CS 224W Final Project Report
Link Prediction for Pinterest Board Recommendation

Andi Yang, Michael Fang, Ayooluwakunmi Jeje
Department of Computer Science, Stanford University

December 11, 2016

1 Introduction

Link prediction and user recommendation are important in a wide variety of areas. It is a particularly key area of research in social networks because of their large and dynamic nature. In this paper, we evaluate the performance of different link prediction methods on graphs constructed from Pinterest's network. Pinterest is one of the largest drivers of traffic on the internet and hence has fascinating temporal graph data. Link prediction is the task of picking out graph edges that will exist in a graph in the near future. Specifically, given the state of the graph at a specific time , we seek to predict the new edges that will be formed within a given time period afterwards.

This paper starts off by summarizing some literature that we have explored on existing link prediction techniques. We then discusses graph models that are created from using Pinterest data and prediction methods that are explored for each graph. Finally, we present the results of the methods outlined and summarize our findings.

2 Literature Review

2.1 The Link Prediction Problem for Social Networks (*1*)

Nowell et. al. studied link prediction in social networks. The paper formulates the link prediction problem as follows: Given a graph of, $G[t_0, t_0'] = (V_0, E_0)$, consisting of all edges formed in time interval $[t_0, t_0']$, we seek to predict the list of edges that are not in $G[t_0, t_0']$ but will appear in $G[t_1, t_1'] = (V_1, E_1)$, where $(t_0 < t_0' < t_1 < t_1')$. The authors surveyed a number of collaborative filtering methods for link prediction which include the use of methods based on node neighborhoods (e.g. common neighbors, Jaccards coefficient and Adamic/Adar, and preferential attachment in growth networks), methods based on the ensemble of all paths (e.g. Katz, Hitting time, PageRank, and SimRank) and higher-level approaches which can be combined with the methods mentioned previously (e.g. Low-rank approximation and Unseen bigrams).

2.2 Link Prediction Approach to Collaborative Filtering (*2*)

This paper explores how link prediction techniques similar to those presented in [1] can be applied to bipartite graphs. The scores considered neighbors of neighbors for one of the nodes rather than just neighbors of both nodes. Huang et. al demonstrated the effectiveness of these modifications on Chinese online book sales data with measures such as precision, recall, f measure and rank score. The Katz measure, preferential attachment and common neighbors performed best.

2.3 Link prediction in bipartite graphs using internal links and weighted projection (*3*)

Allali et. al. take yet another approach to predicting links in bipartite graphs. A bipartite graph, $G = (V, E)$, is projected to unipartite graphs by connecting all pairs of nodes such that $(x, y) \in V$ and $N(x) \cap N(y) \neq \emptyset$, where $N(x)$ denote the neighbors of x. That is, nodes in the projected graph, G', have an edge between them as long as they have at least one common neighbor in G. Internal links are then defined as edges that produce the same graph G' when they are added to graph G. When two nodes do not have any nodes in common in the present time frame, it is reasonable to guess that they will still have no nodes in common within the next period and hence an edges between them can be excluded from the prediction list. Selecting nodes exclusively from the internal links does exactly this. Weighting functions similar to those from [1] and [2] are used and edges that weigh above a threshold are selected. They were able to demonstrate substantial improvements on precision and recall over other collaborative filtering methods previously discussed.

2.4 Link Prediction using Supervised Learning (*4*)

This paper uses simple features to do supervised learning for the link prediction task, based on two datasets of academic co-authorship graphs (dblp and BIOBASE). They set up their task by selecting two non-overlapping time ranges, the earlier one being the training set and the later as test set. The test set is selected as the links of authors that appeared in the training set but did not collaborate. Each link is either positive or negative in the test set if there is or is not a link between the authors (and thus is a binary classification task.). Features used include proximity features (based on similarity of keywords in work), aggregated features (number of papers published by authors, number of neighbors, number of keywords, log number of secondary neighbors), and topological features (shortest path, clustering index, shortest distance in author-keyword graph [computed by extending the author graph to include keywords as nodes]). They used a number of classification algorithms, and SVM with RBF kernel achieved the highest accuracies on both datasets (83 and 90% for dblp and BIOBASE, respectively). They found the keyword similarity for BIOBASE and shortest path for dblp to be the most effective features.

3 Model

The Pinterest dataset with the following characteristics was used.

- Pinner: 18,578,352 Pinterest user ids.
- Food boards: 12,466,754 Pinterest board information (id, name, description, user id and creation time)
- Pins: 736,030,700 entries for 20,049,957 pins to various boards. (creation time, board id and pin id)
- Follows: 48,309,378 entries for 14,097,700 users following boards. (board id, user id, creation date)

We were mainly interested in board followership. Therefore, all our data processing was limited to the food boards and their corresponding follows. Board information and follow information was combined to create one file sorted in chronological order with the following schema: follower id, board id, board owner id, follow date. The main training and testing periods were taken to January 12, 2012 to March 31, 2012 and January 12, 2012 to June 30, 2012 respectively. The following graph models have been created from the dataset (we have created several versions of these graphs for different training and testing periods).

3.1 Follower to Board bipartite undirected graph (Model 1)

This graph contained two kinds of nodes: followers and boards. An undirected edge is created between a follower and a board when the follower follows the board. We constructed the graph by taking the maximum connected component of the full graph ending on March 31, 2012. The nodes are then filtered leaving only nodes that had at

least 10 new edges between March 31, 2012 and June 30, 2012. Graphs ending at each period then captured only edges between the filtered nodes.

Property	Training	Testing
Number of followers	3962	3962
Number of boards	3888	3888
Number of nodes	7850	7850
Number of edges	29599	109851

Table 1: Follower to Board bipartite undirected graph properties

3.2 Co-follower undirected graph (Model 2)

This graph contained Pinterest user nodes. An undirected edge is created between two users if they follow more than 3 of the same boards. We first filtered the full graph to include only nodes that were not present in the follower to board bipartite graph above. Nodes having degree of zero were then filtered out.

Property	Training	Testing
Number of nodes	1512	1512
Number of edges	17093	64770

Table 2: Co-follower undirected graph properties

3.3 Board follower to board owner directed graph (Model 3)

This graph contained Pinterest user nodes. A directed edge is created from user A to user B when user A follows one of user B's boards. We constructed the graph by taking the full graph ending on March 31, 2012. The nodes are then filtered leaving only nodes that had at least 10 new incoming or outgoing edges between March 31, 2012 and June 30, 2012. Graphs ending at each period then captured only edges between the filtered nodes.

Property	Training	Testing
Number of nodes	6574	6574
Number of edges	23948	63599

Table 3: Board follower to board owner directed graph properties

4 Method/Algorithms

Each link prediction algorithm implemented assigns scores to candidate edges and outputs the set of edges predicted edges to appear in the future graph. We have implemented the following scoring algorithms for link prediction:

Note:
In the bipartite undirected graph of followers to boards (Model 1), given user u and board b, we compare the set of boards u follows to the set of boards that all followers of b also follow. This is captured by comparing the neighbors of u to the neighbors of neighbors of b.

In the directed graph of followers to board owner (Model 2), given potential follower f and board owner o, we compare the set of board owners that f follows to the set of board owners that all the users that follow o also follow. This is captured by comparing the outgoing neighbors of f to the outgoing neighbors of the incoming neighbors

of the o.

$N(x)$ denotes the set of all neighbors of node x in the an undirected graph.
$N'(x)$ denotes the set of neighbors of neighbors of x in an undirected graph
$N_{in}(x)$ and $N_{out}(x)$ denotes the set of incoming neighbors and outgoing neighbors of node x respectively in a directed graph.
$N'_{out,in}(x)$ denotes the set of outgoing neighbors of incoming neighbors of node x.

4.1 Random Predictor

The algorithm assigns a random score to each candidate edge. It serves as a baseline when we evaluate different predictors.

4.2 Common Neighbors

$$score(x,y) = \begin{cases} |N(x) \cap N(y)| & \text{Model 2} \\ |N(x) \cap N'(y)| & \text{Model 1} \\ |N_{out}(x) \cap N'_{out,in}(y)| & \text{Model 3} \end{cases}$$

This scoring method serves to capture the hypothesis that the more neighbors two nodes share, the higher the chance that they will have an edge between them in the future.

4.3 Jaccard's Coefficient

$$score(x,y) = \begin{cases} \frac{|N(x) \cap N(y)|}{|N(x) \cup N(y)|} & \text{Model 2} \\ \frac{|N(x) \cap N'(y)|}{|N(x) \cup N'(y)|} & \text{Model 1} \\ \frac{|N_{out}(x) \cap N'_{out,in}(y)|}{|N_{out}(x) \cup N'_{out,in}(y)|} & \text{Model 3} \end{cases}$$

Jaccard's coefficient measures the probability that both node x and node y share a neighbor for a randomly selected neighbor that x or y has. Higher probability indicates that x and y are more strongly related.

4.4 Adamic/Adar

$$score(x,y) = \begin{cases} \sum_{z \in N(x) \cap N(y)} \frac{1}{log|N(z)|} & \text{Model 2} \\ \sum_{z \in N(x) \cap N'(y)} \frac{1}{log|N(z)|} & \text{Model 1} \\ \sum_{z \in N_{out}(x) \cap N'_{out,in}(y)} \frac{1}{log|N_{out}(z)|} & \text{Model 3} \end{cases}$$

Adamic/Adar is similar to Jacard's coefficient, but rare neighbors are weighed more.

4.5 Preferential Attachment

$$score(x,y) = \begin{cases} |N(x)| \cdot |N(y)| & \text{Model 2} \\ |N(x)| \cdot |N'(y)| & \text{Model 1} \\ |N_{out}(x)| \cdot |N'_{out,in}(y)| & \text{Model 3} \end{cases}$$

Preferential attachment is good at modeling the growth of networks. It says that the probability of a new edge involving some node x is proportional to the current number of neighbors of x. Newman et al. further proposed that the probability of an edge between node x and y is correlated with the product of $|N(x)|$ and $|N(y)|$.

4.6 Katz

$$score(x, y) = \sum_{l=1}^{\infty} \beta^l \cdot |\text{paths}_{x,y}^{<l>}|$$

Note that $\text{paths}_{x,y}^{<l>}$ is the set of paths of length l from x to y. Katz scores a candidate edge by summing over all paths between two end nodes, exponentially damped by length. In this way, shorter paths are counted more heavily.

Given the size of our graphs, it is not feasible to perform BFS starting from every node. Thus, we limited the path depth to 7 for each BFS. As a second implementation, we computed scores from $(I - \beta M)^{-1} - I$, where M is the adjacency matrix of the graph. Comparing to the estimation in the first approach, we found that the second approach yields a much better result (for $\beta = 0.05, 0.005, 0.0005$).

4.7 Rooted PageRank

Score(x,y) is defined as the probability of y in a random walk that returns to x with probability α each step, moving to a random neighbor with probability $1 - \alpha$.

We implemented rooted PageRank using two libraries, networkx and igraph (with $\alpha = 0.15$). After testing on smaller dataset, we chose to use igraph, which create the graphs and compute the scores much more quickly than the personalized pagerank method in networkx, and at the same time uses less memory.

4.8 Hitting Time & Commute Time

$$HittingTime: \ score(x, y) \propto \ -H_{x,y}$$

$$CommuteTime: \ score(x, y) \propto \ -(H_{x,y} + H_{y,x})$$

Note that $H_{x,y}$ is the expected number of steps for a random walk from x to reach y. Hitting time and commute time both measure proximity between pairs of nodes. Commute time additionally deals with unsymmetric hitting time by summing both $H_{x,y}$ and $H_{y,x}$. It turned out that for our graphs, Hitting Time and Commute Time required far more iterations and it was not possible to obtain any reasonable predictions if we restrict the number of iterations to a much smaller number to run in a reasonable amount of time.

5 Supervised Learning

We extend our results to include supervised learning. We run our experiments only on the directed board-follower to board-owner graph. Furthermore, unlike the previous experiments, we limit the nodes to potentially follow to those that are within n-hops of each node in the graph from the first three months. These are the edges that are most likely to occur, and focusing on them allows us to maintain a tractable datasets. Then, we construct training and test sets by randomly partitioning the nodes in two, assigning the edges that appear in the second time period that originate in one partition to the training set and the rest in the test set. We use time periods of three months and one year after the first cutoff. Our dataset is summarized in the table below.
We use the graph from the first three months to featurize each edge potential pair. We use a mix of standard graph topological features as well as some of the unsupervised metrics from above. They are: source and destination node in-degree and out-degree (all as separate features), source and destination in- and out-volume, and then

common neighbors, Jaccard similarity, Adamic-Adar, Preferential Attachment, and Katz (with $\beta = 0.005$). We normalize the features to have zero mean and one standard deviation based on the training set. We train a linear SVM to predict whether a potential edge exists or not. Then, to test our results, we rank each potential edge in the test set based on the confidence of the model to predict its existence. We evaluate our model against the unsupervised measures in the feature set using receiver operating characteristic (ROC) curves, which summarizes the performance by a sliding the discrimination threshold over the confidence scores.

	3 Months	1 Year
Training (+)	558	843
Test (+)	665	962

Table 4: Number of new edges created three months and one year the first cutoff. In total, there are 35967 possible edges for the training set and 35917 edges for the test set.

6 Results

6.1 Scoring Methods

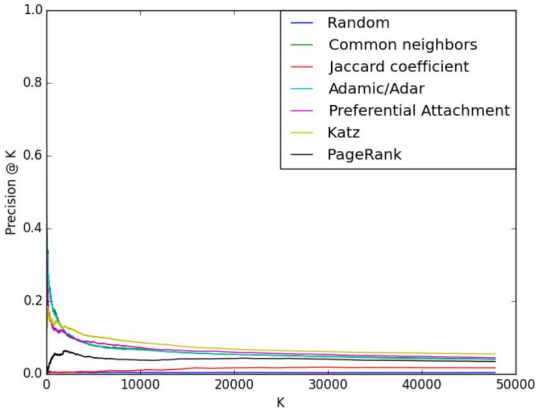
To evaluate the relative performance of the different methods of scoring, we compared the Precision@K and number of correctly predicted edges as the number of prediction, K, increased.

$$\text{Precision@K} = \frac{\text{Number of top K predictions that are correct}}{K}$$

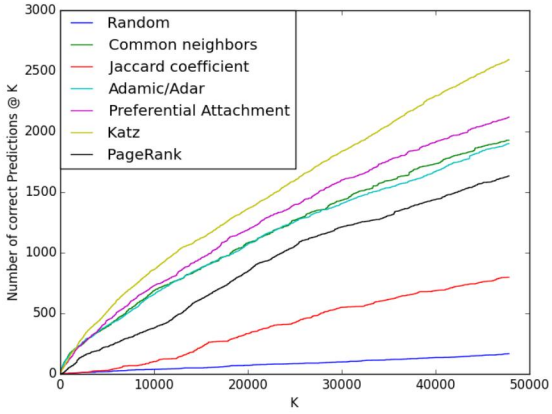
Follower to Board bipartite undirected graph The results for this model can be seen in Fig 1. In this model, Katz and Preferential Attachment perform the best. Since Katz score is based on the emsemble of all paths, if there are more paths between a board and a user, it is more likely that the user will follow that board. This gives Katz an edge over neighbor based scoring methods as they are quite restricted to only capturing relationships between nodes that are a few hops away. Also, Preferential Attachment performs better in this model because it captures the fact that there are popular boards in Pinterest which are much more likely to be followed than other boards. Even though Jaccard coefficient is similar to common neighbors, the normalization dampens the effect of followers being exposed to boards and hence it performs the worst.

Co-follower undirected graph The results for this model can be seen in Fig 2. This network is the most similar to the five co-authorship networks evaluated in this paper (*1*). Similar to the results presented in the paper(in Figure 5), we also observe Adamic/Adar and Katz as the top link predictors. As noted in the paper, we notice that Katz and Adamic/Adar are very similar in predictions, while rooted Pagerank and Jaccard are less similar to the rest. Similar to the follower to board bipartite undirected graph, we observed how the dampening effect of the normalization in the cause much worse performance than other scoring methods.

Additionally, most of the scoring methods achieved much higher prediction precision than in the other models explored. Adamic/Adar, Common neighbors, Preferential Attachment, PageRank and Katz achived greater that 20% accuracy on this model while none of the methods produced 20% on the other models.

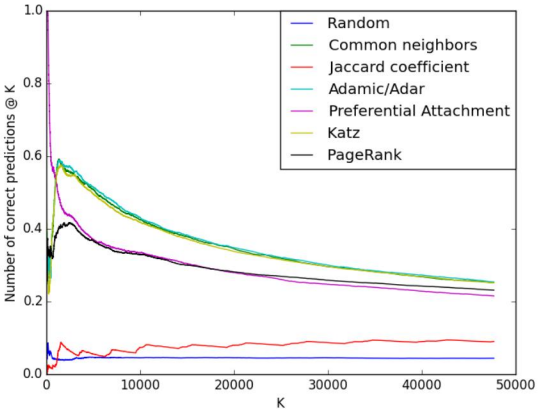


(a) Precision @ K vs K

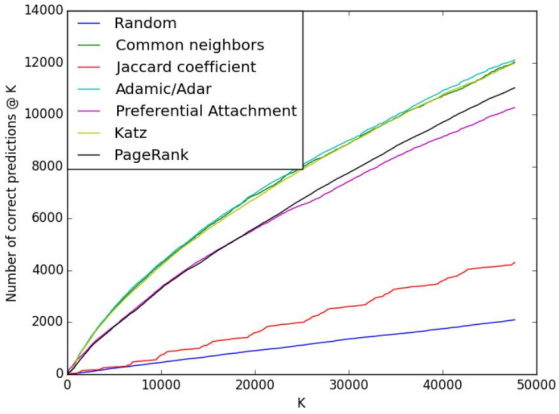


(b) Number of correctly predicted edges vs K

Figure 1: Perfomance of scoring methods on the Follower to Board bipartite undirected graph. (Parameters in the following graphs are set as: (1) for Katz(unweighted), $\beta = 0.0005$; (2) for rooted Pagerank, $\alpha = 0.15$)



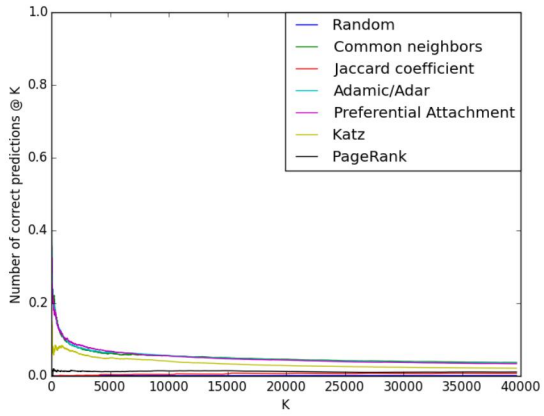
(a) Precision @ K vs K



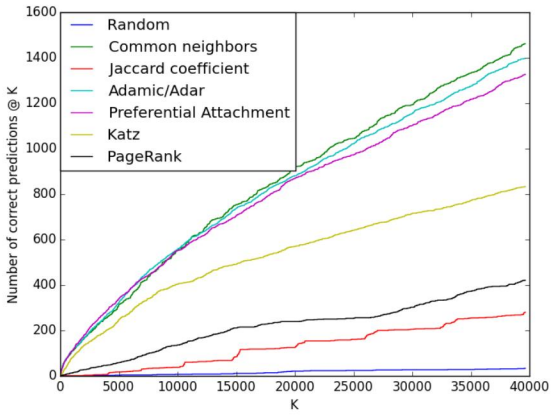
(b) Number of correctly predicted edges vs K

Figure 2: Perfomance of scoring methods on the Co-follower undirected graph. (Parameters in the following graphs are set as: (1) for Katz(unweighted), $\beta = 0.0005$; (2) for rooted Pagerank, $\alpha = 0.15$)

Board follower to board owner directed graph In this network, we observe that methods based on node neighborhoods tend to outperform the others. This can be naturally explained by the fact that two users sharing more neighbors are more likely to follow one or the other. This model captures the interaction of users and so proximity of nodes in the graph will be more important.



(a) Precision @ K vs K



(b) Number of correctly predicted edges vs K

Figure 3: Perfomance of scoring methods on the Board follower to board owner directed graph. (Parameters in the following graphs are set as: (1) for Katz(unweighted), $\beta = 0.0005$; (2) for rooted Pagerank, $\alpha = 0.15$)

6.2 Supervised Learning

Figure 4 shows the curves of the supervised model compared to the unsupervised methods when the the evaluation period is three months. The SVM dominates the unsupervised metrics, performing better than the others for nearly all of the thresholds. Table 5 summarizes the results in terms of the area under the curve, which shows that the superiority endures over a longer time period as well. Note that these results don't compare directly to the previous experiments due to the difference in experimental setup.

AUC	3 Months	1 Year
Linear SVM	**0.729**	**0.708**
Katz	0.637	0.630
Common Neighbors	0.548	0.537
Preferential Attachment	0.584	0.562
Adamic-Adar	0.548	0.538
Random	0.499	0.504

Table 5: Area under the curve (AUC) for the models with three month and one year.

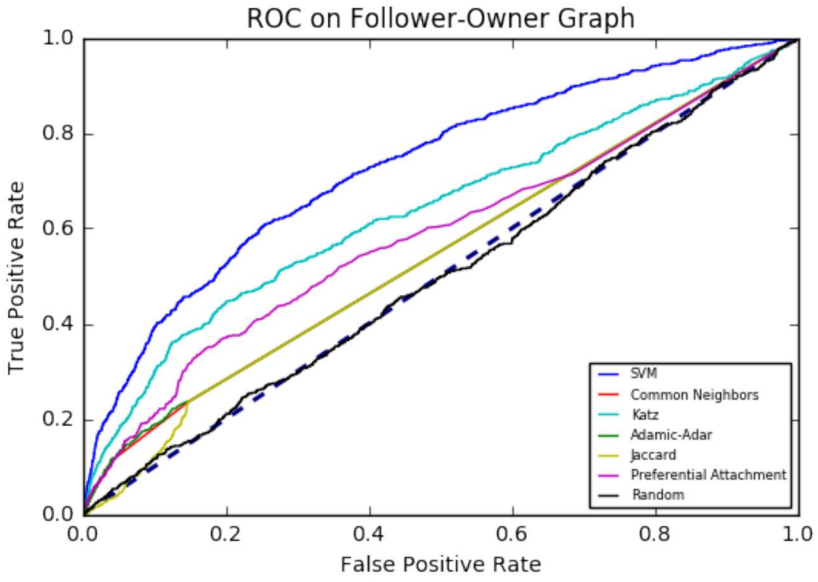


Figure 4: ROC curve of the SVM compared to the unsupervised metrics that are used as features. This is for an evaluation period of three months.

7 Conclusion

Link prediction based solely on graph structure is a difficult task. In this project, we have implemented eight different link prediction methods to evaluate on the following networks that we created: (1)Follower to Board bipartite undirected graph; (2) Co-follower undirected graph; (3) Board follower to board owner directed graph. We successfully limited the size of the graph to compute on by taking the maximum strongly connected component and further pruning down inactive edges. To evaluate the results, we split the graph into training and testing periods and rank all potential edges in descending order of their scores. We were able to obtain reasonable results using various link predictors that we implemented. It is interesting to see that a measure as simple as common neighbors performs better than other slightly more sophisticated methods.

Furthermore, we showed that a supervised learning method could achieve improved results over the unsupervised methods on the board follower to board owner directed graph. This could be potentially useful for recommending additional people for users to follow.

References and Notes

1. Liben - Nowell, D., and Kleinberg, J. (2007). The link prediction problem for social networks. Journal of the American society for information science and technology, 58(7), 1019-1031.

2. Huang, Z., Li, X., Chen, H. (2005) Link Prediction Approach to Collaborative Filtering. In *Digital Libraries, 2005. JCDL '05. Proceedings of the 5th ACM/IEEE-CS Joint Conference on*

3. Allali, O., Magnien, C., and Latapy, M. (2011, April). Link prediction in bipartite graphs using internal links and weighted projection. In *Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference* on (pp. 936-941). IEEE.

4. Al Hasan, M., Chaoji, V., Salem, S., and Zaki, M. (2006, April). Link prediction using supervised learning. In SDM06: workshop on link analysis, counter-terrorism and security.