# Rating Prediction with Latent Semantic Analysis

CSE 258 Assignment 2

Shiwei Song
A53206591
shs163@eng.ucsd.edu

## 1 INTRODUCTION

Rating prediction is a common task in web mining. In this project, I focused on rating prediction with only review text. I built several models and compared their effectiveness on the Yelp challenge dataset. The three models used are Bag-of-Words model, TF-IDF model and Latent Semantic Analysis (LSA) model. The results shown that LSA model outperformed the other two models.

## 2 DATASET

The dataset I used is the Yelp challenge round9 dataset. The dataset contains 4.1M reviews and 947K tips by 1M users for 144K buisiness [1]. For this project, I used 100K reviews and their corresponding star ratings from the dataset.

### 2.1 Exploratory Analysis

I did some analysis on the 100K reviews to get some insight about the dataset.

The first analysis is about the length of the review and star rating. The mean length of the review is 120.76 and the mean star rating is 3.599. Figure 1 shows the star rating vs. length of reviews plot.

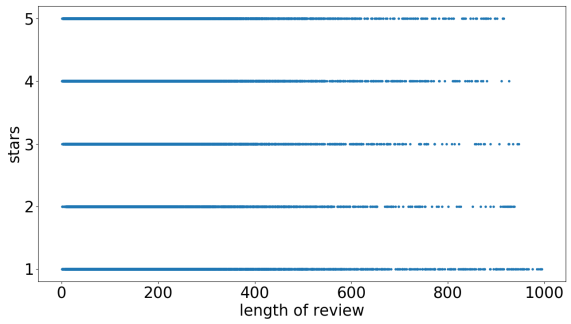


**Figure 1: Star rating vs. review length**

I also calculated the mean length for reviews with different ratings. The results are shown in Table 1.

**Table 1: Mean length of reviews for different star ratings**

| stars | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| mean length | 146.20 | 146.69 | 130.95 | 121.75 | 99.77 |

From Figure 1 and Table 1, we can see that users tend to write longer reviews if they give lower ratings. Intuitively, this phenomena can be explained by customers' psychology. As you are not satified with the service, you would have a lot to complain. On the other hand, if you are satisifed with the service, you won't have much to say.

The second analysis is about the words apeared in the reviews. After removing capitalization, punctuations, typical stop words in English and unigrams only appeared once, there are 75161 unique unigrams. Then I found the 5 most frequently appeared words for different star ratings. The results are shown in Table 2.

**Table 2: Most frequently appered words for different star ratings**

| stars | words |
|---|---|
| 1 | place, one, just, get, back |
| 2 | place, just, like, food, good |
| 3 | good, place, like, food, just |
| 4 | good, place, great, like, food |
| 5 | great, place, time, good, get |

We can see the most common words reflect the star ratings to some extent. For example, 'great' and 'good' become more frequent for higher ratings. However, some frequent words have little correlation with ratings. For example, 'place' is common in all five ratings. This somehow shows the potential problem about the BoW model which will be discussed in the Model part.

## 3 PREDICTIVE TASK

The main task is to predict star ratings with only the review texts.

### 3.1 Evaluation

The dataset will be splitted into training and validation sets in a 9:1 ratio. The results will be mainly evaluated on the validation sets using the Mean-Squared-Error (MSE).

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - M(x_i))^2$$

On the other hand, to get some more direct insight, the prediction will also be rounded to integer to caculate accuracy percentage.

### 3.2 Baseline

I will use a baseline model as reference to evaluate and compare the performance of my models. The baseline model would be using the mean star rating of all training samples as prediction.

$$y_{pred} = \frac{1}{N}\sum_{i=1} y_i$$

### 3.3 Preprocess on Review Texts

I did the following preprocess on the review texts:

- Remove capitalization and punctuations.
- Remove typical stop words in English. (e.g. me, what, it, was, out...)
- Keep only 2000 most common words.

## 4 MODEL

For the model, the input will be review texts and the output will be the predicted star rating. The model can be divided into two parts:

(1) feature extraction part will transform review texts into numerical vectors
(2) prediction part will use the extracted features to predict rating

### 4.1 Feature Extraction

I used three different methods to map the review texts into numerical vectors. The models are Bag-of-Words model, TF-IDF model and Latent Semantic Analysis (LSA) model.

*4.1.1 Bag-of-Words Model.* Bag-of-Words model is a simple model for text mining. It uses the frequency of words in the dictionary as the feature vector. I choosed BoW as the first attempt to the predictive task because of its simplicity.

*4.1.2 TF-IDF Model.* For BoW model, we may face the problem that most frequent words are not the most decisive ones for rating prediction. So I used the TF-IDF model as an improvement. TF-IDF model solves the problem by considering the relative frequency of words in the document.

*4.1.3 LSA Model.* Another problem about BoW and TF-IDF model is the high dimensionality of the feature. The dimension of the feature would equal to the number of words in the dictionary. Even though I choose the 2000 most common words, it's still a relatively large dimension. So I used the Latent Semantic Analysis model which will run a Singular Value Decomposition (SVD) on the word-document matrix and find out the dimensions that contain most variance. This enable us to lower the dimensionality of the feature vector.

One important consideration of LSA model is to choose the number of dimension we want to keep. I tried some different values and the results would be shown in the Results part.

One issue about LSA is about the size of the dataset. Since we need to do SVD on the whole data matrix, it would be very costly on large dataset. I faced the memory problem during implementation. I solved it by representing the matrix as a sparse matrix in Scipy. However, if the dataset gets even larger, the SVD computation would be very difficult. So this is a potential restriction on the application of LSA.

### 4.2 Prediction

For star rating prediction, it can be considered either as a regression problem which will generate continuous output or as a classification problem which will classify into five categories (1 star to 5 star). For regression, I used linear regression. For classification, I used clustering.

*4.2.1 Linear Regression.* Linear regression is the most simple and direct approach to the prediction task. A linear combination of all features will be used to predict rating. So each feature term will have a corresponding weight. I added a bias term and ran linear regression to fit the models. To prevent overfitting, I added a regularizer with parameter $\lambda = 1$.

*4.2.2 Clustering.* The intuition of clustering comes from the LSA model. Since we map the feature into a lower dimension space, we can cluster similar reviews into groups to find out some patterns. Then we can use the cluster to classify new reviews. For prediction, we assign it with mean rating value of the cluster it was assigned to. The clustering algorithm I used is the Kmeans algorithm with Euclidean distance.

## 5 LITERATURE

The dataset I used is the Yelp challenge round9 dataset [1]. It contains business and user information from Yelp. Hundreds of studies have been carried out on Yelp's dataset challenge covering different fields and approaches like location mining, social graph mining and cultural trends. Some similar datasets are Amazon reviews [2] which contains 35 million reviews from Amazon and OpinRank review dataset [3] which contains car reviews and hotel reviews.

A lot of researches focused on rating prediction. Methods like Support Vector Machine (SVM) [4] and Bayesian probability [5] are common approaches for the task.

The use of semantic analysis has become more and more popular on this task. Latent Dirichlet Allocation (LDA) is a method for topic extraction [6]. In [7], a modified LDA model was described to fit the prediction task. The rating was incorporated into the model as a new plate connected with the distribution of terms. The use of codeword reduces the variation of positive and negative words used. This enables the model to learn connection of words and ratings better.

In [8], the model combined semantic analysis and traditional latent factor model to get better results.

## 6 RESULTS

In this part, I will discuss about the results of the model. I will evaluate all models based on the MSE experiments. I will also discuss on some insights and findings on the models.

### 6.1 MSE and Accuracy

Among the 100K reviews, 90K of the reviews are used as training set and 10K of the reviews are used as validation set. I used MSE to evaluate three models. Lower MSE means better results.

For TF-IDF model, I normalized the tf-idf vector sample-wise before linear regression.

For LSA model, the SVD is done on the tf-idf features instead of the BoW features. Normalization is done sample-wise after SVD. The reduced dimension is $k = 1000$ for the data shown in the following tables.

For clustering, the reduced dimension is $k = 20$ and the number of clusters is $c = 2$ for the data shown in the following tables

Table 3 shows the MSE on both training set and validation set for the four models and the baseline results which use the mean rating to predict.

**Table 3: MSE results**

| Model | Train MSE | Valid MSE |
|---|---|---|
| Baseline | 2.0818 | 2.1010 |
| BoW | 1.0068 | 1.0984 |
| TF-IDF | 0.9354 | 1.0278 |
| LSA with linear regression ($k = 1000$) | 0.8640 | 0.9103 |
| LSA with clustering ($k = 20, c = 2$) | 1.7584 | 1.7742 |

First, we can see all models perform better than the baseline solution.

For linear regression, BoW reduced the MSE for baseline by half. TF-IDF had an improvement on BoW and LSA had an improvement on TF-IDF as expected.

For classification, the LSA with clustering model was a little better than the baseline solution. However, it was outperformed by the regression models. Hence, regression methods will lead to better results than classification methods for rating prediction problem in general.

Table 4 shows the accuracy results.

**Table 4: Accuracy results**

| Model | Train Acc | Valid Acc |
|---|---|---|
| Baseline | 0.2477 | 0.2429 |
| BoW | 0.3727 | 0.3603 |
| TF-IDF | 0.3932 | 0.3860 |
| LSA with linear regression ($k = 1000$) | 0.4153 | 0.4189 |
| LSA with clustering ($k = 20, c = 2$) | 0.2018 | 0.1989 |

As we can see, the accuracy results generally agreed with the MSE results. LSA with linear regression has a best accuracy of 0.4189 on the validation set.

## 6.2 Discussion on BoW Result

In this section, I will discuss some interesting results from the BoW model. Linear regression will return a weight for each feature term. I sorted the terms with respect to their weights. Table 5 and Table 6 show the words with largest weight (most positvie) and smallest weight (most negative).

**Table 5: 10 most positive words**

| word | weight |
|---|---|
| outstanding | 0.385 |
| saved | 0.356 |
| excellent | 0.310 |
| amazing | 0.308 |
| thank | 0.304 |
| knowledgeable | 0.304 |
| highly | 0.299 |
| fantastic | 0.296 |
| notch | 0.291 |
| awesome | 0.285 |

**Table 6: 10 most negative words**

| word | weight |
|---|---|
| worst | -0.706 |
| beware | -0.648 |
| horrible | -0.591 |
| terrible | -0.544 |
| waste | -0.542 |
| rude | -0.509 |
| unprofessional | -0.491 |
| overpriced | -0.437 |
| avoid | -0.434 |
| disgusting | -0.431 |

I noticed that the absolute value of the weights for negative words are much larger than those for positive words. So, negative words express stronger feelings than postive words.

Figure 2 and Figure 3 show the word cloud for the 50 most positive and most negative words respectively.


**Figure 2: Word cloud for 50 most positive words**

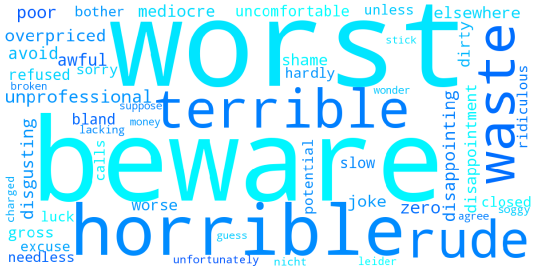


**Figure 3: Word cloud for 50 most negative words**

As we can see, these results make sense because the words with large weights are positive and the words with small weights are negative.

### 6.3 Dimension Selection for LSA

One important hyperparameter for LSA is the number of dimension after SVD. Figure 4 shows the MSE under different values of dimension $k$.

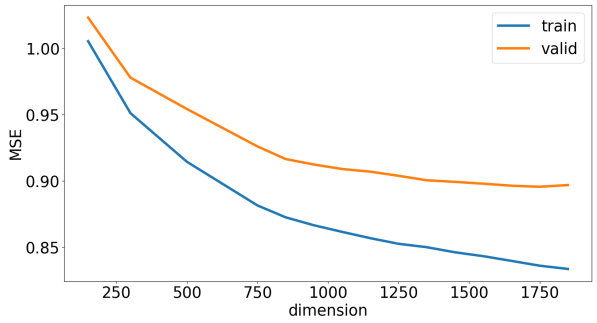


**Figure 4: MSE vs. reducded number of dimension**

From Figure 4, we can see that as number of dimension $k$ increases, MSE decreases until the dimension reaches around 1900 (original dimension is 2000). The best value of MSE on validation set is 0.8955 with $k = 1850$. This shows that part of the original feature can have negative impact on the prediction results. Also, we can see MSE starts to decrease slowly at around $k = 1000$. Hence, for LSA, we may need to take a consideration on the performance-dimentionality balance.

The MSE on validation set with $k = 150$ is 1.022964 which is still lower the MSE we get using TF-IDF model. This shows the advantage of LSA model: with lower dimension of feature (150 compared with 2000), we still got better MSE on the validation set.

### 6.4 Clustering with LSA

I did clustering with different dimensions $k$ and number of clusters $c$. Figure 5 shows the MSE on validation set for $c = 2, 3, 4, 5$ when dimension changes.

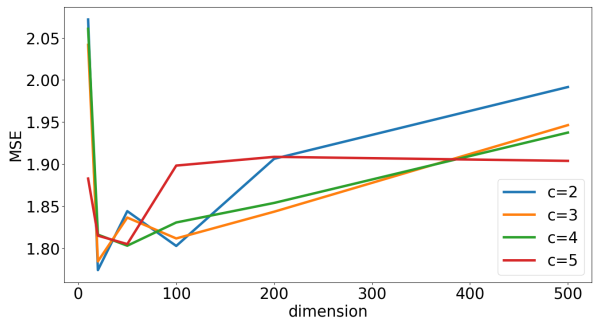


**Figure 5: MSE vs. number of dimension for $c = 2, 3, 4, 5$**

We can see that as dimension becomes too large or too small, the MSE will become very large. The best MSE is reached at $k = 20$. For lower dimension, $c = 2, 3$ had better MSE results and for higher dimension, $c = 4, 5$ had better MSE results.

Table 7 showed the mean rating of the clusters when $k = 20$ with $c = 2, 3, 4, 5$.

**Table 7: Mean rating of the clusters when $k = 20$**

| Cluster | Mean rating |
|---|---|
| 1 | 4.289 |
| 2 | 3.131 |

| Cluster | Mean rating |
|---|---|
| 1 | 3.656 |
| 2 | 4.490 |
| 3 | 3.388 |
| 4 | 3.163 |

| Cluster | Mean rating |
|---|---|
| 1 | 3.643 |
| 2 | 4.401 |
| 3 | 3.137 |

| Cluster | Mean rating |
|---|---|
| 1 | 4.486 |
| 2 | 3.654 |
| 3 | 3.155 |
| 4 | 3.687 |
| 5 | 3.389 |

As we can see, the mean rating reflects not much about the difference in star ratings. Besides, the MSE result is not very good using clustering. Some modification may help to improve the results, e.g. using cosine similarity instead of Euclidean distance. In general, clustering is not a good approach for the rating prediction task. However, it may help us to analyze the texts in another way.

## 7 CONCLUSION

For this project, I had the task of rating prediction using the review text. Three models (Bag-of-Words, TF-IDF, Latent Semantic Analysis) are used for feature extraction and two methods are used for prediction (linear regression, clustering). The models were trained and evaluated on 100K reviews from the yelp challenge round9 dataset. MSE is used to evaluate models. Tests results showed that LSA > TF-IDF > BoW in terms of performance. Linear regression outperformed clustering a lot. The best MSE was 0.8955 obtained from LSA with linear regression ($k = 1850$).

In conclusion, Latent Semantic Analysis is a useful tool for prediction task. It reduces the feature space to lower dimension and improves the prediction results.

## REFERENCES


[1] Yelp, `https://www.yelp.com/dataset_challenge`

[2] Amazon, `https://snap.stanford.edu/data/web-Amazon.html`

[3] Ganesan, K. A., and C. X. Zhai, "Opinion-Based Entity Ranking", Information Retrieval.

[4] Lee, Young-Chan. "Application of support vector machines to corporate credit rating prediction." Expert Systems with Applications 33.1 (2007): 67-74.

[5] Lim, Yew Jin, and Yee Whye Teh. "Variational Bayesian approach to movie rating prediction." Proceedings of KDD cup and workshop. Vol. 7. 2007.

[6] Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent dirichlet allocation." Advances in neural information processing systems 1 (2002): 601-608.

[7] Linshi, Jack. "Personalizing Yelp star ratings: A semantic topic modeling approach." Yale University (2014).

[8] McAuley, Julian, and Jure Leskovec. "Hidden factors and hidden topics: understanding rating dimensions with review text." Proceedings of the 7th ACM conference on Recommender systems. ACM, 2013.