CS224W Reaction Paper and Proposal
Supervised Link Prediction in Bipartite Networks

Arathi Mani
arathim@stanford.edu

Kameshwar Chinta
kamchinta@gmail.com

Kevin Clark
kevclark@stanford.edu

October 16, 2014

1 Introduction

A ubiquitous challenge in the analysis of networks is predicting what new connections will be created in a particular graph at some point in the future. Being able to predict such connections with accuracy has many important practical uses including national defense, where links between terrorists can be deduced from an existing network [?], security, where links might be able to predict future credit card frauds, and social networks, where potential friend relationships could be predicted [?]. Link prediction also can be mutated into variants such as link recommendation (e.g. the "Find Friends" feature in Facebook listing potential acquaintances), link anomaly detection, and other applications.

We chose three papers each predicting new edges in a graph, each using a different algorithm. Yan and Gregory [?] apply node similarity measurements to weight edges before traditional community detection algorithms were applied. Nowell and Kleinberg [?] used network features to calculate various similarity measures between nodes and then predict future edges in a collaboration graph. Backstrom and Leskovec [?] used both node and edge feature data as well as information about the network structure in a supervised random walk algorithm to predict edges.

From these papers, we decided to analyze the algorithms presented when applied to bipartite graphs. Some of the papers suggest applying the link prediction algorithms to graphs that now contain two different types of nodes as future work. We became interested in researching how effective the algorithms would now be and whether modifications in the algorithms could be done to create equally, if not more, accurate predictions. For our experiments, we decided to use Yelp data, which links users to business through reviews. Users and businesses represent nodes in our bipartite graph while the reviews act as edges.

2 Literature Summary

2.1 The link prediction problem for social networks [?]

The setup under which the link prediction problem is studied consists of taking a snapshot of the network at different time points. The links are predicted from the prior network snapshot and evaluated against the links manifested in the subsequent time point (considering only the nodes present in the initial snapshot). The question that this paper tried to answer is how to find the missing links between members of a network to suggest collaboration sooner and/or to infer any missing links from a network of security context involving terrorist activities to study the plausible links and network evolution.

There are several methods discussed for link prediction and they are all intrinsic to the network properties and hence can be applicable for all networks. Different methods proposed and studied under the experimental setup are Graph Distance, Common Neighbors, Jaccard's Coefficient, Adamic/Adar

Score, Preferential Attachment, Katz, Hitting Time, Page Rank and Sim Rank. Many of the methods rely on similarity metrics produced by examining node neighbors (Common Neighbors, Adamic/Adam Coefficient, and Preferential Attachment). The remaining methods deal with the ensemble of all the paths in the graph.

These methods are tested on five datasets, each from a different section of the co-authorship network arVix. The results suggest that Adamic Score and Katz method (or its variants) are relatively close in their performance and provided consistent results. To a lesser extent, SimRank, Page Rank and Jaccard's Coefficient have provided consistent results as well. The number of common "correct" predictions shows qualitatively similar behavior suggesting that there is a large overlap counter-intuitive to the definitions of each one of these methods provided by their definitions.

Apart from this unexpected behavior, the small world problem in observed in collaboration networks. To elaborate, there is often a very short path between any two random scientists, at times between scientists of unrelated fields. For example, the paper mentions how developmental psychologist Jean Piaget has a surprisingly small Erdős number of three. The most successful link predictors are using measures of proximity that are robust to few based based on a rare collaboration between nodes (scientists, in this case of collaboration). Many pairs that collaborate are at a distance of more than two. The study also tested disregarding the distance two pairs, which renders the common neighbor methods irrelevant (as there are no common neighbors). The methods based on shortest path have been applied, out of which unweighted Katz method has the best performance closely followed by weighted Katz, SimRank, Page Rank, etc.

Even the best of the methods, Katz, is only performing correctly 16% of the times. This leaves a lot of room for improvement. Because the networks studied have a loss of information by only using one class of node (only scientists but not articles), viewing collaborations as a bipartite graph was considered instead. This increased in magnitude of data, causing some of the algorithms to become computationally prohibitive, so they were tested on a smaller dataset. The results indicate no improvement.

2.2 Supervised random walks: predicting and recommending links in social networks [?]

One challenge of link prediction is building a model that combines information about node attributes and edges with network structure in a principled way. A common approach is to extract network features from two nodes and combine them with node attributes, but this can be cumbersome and and requires dealing with highly unbalanced data (networks tend to be sparse so very few pairs of nodes have an edge between them). Another common approach to link prediction is using a PageRank style algorithm that assigns scores to nodes based on the stationary distribution of a random walk. This provides a measure of distance between nodes that can be used for link prediction. Although this takes better advantage of the network topology, it does not take into account node or edge attributes.

Backstrom and Leskovec propose an algorithm based on supervised random walks that combine ideas from both these approaches. Predictions are made based off of the scores of a random walk, but the edge weights for this walk are learned from node and edge attributes in a supervised way. Edge weights are assigned as a function f of the attributes of the edge and the involved nodes. Ideally, the parameters of this function would be set so the nodes that will link up to a node v in the future are given a higher score by the random walk than nodes that won?t. However, this is a highly constrained optimization problem that may not be solvable. Instead, the problem is relaxed to learning parameters for f that minimize a loss function that captures these constraints in a soft way. This loss function is minimized with a gradient-based optimization technique. Although some of the derivatives for the loss function cannot be compute exactly, they can be approximated with an iterative process. To make a prediction, edge weights are assigned to the network based off of the learned function and then a random walk is done to predict new links.

This approach is evaluated on both synthetic and real-world data. The synthetic network was created using the Copying graph model. Edge weights are assigned according to an f with predefined parameters plus some noise. Then future edges are generated based off the results of a random walk with these edge weights. The algorithm was able to learn the predefined parameters for f when there was no noise and actually outperforms a model using the predefined parameters even when noise is added. The algorithm was also evaluated on four physics co-authorship networks and a complete Facebook network of Iceland. The algorithm was compared with four unsupervised link prediction techniques as well as simple supervised classifiers that make use of both network features and node attributes. The supervised random walk algorithm outperformed all other approaches on both datasets.

2.3 Detecting community structure in networks using edge prediction methods [?]

This paper aims to take a novel view of harnessing link prediction techniques to effectively detect communities in the network. It is proposed that before applying community detection algorithms like Common Neighbors (CN), Reposition by Fine Tuning (RFT), and Community Overlap Propagation algorithms, a node similarity function should be applied. This function is defined as the number of common neighbors for the nodes of a given edge. This is added as a weight to the edge (for edges with no common neighbors, 1 is added). The similarity score of the vertices suggests whether the nodes can be neighbors or not (thus predicting the edge between them). The intuition is the weight assignment will help identify potential strong bonds between the nodes to help identify the communities. Once the weights are added, the community detection algorithms which are capable of determining communities in a weighted networks are employed.

The experimental setup to test the proposal entails tests on both the artificial and real-world networks. For artificial networks, a normalized mutual information measure is used to compare against the known networks. In the case of real-world networks, since there is no knowledge of the true number of communities existing, modularity measure is used to assess the quality of the partition. From a data standpoint, the real-world datasets used to test the proposal came from social, collaboration and citation types and are 10 in total.

The result of these tests on different datasets only proves the point that the communities are more accurately identified when using the idea of node similarity to augment the community detection. Since the technique of vertex similarity is local to the graph, it does not reduce the speed of community detection algorithms but increases the accuracy.

3 Literature Critique

3.1 The link prediction problem for social networks [?]

The study of link prediction in this paper is rather interesting. The exploration of similarity measures have been applied well on the data, however, it would have been interesting to see the results of applying these techniques in a directed graph setting as well as to comparing the algorithm's performance on undirected networks.

The small world problem discussed in the study along with no links predicted for common neighbor score of two have not been explored in detail to propose countermeasures rather than avoid the node based similarity measures to path based similarity measures. It could be the case that the choice of dataset can influence whether choosing to discard a common neighbor score of two is a valid filter or not. For example, in Backstrom and Leskovec's paper [**?**], the co-authorship graph benefited from giving less weight to triangular relationships where a friend two hops away is less likely to be a new edge if the source node's friends were friends, but the Facebook network benefited in accuracy for acknowledging such relationships with higher weights. Also, what is not made clear is whether the subset of bipartite network is representative of the overall network or if it is missing the links of high prediction value.

It is suggested towards the end of the paper that, perhaps weighting the graph based on temporal aspects might have increased the accuracy. Since all the methods studied are intrinsic to the network properties, a study considering the node attributes like, geographical aspect, should have been studied to see how any of these algorithms would have performed when treated with "node attributes".

3.2 Supervised random walks: predicting and recommending links in social networks [?]

The algorithm proposed in this paper proved to be very effective in link prediction. However, there was not much motivation for why the approach works well. Edge weights in random walks can be considered transition probabilities, so learning these weights corresponds to learning the probability of moving between nodes. Random walks themselves are well understood probabilistic models, so it would have been nice to see a probabilistic interpretation of what the algorithm is learning and what assumptions are going into that model. The paper also does not investigate how the graph structure relates to the algorithm's convergence and accuracy. The approach is only evaluation on small-world networks. It would be interesting to investigate how the algorithm works on other kinds of graphs, such as for bipartite link prediction. Finally, although the algorithm uses node and edge attributes to make predictions, it is only tested on networks with very few (less than 10) features. It would be useful to know how well the algorithm scales and performs on datasets with much richer node and edge attributes.

3.3 Detecting community structure in networks using edge prediction methods [?]

The context under which the paper introduces the proposal is tested on an undirected and unweighted graph structure. If this proposal is going to hold for other network topologies, it is yet to be tested before adopting this as generalized algorithm to enhance community detection. The details of what needs to be changed in the similarity function when this is extended to other graph models remains a limitation.

Instead of trying to propose just the vertex similarity, harnessing node properties (when available) to construct a rather sophisticated similarity function would provide better results. Perhaps, this takes a narrow viewpoint of defining the function "local" to the network which will limit the ability to leverage other node properties from consideration. The other network measures such as node centrality and betweenness measures are not considered to better understand the ”influence” of a node in forming an edge with the other.

4 Discussion

Nowell and Kleinberg?s work with link prediction was a particularly good source for brainstorming ideas for a project. Some of the ideas they mentioned as possible topics for future experimentation included using a bipartite graph to include information about a dual relationship, assigning weights to edges based on their timestamp on when a particular relationship formed, and, in general, finding ways to better utilize information that is available in the nodes themselves rather than just focusing on the network structure.

One idea from the Nowell and Kleinberg paper was that the experiments disregarded any new nodes that appeared in the network at time $t' > t$. One of the reasons behind this was because the experiments assumed that these new nodes came in with no node attribute information and thus predictor algorithms such as Jaccard?s coefficient, PageRank, and common neighbors would fail for these new nodes. However, in real life, when a new node joins a particular network, it is very likely that the new node will come with certain attributes defined. For example, a new Facebook user must input their email, name, birthday, and gender to create an account. These node attributes can already provide a great deal of information about the placement of a node in the network and links could be possibly predicted.

Another idea stemming from the Nowell and Kleinberg paper was to evaluate the same idea of trying to predict new edges of the already existing graph at a later time t? (ignoring new nodes) by weighting edges in the original graph by recency. We can partner this edge attribute with node attributes such as locality, and even further incorporate network features such as finding common neighbors or finding a shortest path from a source node to a target node. Backstrom and Leskovec proved in their paper [**?**] that using both network features and node/edge attributes outperformed models that used either network features [**?**] or node attributes [**?**] alone.

The three papers analyzed for the literary search were all focused on trying to predict edges in homogeneous graphs where the nodes were all of one type (i.e. the nodes represented authors [**?**, **?**] or just people in a social network [**?**]). The papers also mentioned the link prediction problem appearing in graphs with multiple types of nodes as well which could be modeled using a bipartite graph. For example, we could test whether the algorithms proposed in the papers could be equally as effective for a bipartite network where we try to predict edges going from one group to another[**?**]. An example where such a prediction could be useful is where one group represents a set of users and the other group represents videos and we try to predict which users will be watching which videos in the future.

5 Project Proposal

5.1 Problem Statement

The goal is to infer missing edges in a bipartite graph. In particular, we will try to predict which businesses a given Yelp user will review. If we restrict the graph to only including favorable and unfavorable reviews, this could be used to predict which businesses a user is likely to go to and like or go to and dislike. This kind of information could be useful for businesses (for example in improving ad targeting), and for users (for example in providing recommendations).

Although much work has been done on predicting edges in homogenous graphs like social networks, less has been done in the bipartite setting. We plan on extending algorithms commonly used for link prediction in homogenous graphs to be used on bipartite graphs. We are particularly interested in combining the information from external node and edge attributes with the network structure, as the Yelp dataset contains rich metadata and textual information about the involved users and businesses.

5.2 Dataset

Yelp provides a Challenge dataset consisting of three different types of objects: business objects, review objects, and user objects. The dataset contains 42,153 business objects, 252,898 user objects, and 1,125,458 review objects. The dataset also contains information on check-ins and tips (two methods aside from reviewing with which a user may interact with a business), but for the purpose of this project, we will ignore this data. The Yelp dataset also contains information about a social graph with 955,999 edges that shows which users are friends with other users.

A business object contains information about the business location (city, state, neighborhoods nearby, schools nearby, latitude, and longitude), reviews information (review count and average stars), whether or not the business is still open, and other identification information (name, photo url, and id).

A review object contains information about which objects are connected by the review (a user id and a business id), the number of stars, the date, the text of the review, and information about how other users have rated the review (counts of votes as “useful”, “funny”, or “cool”).

A user object contains information about the number of reviews he/she has written, the average number of stars that he/she has doled out to businesses, information about how other users have rated their reviews (counts of votes as “useful”, “funny”, or “cool”), and user identification information (name and user id).

5.3 Network Representation

We will represent the network with a bipartite graph. One side of the graph will represent users and the other side will represent businesses, with an edge going between a user and a business if that user has written a review about the business.

5.4 Algorithm

5.4.1 Similarity Measures

Many of the unsupervised link prediction approaches described by Liben-Nowell and Kleinberg [**?**] could be applied to bipartite networks with a few modifications. In particular, we will try using various similarity measures between reviewers and businesses as a way of ranking the likelihood of new edges appearing. Since we are dealing with a bipartite network, some of the similarity measures listed in [**?**] may not be directly applicable. However, we believe they could be still be useful if we adapt them to be used for bipartite graphs. For example, instead of using common neighbors (for which there will be none), we could look at how many neighbors of one node are neighbors-of-neighbors of the other node (excluding the first node when computing the neighbors-of-neighbors).

5.4.2 Pairwise Classifier

We will train a classifier to predict the presence or absence of a future edge appearing between two nodes using logistic regression. We will use both features from the topology of the network as well as attributes from the involved nodes. Our network features will differ slightly from the ones commonly used for social network link prediction because we are dealing with a bipartite network, but many of those features could be applicable if they are modified slightly. We also plan to use the similarity metrics we came up with as features. Due to the size of the network, it may be computationally infeasible to make a prediction for every user-business pair. To handle this, we could prune the set of possible future edges being considered using heuristics based on our similarity measures.

5.4.3 Supervised Random Walks

Backstrom and Leskovec [**?**] used supervised random walks to use node and edge attributes as well as the network topology to predict links in social networks. They train a model to learn edge weights based on the attributes so a random walk over the dataset will highly rank nodes likely to be connected with a new edge. We plan to investigate if this sort of approach will be effective for bipartite link prediction. To start with, we will evaluate the effectiveness of a PageRank approach using unweighted edges. We could also evaluate the effectiveness of heuristically chosen edge weights, or using the scores produced by our pairwise classifier as weights. Finally, we will re-implement the supervised random walk algorithm and compare its effectiveness with our other approaches.

5.5 Evaluation

Link prediction can be viewed as a binary classification task on whether there will be an edge between two nodes at some future time. To evaluate this, we will create two “snapshots” of the network at time t and a later time t' and try to predict which pairs of nodes in the earlier snapshot will have an edge between them in the later snapshot. We will construct these by considering the whole available dataset to be the snapshot at time t' and then delete all nodes and edges created after a particular time to get the earlier snapshot. Our algorithms will assign a score to the existence of each possible edge at t'. Using this, we can get a ranking of which edges are likely to occur in the future, and then evaluate this ranking with the AUC (area-under-curve) metric. We will also score out approaches using the Prec@10 metric (what proportion of the top 10 highest scoring predicted edges will exist at t'). Prec@10 better reflects the quality of our algorithm as a recommender since it measures how accurate we will be if we suggest 10 new businesses for a user to visit.

5.6 Deliverables

We will create two graphs from the Yelp data at two different points in time based on the timestamp of reviews. After training the model, we will predict edges on the data and compare the results against baseline results using the same algorithm but with a homogeneous graph. This will allow us to asses the effectiveness of each algorithm in a bipartite graph and also allow us to propose modifications to algorithms in order to better the accuracy.

References

[1] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *12th CIKM*, pages 556-559, 2003.

[2] L. Backstrom and J. Leskovec. Supervised random walks: predicting and recommending links in social networks. In *WSDM?11*, pages 635?644, 2011.

[3] B. Yan, S. Gregory. Detecting community structure in networks using edge prediction methods. In *Journal of Statistical Mechanics: Theory and Experiment 2012* 2012 P09008.

[4] S. Myers and J. Leskovec. On the convexity of latent social network inference. In *NIPS ?10*, 2010.

[5] Valdis Krebs. Mapping networks of terrorist cells. *Connections*, 24(3):43?52, Winter 2002.

[6] N. Benchettara, R. Kanawati, C. Rouveirol. Supervised Machine Learning applied to Link Prediction in Bipartite Social Networks. In *Proceedings of the International Conference on Advances in Social Network Analysis and Mining*. IEEE, Los Alamitos, CA, 326-330, 2010.