CS224W: Social and Information Network Analysis
Project Report: Edge Detection in Review Networks

Archana Sulebele, Usha Prabhu, William Yang (Group 29)

Keywords: Link Prediction, Review Networks, Adamic/Adar, Social Networks, Bipartite Networks

1 Introduction

The goal of this project is to study review networks with the intention of predicting which product a reviewer is going to review next. The question is framed as a link prediction problem, which has been extensively studied. Our primary contribution is in three areas: understanding how product review graphs differ from other types of graphs which have been studied by others, adapting distance measures that have been used in unipartite graphs to bipartite graphs such as a review graph, and finally, understanding what advantage knowledge of other features of the reviewer and the product (such as product similarity) play in link prediction.

2 Prior Related Work

In Liben-Nowell and Kleinberg[1], the authors try to solve the link prediction problem based solely on the characteristics of the network itself. They exhaustively study a number of unsupervised methods of link prediction and compare the results. We have used the algorithms outlined in this paper,namely, Common Neighbors, Adamic/Adar, Katz and Preferential Attachment as the basis for our analysis.

In Murata et al.[2], the authors utilize structural properties of networks to predict network evolution. Improving over the existing approaches based on structural properties like Newman's Common Neighbors and Adamic/Adar which do not take weights of links into account, they present an improved method for predicting links based on weighted proximity measures of social networks. We closely followed their methodology.

In Huang et al.[3], the authors employ link prediction approaches to the realm of recommendation systems. In order to infer user interests and preferences and suggest the right products and services to them, they explore correlations between user-product interactions. They derive user neighborhoods by identifying similar users and then make recommendations based on user's neighbors' experiences. The authors have extended six proximity measures commonly used in social network link prediction for making collaborative filtering recommendations. We have extended the linkage measures outlined in the paper.

In Benchettara et al[4], the authors attempt to solve the link prediction problem with special emphasis on bipartite networks. They attempt to predict links in the bipartite network and the unimodal network obtained by projecting the bipartite graph on one of its node sets. Using Classical supervised machine learning approaches, they model the problem as a classification problem and learn prediction models. We have closely followed the approach outlined in the paper to compute direct linkage attributes on the bipartite graph and indirect topological linkage attributes on the projected graph. The computed metrics are used to estimate the likelihood of a link between two nodes in the bipartite graph.

3 Data Collection - Taming the beast

3.1 Dataset information

We use the Amazon product co-purchasing network metadata available at http://snap.stanford.edu/data/amazon-meta.html. This dataset contains product metadata and review information for 548,552 different products (Books, music CDs, DVDs and VHS) with 7,781,990 reviews. For each product, the following information is available: title, sales rank, list of similar products (that get co-purchased with the current product), product categorization and product reviews, including time, customer, rating, number of votes and number of people that found the review helpful. The data(collected in summer 2006) is interesting because it contains information on reviewers and products reviewed, but also contains information on similar products.

3.1.1 Dataset Analysis

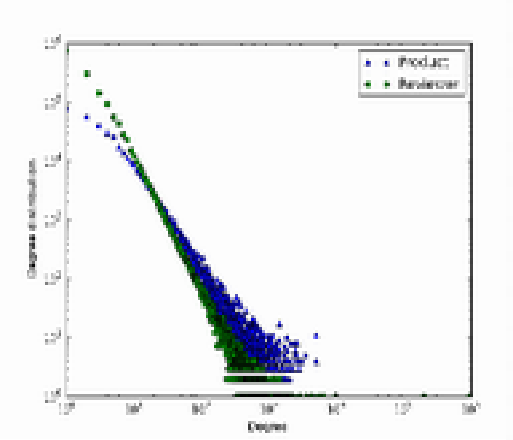

Fig 1. Degree Distribution on entire dataset

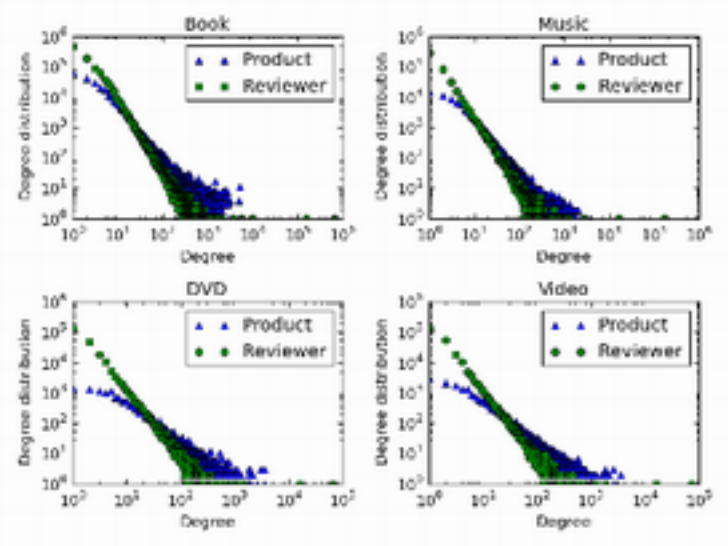

Fig 2. Degree Distribution by major category

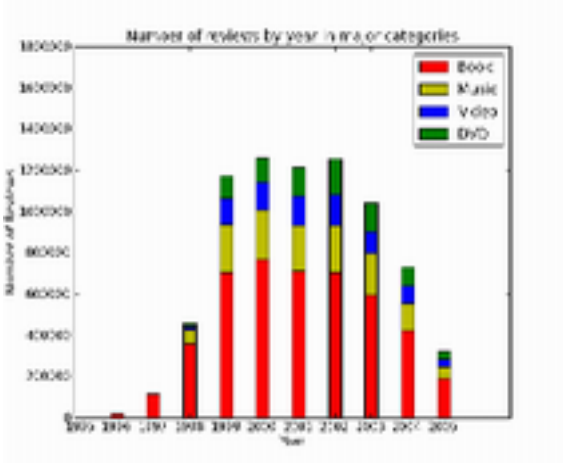

Fig 3. Number of review by year in major category

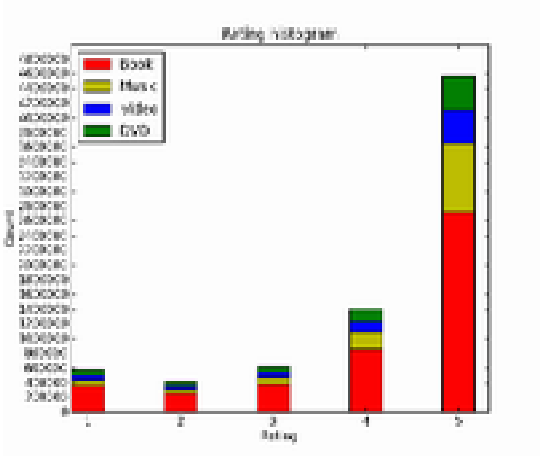

Fig 4. Review rating by major category

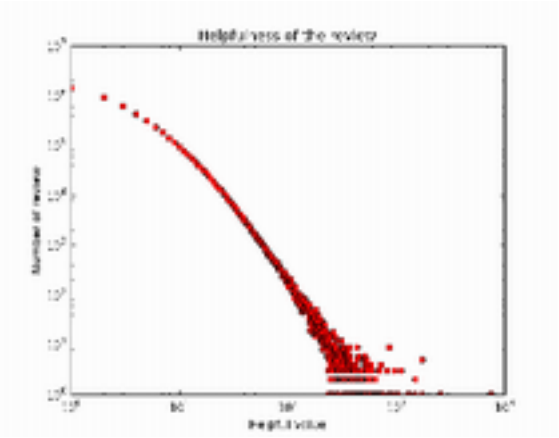

Fig 5. Distribution of helpfulness of the review

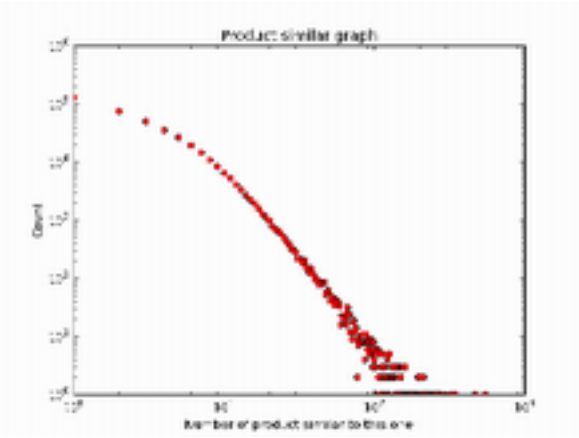

Fig 6. Product similarity graph

3.2 Core Creation

Social Networks constantly evolve over time. Therefore it is essential that we seek predictions for edges whose endpoints are present in both the training interval and the test interval.
The set *Core* is defined to be the set of all edges that are present in both the training dataset and the test dataset. This is similar to the approach followed in [1]. Since ours is a bipartite graph, we define *Core* to be the dataset which has reviewers and products that are common to both training and test datasets. We aim to predict new edges between the nodes in *Core* and evaluate how accurate our predictions are.

Needless to say, the Amazon Dataset is huge. Moreover, to compound the immensity of the dataset, we also have a data sparsity problem where in most reviewers review only a few products. Some of our algorithms, noticeably, Katz requires computation on all graph paths so we could not do the computation on large graphs. We were forced to find a size that works for all algorithms. Therefore, we created a dataset that has only 10,000 products or less and used this as the basic dataset to define *Core*

4 Project Methodology

4.1 General Outline

Core data is divided into training period and testing period according to the timestamp associated with the customer review posted on Amazon. The training and testing data is then used to create training and test bipartite weighted networks. The set of top nodes in the bipartite graph represents the reviewers and set of bottom nodes represents the products. An edge from reviewer A to product 123 specifies that reviewer A reviewed product 123 - and is weighted by the number of such reviews. For every pair of nodes(x,y) where x is a Reviewer and y is a Product, graph proximity measures like Common Neighbors, Adamic/Adar, Preferential Attachment, Katz etc are calculated. Candidate edges that connect pairs of nodes are sorted in decreasing order of the graph proximity scores. Then E_{new} high-ranking link candidates are predicted, where $E_{new}$is the number of newly generated edges in testing period. Accuracy of the prediction is calculated as the number of correctly predicted links divided by E_{new}

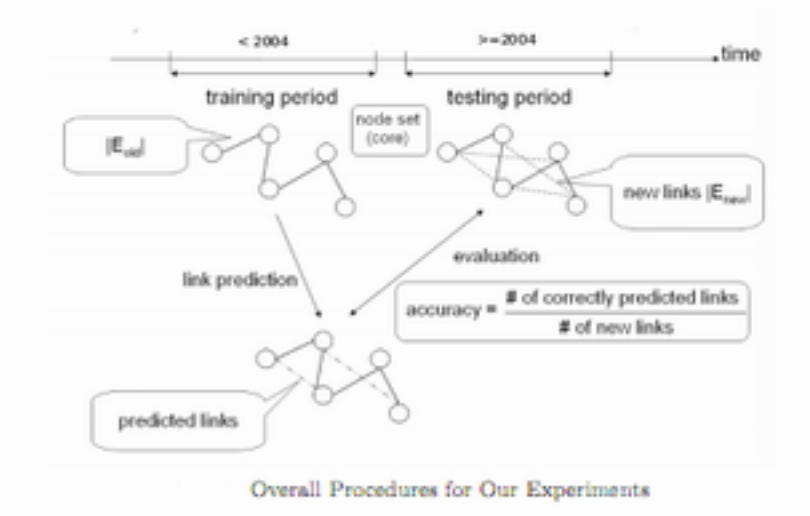

Fig 7 Image Courtesy of [2]

4.2 Details

- G_{amazon} <X,Y,E> = *Amazon Reviewer - Product Bipartite Graph. X* is the set of Reviewers and *Y* is the set of Products. *X* and *Y* are mutually exclusive and *E* is a subset of *X x Y.*
- Define G_{train} <X,Y,E> a training data set and G_{test}<X,Y,E> a test data set. We found that dividing the data set temporally was the best. All reviews before 2004 are part of training data set and all reviews after 2004 comprise test data set.
- For both G_{train} and G_{test} calculate $G_{train\text{-}core}$ and $G_{test\text{-}core}$ This is the set of reviewers and products which are present in both training and test data sets.
- For both $G_{train\text{-}core}$ and $G_{test\text{-}core}$ calculate the number of edges, E_{old},in the graph.
- Compute the number of edges,E_{new},, that are present in the test set but not in the training set.

- Obtain unimodal graphs, which are projections of the bipartite graphs, over its 2 node sets. The projection over the X set is defined by a unimodal graph where nodes from X are tied if they are linked to at least n common nodes in the initial bipartite graph. $\Gamma_{G(x)}$ is the projection of Reviewers and represents the neighbors of *x* (x is a reviewer) in G, the bipartite graph. $\Gamma_{G(y)}$ is the projection of Products and represents the neighbors of *y* (y is a product) in G, the bipartite graph.
- Compute Common Neighbors, Adamic/Adar, Preferential Attachment, Katz scores on the training sets in both weighted and unweighted modes
- Sort the scores in decreasing order and pick the top E_{new} scores as the new links predicted.
- Test the new links predicted against the test set to see how accurate the prediction was

For linear regression, we found that, of the graph based algorithms, Katz outperformed the other algorithms by such a high margin, that it did not perform much better than Katz alone. All the other features we propose require a graph of high degree, which the above process did not give us. Hence we used the following process for linear regression:

- Pick a subset of the Amazon dataset. Compute features for all combinations of reviewer and product.
- Recursively remove all products and reviewers who have degree less than 5. The remaining bipartite graph is the graph we used for our experiments.
- Randomly remove 20% of the edges - this now constitutes our Test set, while the remaining edges constitute the Training set. The edges in Test constitute E_{new}
- Perform the training portion of linear regression using the Training set. Randomly include an equal number of negative test cases as well.
- Given the computed weights for each feature, compute predicted values for the items in the test set, as well as all remaining negative test cases.
- Sort the predicted values in decreasing order and pick the top E_{new} scores as the predicted links.
- Test the new links predicted against the test set to see how accurate the prediction was

$$Accuracy = \frac{\#\ of\ correctly\ predicted\ links}{\#\ of\ links\ in\ Test}$$

4.3 Degree Distribution

Degree distributions for the training and test dataset is outlined below. Clearly the distribution of reviewers and products in G_{amazon} *<X,Y,E> = Amazon Reviewer - Product Bipartite Graph* follows a power law.

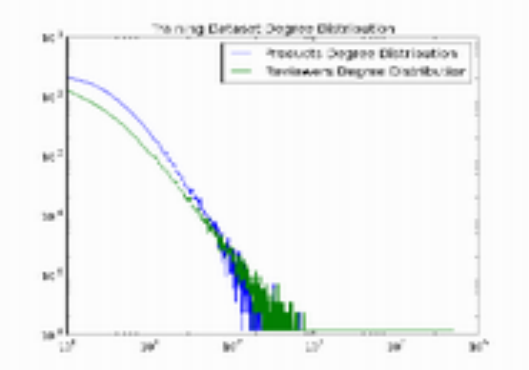

Fig 8. Degree Distribution of Training Dataset

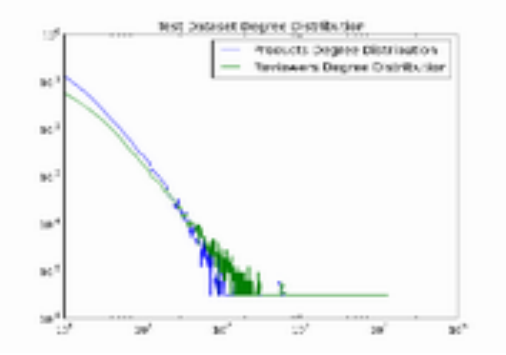

Fig 9. Degree Distribution of Test Dataset

5 Descriptions of Algorithms

We start with several basic methods for link prediction like Common Neighbors, Adamic/Adar, Katz Clustering and Preferential Attachment. Below we discuss the algorithms we use in detail.

Methods based on node neighborhoods.

1) Common Neighbors: In a unipartite graph, the number of common neighbors of x and y is

$$score(x,y) = |\Gamma(x) \cap \Gamma(y)|$$, where $\Gamma(x)$ represents the neighbors of x

$\Gamma(y)$ represents the neighbors of y

For our bipartite graph, we adapt this to

$$score(x,y) = max\,|\Gamma(x) \cap \Gamma(x_i)|$$, where $x_i \epsilon\, \Gamma(y)$ (or common neighbors of the reviewer)

$$score(x,y) = max\,|\Gamma(y) \cap \Gamma(y_i)|$$, where $y_{i\epsilon}\, \Gamma(x)$ (or common neighbors of the product)

2) Adamic/Adar- The Adamic/Adar score computes the features shared by objects and defines the similarity between them as

$$score(x,y) = \sum_{z:feature\,shared\,by\,(x,y)} \frac{1}{log(frequency(z))}$$

In our context, the object is a node and its features are its neighbors. However, we cannot use the formulation as is, since we are working with a bipartite graph. We have adapted the formulation accordingly.

Formulation for the direct topological attribute computed in the projected graph G_x

$$score(x,y) = max\, x_{i \in \Gamma_G}(y) \sum_{z \in (\Gamma_{Gz}(x_i) \cap \Gamma_{Gz}(z))} \frac{1}{log|\Gamma_{Gz}(z)}$$

where $\Gamma_G(x) - Neighbors\ of\ x\ in\ graph\ G$

$\Gamma_G(y) - Neighbors\ of\ y\ in\ graph\ G$

$\Gamma_{Gx}(x) - Neighbors\ of\ x\ in\ the\ projection\ of\ graph\ G\ on\ X$

$\Gamma_{Gy}(y) - Neighbors\ of\ y\ in\ the\ projection\ of\ graph\ G\ on\ Y$

A high score is computed if a reviewer who has written on y has a lot in common with x, i.e, has reviewed the same products as x with reviewers who rarely review being weighed higher.

Formulation for the direct topological attribute computed in the projected graph G_y

$$score(x,y) = max\, y_{i \in \Gamma_G}(x) \sum_{z \in (\Gamma_{Gy}(y_i) \cap \Gamma_{Gy}(y))} \frac{1}{log|\Gamma_{Gy}(z)|}$$

A high score is computed if a product x has reviewed has also been reviewed by a lot of people who have reviewed y with rarely reviewed products being weighed higher.

3) Preferential attachment: has received considerable attention as a model of the growth of network. The basic premise is the probability that a new edge involves node x is proportional to |Γ(x)|, the current number of neighbors of x. The probability of customer x reviewing product y is correlated with common reviews of other customers who have already review the product y.

$$score(x,y) = |\Gamma(x) * \Gamma(y)|$$

where $\Gamma(x)$ - neighbors of x , $\Gamma(y)$ - neighbors of y

Methods based on the ensemble of all paths

1) Katz: defines a measure that directly sums over the collection of paths. exponentially damped by length to count short paths more heavily.

$$score(x,y) = \sum_{i=1}^{\infty} \beta^i * |paths_{x,y}{}^{(i)}|$$

Where $paths^{\langle i \rangle}_{x,y}$ is the set of all length *i* path from x to y.In our amazon reviewer-product bipartite graph x is reviewer and y is the product. We only consider unweighted Katz measure, which $paths^{\langle i \rangle}_{x,y}$**=** 1 if x has reviewed y. Otherwise it is 0.

Linear Regression

Our linear regression model: $$z(x,y) = \sum_{k=1}^{p} \alpha_k score_k(x,y)$$ where $score_k(x,y)$ are properties of the review or of the reviewer and the product..

We used the following properties of the reviewer, the product and the review in the linear regression.

1) Distance on the similarity graph: We propose that the probability of a reviewer reviewing a product is proportional to how similar this product is to products the reviewer has already reviewed. Hence,

score(x,y) = min(length of shortest path on product similarity graph(neighbors(x),y))

2) Category: We hypothesize that reviewers tend to review items in the same category. Hence,

$$score(x,y) = \frac{|category(\Gamma(x)) \cap category(y)|}{|\Gamma(x)|}$$

3) Salesrank: Salesrank measures how popular an item is. We hypothesize that people who have bought (and reviewed) popular items will probably do so again. Hence,

$$score(x,y) = 1 / abs(salesrank(y) - \frac{\Sigma salesrank(\Gamma(x))}{|\Gamma(x)|})$$

4) Year: Most reviews get written at the peak of a product's popularity, and most reviewers tend to write reviews in a single timeframe. Hence,

score(x,y) = 1/abx(avg year(reviews(x)) - avg year(reviews(y)))

5) Rating: We postulate that there is a correlation between the average rating given by a reviewer, and the average rating of a product - i.e., if a product is highly rated, and the reviewer tends to write only negative reviews, it is unlikely they have reviewed this product. Hence,

score(x,y) = 1/abs(avg(ratings of reviews written by x) - average rating(y))

6) Degree: This measures the similarity of the reviewer and the types of people who have reviewed y, as well as the similarity between the product and the types of products x has reviewed. Hence,

$$score(x,y) = \frac{1}{abs(deg(x) - avg(deg(\Gamma(y)))} \times \frac{1}{abs(deg(y) - avg(deg(\Gamma(x)))}$$

In addition, we used properties of the graph we have computed already: common neighbors, Adamic-Adar, and Katz.

6 Experimental Results

Based on the computed Core Dataset from the Amazon meta data, we have the following results for our experimental study. Table 1 shows our experimental results. We experimented with different sizes of data. The number in the heading corresponds to the number of nodes in our data set. **Bold** means the highest performance.

Algorithm	**500**	**1000**	**2000**	**3000**	**4000**	**5000**
Random	3.85%	1.99%	0.75%	0.62%	0.46%	0.16%
Common Neighbors of the reviewer	0%	2.98%	4.27%	4.01%	3.90%	4.01%
Common Neighbors of the product	0%	0.99%	3.01%	1.85%	11.72%	9.83%
Preferential Attachment	15.6%	6.32%	2.98%	3.55%	3.37%	6.67%

Adamic/Adar for reviewers	15.7%	10.12%	4.18%	4.67%	4.1%	8.45%
Adamic/Adar for products	7.84%	4.4%	6.8%	4.2%	3.15%	3.67%
Adamic/Adar similar products reviewer	0%	0%	0.9%	1.24%	2.1%	7.08%
Katz (unweighted) $\beta = 0.05$	**20.5%**	15.9%	11.56%	7.25%	6.55%	6.77%
Katz (unweighted) $\beta = 0.005$	**20.5%**	**17.4%**	**14.31%**	9.87%	9.66%	9.60%
Katz (unweighted) $\beta = 0.0005$	**20.5%**	**17.4%**	14.07%	**10.4%**	**9.89%**	**10.4%**

Table 1. Experimental Results

Linear Regression

We ran Linear Regression on the 500 size dataset, using the results of Katz, Adamic-Adar and Common Neighbors. This gave us results comparable to Katz. We could not compute the other features on this graph because most of the nodes in the graph had low degree.

We then ran Linear Regression, starting with a 10,000 node dataset. When pruned, it had 1170 reviewers, 673 products, and 10,357 reviews. We could correctly predict **9.3%** of new reviews, compared to **0.04%** which could be randomly predicted. We discovered that the highest correlation existed between distance on the product similarity graph, followed by category of product - showing that reviewers tend to review similar products, and tend to review products in the same category. High correlations were also found with the salesrank feature - people who bought and reviewed niche products tended to continue doing so. The lowest correlation was on year - we found that people who wrote a lot of reviews continued to do so over time.

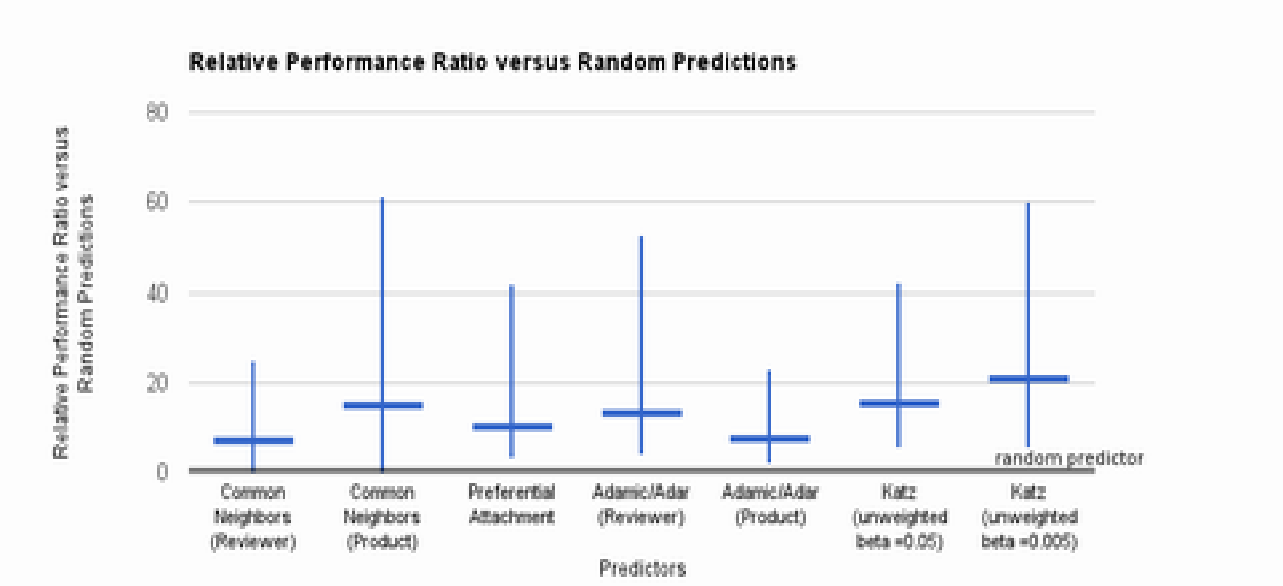

Figure 10: Relative average performance of various predictors versus random predictions. The value shown is the average ratio over the various datasets of the given predictor's performance versus the random predictor's performance. The error bars indicate the minimum and maximum of this ratio over the datasets

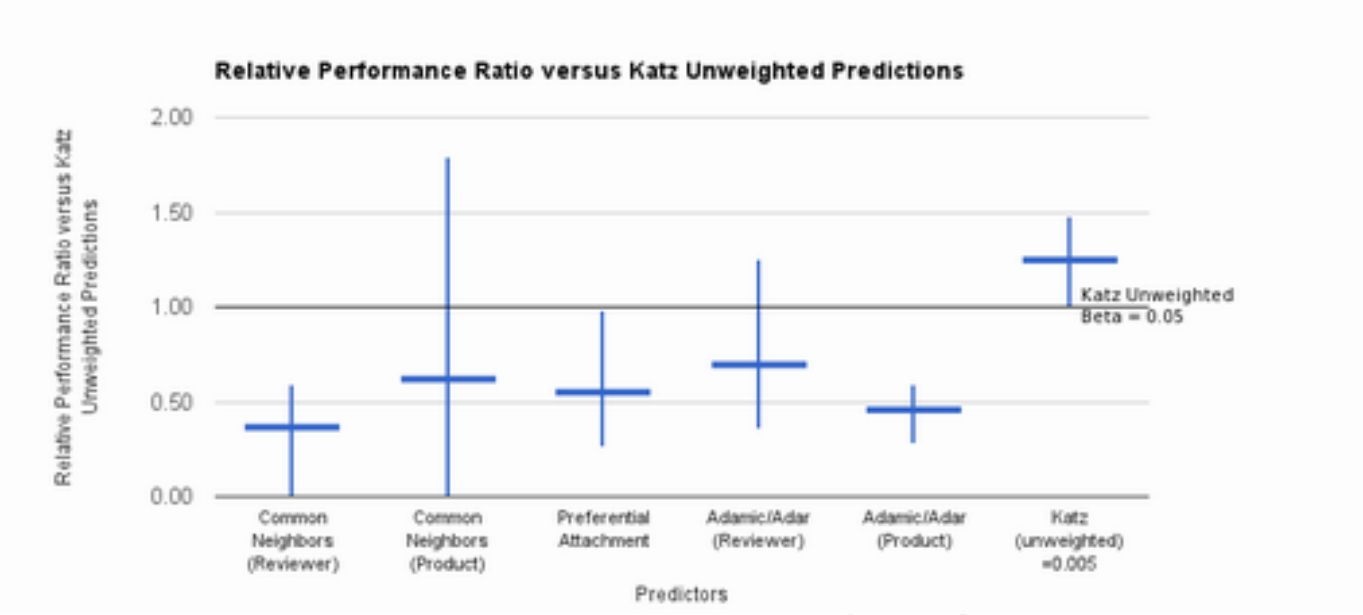

Figure 11: Relative average performance of various predictors versus Unweighted Katz ($\beta = 0.05$) predictions. The value shown is the average ratio over the various datasets of the given predictor's performance versus the unweighted Katz predictor's performance. The error bars indicate the minimum and maximum of this ratio over the datasets

7 Conclusions and Future Work

We have modified standard network distance measures for our bipartite graph. Katz consistently gives us the best performance, even on relatively sparse graphs - similar to the results seen in [1]. However it is computationally expensive. We found that for denser graphs, we could use measures that were computationally inexpensive, and still get good results for large graphs. We noted that product similarity was highly correlated with the probability of a new review being written.

Most link prediction papers we have read use either citation networks, which are relatively small, or subsets of social networks, like Facebook. Both types of graphs differ from review networks, in that review networks are usually much larger than citation networks, and have different characteristics from social networks. Measures, like common neighbors, which work very well with social networks, did not do very well with review networks. We found that for review networks, the best predictors were information about the reviewers and the products that could be recomputed as the network grows.

We would like to extend the Linear Regression measures to the entire Amazon dataset to see how it scales with size. We would also like to compare these results with Yelp reviews, where we may not have access to a similar product graph, but may have access to many of the same features we had with the Amazon dataset.

8 References

[1] D. Liben-Nowell, J. Kleinberg. The Link Prediction Problem for Social Networks. *Proc. CIKM,* 2003
[2] T. Murata, S. Moriyasul. Link Prediction based on Structural Properties of Online Social Networks
[3] Z. Huang, X. Li, H.Chen. Link Prediction Approach to Collaborative Filtering.
[4] N. Benchettara, R. Kanawati, C. Rouveirol. Supervised Machine Learning applied to Link Prediction in Bipartite Social Networks