    - Common Neighbour (CN)
    - Jaccard Coefficient (JC)
    - Adamic Adar (AA)
    - Resource Allocation(RA)
    - Preferential Attachment (PA)

- ## Global measures
    - Katz
    - Rooted PageRank

For performance evaluation of these topological methods, we **considered one fold of the YES edges and entire NO edges as test dataset,** and computed **AUC score** for each of these local and global measures to analyse the performances of these prediction models.

AUC computation: Provided the rank of all non-observed links, the AUC value can be interpreted as the probability that a randomly chosen missing link is given a higher score than a randomly chosen nonexistent link (i.e., a link in U − E). In the algorithmic implementation, we usually calculate the score of each non-observed link instead of giving the ordered list since the latter task is more time-consuming. Then, each time we randomly pick a missing link and a nonexistent link to compare their scores, if among n independent comparisons, there are n' times the missing link having a higher score and n'' times they have the same score, the AUC value is

$$\mathrm{AUC} = \frac{n' + 0.5n''}{n}.$$

If all the scores are generated from an independent and identical distribution, the AUC value should be about 0.5. Therefore, the degree to which the value exceeds 0.5 indicates how much better the algorithm performs than pure chance.

Below is a table summarising the observed AUC scores for both datasets.

Similarity measure	AUC score (Foursquare dataset)	AUC score (BlogCatalog dataset)
Common Neighbour(CN)	***0.9595***	***0.9436***
Jaccard Coefficient(JC)	***0.9693***	***0.7708***
Adamic Adar(AA)	***0.9666***	***0.9481***
Resource Allocation(RA)	***0.9767***	***0.9539***
Preferential Attachment(PA)	***0.8416***	***0.9513***
Katz	***0.9554***	***0.8188***
Rooted Pagerank(RP)	***0.9597***	***0.7885***

Classification Models (Supervised Approaches)

- ### With explicit features

  Represented the edges in a network considering the above similarity estimates. Created both the YES edges and NO edges using the same feature space. In addition to these

features,we added two extra features namely **'in_same_component'** and **'friends_measure'**.

$$friends - measure(u, v) = \sum_{x \in \Gamma(u)} \sum_{y \in \Gamma(v)} \delta(x, y)$$

where we define the function $\delta(x, y)$ as:

$$\delta(x, y) = \begin{cases} 1 & \text{if } x = y \text{ or } (x, y) \in E \text{ or } (y, x) \in E \\ 0 & \text{otherwise} \end{cases}$$

Talking about **Friends-measure** , when looking at two vertices in a social network, **we can assume that the more connections their neighborhoods have with each other, the higher the chances the two vertices are connected.**

$$Cm_{xy}^{Comm} = \begin{cases} 1 & \text{if } x \text{ is in the same community as } y \\ 0 & \text{otherwise.} \end{cases}$$

The same community metric calculates the intuition that people who belong to the same community will probably be linked.

Using the data representation thus constructed, the link prediction problem is a binary classification problem. Thus, investigated the performance of various classification models.

- Naive Bayes
- SVM
- Decision Tree (ID3)
- Random Forest

Embedding Features

Generated low dimensional feature representation using following embedding techniques and repeated the above classification tasks.

- PCA
- SVD
- Deepwalk
- Node2Vec

For deepwalk, we did **10 random walks per source of length 80**. The node embedding was of **128 dimensions**. **Edge embedding for (u, v)** was taken to be the **hadamard product** of node embeddings for u, v. i.e., the numpy vector multiplication of the two embeddings.

For Node2vec also, the embedding was generated in similar fashion as in the deepwalk described above.For node2vec different combinations of **hyperparameters(P and Q)** were tried for P,Q in {0.25, 0.5, 1, 2, 4}. Total 13 combinations were tried, all had almost similar scores.

For performance evaluation of these classification methods, we used **five-folds cross validation** and evaluated the performance of the prediction model using **AUC** (Area Under Curve), **Precision** and **Recall.**

The table below lists the observed scores for the **foursquare dataset**.

Classification method	AUC score	Precision score	Recall score
Naive Bayes	***0.8223***	***0.72797***	***0.9893***
SVM	***0.951474***	***0.9993***	***0.042186***
Decision Tree	***0.5386***	***0.998***	***0.0773***
PCA with SVC (95% variance with 4 components)	***0.96325***	***0.9737***	***0.7794***
SVD with SVC(4 components)	***0.9566***	***0.97388***	***0.7659***
Deepwalk(Hadamard product)	***0.8668***	***0.8609***	***0.6609***
Node2Vec(P=4,Q=0.25)	***0.8671***	***0.8499***	***0.6803***
Random Forest Classifier	***0.96555***	***0.99772***	***0.1005***

Deepwalk with node similarity matrix computed by dot product of node embeddings matrix gave **AUC score 0.8479** and **average precision score 0.8607**.

From **random forest classifier**, which is an ensemble based classification framework, we got the feature importance values for the features used. We used **100 estimators** in the classification task. The table below lists the feature importance values(top 5) averaged over calculations from 5 iterations of 5-fold.

Feature	Importance Value
Katz score	*1.6934*
Rooted Pagerank(RP)	*1.4765*
Resource Allocation(RA)	*0.74710*
Adamic Adar(AA)	*0.47752*
Jaccard Coefficient(JC)	*0.329*

The table below lists the observed scores for the **BlogCatalog dataset**.

Classification method	AUC score	Precision score	Recall score
Naive Bayes	*0.8835*	*0.8138*	*0.9442*
SVM	*0.9558*	*1.0*	*0.0024*
Decision Tree	*0.5022*	*0.9988*	*0.0044*
PCA(with SVC)	*0.9574*	*0.9541*	*0.7361*
SVD	*0.9572*	*0.9459*	*0.7761*
Deepwalk	*0.8206*	*0.824*	*0.6120*

Network Deletion

It is also known as **disappearing link prediction**. Link prediction methods are generally used for network completion tasks. We used link prediction methods for network destruction task as well. We formulated an algorithm for network destruction using link prediction method motivated from the use of similarity scores in the link prediction problems.In short, **dissimilarity index can be induced by smallness of similarity index**. We Investigated the effectiveness of topological methods for this task of network destruction using AUC scores.

Algorithm pseudocode :

1. Compute similarity scores of all edge pairs.
2. Sort these scores from smallest to largest value.
3. The edges with less similarity(i.e., more dissimilar) scores tend to lose the connection.

Effectiveness analysis

To test the effectiveness of these topological methods, we randomly chose some non-existent edges equal to the count of actual edges(= m) in the network, and build the training graph out of these 2m edges and computed the similarity scores of all the edges the train graph according to all those topological methods and computed AUC scores for each of the methods.

The table below lists the observed AUC scores for both datasets.

Similarity measure	AUC score (Foursquare dataset)	AUC score (BlogCatalog dataset)
Common Neighbour(CN)	***0.945***	***0.9566***
Jaccard Coefficient(JC)	***0.9612***	***0.8743***
Adamic Adar(AA)	***0.9473***	***0.9582***
Resource Allocation(RA)	***0.9525***	***0.9456***
Preferential Attachment(PA)	***0.8280***	***0.9487***
Katz	***0.90593***	***0.95734***
Rooted Pagerank(RP)	***0.3835***	***0.1246***

Conclusion

From what we observed, using a supervised method which combines unsupervised methods seems to be more efficient. This was an “easy” network as we got really **high AUC scores**, but depending on the complexity and noise in the graph you are working on, this can be more or less difficult to predict accurately future links. **The gap between supervised and unsupervised scores can also widen with complex networks.**