Link Prediction Approaches Analysis

Ankit Agrawal (194101006)
Deepen Naorem (186155102)
Sujit Kumar (186101107)
Indian Institute of Technology, Guwahati

Abstract

Various link prediction approaches were analyzed and their performance was measured, we used 2 datasets publicly available for our analysis.

Report is divided into 3 sections:

1. Unsupervised Approaches: Various local/global measures listed below are used for this purpose. How good each approach is able to predict link is measured by calculating their AUC score.
   a. Common Neighbor
   b. Jacard Coefficient
   c. Adamic Adar
   d. Preferential Attachment
   e. Katz
   f. Rooted PageRank
2. Supervised approaches : classical machine learning model (Decision Tree, Naive Bayes and Support vector) where used with 5-fold cross validations.

3. Network Destruction: Finally, we have analyzed how we can destruct the network quickly. This section list various measures used by us to destruct the network and how analyzed performance of each measure for network destruction based on how quickly removing links using these methods increases the number of components in the graph.

Keywords: [Click here to add keywords.]

Dataset:

(1) NYC Restaurant Rich Dataset

a) nodes: 2060
b) yes_edges: 58810

(2) BlogCatalog3

a) nodes: 10312
b) yes_edges: 333983

Theme A: Topological Methods

These methods are unsupervised approaches which predicts whether the link will be formed between 2 nodes in future based on some score. Is value of score is above a certain threshold then we can say that a link is most likely to form between the 2 nodes in future.

For Performance measure of topological classifiers, we have taken all the existing edges and selected equal number of non-existing edges randomly and calculated score for all of them. AUC calculated for each method which is used to analyze the performance of a particular method.

1. **Common Neighbor (CN)**

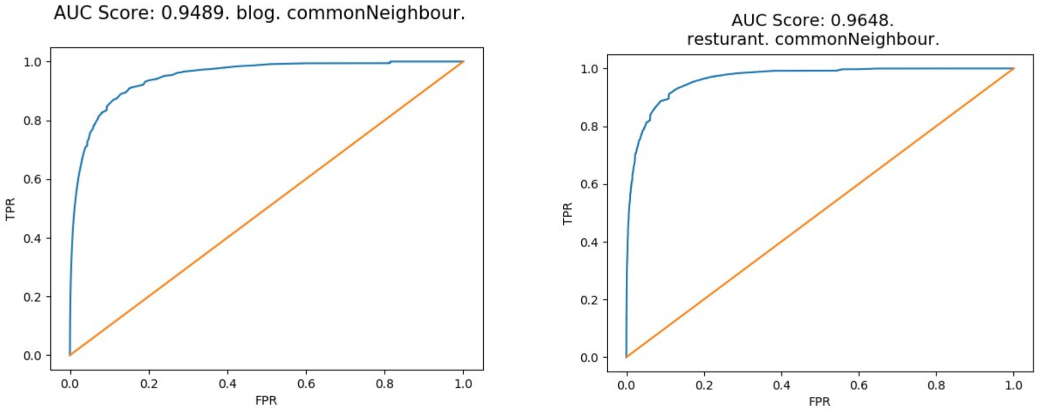


Fig 1: ROC curve

Analysis:

In restaurant dataset if there is a link between 2 nodes it means both persons have visited at least 1 common restaurant. High AUC score of common neighbor means 2 persons are likely to visit a restaurant which is visited mostly by their friends.

Blog dataset also have very high AUC score here it means 94% of the time new friendship will be based on number of common friends.

2. Jacard Coefficient (JC)

It measures the probability that a randomly selected edge (x, y) share a common feature. Here that feature is neighbor nodes.

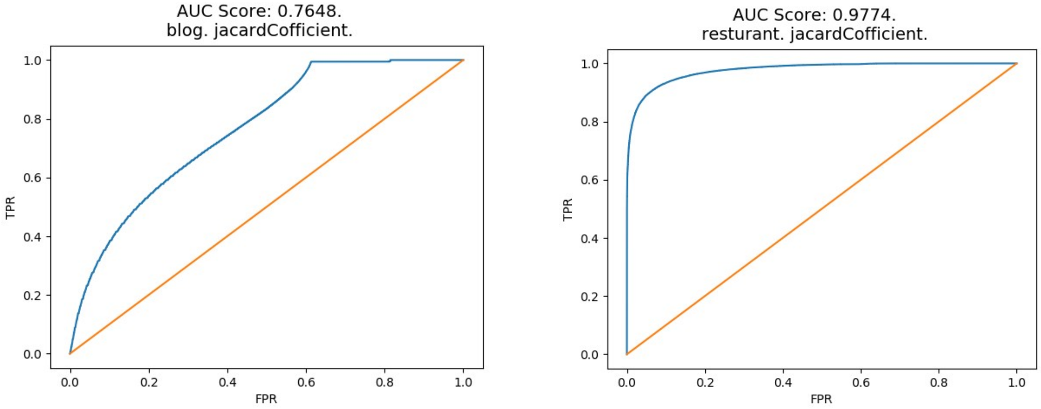


Fig 2: ROC curve

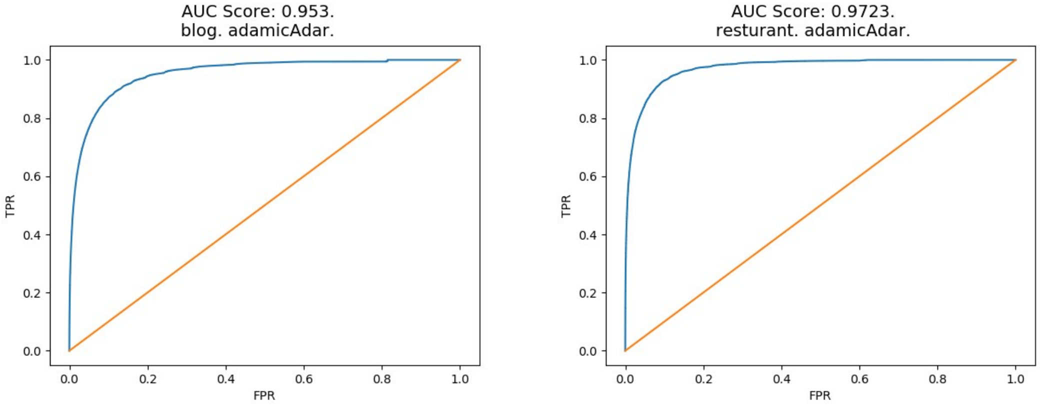
Analysis:

JC is just a normalization of CN, here there is no dramatic change of score in case of restaurant dataset but in case of blog dataset performance has degraded dramatically. This implies that in blogs dataset new friendship is most likely to form between 2 nodes which have high number of common friends but also who do not have a lot of non-common friends.

Meaning AUC score comparison of CN and JC implies that when only denominator part is considered then score of **YES** edges are becoming **low** and **NO** edges are becoming **high**. That implies YES edges are having a greater number of friends than NO edges hence after normalization their scores are falling below NO edges.

3. Adamic Adar (AA)

CN doesn’t account for type of common friends, AA gives more weight to type of common friend which in turn doesn’t have a lot of friends (reserved).


AUC Score: 0.953.
blog. adamicAdar.

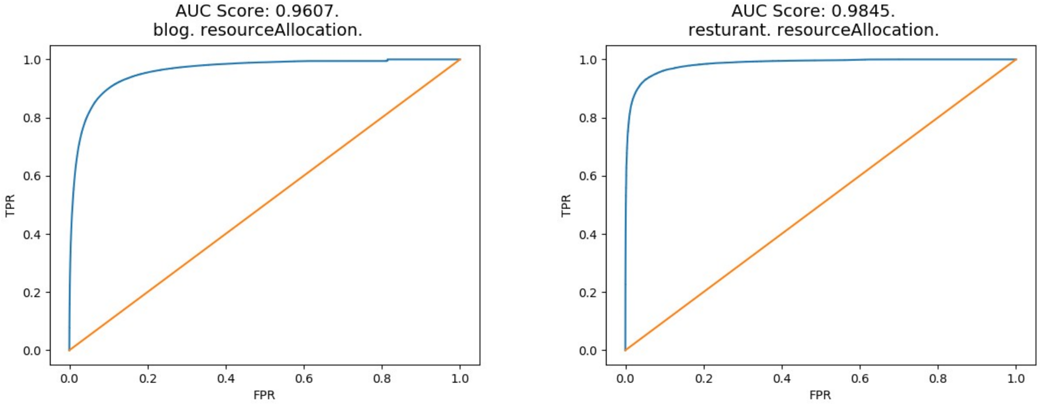
AUC Score: 0.9723.
resturant. adamicAdar.

Fig 3: ROC curve

AA score is able to separate out yes edges from no edges quite well.

4. **Resource Allocation (RA)**

Its similar to AA its just that in some cases RA performs better than AA, while building classifier we can evaluate both of them to decide which one to use.


AUC Score: 0.9607.
blog. resourceAllocation.

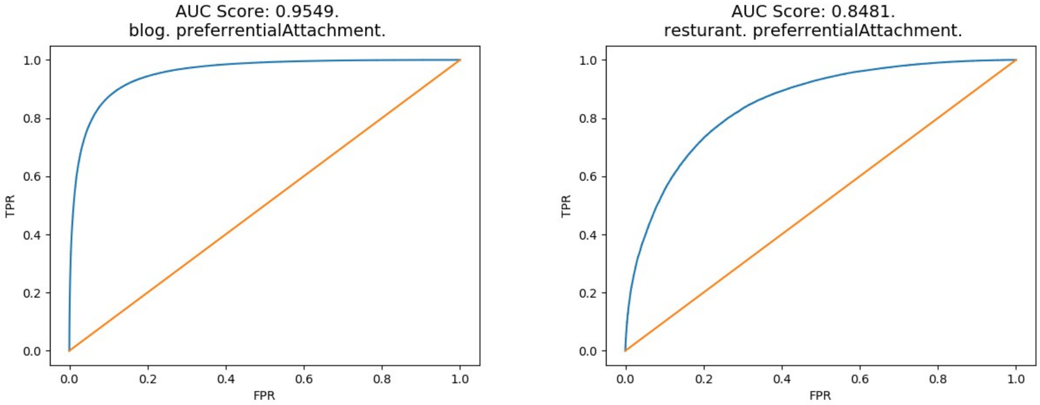
AUC Score: 0.9845.
resturant. resourceAllocation.

Fig 4: ROC curve

In both of our dataset's RA is performing slightly better than AA.

5. **Preferential Attachment**

It says new node is most likely to form between 2 highly popular nodes or 2 nodes with lot of neighbors.



AUC Score: 0.9549.
blog. preferrentialAttachment.

AUC Score: 0.8481.
resturant. preferrentialAttachment.

Fig 5: ROC curve

Analysis:

In blog scenario 2 popular bloggers are more likely to become friends.

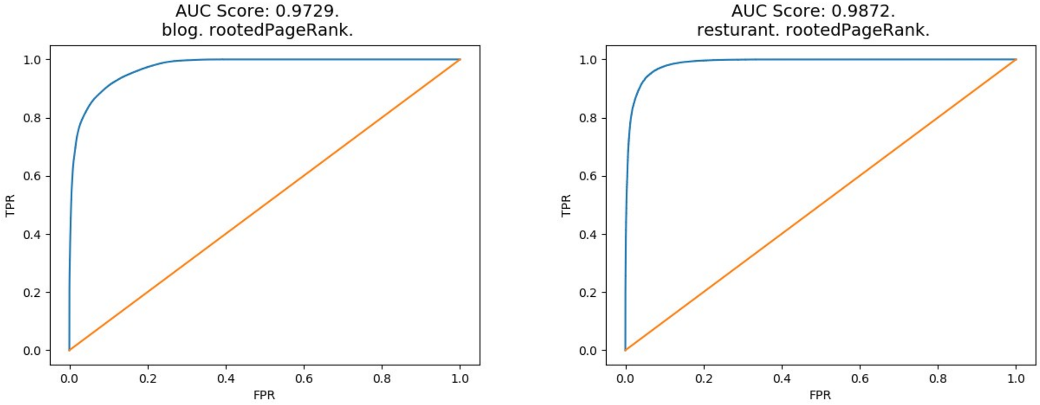
But in case of restaurant dataset where edge between 2 nodes says they have visited a same restaurant PA rather gives poor performance which is quite intuitive because 2 persons may individually have different taste and visit restaurants accordingly **but its highly unlikely that two persons with a lot of neighbors will share same taste and visit same restaurant.** That's why in case of restaurant dataset other methods which incorporated common neighbors' information are performing better than RA.

6. Rooted Page Rank (RPR)

RPR score (x, y) is probability of random walk starting at 'x' and reaching 'y' with probability 'a' and returning to 'x' with probability '(1-a)'.

This is an improvement of Hitting time which is another performance measure. **HittingTime(x, y): Expected step of random walk from x to reach y.**

In hitting time random walk may venture around far away before reaching y, which will add less value, in order to prevent this we use RPR concept.



AUC Score: 0.9729.
blog. rootedPageRank.

AUC Score: 0.9872.
resturant. rootedPageRank.

Fig 6: ROC curve

RPR is a good measure in both the datasets for link prediction. Values says next link is most likely to appear between 2 nodes (x, y) which has higher probability for a random walk to reach from x to y.

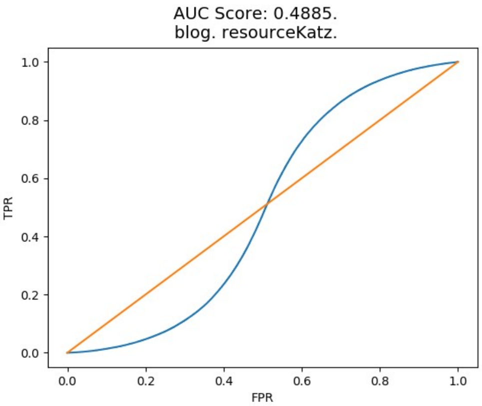
7. **Katz**

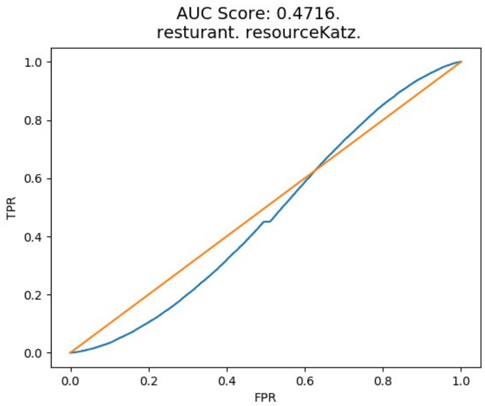
Its another global measure technique which calculates number of paths between 2 nodes to calculate their score. It has a parameter ‘beta’ which is used to penalize the paths which is at higher distances.

If ‘beta’ is very small then paths of length 2 or more will contribute very less towards the score and prediction will be same as CN.

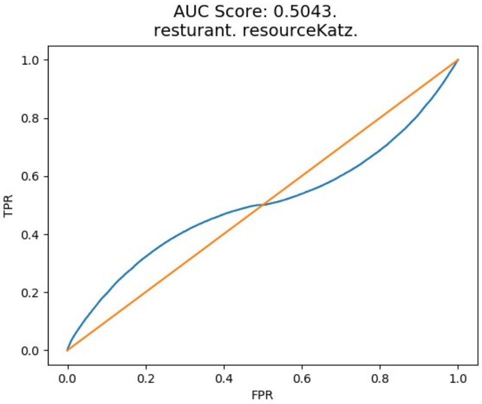
We calculated katz score by varying beta values as depicted below.

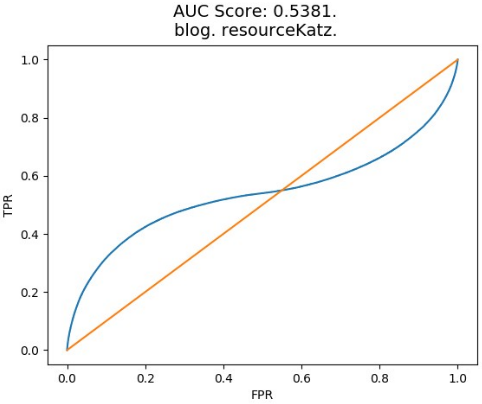
Beta = 0.85


AUC Score: 0.4885.
blog. resourceKatz.

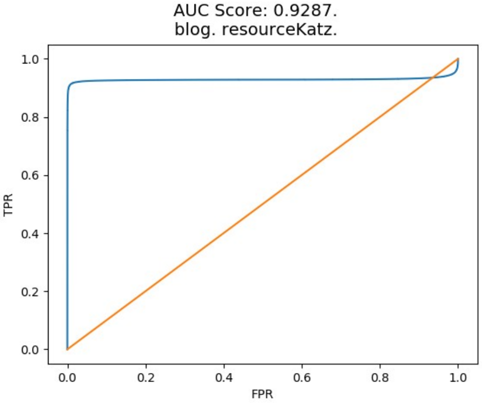

AUC Score: 0.4716.
resturant. resourceKatz.

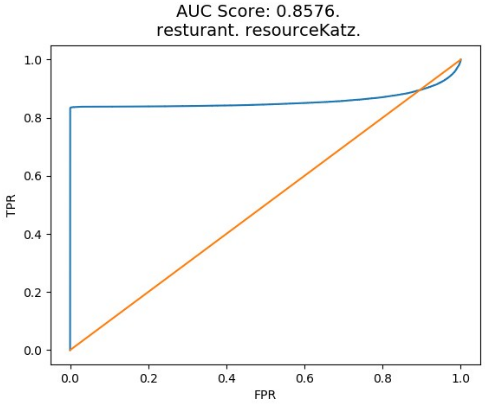
Beta = 0.085


AUC Score: 0.5043.
resturant. resourceKatz.


AUC Score: 0.5381.
blog. resourceKatz.

` Beta = 0.01


AUC Score: 0.9287.
blog. resourceKatz.


AUC Score: 0.8576.
resturant. resourceKatz.

Beta = 0.001

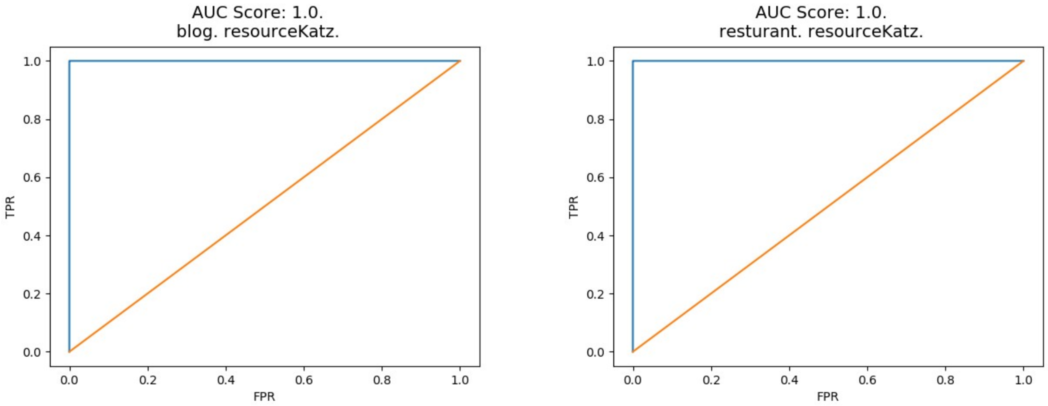


Fig 7: ROC curve

Analysis:

We can see that decreasing beta value is giving better AUC scores for given datasets. This means that nodes which are nearby are contributing more towards similarity measure than the node which are far away.

We are able to achieve perfect classification for both the datasets for beta value 0.001 and below.

Theme A Summary:

Score Name	**Blog Catalog AUC Score**
Katz (Beta = 0.001)	1
Rooted PageRank	0.9729
Resource Allocation	0.9607
Preferential Attachment	0.9549
Adamic Adar	0.953
Common Neighbour	0.9489
Jacard Coefficient	0.7648

Table 1: Blog catalog dataset scores for topological methods

Score Name	**Resturant AUC Score**
Katz (Beta = 0.001)	1
Rooted PageRank	0.9872
Resource Allocation	0.9845
Jacard Coefficient	0.9774
Adamic Adar	0.9723
Common Neighbour	0.9648
Preferential Attachment	0.8481

Table 2: Restaurant dataset scores for topological methods

Theme C: Network Destruction

In this section we have used topological scores obtained above to study using which score network can be dismantled quickly. Idea is to remove edges and break network into many components.

Observation which we got is topological method which is good for link prediction is bad for network destruction in that graph. To analyze performance of the technique used we have plotted **Edges Removed** vs **Number of components** graph and calculated its AUC score to get to know relative performance of same network with respect to various techniques.

Idea behind using AUC score is if a graph is dismantled quickly that is let's say if we are able to break graph into components by deleting few edges that AUC score of such graph will be higher as it will have large y-axis which will hold till end.

For each topological method we have dismantled graph in 2 ways.

First by removing edges in decreasing (natural) order of score i.e. removing edge with higher score first and other way is increasing order of score i.e. removing edge with lowest topological score first.

After each removal of an edge, number of components in the graph is calculated and plotted in the graph.

Restaurant and Blog Catalog graph dismantling:

1. Common neighbor:

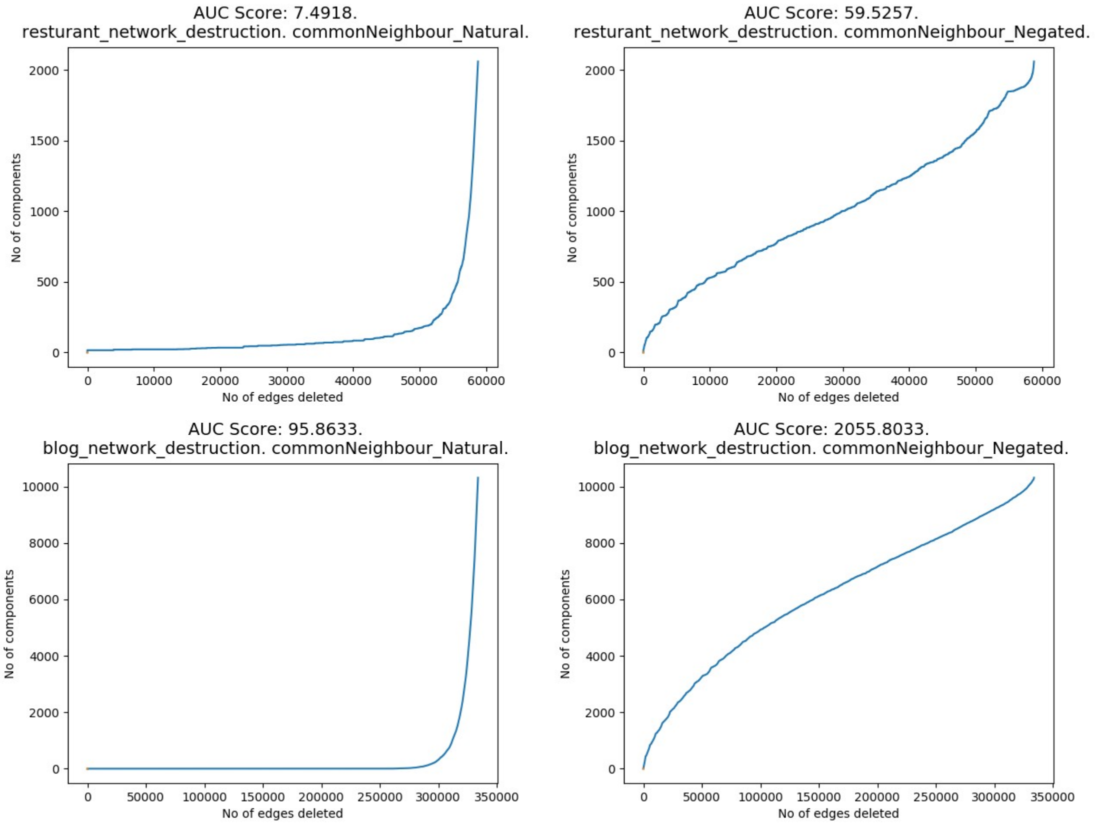
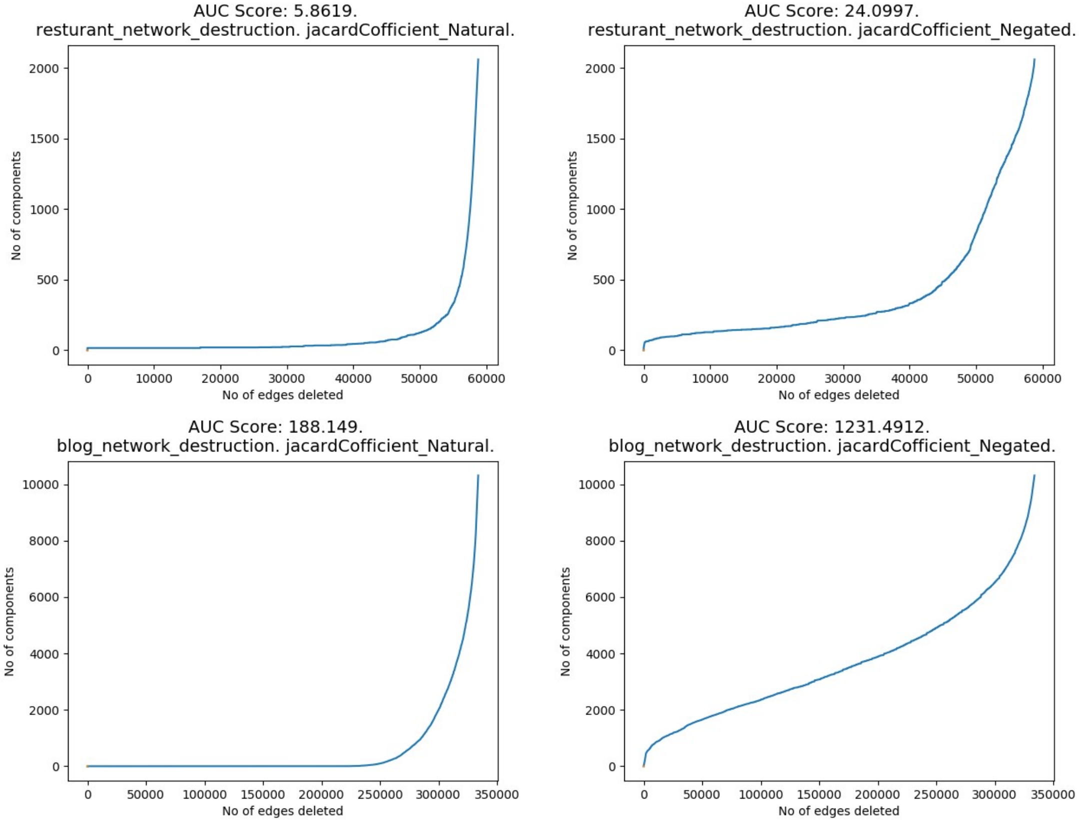


Fig 8: ROC curve

2. Jacard Coefficient:



3. Adamic Adar:

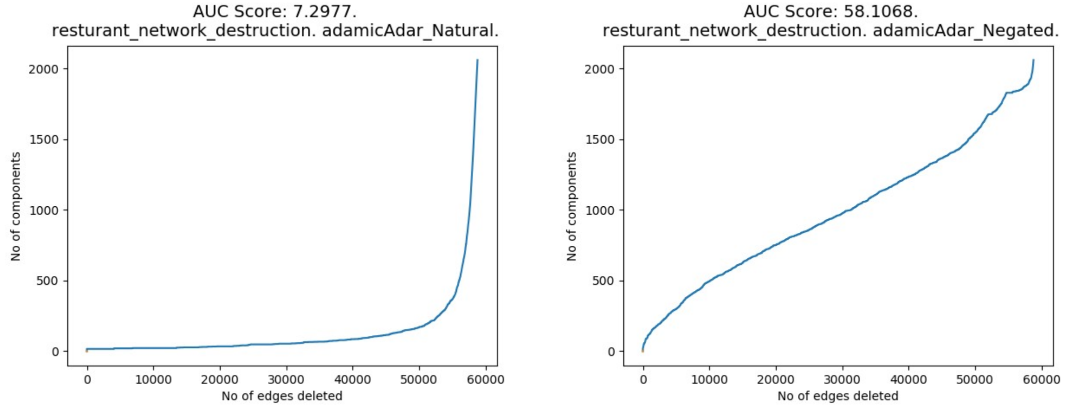


Fig 9: ROC curve

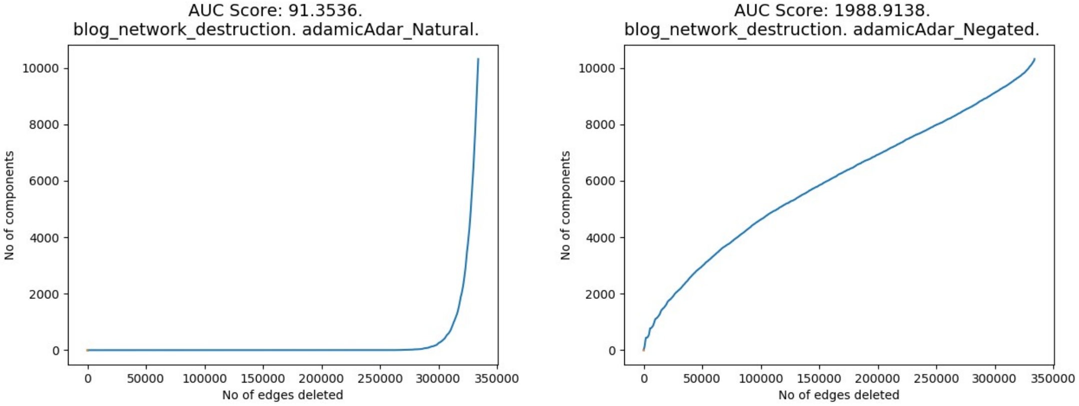


Fig 10: ROC curve

4. Resource Allocation:

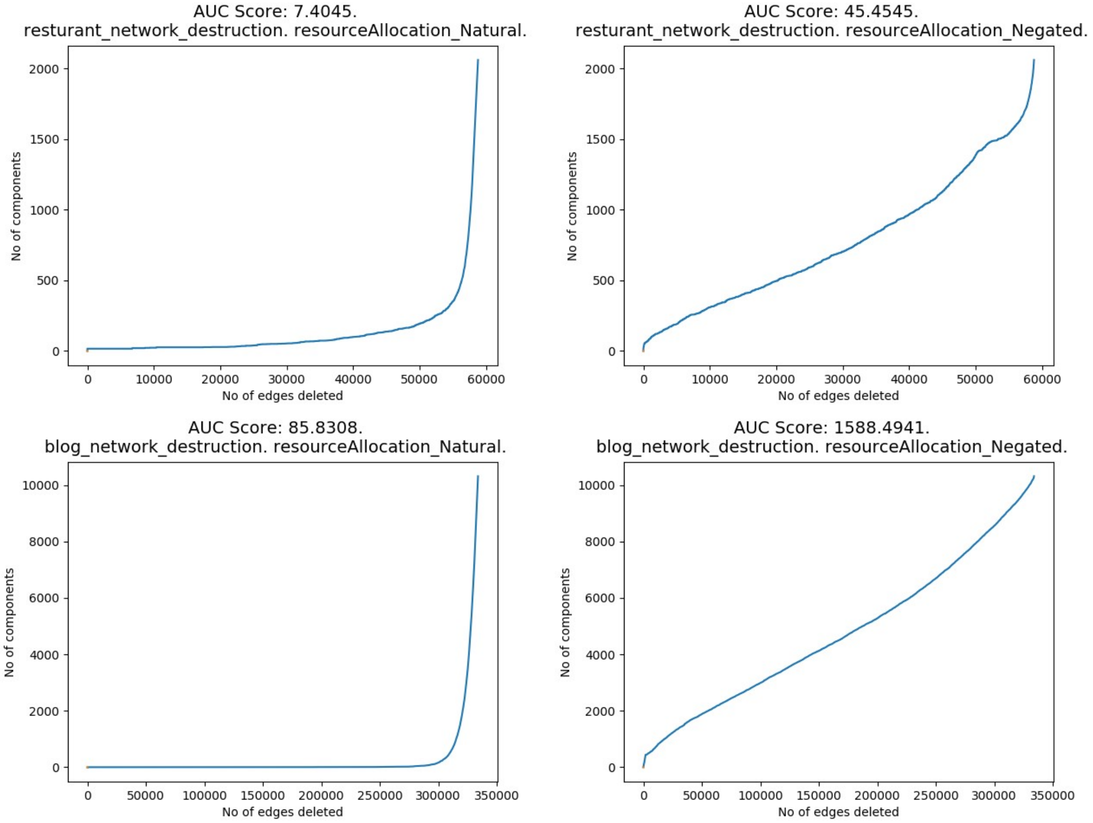


Fig 11: ROC curve

5. Preferential Attachment:

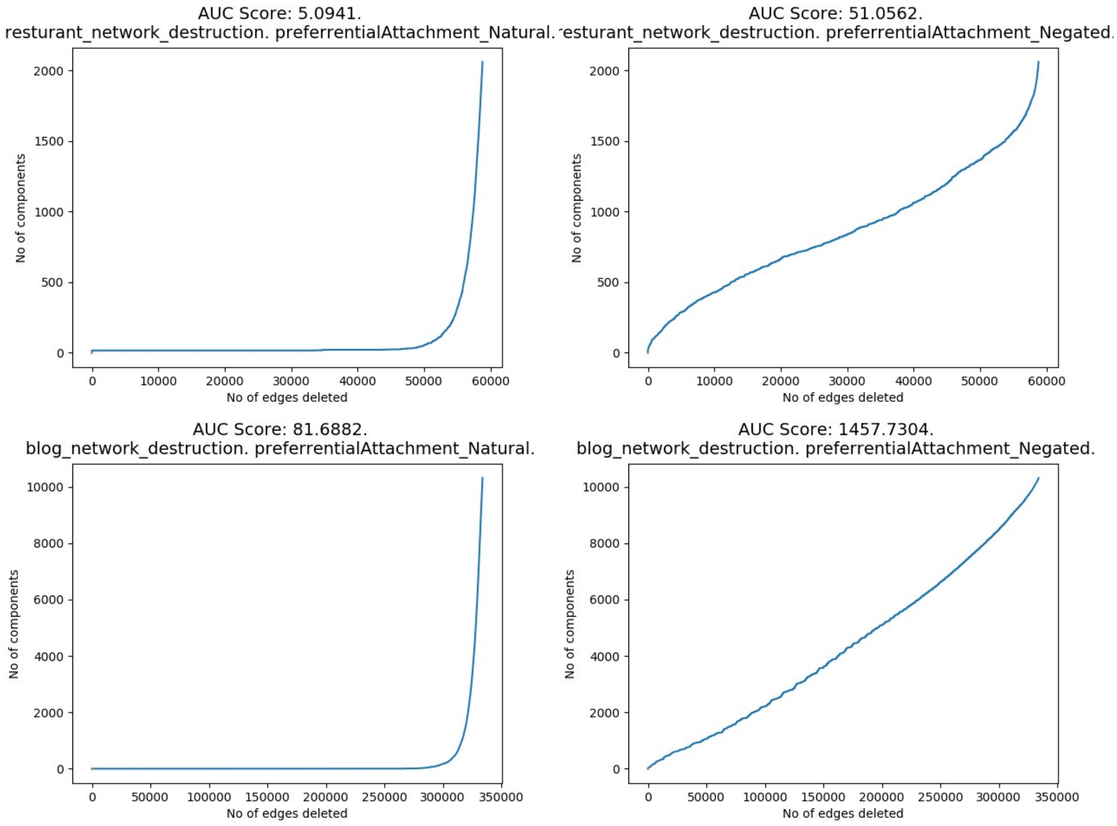
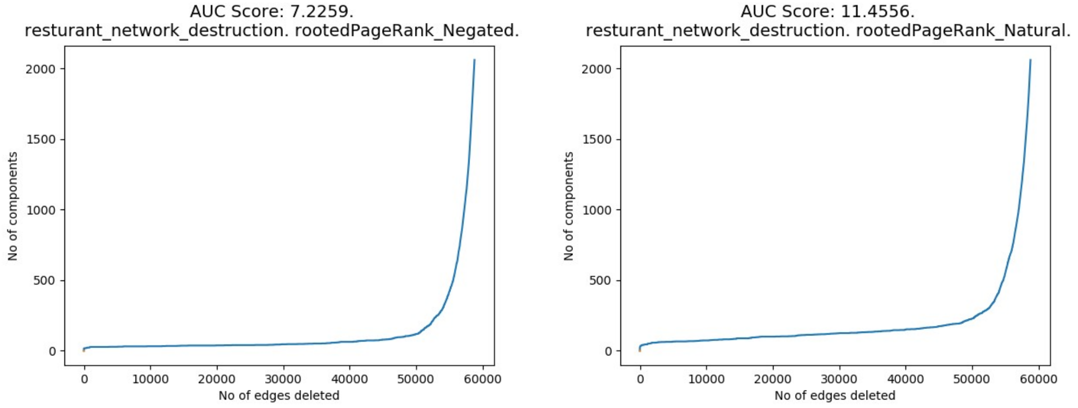


Fig 12: ROC curve

6. Rooted Page Rank:



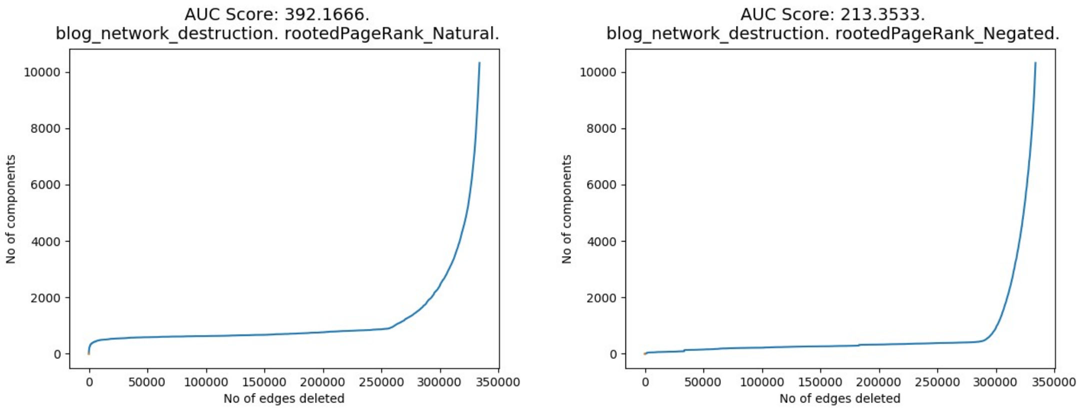


Fig 13: ROC curve

7.

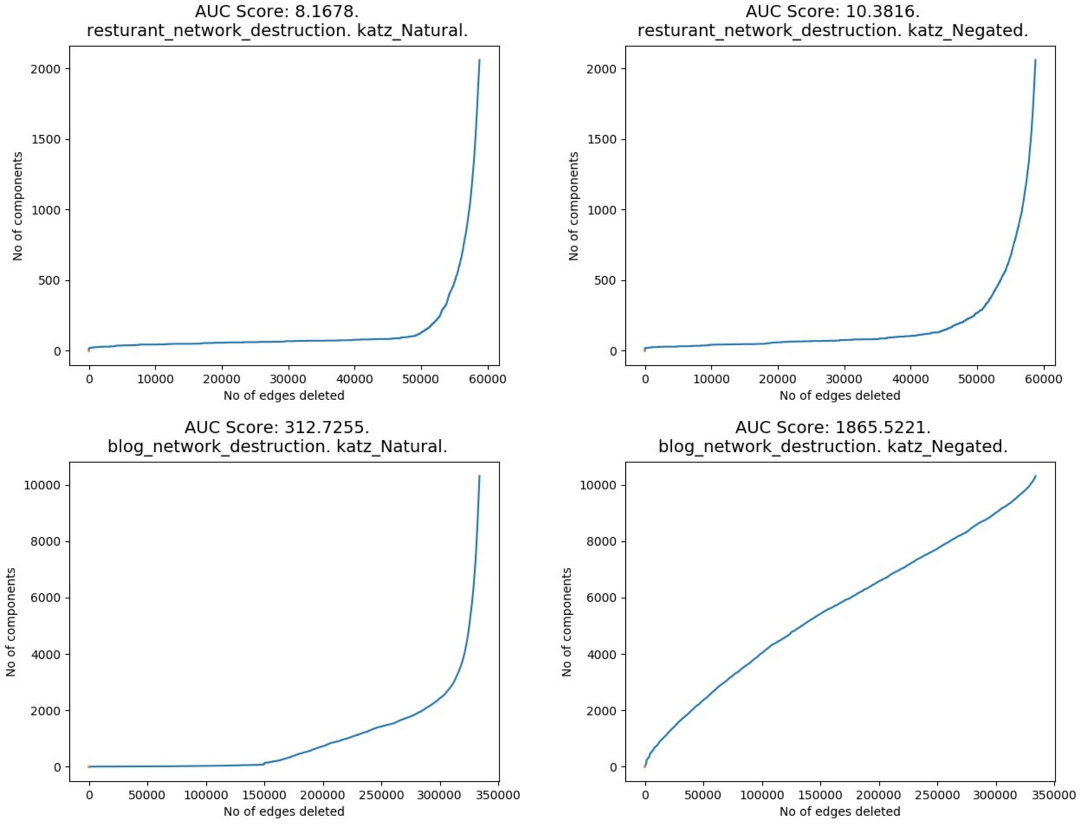


Fig 14: ROC curve

Network Destruction Summary.

Score Name	Restaurant Nw. Destruction AUC Score
Common Neighbour Negated	59
Adamic Adar Negated	58
Preferential Attachment Negated	51
Resource Allocation Negated	45
Jacard Coefficient Negated	24
Rooted PageRank Negated	11
Katz Negated	10
Katz	8
Rooted PageRank	7
Resource Allocation	7
Adamic Adar	7
Common Neighbour	7
Jacard Coefficient	5
Preferential Attachment	5

Table 3: Restaurant dataset graph destruction.

Score Name	Restaurant Nw Destruction AUC Score
Common Neighbour Negated	2055
Adamic Adar Negated	1988
Katz Negated	1865
Resource Allocation Negated	1588
Preferential Attachment Negated	1457
Jacard Coefficient Negated	1231
Rooted PageRank	392
Katz	312
Rooted PageRank Negated	213
Jacard Coefficient	188
Common Neighbour	95
Adamic Adar	91
Resource Allocation	85
Preferential Attachment	81

Table 4: Blog catalog dataset graph destruction

Key Observations:

1. For network destruction local measures are behaving better than global measure.
2. Negated common neighbor is better able to delete edges from corners decomposing graph into components faster than any other method.
3. In all the above methods when we delete in decreasing similarity score order than graph is not decomposed fast but when negated scores are used than graph is decomposed exponentially fast as compared to other way around.
4. In both the case preferential attachment is giving worst score so in any practical scenario if you want an edge deletion ordering which does not disconnects the graph for long time than PA can be used.