The Predictive Power of Bias + Likeness

Combining the True Mean Bias and the Bipartite User Similarity Experiment to Enhance Predictions of a Rating

Hongze Lai (laihz)
Neel Murthy (nmurthy)
Nisha Garimalla (nisha)
CS224W Final Paper
12/11/11

Table of Contents

Introduction

In a survey from 2007, it was reported that 89% of consumers preferred online shopping, and out of these consumers, 91% used online shopping primarily for research purposes[1]. Upon glancing at these statistics, the average reader would perhaps primarily extract the belief that the world has become lazily dependent on it's computing devices to perform such quotidian tasks as shopping; however, a more savvy and perceptive reader would glean that these statistics are merely an indication of a heavily popular consumer trend to research products and services online that are prevalent not only in shopping, but also in several other areas such as travel, finance, and networking.

A fantastic demonstration of this consumer exploration, probing, and investigation phenomena is the Amazon recommendation and review system. This system consists of a community of Amazon members that are bound as a team by their goal of using Amazon as a platform to better buy, sell, and browse items of interest. This community is a pertinent example of a group that is to a large degree naked without the concept of intra-group collaboration; without the reviews of fellow Amazon members, the site would be no more than a bare-bones retailer as it would only contain generic information like size, type, genre, author, etc. The Amazon system of recommendation and review, therefore, is the focal point of informed purchases and investments.

In addition to having access to reviews, many companies like Amazon take it one step further by providing helpfulness ratings for each review. Though this measure may seem slightly too metaphysical and abstract to be meaningful, it has, in reality, proved helpful in weighing the relevance and appropriateness of a review and is a quite in demand and banked on feature of commercial evaluation systems. To demonstrate the value of this data, take the example of the common consumer action of sorting reviews based on rating while browsing reviews for a product. By doing this, consumers hope to control for and filter out irrelevant reviews. To a large extent, this approach works; it seem that we can trust that the highest rated ratings are a reflection of relevant content. However, to what degree can we trust that the lower rated reviews are a reflection of poor content and are we in fact missing out on more helpful reviews due to biased helpfulness ratings?

Motivation

In order to explore these questions further, we browsed several data sets and tried to seek out a general answer to the question: is there a linear relationship between the quality of a review's content and a review's rating? After analyzing the data from a birds-eye view, we became specifically interested in the relationship between a review's rating and it's creator's rating. This notion of a creator's rating is present in many popular evaluation systems and functions as a trust metric for a user, and in this case, for a user that has uploaded a review. Browsing different sites and data sets led us to form the hypothesis that that reviews might be skewed based on its' creator's rating. We referred to this belief as the *status-quo-bias-based intuition* (since we believe that status quo is the factor that makes consumers behave the way they do in this hypothesis). For example, even though review A and B might have the same content, A might be rated more helpful because A's uploader has a higher creator's rating than B's creator's rating. We had the notion that a person might review A to be more helpful because they see that it's reviewer seems to be a good reviewer and consequently, they are biased to the idea that the review is helpful without even seeing the content.

Consequently, our original project proposal was an analysis of how a user's rating will skew the perceived helpfulness associated with that rating. However, after feedback and further analysis of the project and available data, we decided that some our assumptions were not correct. The biggest obstacle we faced was that our original proposal assumed that the RevRank algorithm proposed by Tsur et al. was to be held as a true measure of content helpfulness. However, due to the fact that this algorithm is still young and we don't actually know its validity with respect to content, it is hard to base our project upon it. Furthermore after reading other papers that described the difficulty of controlling reviews for content, we decided that our project would turn into more of a Natural Language Processing analysis than a network analysis. Ultimately, since we could not use RevRank as a measure of content standardization or reliably standardize helpfulness based on true content, we decided against our original project proposal. Upon further research, we found a phenomena that was even more striking, the true mean bias. This is described below:

Work that Inspired Our New Proposal

Danescu-Niculescu-Mizil et al. 2009

In Danescu et al., the authors investigated the process of deducing how personal opinions are received and evaluated. In other words, more important than Y's opinion about Z, is X's opinion about Y's opinion about Z. Danescu et al. decided to focus on an Amazon.com data-set to examine tendencies in how "helpfulness ratings", ratings that measure the proportion of people that found a specific review helpful, acted with respect to issues like variance in overall ratings. The goal of this investigation was to produce a model for predicting what helpfulness ratings should look like. To begin, Danescu explored social psychology concepts, namely the conformity hypothesis, individual-bias hypothesis, brilliant-but-cruel hypothesis, and quality-only straw-man hypothesis. The conformity hypothesis results in higher helpfulness ratings when the ratings assign a star rating close to the mean star rating for the product. The individual-bias hypothesis states that a higher helpfulness rating is given to those that share the opinion of the review rater. The brilliant-but-cruel hypothesis illustrates that negative reviews tend to be thought of as more insightful and consequently receive higher helpfulness ratings. By contrast, the quality-only straw-man hypothesis stresses that the only real factor in helpfulness ratings is the actual helpfulness of the review.

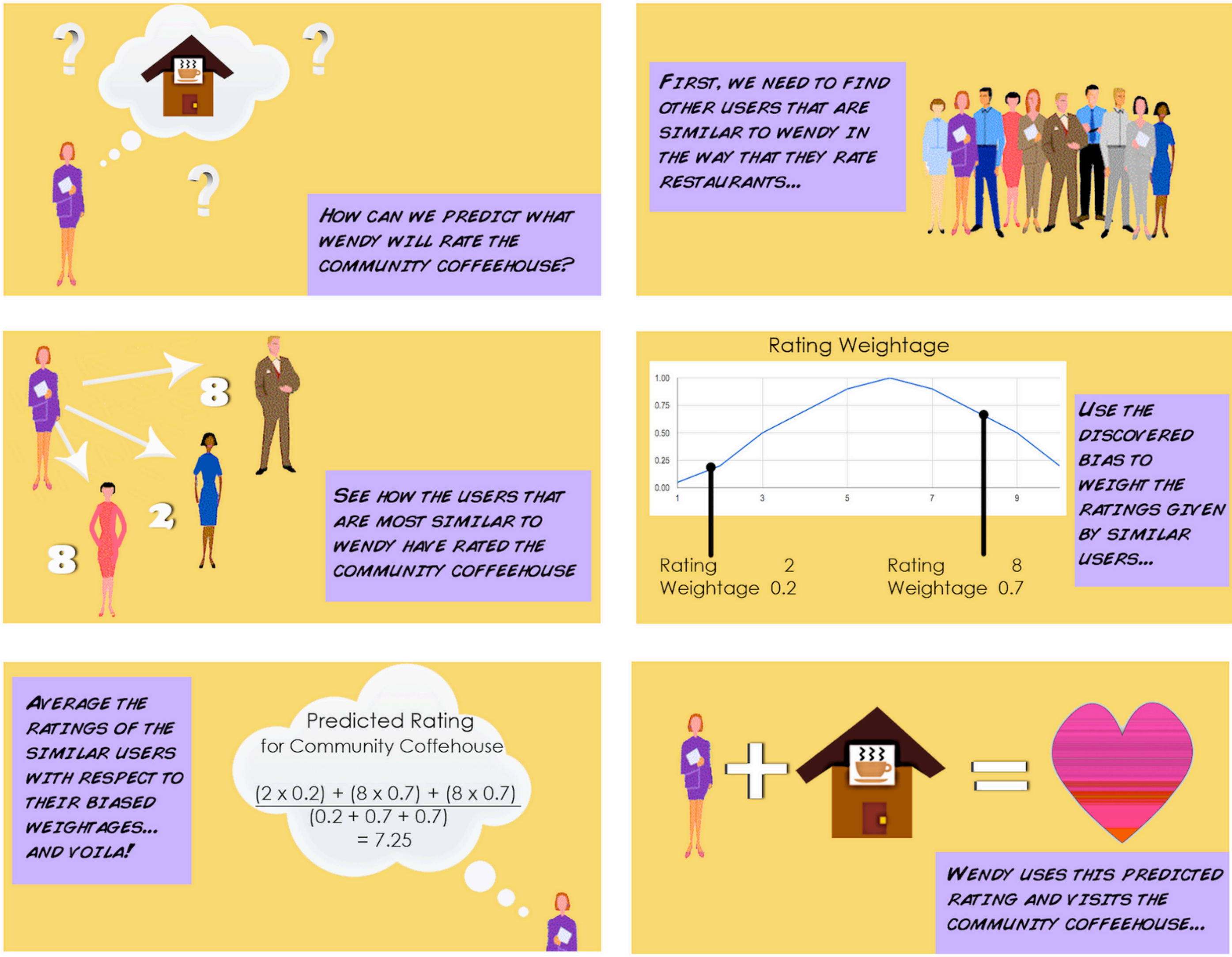
By examining a data-set of 4 million reviews on 675,000 books, Danescu et al. then tried to support the conformity hypothesis in an effort to create a global model for this helpfulness metric. The findings resulted in the conclusion that while none of these hypotheses were entirely correct, each could contribute specific components to the desired model. The major findings were that helpfulness ratio did in fact decline with the absolute value of a review's deviation from the average star rating (which supported conformity), and the helpfulness ratio depended on the signed deviation from average, that is, slightly negative reviews were actually punished more than slightly positive reviews (which slighted both conformity and brilliant-but-cruel). From these results, the authors were in fact able to come up with an appropriate model that utilized conformity with a positive bias, as well as a variance-based adjustment.

Problem Definition

We wanted to extend the Danescu-Niculescu-Mizil et al. work by utilizing a more generalized network, the user/item bipartite graph where users are linked to the items they review. We used the network to generate similarity scores between users or items. With such a similarity metric, we wanted to ask the question of whether this similarity has any effect on the helpfulness of reviews.

Thus, for a given category (for instance, type of place in Yelp), we wanted to compare the correlation between the perceived helpfulness of a rating and its deviation from the average. So, we took the rating associated with the review and looked at the difference from the average rating and plotted helpfulness against the standard deviation from the average.

After discussing our project milestone report with Professor Leskovec, we decided to extend this project from a discount-factor-oriented project to a predictive power project. Essentially, the goal was to be able to predict a user's rating of a reviewable object/place. We wanted to use the average of the most similar users ratings of the given object/place, weighted by the true mean bias, to predict what a certain user's rating of a restaurant would be.
Here is a very basic sketch of what we planned to be able to do:

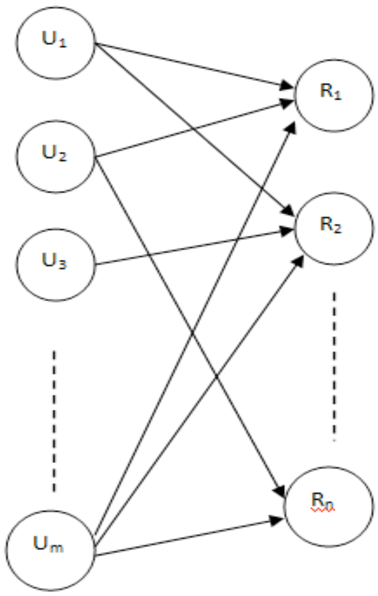


Models, Algorithms, Methods

In order to concretize our methods for this project and formulate how we would evaluate our project, we used the following four models:

Bipartite Graph

Many real-world networks are naturally bipartite graphs where their nodes are separated into two different classes with edges linking only nodes from both classes. They are also called 2-mode networks. For example, online communities where users give reviews on restaurants, p2p networks where users share files, and the actor affiliation network in movies are this kind of networks.



The bipartite graph above can be defined as $G = \{U, R, E\}\ where\ E \subseteq U \times R$. The difference between a bipartite and a classical graph is that in the former, edges can only link nodes from different classes but never those in the same class.

Bipartite networks have some interesting properties but are generally more difficult to analyze than classical 1-mode networks. There are two approaches: the first one is to project the 2-mode network to either class of nodes then use tools and methodologies developed for classical networks. However, as pointed out by Latapy et. al, the projection can introduce some subtle yet important changes to network characteristics. The second approach is to try to adopt or extend concepts and methods from 1- mode networks, and to perform analysis directly on bipartite graphs. In our analysis, we decide to take the second approach.

SimRank

SimRank is a similarity measurement between nodes derived solely from the network structure. The basic principle of SimRank is simple: two nodes are similar if they reference or being referenced by similar nodes. The definition is recursive in nature but the algorithm converges quickly for most networks as shown in Jeh et. al. SimRank equations applied to the bipartite graph can be written below. For $U_i \neq U_j$

where $O_i(U_i)$ represents the *i*th node in U_i's out-neighborhood.

and for $R_i \neq R_j$

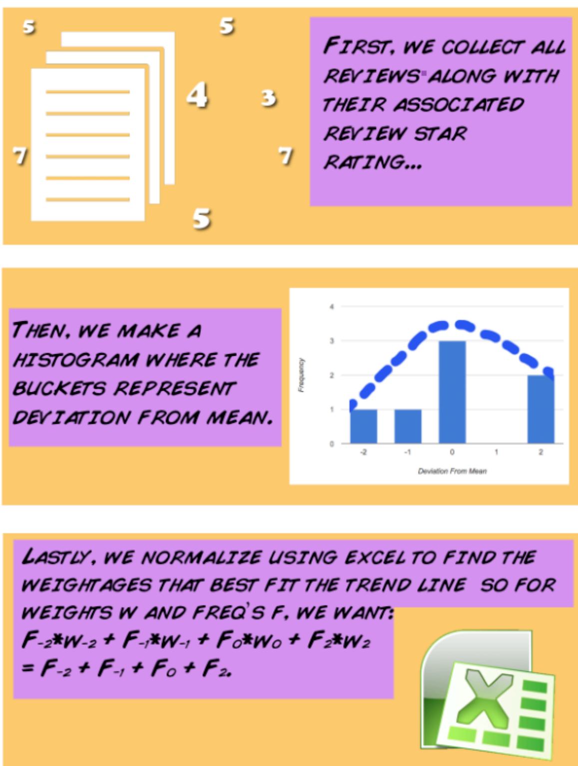
where $I_i(R_i)$ represents the *i*th node in R_i's in-neighborhood.

If $U_i = U_j, s(U_i,U_j) = 1$, and the same is for $s(R_i,R_j)$. $C_1\, and\, C_2$ are decaying constants which can be set to *0.8* for practical purposes.

It was a natural extension to the previous experiments to use bipartite analysis on the Yelp dataset. We also used the SimRank algorithm to find the top 5 or 10 most similar users to a user associated with a randomly picked edge (review) and then tried to predict the rating score on the edge. The relevance of this analysis is based on the following hypothesis.

User Consistency Hypothesis: The assumption is that similar users tend to give same rating to similar restaurants. Since users are consistent in giving scores, it should be possible to predict the rating given by other similar users to the same or similar restaurants.

Weighting



We generated a system in order to weight reviews according to the bias. Here is the procedure:
First, we create a bucket for each discrete value (for example, -1, 0, 1, …). Then, we calculate the signed difference between star rating for each business and each review of that business and assign it to the appropriate bucket. After that, we plot the average helpfulness (or quartile values) against buckets of rating differences and normalize the weights to be centered around 1. That is to say, when the weights are applied to the aggregate totals, we should generate the same review ratings as the case where there are no weightings at all.
For the Danescu model weights, we followed a similar procedure but changed the trend-line so that it reflects the expected behavior as dictated by Danescu et al.

Accuracy Method
We decided to measure accuracy of our predictions by using the percentage error between our predicted ratings and our actual ratings of a restaurant:

$$\%\,Error = |predicted\ star\ rating - actual\ star\ rating| \div actual\ star\ rating$$

For instance, if our algorithm gives us an expected rating of 10 and the actual rating is 5, our delta is 5, and the percentage error is 100%. Our accuracy measurement is the average of the percentage error computed for several iterations.

Set of Experiments
Using these methods, we structured our project into three experiments, the True Mean Bias Experiment, the User Similarity Prediction (without weighting), and the User Similarity Prediction (Enhanced with True Mean Bias Weighting).

Experiment 1: True Mean Bias
Goal: We wanted to confirm that the notion of the true mean bias (from Danescu et al's findings) held in the Amazon and Yelp data sets.
Methods: Before starting the main experiment and trying to reach our goal, we first had to get the data, and then clean and sanitize it. Yelp is the ideal data set (composed of 65888 users, 6898 restaurants, and 152327 reviews with star ratings) in terms of extending previous experiments. The original problem with this data set was that the API limits results of requests to 20. Consequently, getting a meaningful data set was difficult as they could block a script from running and we need to gather and sanitize 1000's of objects. However, later we got access to an academic data set from Yelp.

We also wanted to use the Amazon data set to corroborate our findings from Yelp and prove that the trends found in Yelp have some basis of comparison in other data sets.

After getting these two data sets, we wrote a program to validate the data sets for relevant data, checking if both user id and item id were present for every object. After this, we cleaned the data using a python script. Here is what the data looked like before:

{"votes": {"funny": 1, "useful": 8, "cool": 1}, "user_id": "TFm6azL9LLpljgVKx-Vfkw", "name": "E L.", "url": "http://www.yelp.com/user_details?userid=TFm6azL9LLpljgVKx-Vfkw", "average_stars": 3.5, "review_count": 12, "type": "user"}

and we sanitized it to look like this:

funny useful cool stars date categories user_id review_id business_id
1 1 1 1 2008-01-28 Restaurants/Delis jqhP9mV2rYvmPdKvlOfp0g
KhUJX_0L7Mf-iArcFaTpAg iZYDZvXoIT648EZOnEP0pQ

In order to generate a plot to compare with the Danescu model, we looked at the difference between review rating and the mean rating assigned to the reviewable object. We discretized these values and created different buckets to correspond to the values in the following procedure:

1. *Create a bucket for each discrete value (for example, -1, 0, 1, …)*
2. *Calculate the signed difference between star rating for each business and each review of that business and assign it to the appropriate bucket*
3. *Plot the average helpfulness against buckets of rating differences*

Then, we looked at trends in useful (aka helpful), funny, cool votes assigned to reviews in this category.

Here is the graph that was generated using the bucket algorithm described above:

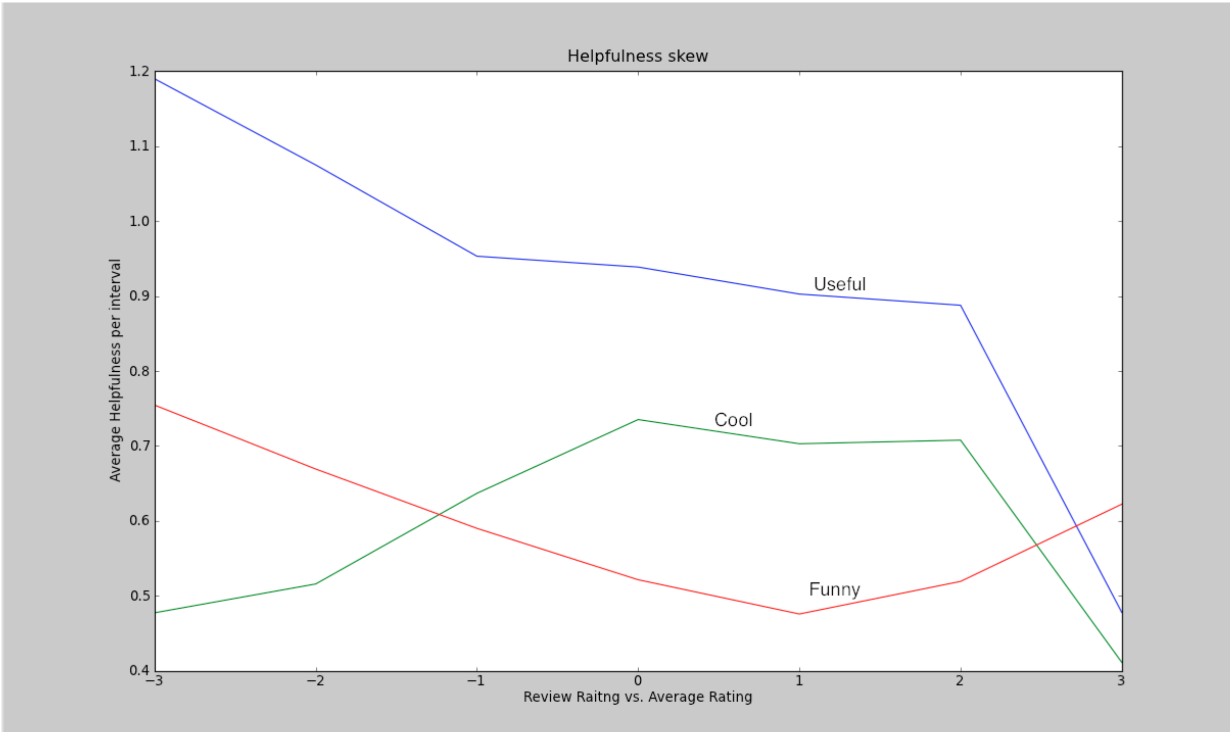


Figure 1: The skew in perceived useful (aka helpfulness), cool, and funny votes as demonstrated by the bucket algorithm. As we explained above, useful follows an interesting negative mean bias. Cool follows the true mean bias, while funny is the inverse of that.

<u>Analysis</u>

In analyzing this data, we found some interesting results. We were able to duplicate the findings of Danescu et al. in certain cases by verifying that there is a spike in the curve of helpfulness vs. rating where the rating is closest to the mean rating. For example, the cool rating that Yelp allows its users to bestow on the reviews of others follows this predicted trend. However, one of the interesting observations was that the funny rating (when a user marks the review funny) followed an inverse relationship, that is reviews that assigned a rating further away from mean were found to be more funny. We explain this finding by postulating that there is a correlation between assigning a rating further off than expected and written text that entertains other users. That is someone that disparages a restaurant with high ratings is likely to do so in a dramatic fashion as is someone that praises a restaurant with low ratings. The most unexpected result, however, was that helpfulness ratings actually didn't follow the Danescu "spike" pattern. The trend we observed was that as the reviews became more negative with respect to the actual rating of the object, the review was perceived to be more helpful. For that reason, we decided to examine the Amazon data set to look into this issue further.

Running the Algorithm with different Data

In analyzing the Amazon data set we were able to come up with a conclusion that bridged the gap between our finding and Danescu et al. With the Amazon data set when we looked at average helpfulness percentage (that is the number of helpful votes/ total votes on a review), we we're able to reproduce the spike in helpfulness around mean.

Graphs

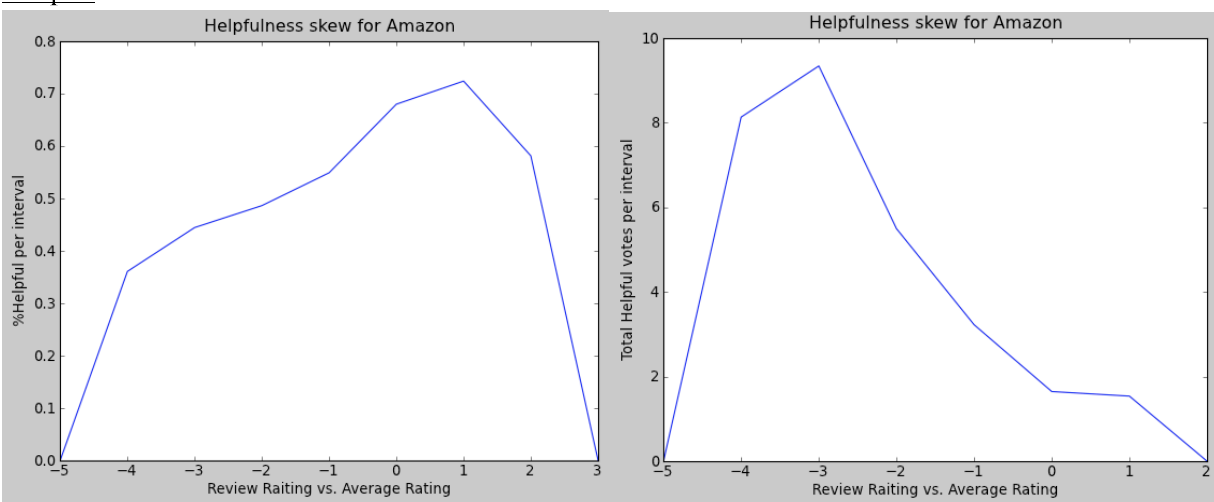


Figure 2 (a) (b): Plots of the helpfulness skew for Amazon (used to corroborate the Yelp findings). Figure 2a maps %helpfulness versus the discrete buckets, whereas 2b plots absolute helpful votes versus the buckets.

Analysis

In this analysis, we were able to corroborate both the negative mean bias (from the Yelp findings) and spike bias (from the Danescu et al. paper). When we looked at overall %helpful for Amazon reviews (that is helpful votes out of total votes), we are able to see the trend proposed by Danescu. However, when we looked at total helpful votes per discrete interval, we saw the same trend that we saw in the Yelp data set. That is, we saw a negative bias where there were more helpful votes as the review got more negative as compared to the average rating. We hypothesize that the negative bias trends for helpfulness are based on the fact that people tend to be more vocal about negative feedback. In Amazon, there were not only a lot more people that found negative reviews helpful, but also a lot more people that found them unhelpful (as shown by the fact that absolute votes were much higher than positive, but average helpfulness followed the Danescu spike). Similarly, in Yelp helpfulness was perceived to have the same pattern.

Experiment 2: User Similarity Prediction

Goal: We wanted to predict what a given user would rate a reviewable business based on how other similar reviewers rate similar businesses.

Method:

1. Cleanse Yelp academic data set.
2. Generate a bipartite graph with users and restaurants as two classes of nodes, and with reviews as edges.
3. Plot the degree distribution of the bipartite graph.
4. Remove users who wrote less than 5 reviews and restaurants with less than 10 reviews. Those nodes are not well connected so should be safe to ignore in similarity score calculation.
5. Choose an edge at random. Mark the user and restaurant for this edge.
6. Find the top 5 most similar users to the randomly chosen user with the SimRank algorithm.
    a. find all restaurant nodes that the user links to. They are the user's first neighbors in the bipartite graph.
    b. find all users that link to all restaurants in the user's first neighborhood. They form the user's second neighborhood.
    c. remove the restaurant node and associated edges so we are unbiased
    d. use SimRank to calculate similarity score between users in the second neighborhood and restaurants in the first neighborhood. We can assume that users outside of the neighborhoods are very different and set the similarity scores between the chosen user and any user outside the neighborhood to 0.
    e. sort the similarity scores in descending order and get the top 5 or 10 most similar users nodes.
7. Predicting user ratings
    a. Take a given user review
    b. Compute the average rating that top 5 or 10 similar users assigned to generate an unweighted rating prediction.
    c. Compare to the actual rating the selected user assigned and find the delta
    d. Calculate the average delta to get the predication error
8. Repeat the process of picking a random review many times (we chose 1000 iterations)
9. Calculate overall accuracy of predictions using the accuracy method proposed above

Graphs:

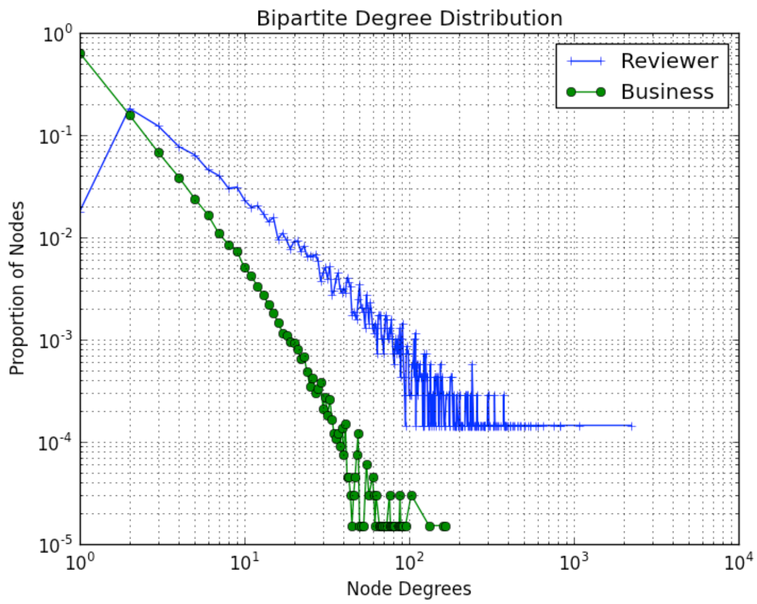
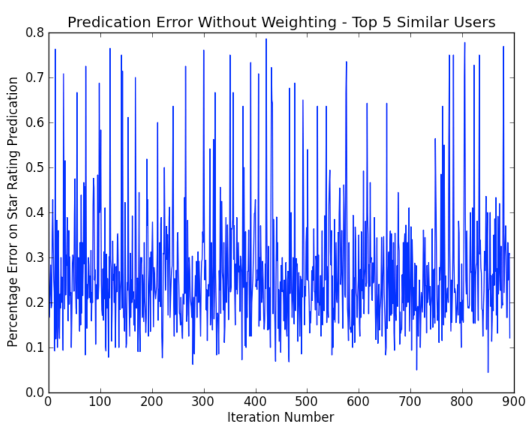


Figure 3: The degree distribution of reviewers and restaurants in the Yelp data set. The degree distributions of both reviewers and restaurants follow power law as shown figure 1. However, they have different scaling factors:

$$\alpha_{Rev} = 1.73,$$

$$\alpha_{Bus} = 2.59$$



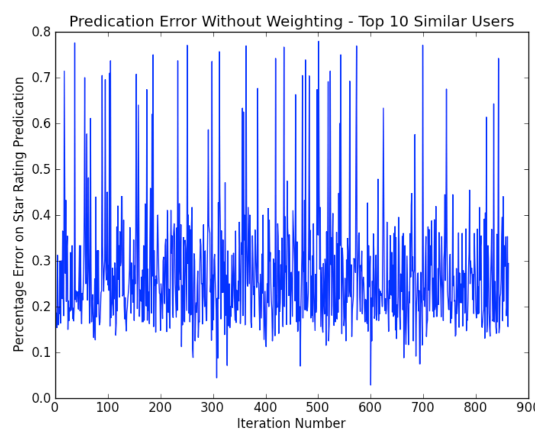


Figure 4 (a) (b): Plots of the percent error in the generated prediction as compared to the actual rating a review assigns. Figure 4a is the results when the top 5 similar users are selected and Figure 4b is the results when the top 10 similar users are selected.

Results and Interpretation:

After running this basic prediction algorithm, we were able to generate fairly accurate predictions. However, one important distinction between this experiment and the latter *Experiment 3* is that this version generated significantly more outliers. Almost 11% of our predictions had to be thrown out since the percentage error made the actual and predicted values polar opposites. After removing those outliers, the percentage error was centered around 27%, which is a pretty decent prediction rate. In Yelp, that translates to a little more than ½ a star off. While the aggregate predictive quality was quite good, we wanted to see what effect adding weighting to the reviews could improve predictability and decrease the number of outliers.

Experiment 3: User Similarity Prediction (Enhanced with True Mean Bias Weights)

<u>Goal:</u> Our goal was to combine the results of experiments 1 and 2 in order to enhance our predictive power.

<u>Methods:</u> Here, we followed the basic structure of *Experiment 2*, but also weighted the users using the negative True Mean Bias (found as a trend in Yelp data) and Danescu models.
Follow steps 1-6 from *Experiment 2*

7. Predicting user Ratings
    a. Take a random user review
    b. Find the top 5 most similar users who rated the same restaurant
    c. Apply a weighting based on the difference between the the restaurant rating and the rating the reviewer assigned (see Table 1 below for generated weightings)
    d. Compare to the actual rating the selected user assigned and find the delta
    e. Calculate the average delta to get the predication error
8. Repeat for many different reviews (we chose 1000 iterations)
9. Calculate overall accuracy of predictions using the accuracy method proposed above

Generated Weightings (applied in step 7c):

Difference between review rating and rating of the object	Method 1: Weighting informed by the Danescu model	Method 2: Weighting informed by normalized Yelp data	Method 3: Weighting informed by the raw Yelp model
<= -2.5	0.8	1.1	1.12
<= -1.5 and > -2.5	0.9	1.06	1.08
<= -0.5 and > -1.5	0.95	1.02	0.95
<= 0.5 and > -0.5	1.1	1.0	0.94
<= 1.5 and > 0.5	0.95	0.95	0.90
<= 2.5 and > 1.5	0.9	0.88	0.89
> 2.5	0.8	0.7	0.6

Table 1: The weightings we generated to adjust review influence. The first method reflects the Danescu et al. model for true mean bias and is normalized to a weighting of 1 (based on number of reviews in each category). The second method is normalized values for the Yelp negative mean bias that we observed. The last method is generated using the raw trend values that we observed in Experiment 1. Finally when applied to the review rating we actually use the inverse of these weightings since we are correcting for bias.

Graphs:

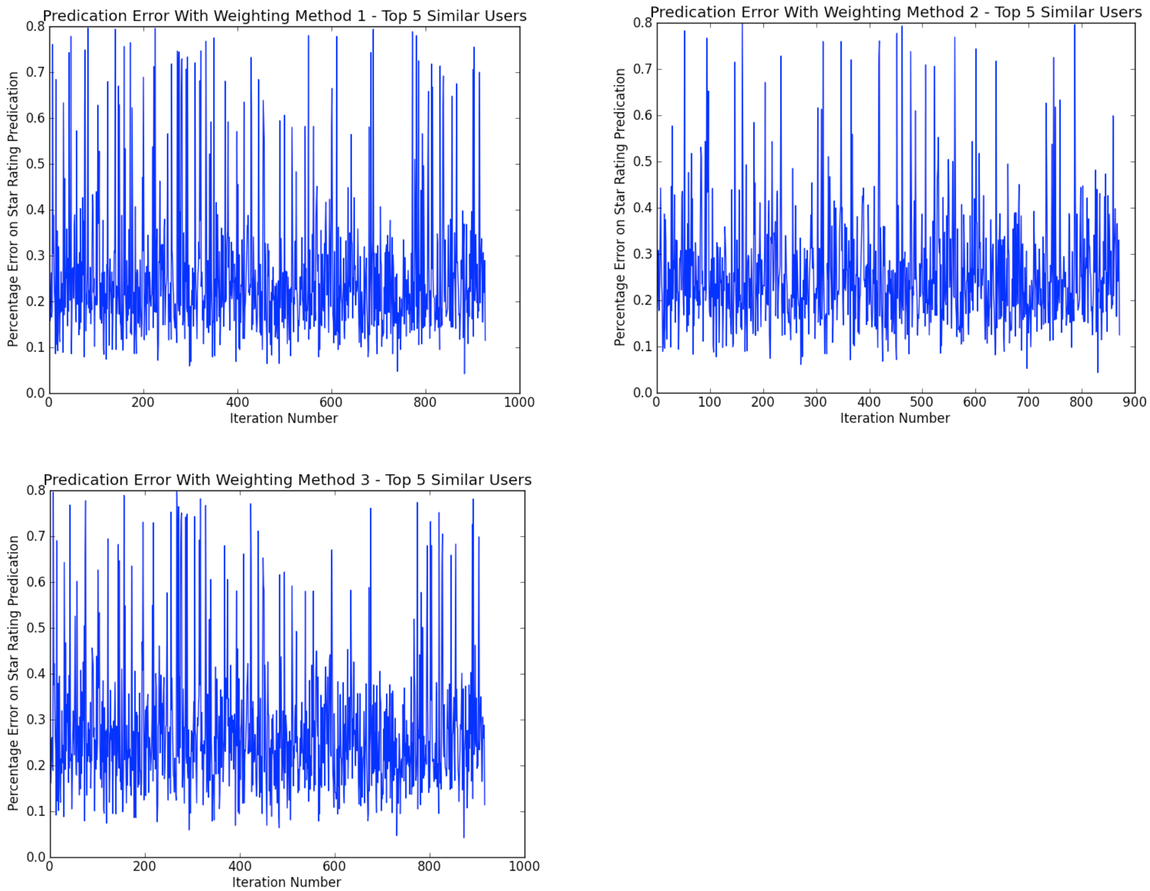
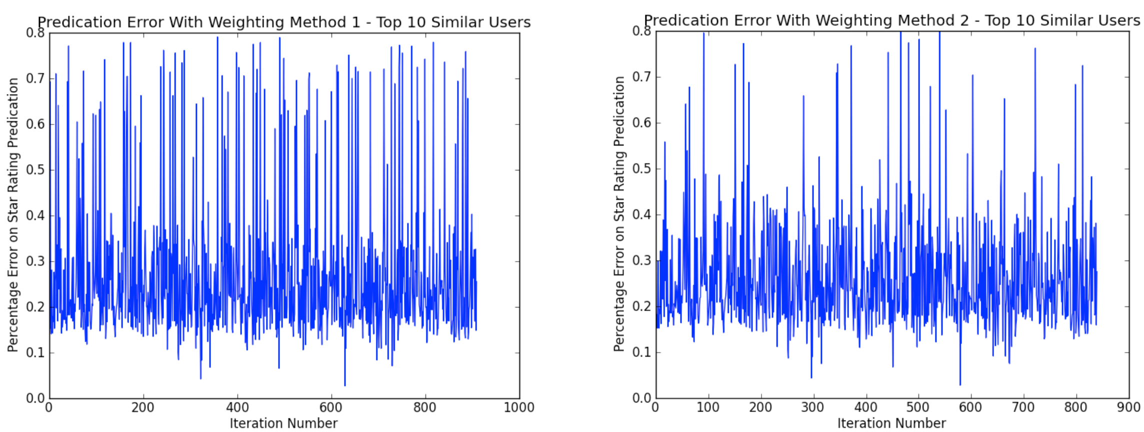


Figure 5 (a) (b) (c): Plots of the percent error in the generated prediction as compared to the actual rating a review assigns and weights them according to 3 different models (uses Top 5 similar users). Figure 5a is the results when the Danescu model is applied to the data Figure 5b is the results when the normalized negative mean bias trend is applied to the data and 5c is the results when the raw negative mean bias trend is applied to the data.



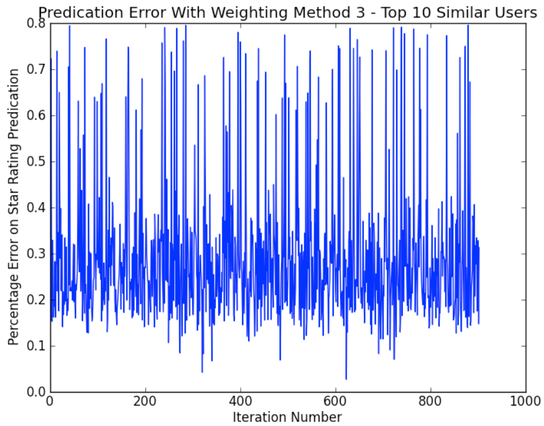


Figure 6 (a) (b) (c): Plots of the percent error in the generated prediction as compared to the actual rating a review assigns and weights them according to 3 different models (uses Top 10 similar users). Figure 6a is the results when the Danescu model is applied to the data Figure 6b is the results when the normalized negative mean bias trend is applied to the data and 6c is the results when the raw negative mean bias trend is applied to the data.

<u>Results and Interpretation:</u>

- Methods 1 and 2 decreased our percent error (only slightly) to 26%. Method 3 increased our percent error to almost 28%.
- Method 1 and 3 both decreased the number of outliers to ~7%
- Method 1 and 2 also both increased the number of times we predicted a star rating with error <25%. Method 1 did so more accurately with almost a 20% increase in number of predictions.

While these results weren't as strong has we hoped in terms of overall predictability, we were able to use them to finely tune the accuracy prediction and get fewer outliers (>40% reduction in outliers in the case of method 1). The interesting observation is that the true mean model was actually a better weighting system than the negative mean bias one. To explain this, we went back to our findings in the Amazon data set. While Amazon's data seemed to follow a Danescu model, we did notice that when looking at absolute votes it followed the negative mean bias that we observed in Yelp. Since Yelp doesn't allow users to negatively vote on a review (i.e. find it unhelpful), we hypothesize that if there was this ability, Yelp would also follow the Danescu model. Consequently, the Danescu model is actually a better representation for yelp data as well. Ultimately, we are able to in almost all cases (93% of the time) predict what a user will think of a restaurant within a ½ star.

Overall Conclusion

From our experiments, we were able to demonstrate 3 things:

1. *Negative mean bias leads to higher helpfulness*
2. *Danescu's model for mean bias can ultimately be corroborated with other data sets*
3. *Using our algorithm, we can predict user ratings with 74% accuracy (1/2 a Yelp star)*

When we initially set out to corroborate the Danescu model in other data sets, we found an interesting relationship, contrary to what they hypothesized would exist. We termed this trend negative mean bias, which seems to indicated that since people are more vocal about overall negative statements, we see higher levels of helpfulness in those interval. Even though these initial observations of the Yelp data demonstrated a new relationship between signed difference of a user rating and average rating versus the perceived helpfulness , we were ultimately able to prove the Danescu model does exist in both data sets. In the Amazon data set, we were able to explicitly duplicate the spike. However, in the Yelp data set, we were able to show that given the limitations of Yelp's functionality, we actually expect to see a negative mean bias curve rather than the spike predicted by Danescu. We also further hypothesize that if Yelp allowed users to find reviews unhelpful, then we would see a normal mean bias curve.

This led into the second part of our project where we tried to apply these findings to solve a puzzle - are we able to predict what a user will think about a reviewable object? For this, we used a bipartite graph SimRank algorithm along with a variety of prediction algorithms to determine user ratings. Our efforts led to prediction capabilities that determined what a user would think of an object/place with 74% accuracy (in Yelp, that translates to ~1/2 a star). Ultimately, by combining the analysis from the first part of our project with SimRank generated similarity scores, we were able to apply both our observed trends and those that have already been established, to generate quite accurate predictions.

While we were able to quite accurately predict what a given user would rate a business using our similarity scores and review weightings, there are still a couple issues that would be interesting to investigate as next steps. One interesting variation would be to investigate which categories of business and which types of users are causing the predictions to be wrong, and try to generate a new model to account for this. Perhaps, certain categories of business require different weightings, or don't follow any trend at all. In addition, so far we have only applied the prediction mechanism to Yelp, using a comparatively small amount of data. It would be interesting to try a more established site with a connected network of reviews and reviewers and see how that impacts our findings.

References

[1]Knight, Kristina. "Survey: 89% of consumers prefer online shopping ." <http://www.bizreport.com/2007/12/survey_89_of_consumers_prefer_online_shopping.html>

[2] C. Danescu-Niculescu-Mizil, G. Kossinets, J. Kleinberg, L. Lee. How opinions are received by online communities: A case study on Amazon.com helpfulness votes. In Proc. ACM WWW 2009.

[3] A. Ghose, P. Ipeirotis. Designing Novel Review Ranking Systems: Predicting Usefulness and Impact of Reviews. Proceedings of the International Conference on Electronic Commerce (ICEC), 2007.

[4] Glen Jeh, Jennifer Widom, SimRank: A Measure of Structural-Context Similarity,

[5] Matthieu Latapy, Clémence Magnien, and Nathalie Del Vecchio, Basic Notions for the Analysis of Large Two-mode Networks,