EECS 445 Project Progress Report: Judging Helpfulness from Amazon Reviews

Riyu Banerjee, Justin Kim, Jingzhu Yan
{riyub, jskimmer, yancarol}@umich.edu

December 24, 2015

1 Introduction

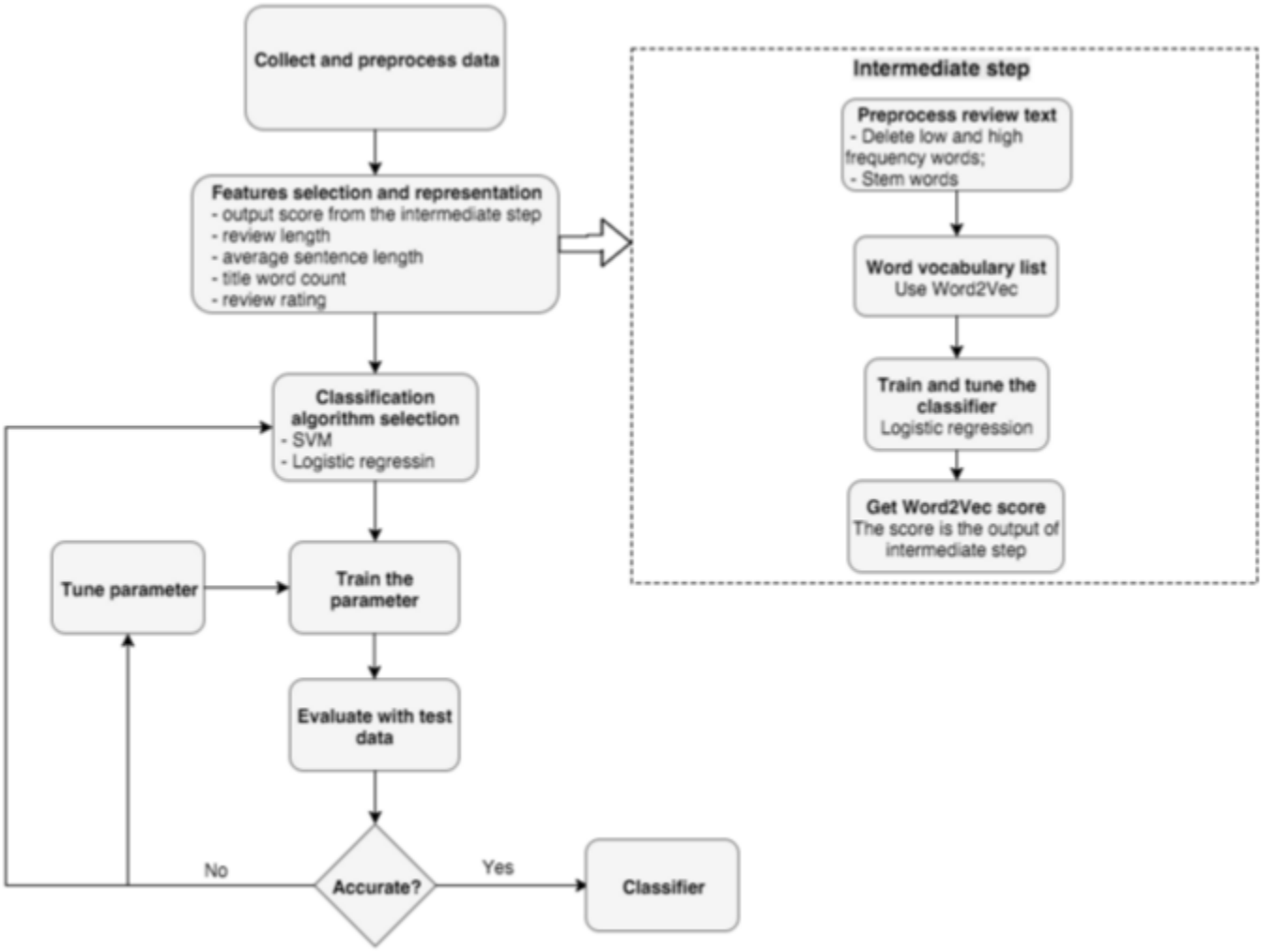
Users have opinions on the products they use. Sites like Amazon offer a way for users to express those opinions in the form of product reviews. Product reviews can guide other users to make more informed decisions about their purchases. However, with a large number of reviews, it can be hard to determine which of those reviews are worth considering when making a purchasing decision. On Amazon, a text review is accompanied by its helpfulness rating, a metric that shows how many people found the review helpful compared to the number of people who did not. In this project, we train a predictor to determine whether or not a review is helpful based on the features of the review.

In detail, our task is a classification problem. We aim at building a classifier that classifies Amazon reviews to two labels – helpful or unhelpful – based on provided information in the review. First, we explore on several potential features in the review, and select features useful to create a feature representation of the data. Then we build the classifier using SVM and logistic regression and compared the results.

2 Proposed Method

In our model, we defined helpfulness as a binary label. A review is considered helpful when the ratio of helpful reviews to total reviews is at least 60%. We considered the following as potential features for our model: review length, average sentence length, title words count, and review ratings. We also trained a word2vec model of the text, and assigned each review a word2vec score. This score is a multidimensional vector which can be used as features into a classifier. Instead of using the vector directly as features, we trained a logistic regression model on the word2vec vector to turn it into a probability that a review is helpful. We then used this probability as a feature along with the features mentioned before. Our motivation for this intermediate probability step was that it would result in feature reduction, but we conducted experiments to see if it would result in any loss in accuracy. We trained a SVM with a radial-basis-function kernel as well a logistic regression model on different selections of features to see which would result in the greatest accuracy. We compared the accuracy of the different models as well as the different feature selections after tuning various hyper parameters. Our criterion for selecting features was that they had to be readily available from just the text. Meta-data about the review, such as overall product score, or how many reviews

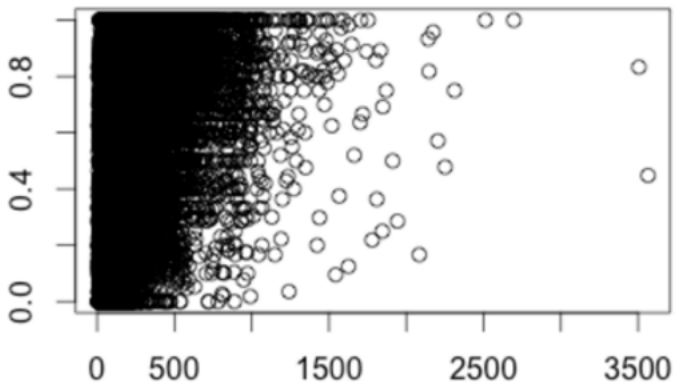
the reviewer has written, was not considered. This makes the classifier more flexible and able to be used for more than just Amazon reviews. The pipeline of our method is shown in the figure below.



Data Exploration and Feature Selection

In this section we explore on the data and show how we decided which features to select. Basically, for each potential feature, we explain our intuition, make exploratory scatter plots, and calculate the correlation between the feature and the helpfulness rate to prove the relationship.

2.0.1 Review length



We select this feature based on the intuition that longer reviews tend to be more useful than shorter ones. Figure 1 can explain this relationship there are more lengthy reviews rated helpful than short reviews. A relatively high correlation also proves the importance of this feature in predicting helpfulness.

2.0.2 Average sentence length

Similar to the first feature, average sentence length is another form to measure how detailed a review is. A correlation of 0.139 indicates that it is relatively correlated to the helpfulness.

2.0.3 Punctuation count

Occurrence of exclamation and question marks are counted. These two punctuations may express reviewers emotion, thus we suppose the count would be useful in the prediction task. However, in fact, the correlation is near zero, which means it almost plays no role in predicting helpfulness.

2.0.4 Capital words count

Reviews with capital words may express strong emotions to the product or experience. The correlation of this feature is also not high.

2.0.5 Title words count

The title as a summary of the review, concludes reviewers opinion in few words. A more descriptive title gives concrete information of the product from the beginning. We include the length of title as a potential feature, trying to figure out whether longer title could be useful in making the review helpful.

2.0.6 Review rating

Rating	Avarage helpfulness
1	0.3436113
2	0.3883592
3	0.5002236
4	0.7031294
5	0.7478761

The helpfulness rate is aggregated on ratings of 1 to 5. From the result table, we can see that the rating has a strong correlation with helpfulness. 5-star reviews tend to be more helpful to other users.

2.0.7 Word2vec score

Word2vec score is the score derived from an intermediate layer. We use this layer in purpose of reducing the dimensionality of feature matrix. We will exam whether this dimension reduction will cause reduction of accuracy in the following sections.

Correlation of Features

Feature	Correlation
Review Length	.24084
Average sentence length	.13921
Punctuation count	.00804
Capital words count	.05278
Title words count	.13104
Review rating	.53472
Word2Vec score	.31628

The correlations of each feature are listed in the following table. The features with a correlation less than 0.1 are dropped in this task. After all, the features we used in prediction are review length, average sentence length, title words count, review ratings, and the score we get from Word2Vec model.

3 Related Work

There has been a lot of work relating to analyzing the sentiment of a text using machine learning techniques. Many of these methods use the bag-of-words approach or the word2vec approach to classify the text as having either a positive or negative emotion. We wanted to see whether the same approach would work in determining whether or not a review was helpful or unhelpful.

4 Experimental Results

We have a dataset of sample size 20000, in which 40% are training data, 30% are validation data, and the rest 30% are testing data. If accuracy rate (number of votes being helpful/total votes) is greater than 0.6, the review is considered as helpful (positive). In 8000 training data, 4270 reviews are labeled helpful and 3730 are labeled unhelpful, with positive rate being 0.534. The threshold 60% is selected in purpose of making the label balanced. Hence we set the minimal accuracy rate to be 0.534, which happens when the classifier labels all testing data positive. Meaning, if after all the classifier get an accuracy higher than 0.534, we can tell that it is performing above the baseline.

In feature selection, we consider mainly three categories of features, and in the experiment, we try a feature matrix of combination of the three categories. The three categories are: Text features: text features are text information in reviews. It includes review length, average sentence length, title words count, and review ratings. Word2vec probability: Word2vec probability is the Word2vec vector derived directly from the review text. Word2vec score: Word2vec score is the result score derived from the intermediate layer explained in section 4.

We run experiments using different combination of features and different classification models. We train each combination with various hyper parameters and selected the highest value for each across multiple validation sets. We see that in most cases SVM with rbf kernel has a higher accuracy than logistic regression. We also observe that using the Word2vec vector directly as a feature yields similar accuracy with using text features alone. While when using the combination of these two, only Logistic regression gets a good accuracy. Using Word2vec score and text features together get good results too, while using the Word2vec score alone gives the lowest accuracy in our experiments.

In conclusion, using combination of Word2vec and text features always yields better result than using each of these two alone. We try to use the intermediate probability to reduce the number of features, we found that in most training size conditions, this increases accuracy compared to directly using the word2vec vector.

	Text features		**Word2vec vector**		**Word2vec vector and text features**		**Word2vec score and text features**		**Word2vec score**	
Training Size	SVM-rbf	Logistic regression	SVM-rbf	Logistic regression	SVM-rbf	Logistic regression	SVM-rbf	Logistic regression	SVM-rbf	Logistic regression
1000	0.496	0.517	0.6026	0.5973	0.504	0.5866	0.544	0.6426	0.542	0.5766
5000	0.5005	0.5349	0.6282	0.6392	0.5152	0.6485	0.6037	0.6909	0.6125	0.6482
10000	0.60853	0.6786	0.7001	0.7005	0.613	0.7089	0.6442	0.726	0.6365	0.7134
20000	0.7343	0.7283	0.7394	0.7393	0.6295	0.7424	0.7383	0.7342	0.7283	0.7132

5 Future Milestone

While we were able to achieve an accuracy score of around the mid 70s we found that many of the feature additions lead to only marginal increases. Future work on the problem would involve a combination of finding new features that correlate with the helpfulness score better than our current set as well as rethinking previous methods. Perhaps a better feature to replace Word2Vec exists or maybe Logistic Regression and SVM were not the best models to describe the data. Much of the process involves both creativity and experimentation in order to find the best predictors.

6 Conclusion

We were able to train a classifier that took as features the combination of review length, average sentence length, punctuation count, title word count, review score, Word2vec vector, and Word2vec score to predict whether a review was helpful or not helpful. It was able to predict a reviews helpfulness with an accuracy of 74%. We obtained the highest accuracy by selecting the word2vec score and other text features as our features using logistic regression as our model. This suggests that taking the word2vec score as features does not require a reduction in dimensionality to get the most accurate answers. This also shows that machine learning can be used to predict with reasonable accuracy the helpfulness of a review given its basic information. Because we only used features readily available from the most basic parts of a review, this method can be applied to reviews on any type, not just on Amazon. For future work, we would like to see if other techniques can increase the accuracy to 80%, comparable to that of traditional positive or negative sentiment analysis.

Author Contributions

R.B., J.K., and J.Y read the existing literature and conceived of an approach to classify Amazon reviews. J.K. wrote the proposed method, and J.Y. created the project workflow figure. R.B. obtained the dataset for use in our project. R.B. worked on initial prepossessing on review sets. J.K. worked on the various models and experiments to find the best procedure. J.Y added more features based on correlation to increase the accuracy and optimize the results.

References

[1] Julian McAuley. *Amazon Review Data.*

[2] Bo Pang and Lillian Lee. *Opinion Mining and Sentiment Analysis.*

[3] Bo Bang and Lillian Lee. *Seeing Stars: Exploiting class relationships for sentiment categorization with respect to rating scales.*

[4] S.B. Kontsiantis. *Supervised Machine Learning: A Review of Classification Techniques.*

[5] B. Pang et al. *Thumbs Up? Sentiment Classification Using Machine Learning Techniques.*