Amazon Digital Music: Sentiment Analysis and Text Mining

[CSE 190 Assignment 2]

Aaron Wong
University of California, San Diego
azwong@ucsd.edu

Brittany Factura
University of California, San Diego
bfactura@ucsd.edu

ABSTRACT
In this paper, we aim to analyze the sentiment of review text in order to accurately predict product ratings. We extract features from text reviews and use regression models to predict review ratings. By using unigrams and bigrams as features, we build a rich predictor that identifies phrases and their polarities and how they could predict product ratings.

1. INTRODUCTION

We explore product ratings and its relation to review text in the context of music recommendation systems through the use of the Amazon product dataset's Digital Music category. Digital music has become so much more accessible than ever before all thanks to the increasing popularity of music streaming platforms and services such as Pandora, Spotify, SoundCloud, and Apple Music. A recommender system attempts to predict the rating or preference that a user would give to an item. In this case, the recommender system will predict whether or not the user will like a certain song or album. Using knowledge about a user, such as their preferences and previous actions, recommender systems like the ones used at Netflix or Amazon provide users with ideas of new products to consume. The challenges that come to mind when building a recommender system are the time it takes to make suggestion as well as the accuracy.

The Digital Music Amazon product dataset differs from regular music recommendation since it focuses on the feature of text reviews along with a 5 star rating rather than just binary categorization of whether a user liked the item or not. Using this we can perform sentiment analysis or opinion mining to better understand what constitutes a good rating for music records. Essentially, our problem is just a product review based recommendation system that is catered towards digital music sales. We will analyze user opinions and sentiment on this kind of dataset and its correlation on product review ratings.

Motivation to study this relationship between review text and user rating comes from our observation of people's incessant need for fast paced results especially when text is involved; they simply want a "too long, didn't read" summarization of long texts.

Our model here proposes to determine what kind of prediction tasks we can solve using text. Examining text allows us to discover relevant interests specific to a user, best summarize a user's opinion, and identify certain aspects of a product. We choose to analyze review text to capture positive or negative sentiment about the product being discussed, thus increasing the accuracy of rating prediction.

2. EXPLORATORY ANALYSIS

We studied the 'Digital Music' category from the Amazon product dataset [1] which contains product reviews (reviewerID, reviewerName, ratings, text, helpfulness votes, review time) specifically for Digital Music purchases.

Some general statistics about the data:

Number of reviews	836015
Number of unique users	478243
Number of unique items	266416
Mean rating	4.540253
Average length of text review	416
Mean helpful ratio	0.652559
Length and Helpful ratio	0.92301806
% 1 star rating	0.360
% 2 star rating	0.024
% 3 star rating	0.049
% 4 star rating	14.7
% 5 star rating	74.4

Table 1: Data Statistics

The large distribution of 4 to 5 star ratings corresponds to a high mean rating of 4.54.

The following figure is a word cloud of the most frequently generated words from all text reviews. The words in the figure have largely positive connotations which correlates with the average rating of all reviews, 4.54 out of 5. Since most of the reviews were given a positive rating, it makes sense that the most frequently occurring words in all of the reviews are positive or words related to the subject being reviewed.

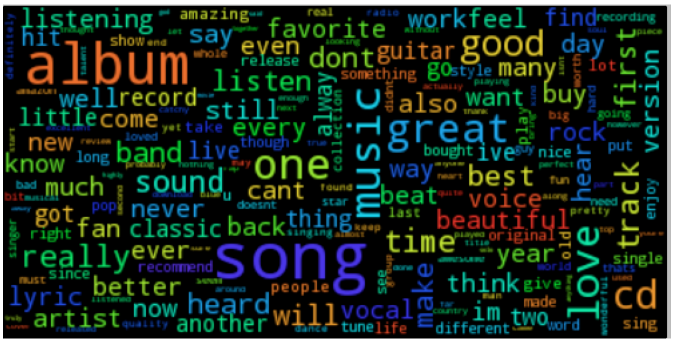

Figure 1: Word Cloud of Frequent Words

3. PREDICTIVE TASK

We started off attempting to predict the overall rating of each review by building simple linear regression models using features such as length of review and helpfulness ratio. To measure the performance of each model, we used mean squared error and a baseline which predicted global mean rating, which resulted with an MSE of 0.7119 on the whole set.

After looking at the length of review distribution against the rating, there seems to be no correlation between length of a review on the review rating as both low and high ratings could have the same review length. However, since most of the reviews received favorable ratings of mostly 5s and 4s, a model built with a feature of review length seemed to give favorable MSE result that beat the baseline by an insignificant amount, as shown in the table below.

As noted below, the reported MSE on the whole set for helpfulness ratio was extremely high therefore it should not be used as a feature to predict review ratings.

Feature	MSE
Global mean rating	0.7119043190
Length of review	0.7071163185
Helpfulness ratio	2046.0820983

Table 2: Baselines

Now we will use the text reviews to train a model. Our features will be the 1000 most frequently occurring unigrams and bigrams. We split up our data into a training and testing set with 90/10 split to measure the performance in terms of training and testing error.
The simple linear regression model will also be used to compare against this text based model.

4. LITERATURE

The dataset we consider is a subset of the Amazon dataset obtained by McAuley and Leskovec. Many works have studied Amazon review corpora in similar attempts to predict product ratings through the information present in review text. McAuley and Leskovec model user product evaluations by understanding the hidden dimensions and factors of review text to help justify user reviews. Their statistical models combine latent dimensions in rating data with topics in review text using an objective that combines the accuracy of rating prediction with the likelihood of the review corpus. By understanding how hidden rating dimensions relate to hidden review dimensions, their models facilitate a variety of novel tasks, such as genre discovery and identification of informative reviews. Their HFT model combines the two ideas of Latent-Factor Recommender Systems and Latent Dirichlet Allocation to discover topics that are correlated with the hidden factors of products and users [3].

Qu et al. also studied Amazon review corpora to learn and predict a user's numerical rating of a review from review text, and model this as a metric regression problem over a defined feature space. In their research, they found that unigrams fail to capture important expressions for prediction models. Furthermore, while N-grams are able to capture such expressions, they occur too sparsely in the training set to yield strong predictors. They introduce a bag-of-opinions representation that splits an opinion into three components: a root word, a set of modifier words, and one or more negation words. A ridge regression algorithm is used to learn opinion scores. Their method, CRR-BoO (Constrained Ridge Regression for Bag-of-Opinions) outperformed alternative methods including unigrams, unigrams and bigrams, n-grams, and cumulative linear offset model by a significant margin in terms of mean square error measurement. This bag-of-opinions representation is thus able to achieve the expressive power of n-grams as well as overcome the sparsity bottleneck [4].

In other related literature, Michael Tran attempted to fit a linear regression model using a feature vector of the td-idf counts of the 1000 most popular words, however the result was unfavorable as the reported mean squared error across validation and test sets were around 60. Tran also uses a similar baseline of predicting the global mean of all the review ratings achieving a mean squared error of 1.5675. Though we also used this as a baseline, our reported mean squared error of 0.71190 was significantly better which may be due to the data set used, his being in the category of Grocery and Gourmet food which had a lower global mean rating and more sparse amount of reviews. Our chosen category, Digital Music, had over 89% of the rating distribution with a value of 4 to 5.

5. RESULTS AND DISCUSSIONS

First, we observe the frequently occurring unigrams and bigrams on the review text from the Amazon product dataset.

From this we build a predictor of the form:

$$f(\text{text}) \rightarrow \text{rating}$$

using a model based on ridge regression

$$\text{rating} \simeq \alpha + \Sigma_{w \in text}\ \text{count(w)} \cdot \theta_w$$

We perform a regularized regression with a regularization parameter of 1.0 on 1000 features of just unigrams and on 1000 features of a combination of unigrams and bigrams. Using this model, we can identify the unigrams and unigrams + bigrams with the most positive and negative assocaited weights. The following tables outline the steps of our results.

Tables 3 & 4 show the 10 unigrams with the most positive associated weights and the 10 with the most negative associated weights from the 1000 most common unigrams. These tables display the theta values representing such weights and the corresponding term-Document frequency of each word.

unigram	associated weight	term-Doc freq
rocks	0.1245044494	8044
thanks	0.1285274625	14084
amazing	0.1287764882	48413
whether	0.1329483064	6960
bonus	0.1380806981	7800
fantastic	0.1415867658	17654
glad	0.1444378710	19670
wait	0.1532271754	30464
awesome	0.1640065690	36611
hooked	0.1655398855	5781

Table 3: Most Positively Weighted Unigrams

unigram	associated weight	term-Doc freq
worst	-0.8063026544	5840
boring	-0.5549747278	6705
money	-0.4607158545	17569
sorry	-0.4222718724	7068
tried	-0.3602179511	5460
unfortunately	-0.3495196822	6646
disappointed	-0.3045451763	16957
decent	-0.2641513280	6758
ok	-0.2600767688	129339
okay	-0.2590140786	6571

Table 4: Most Negatively Weighted Unigrams

We further took our model to be 1000 features including a combination of the most common unigrams and wbigrams. We again found the 10 unigrams/bigrams with the most positive associated weights, and the 10 unigrams/bigrams with the most negative associated weights. The following tables display the theta values representing such weights as well as their corresponding term-Document frequency for all the text reviews in the corpus.

unigram/bigram	associated weight	term-Doc freq
love this	0.1398009539	51191
wait	0.1400610097	30464
fantastic	0.1444542965	17654
glad	0.1535397431	19670
not to	0.1543851966	14533
better than	0.1617524916	19991
awesome	0.1660254639	36611
worth the	0.1668395504	11601
you dont	0.1836757934	1425
a must	0.2281932510	18417

Table 5: Most Positively Weighted Uni/Bi

unigram/bigram	associated weight	term-Doc freq
money	-0.4878278263	17569
disappointed	-0.3144375315	16957
ok	-0.2620724989	129339
at all	-0.2617440163	15330
trying	-0.2528443594	13561
instead	-0.2361090468	12003
not the	-0.2172325017	16267
sounds like	-0.1965960387	14657
please	-0.1890123601	15951
bad	-0.1879811533	37071

Table 6: Most Negatively Weighted Uni/Bi

The unigrams and bigrams from Tables 3, 4, 5 & 6 are sensible words and phrases that accurately identify the polarities of review text. The words justify the positive or negative rating corresponding with the text. The combination of unigrams and bigrams can add a richer prediction, as seen in the phrase with the most positively associated weight: 'love this', which truly captures the positive sentiment of the review text.

Feature	MSE
Unigrams	0.5544962807
Unigrams/bigrams	0.5571903039

Table 7: Mean Squared Error

Table 7 shows the mean square error (MSE) from each of the two test sets. Our model which uses unigram features improved on the baseline by approximately 16%. The unigram and bigram feature model performed about the same as the unigram model. These results were achieved by performing regularized regression with a regularization parameter of 1.0 to prevent overfitting on the complex model with such a large feature vector. This was a likely issue when performing least squares regression.

6. FUTURE WORKS

To improve upon our selected regression models, we shall remove stopwords as some do appear in the list of most weighted unigrams and combination of unigram/bigrams. These should not add value to the sentiment analysis and thus by removing them should increase the accuracy of our predictors. In addition, we will merge different inflections of words in a process called stemming. While this method can sometimes throw away similarly stemmed words with different meanings, we can see if it leads to any improvements. We can also attempt discarding extremely rare words to further pre-process our review text. Furthermore, we would use Qu et al's bag-of-opinions model, which would outperform the unigram and the unigram/bigram model. This algorithm would still capture the expressive capabilities of n-grams that we have already seen while overcoming the sparsity bottleneck.

Another model worth looking into would be the latent factor model. A lot of music recommender systems are built using this type of model through collaborative filtering. Though this kind of model does not incorporate the review text in the corpus. Matrix factorization with user attributes model may also be used in conjunction with sentiment analysis techniques to create better models as Seroussi et al concluded that improvements were gained after inferring attributes from user-generated texts. This could be a solution to the cold start problem in which it may be difficult to give predictions for new users who have given no ratings. By personalizing predictions based on user attributes, such as genre preferences, age, location, it would encourage user participation and thus improving prediction systems. [5]

Trying greater n-gram models might possibly be useful in evaluating improvements in performance on this data set as people might use greater than 2-gram catch phrases such as 'worth the wait,' 'a must buy,' or 'sounds like [something negative]' to express positive or negative sentiment.

For future works, we would like to consider sentiment analysis techniques we utilized to create better models for text based recommender systems. Using the results found from applying sentiment analysis in conjunction with collaborative filtering techniques would make it possible to create rich predictor models for services that focus on text and rating based reviews as a new way to recommend music or other media to users. Metacritic, a critic and user review aggregating service which features text reviews on music, movies, television, and games could potentially use models like these to create personal recommender systems for its users.

7. REFERENCES

[1] J. McAuley, C. Targett, J. Shi, A. van den Hengel Image-based recommendations on styles and substitutes SIGIR, 2015

[2] M. Tran Predict Product Ratings From Review Text UCSD, 2015

[3] J. McAuley and J. Leskovec. Hidden Factors and Hidden Topics: Understanding Rating Dimensions with Review Text. RecSys, 2013.

[4] L. Qu, G. Ifrim, and G. Weikum. The Bag-of-Opinions Method for Review Rating Prediction from Sparse Text Patterns. In Proceedings of the 23rd International Conference on Computational Linguistics, pages 913-921. Association for Computational Linguistics, 2010.

[5] Y. Seroussi, F. Bohnert, I. Zukerman. Personalised Rating Prediction for New Users Using Latent Factor Models. Faculty of Information Technology, Monash University, 2011.