CS224W Project Report: Predicting Online Social Influence through User Reputation

Group 24: Clifford Huang, Daniel Jih, Chang Liu

1. Introduction

In 2012, the revenue from online transactions around the world exceeded $1.298 trillion [5]. Many companies are turning to online marketplaces as a platform to reach more consumers as well as collect user feedback from product reviews. Marketplaces such as Amazon.com and Newegg.com allow shoppers to indicate whether a particular review was helpful or not helpful in their decision to purchase a product. The reviews that are voted most helpful are more prominently displayed and can significantly influence the success of a product. Thus, it is beneficial for companies to identify influential reviewers and to understand the nature of their influential reviews.

In this project we examined previous studies that tried to quantitatively define the qualities of helpful reviews. The two papers we found proposed competing definitions of what these qualities are. We tested their proposed qualities against Amazon Fine Foods review data to see which model was more accurate. Furthermore, we propose our own user-ranking system that orders users based on a normalized reputation score in an effort to better identify helpful users. By applying the Random Forest classification algorithm on this system combined with the feature qualification efforts of previous works, we were able to accurately predict the helpfulness of an Amazon Fine Foods review with roughly 92% accuracy.

2. Previous Work

2.1 Influences of Negativity and Review Quality on the Helpfulness of Online Reviews [4]

By using a linear classifier based learning algorithm, Heijden et al. concluded that review quality (best shown with word count) and the review's product rating are the best indicators for how helpful the review is. They propose that review quality and helpfulness are proportional to word count, and that the reviews which give a product rating between three and four stars have the highest level of helpfulness to the community.

2.2 How Opinions are Perceived by Online Communities [6]

Danescu-Niculescu-Mizil et al. argue that a review's perceived helpfulness depends solely on the difference of its rating from the product's overall average ratings. The more that a review's rating deviates from the product's average rating, the less helpful the review is.

From these two papers, we found the proposed features (word count and rating deviation) to be a good starting point for our project. We will see which feature set is the better predictor, and use the results as benchmarks for our own feature set.

3. Network Models and Algorithms

3.1 Data Model

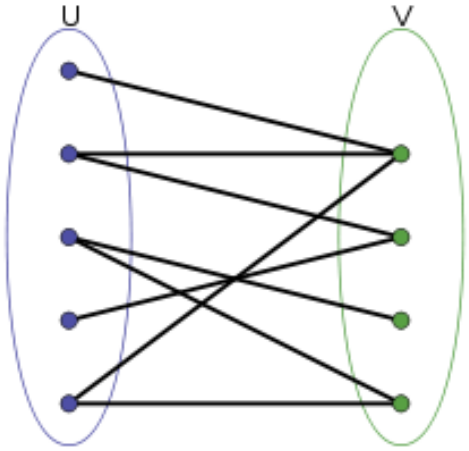

Fig. 1 Bipartite graph representation of our data

We parsed the Amazon Fine Foods data into a bipartite graph as depicted in Fig. 1. The Users and Products are divided into two sets of nodes, U and V, and an edge exists between a user and a product if a user writes a review for that product. This bipartite data model allows us to quickly calculate feature values and easily partition the data into different training and testing sets.

3.2 Normalized Helpfulness Score

Heijden et al.[4] and Danescu et al.[7] calculate review helpfulness as the number of helpful votes minus the number of not helpful votes. The issue with this metric is that it does not take into account the total number of reviews for a product. Thus we propose a normalized helpfulness score metric:

$$h(x) = \frac{(p(x) - q(x))}{r}$$

$p(x)$ = Number of helpful votes
$q(x)$ = Number of not helpful votes
r = Total number of reviews for the product

To demonstrate this scoring algorithm, let us assume a review *A* that received a feedback of "*4 out 5 people found it helpful*", to a product that has 10 reviews. The normalized helpfulness score of review *A* is given by:

$$h(A) = \frac{4 - 1}{10} = 0.3$$

We also have review *B* for a different product with 1000 reviews. Review *B* also has 4 helpful votes out of 5. Using the original metric, they are equally helpful. However,

since review *B* is only one of 1000 reviews for the second product, it should have less impact on a consumer's decision to purchase the product. By defining a normalized value, *B* is still considered helpful, but not as helpful as *A*. The normalized helpfulness of review *B* is given by:

$$h(B) = \frac{4-1}{1000} = 0.003$$

A final thing to notice is that a review with more than 50% of its votes being *helpful* will always generate a positive normalized helpful score, while those with less than 50% will always generate a negative score. If a review does not receive any voting feedback then it is considered a particular case that is dealt with in our Data Collection Process section.

3.3 User Helpfulness Rank

The most effective feature we created is the user helpfulness rank, which is a ranking score for each user based on the sum of the normalized helpful scores of the reviews that they wrote. Users have a higher rank if they write more reviews and also if the reviews they write have a higher helpful score. This ranking score is given by:

$$r(u) = \sum_{x} h(x)$$

From our graph model of the data, we were able to quickly sum different subsets of the weighted edges to calculate user ranks for each training set.

3.4 Random Forest Classifier [10]

We compiled all of the different feature subsets and used a random forest classifier to test their accuracy against our different training and testing data sets. This classifier uses a "forest" of multiple decision trees, with each tree using a subset of the features and data passed in. To use the model, we pass in the testing data set's reviews one at a time through each tree. The response from each tree will generally be noise, but hopefully with the noise more biased towards an answer *helpful* or *unhelpful*. The good thing about this noise is that it cancels out against itself nicely, leaving behind a few select trees that give good answers. From the response of these select trees, the Random Forest Classifier makes it decision and its certainty of that decision. If this certainty (given as a probability split between two classes that must sum up to 100%) is above a certain threshold *T* calculated by our group, then our prediction algorithm will output that decision. After tuning the hyper-parameters for our data set (with the most effective modifications being the number of trees to use and minimum node splitting size), we ran each feature subset of our *Results* section through this classifier to obtain the respective error rate.

3.5 Preparing Amazon Data for Analysis

Since we are only interested in predicting helpfulness, we used *SNAP.PY* functions to remove all unrated reviews (ones with no votes) from the bipartite graph. We have no way of knowing whether those reviews are helpful or unhelpful until at least one person rates it. From Table 1 below, we can see that roughly 50% of the data is Unrated.

Next, we needed to differentiate helpful reviews from unhelpful ones. We define a helpfulness threshold T such that a review is considered helpful if:

$$\frac{\#\ helpful\ votes}{total\ \#\ of\ votes} \geq T$$

For example if the threshold = 0.5, then a review is helpful only if its number of helpful votes is greater than or equal to its number of not helpful votes.

With the threshold set to 0.5, we can see that the ratio of helpful reviews to unhelpful reviews in our data set is almost 5:1, which follows Heijden et al's observation that people tend to rate reviews favorably [4]. We believe, however, that in realistic scenarios, consumers expect the threshold to be much higher. For example a review with 5 helpful votes and 5 not helpful votes would be ignored in favor of a review with 8 helpful votes and 2 not helpful votes. For that reason we also tested our model with threshold = 0.8.

	Threshold = 0.5		Threshold = 0.8	
	Reviews	Percentage	Reviews	Percentage
Helpful	244,441	44%	198,414	35%
Not Helpful	49,544	9%	95,571	17%
Unrated	266,819	48%	266,819	48%

Table 1: Review Count Breakdown

The last step was to divide the data into training and test sets for the classification algorithm. We used cross-validation to create 10 different training and test sets for each helpfulness threshold, with a split of roughly 50,000 reviews in the training set and 70,000 in the test set.

Each point on the graph below represents a different subset of features used, with further detail about which features were used at which data point being highlighted in Table 2. The idea behind each feature subset is also given below.

	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	ϕ_6	ϕ_7	ϕ_8	ϕ_9	ϕ_{10}
S_A	■									
S_B		■								
S_C	■	■								
S_D									■	
S_E	■			■					■	■
S_F					■					
S_G		■			■					
S_H		■			■	■				
S_I						■				
S_J						■			■	
S_K		■				■				
S_L	■	■	■	■	■	■	■	■	■	■

Table 2: Features in each Feature Subset

List of Features, ϕ:

1) Review's Rating of Product
2) Word Count
3) Average Product Rating Given by User
4) Product's Average Rating
5) Indicator for 3 or 4 Star Rating
6) User Ranking Score
7) $\phi_3 - \phi_1$
8) $|\phi_7|$
9) $\phi_4 - \phi_1$
10) $|\phi_9|$

Feature Subsets, S:

A) Rating
B) Word Count
C) Rating + Word Count
D) Danescu
E) Danescu-2 (Includes subcomponents of his original feature as separate features)
F) Heijden's 3 or 4 Star Rating Indicator value
G) Heijden
H) Heijden + User Ranking
I) User Ranking
J) Danescu + User Ranking
K) User Ranking + Word Count
L) Our Chosen Features: all features 1-10

4. Results

The graph below compares the error rates between the training and test sets at different threshold values *T* for each feature subset.

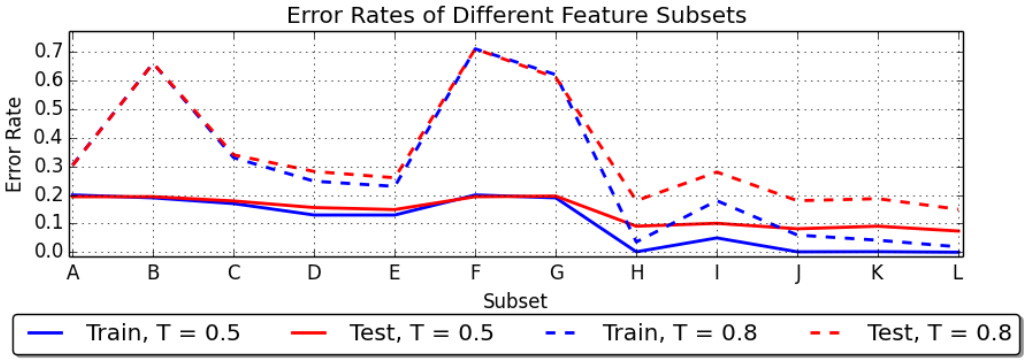

Figure 3: Error Rate for Each Data Set and Helpfulness Threshold

Analysis

As we can see from the error rates for **B**,**F**, and **G** in Figure 3, Heijden's proposed features are not a good predictor for our data set. Furthermore, their paper states that they used a linear classifier, which is only capable of giving a linear decision boundary for each feature. This is the most basic classifier, so it is not surprising to see that their test error rates were 20% and 60% for $T = 0.5$ and $T = 0.8$ respectively.

One interesting thing to note about Danescu's feature set is that while it had roughly same level of accuracy as Heijden's feature set for $T = 0.5$, its error rate for $T = 0.8$ was only 30% - half the error rate of Hiejdan. This is because our dataset has a disproportionately high number of 5-star-rating reviews. As a result, their 3 or 4 Star Indicator feature ϕ_5, was unable to make accurate guesses when the only other feature is word count; it would merely predict everything to be Helpful. This is best illustrated with point **E**, as it gives the highest error across all the tested feature sets.

Feature ϕ_6 is the User Ranking that we developed. As mentioned earlier in the *User Helpfulness Ranking* section, this feature uses normalized helpfulness scores to assign a ranking value to each user. We can see from **I** that this feature by itself is a good predictor; however, it is most remarkable when combined with the feature sets of Heijden and Danescu.

For $T = 0.8$, after adding the User Rank we see in **H** that Heijden's error rate dropped from 60% to 20%. Likewise we see in **J** that Danescu's error rate dropped from 30% to 20%. These results show that the User Rank feature acts as a better decision boundary for both sets.

Features ϕ_1, ϕ_3, ϕ_7, ϕ_8, ϕ_{10} are more features that we believed would be useful, such as the difference between the average rating given by a particular user and their rating for the product we are predicting. Incorporating these features lowered the error

rates even more, with *T = 0.5* and *T = 0.8* having 8% and 16% average error rates respectively. A comparison of our chosen feature set against Heijden's and Danescu's proposed feature sets on our data set is given in the table below.

	T = 0.5 *Prediction Accuracy*	*T = 0.8* *Prediction Accuracy*
Heijden et al.	**80%**	**38%**
Danescu et al.	**80%**	**70%**
Huang, Jih, Liu	**92%**	**84%**

Table 4: Prediction Accuracies of Proposed Feature Subsets

5. Conclusion

Our group was able to effectively use network analysis topics to create a bipartite graph that allowed easy separation of testing and train data for cross-validation. Furthermore, the use of a bipartite graph and weighted edges allowed us to clean the data and quickly gather different permutations of review edges to create the powerful ranking feature to classify the helpfulness of a user's next review. We were able to dramatically reduce the error rates of our data with the inclusion of this network-based feature built on top of Heijden's and Danescu's proposed features.

An example of future work that can continue from this project is to create proprietary rating algorithms that give the ratings of helpful users more weight. With these rating algorithms, product ratings will not be skewed by the ratings of unhelpful users. While our group discussed this idea in the beginning, we set it aside as there is no definitive answer for this problem.

6. References

1. Chen, Bee-Chung, Jian Guo, Belle Tseng, and Jie Yang. "User Reputation in a Comment Rating Environment." *KDD '11: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2011): 159-67. *ACM Digital Library*. Association for Computing Machinery. Web. 16 Oct. 2013. <http://delivery.acm.org/10.1145/2030000/2020439/p159-chen.pdf>.

2. 16% of Yelp Restaurant Reviews Are Fake, Study Says http://eater.com/archives/2013/09/26/study-says-16-of-yelp-restaurant-reviews-are-fak e-1.php

3. Yelp Continues to Defend Against Claims of Review Manipulation http://www.entrepreneur.com/article/226832#ixzz2hvXmnxK7
4. Wu, Philip F., Hans Heijden, and Nikolaos Korfiatis. "The Influences of Negativity and Review Quality on the Helpfulness of Online Reviews." Epubs.surrey.ac.uk. University of Surrey, United Kingdom, 2011. Web. 10 Oct. 2013. <http://epubs.surrey.ac.uk/7533/2/icis2011.final.pdf>.
5. "Ecommerce Sales Topped $1 Trillion for First Time in 2012." *EMarketer*. EMarketer Inc., 5 Feb. 2013. Web. 16 Oct. 2013. <http://www.emarketer.com/Article/Ecommerce-Sales-Topped-1-Trillion-First-Time-201 2/1009649>.
6. Danescu-Niculescu-Mizil, Cristian, Gueorgi Kossinets, and Jon Kleinberg. "How Opinions Are Received by Online Communities: A Case Study on Amazon.com Helpfulness Votes." *WWW* (2009): n. pag. World Wide Web Conference Committee, 20 Apr. 2009. Web. 16 Oct. 2013. <http://www.cs.cornell.edu/home/kleinber/www09-helpfulness.pdf>.

7. J. Liu, Y. Cao, C.-Y. Lin, Y. Huang, and M. Zhou. Low- quality product review detection in opinion summarization. In *Proc. EMNLP-CoNLL*, pages 334–342, 2007. Poster paper.
8. News, Bloomberg. "Alibaba Breaks Sales Record Amid China Singles-Day Rebate."*Bloomberg.com*. Bloomberg, 12 Nov. 2013. Web. 14 Nov. 2013. <http://www.bloomberg.com/news/2013-11-11/alibaba-breaks-sales-record-on-china-sin gles-day-amid-discounts.html>.
9. "Web Data: Amazon Fine Foods Reviews." *SNAP:*. N.p., n.d. Web. 14 Nov. 2013. <http://snap.stanford.edu/data/web-FineFoods.html>.
10. Leo Breiman, Adele Cutler. "Random Forests". Web. 18 Nov 2013. <http://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm>