Rating Prediction for Beer
Assignment 2 of CSE 255, Fall 2015

Xun Jiao [A53048611] and Zhou Fang [A53070197]

Abstract— **The increasing online reviews regarding products and services have provided an important reference for people considering pruchasing a products. The rating prediction thus can further help online shopping portals to shape their recommendation system by recommending products that might get higher ratings. In this project, we conduct the rating prediction for beer review. We conduct the analysis on the data to present the property of the dataset. Based on the properties, we implement several models for predicting the ratings. We compare several models to identify most important features, and select the best model among several models. By using a validation set, we tune the model parameter to maximize its performance compared to alternatives.**

I. INTRODUCTION

The growing of online shopping portals such as Amazon has provided a lot of data regarding to the users and products. Specifically, the reviews of users for particular products have been important factor to people to decide whether or not to buy a product. Besides, the recommendation systems usually recommend products to users which are more likely to get higher ratings from them. Thus, it is very necessary to understand what features of products will impact their reviews rating from user and how will they shape the rating.

In this assignment, we explore the data set 'RateBeer' [3] which contains nearly 3 million review for beers to build models to predict the rating of beers via several different models. The prediction are made from several different angles, with each focusing on different features of the review content.

The first angle would be using the multi-aspect of a particular product and predict the overall rating based on each aspect. From the 'RateBeer' data set, we can easily find there are four different other ratings: aroma, appearance, palate and taste. Each rating among the four should have impact on the overall rating so we would naturally use regression model to make predictions by taking the features derived from these four aspects. In the later section, we use linear regression, support vector machine (SVM) regression and random forest regression. Through the regression we can also see each feature weights differently in predicting overall rating, which will be addressed in later section.

The second angle comes from the text based prediction. In [1], the authors proposed multiscale multiaspect rating prediction for textual reviews via supervised learning methods to train predictive models and use a specific decoding method to optimize the aspect rating assignment to a review. The text-based prediction tries to extract the features from the text review and accordingly predict the rating. Although proved to be effective to some extent, the text based rating still lacks the sense that the text is directly and clearly related to the product rating.

The third angle comes from the user-item modeling approach. People jumps out of the review text itself but trying to extract the user-item model by modeling the user experience, item popularity and user-item preferences. This approach usually considers the user overall rating attribute, i.e according to a user's rating history to measure this user's "generosity" of making rating scores. It also concerns the products overall ratings from people. In other words, it tries to evaluate whether the product is "truly" good based on its overall rating history. If a product gets higher average rating then it will be more likely to get good rating from unseen users. Finally, it also considers the user's preferences onto some features and the corresponding features that could be found in the item itself. This kind of modeling creates a interaction between user and item.

In this report, we predict the overall rating for beer in data set 'RateBeer' [3]. We use mean square error (MSE) to evaluate the model performance. Various regression models and collaborative filtering models have been used with different model parameters. We split the data set into training data set and test data set in order to verify the model accuracy. For each model, we select different features to evaluate their performance.

II. RELATED WORK

In data mining community, researchers have proposed a lot of methods to predict the user-item ratings. First of all, people try to understand what features are important to an product rating. [2] extract the most important feature and opinion from users' reviews to a particular product by using unsupervised learning. In [4], the author proposed using features from several different aspects to predict the overall rating. It tries to understand the "potential" effect of features that lie in different categories on the overall rating such as taste, look and feel. It also access to the review text to extract the sentimental element towards to rating.

In [5], the authors proposed to use a latent factor model to find the hidden features that might potentially attract users. This method proved to be very effective in catching the interaction between users and products. Another review text based model was proposed [6] to predict ratings via the specific words in review text. This models concerns the unigrams and N-grams of words that occurs in the text and how they will affect the overall rating.

III. Data Analysis

In this work we select $1000k$ out of 2,924,127 reviews in the original data. The first $900k$ reviews are used as training data set and the rest $100k$ reviews are used as test data set. At first we analyzed a few features of the data set. Each review is attached a timestamp in UTC time format. We convert each timestamp back to text representation of time. It started from the year of 2000 and ended in 2011. We calculate the number of reviews in each year in between. The result is shown in Figure 1. We find that the amount of reviews increases linearly each year, as the website (http://www.ratebeer.com/) is getting more and more popular. Therefore if the data set is divided and analyzed annually, the result of first a few years are less accurate due to the smaller data set.

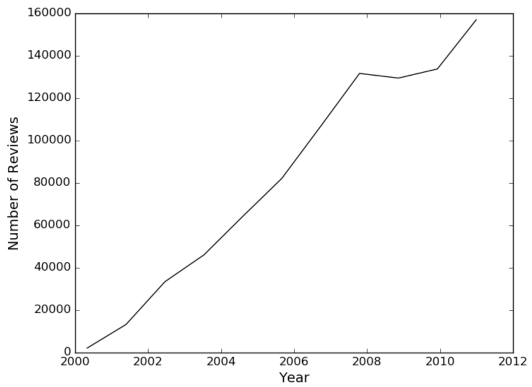

Fig. 1. Number of reviews over time.

Each review contains five rating scores: overall, appearance, palate, taste and aroma. The original data set contains review ratings like this: $6/10$ or $14/20$. Each kind of rating has different score range, to be able to study their relationship, we convert the range of all ratings to 0 to 1. For example, if the rating/overall is $14/20$, we normalize it to 0.7. Then we studied the average ratings in each category in different years. The result is shown in Figure 2. It shows that the average ratings in different years are slightly different from each other. We find that among all the five kinds of ratings, users tend to give the highest score to appearance of beer, whereas aroma gets the lowest. It means that users tend to be satisfied with appearance of beer. The ranking of score of the kinds are (high to low): appearance, overall, palate, taste, aroma. This relationship holds for the whole year range (2000 to 2011).

We are also interested at user's preference on beer of different values of Alcohol by volume (ABV). We calculate averages of five kinds of rating in different small ranges of ABV, as shown in Figure 3. It shows that ABV evidently influence beer's rating. Beers with very small ABV have very low ratings compared to beers with a medium ABV. It shows that users prefer beers with medium ABV (10 to 30).

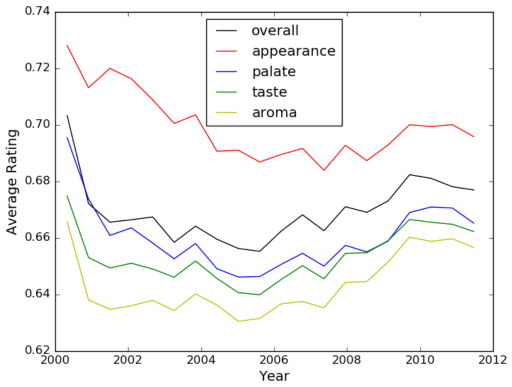

Fig. 2. Average rating over time.

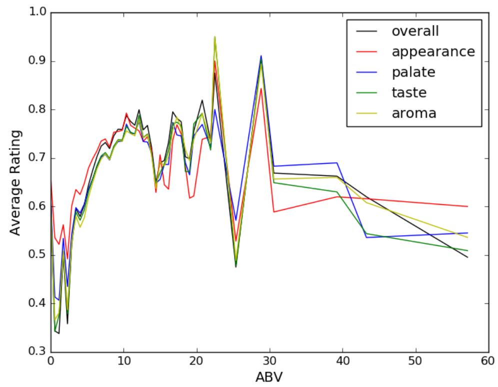

Fig. 3. Average rating over ABV.

IV. Prediction Model

Before we applying any model, we firstly perform preprocessing of the data set. After that, We select several prediction model to predict reviewer's rating of beers.

A. Evaluation

We choose to use mean squared error (MSE) as our evaluation scheme.

$$MSE = \frac{1}{n}\sum_{i \in n}(r' - r)^2 \tag{1}$$

where r' is the predicted output while r is the true output.

B. Regression

In this section, we try to use different regression model to predict the rating such as linear regression, SDG regression and some tree-based regression such as decision tree and random forest regression. We also try to filter out the most important features that could impact the rating of beer. By using a validation set, we also tune the model parameters to achieve highest performance. A discussion section is presented to discuss the feature selection, model selection and parameter selection problems.

1) Linear Regression:

Observing each review contains four categories rating: aroma, appearance, taste and palate, we try to predict the overall rating based on these ratings of these four features. We firstly predict the rating using the most-straightforward prediction model, the linear regression, where it has following forms:

$$y = X\theta \tag{2}$$

where y is the vector of output, X is matrix of features and θ is the regression coefficients determine which features are relevant. In order to solve this equation, we need to solve θ by using:

$$\theta = (X^T X)^{-1} X^T y \tag{3}$$

After solving θ out we can then apply it to unseen data and compute the MSE.

2) SVM Regression:

Support vector machine, (SVM) trying to solve a minimization problem supposed we are given training data $\{(x_i, y_i), ...(x_n, y_n)\}$

$$min_{\omega,\zeta} \frac{1}{2} \omega^T \omega + C \sum_{i \in n} \zeta_i \tag{4}$$

subject to

$$y_i(\omega^T \phi(x_i) + b) \geq 1 - \zeta_i \tag{5}$$

$$\zeta_i \geq 0, i = 1, ...n \tag{6}$$

where C is the regularizer, ζ_i is the non-negative slack variable used to measure the the degree of misclassification of the data x_i. SVM is good at reducing the classification&regression error by handling the boundary cases effectively. SVM generate a separation gap to separate different categories as wide as possible. Other than using a linear kernel for SVM, we can also use non-linear kernels such as polynomials and radial basis function(RBF).

When using non-linear kernel such as RBF the complexity of model increases to trade-off a better performance. Several parameters need to be chooses when applying this model: C and γ where C represents for regularizer and γ represents for the kernel coefficients that will be used.

3) Random Forest Regression:

The random forest method is an ensemble method. Specifically, random forest, as a tree based method, is designed based on decision tree. The decision tree algorithm will cause an overfitting problem when it becomes very deep as it will learn a lot of irregular pattern with a large variance. Random forest try to eliminate this by averaging different decision trees so to reduce the variance. Although this might hurt some "irregular" predictions, overall it will boost the prediction performance, which was later confirmed in experimental results. It is also resistant to redundant features.

In the model construction, we have several important parameter to tune the number of trees in the forest and the number of features to consider when looking for the best split. Increasing the number of trees will likely to increase the prediction accuracy but lower the running speed. The selection of both parameters needs to be aided by using a validation set, which will be addressed later.

C. Collaborate Filtering

We are interested at designing a recommendation system for beer based on the 'RateBeer' data set. The system recommend new beers to users without knowing any information about the user's opinion towards the beers. The system can only obtain recommendations from the current user's history preference and other user's opinion on new beers. We adopt collaborate filtering algorithm as the natural solution for this problem. It uses the preference similarity between users to predict the current user's preference towards unknown item. A nearest neighbor algorithm is applied to select top N_{nn} users who have most similar preference to the current user. The steps of collaborate filtering are:

- Initialize data structures, calculate reviewers' average ratings of all reviews that they gave, build dictionaries storing user-beer relation and associate rating.
- Use nearest neighbor algorithm to select N_{nn} nearest neighbor based on similarity.
- Evaluation the recommendation system, predict rating using this system on user's unknown beers and calculate mean square error (MSE) using label rating score.
- Recommend beers to the user.

The nearest neighbor selection algorithm computes the similarity between the current user and all the other users. Then it selects the N_{nn} users with the highest similarity scores. We adopt a widely used similarity algorithm: the cosine-based similarity, as shown by Equation 7. In order to increase the accuracy of nearest neighbor, we ignore the users who do not have more than N_c common beer reviews with the current user.

$$
\begin{aligned}
simil(x,y) = cos(\overrightarrow{x}, \overrightarrow{y}) &= \frac{\overrightarrow{x} \cdot \overrightarrow{y}}{\|x\| \times \|y\|} \\
&= \frac{\sum_{i \in I_{xy}} r_{x,i} r_{y,i}}{\sqrt{\sum_{i \in I_x} r_{x,i}^2} \sqrt{\sum_{i \in I_y} r_{y,i}^2}}
\end{aligned}
\tag{7}
$$

We test the performance of the recommendation system by the following approach. For the beers which are in the current user's item list, we search in the nearest neighbor list. If it is also in the item list of one or more neighbor, we apply Equation 8 to predict its rating. Then we obtain the error by comparing the predicted rating with the label rating. MSE is computed as Equation 1.

$$
r_{u,i} = \bar{r_u} + \frac{\sum_{n \in N_u} simil(u,n) \cdot (r_{n,i} - \bar{r_n})}{\sum_{n \in N_u} simil(u,n)} \tag{8}
$$

The system can recommend beers based on all the 5 kinds of rating.

V. Evaluation

In this section we evaluate and compare the performance of different prediction models. We also present our unsuccessful attempts and our key findings. Furthermore, we also discuss the selection of parameters tuned via validation set and how they could potentially impact our model. We use $900k$ data as training data set and $100k$ as our test data set.

A. Regression

We use four regression models to predict the overall ratings. The features are chosen based on the multi-aspect rating lies in four categories: aroma, appearance, taste and palate. The MSE of different models are presented in Table I.

TABLE I

MSE of regression models

Regression model	MSE
Linear regression	0.00531790
Decision tree regression	0.00516662
Random forest regression	0.00515258

The above table shows that linear regression cannot perform as good as tree-based regression. We do not show SVM regression results because it turns out that SVM runs very slow to fit $900k$ training data to build the model. We also assess the importance of different features to the overall rating, as shown in Table II. We can see each of these four features matters to the overall rating, but obviously the taste is the most important feature in the overall rating. Once the taste feature is excluded, the prediction accuracy drops a lot.

TABLE II

MSE of linear regression using different set of features

feature	MSE
aroma,taste,palate,appear	0.00532
aroma,taste,palate	0.00538
aroma,taste,appear	0.00556
palate,taste,appear	0.00582
palate,aroma,appear	0.00793

Since random forest outperforms linear regression and decision tree, we choose to focus on random forest and tune the parameters. We use an additional validation set of $100k$ to tune the parameter. There are two parameters impact the model performance: $n\_estimator$ and $max\_feature$. We find the best selection of parameters setting is $n\_estimator = 100$ and $max\_feature = log2$, resulted in the $MSE = 0.00514$.

B. Collaborate Filtering

First we have a glance of how the recommendation system works. We set the number of nearest neighbors N_{nn} to be 5, and the minimal common items between neighbors N_c to be 10. We test the prediction for a selected user 'jcwattsrugger'. The nearest neighbors of the user 'jcwattsrugger' are in Table III.

TABLE III

An Example of Nearest Neighbor

User ID	Similarity Score
'DarkBeer'	0.9983
'ucsbdude'	0.9980
'hefevice'	0.9977
'tkimbrought05'	0.9974
'ajd6c8'	0.9971

We use this set of nearest neighbor to predict the rating given by 'jcwattsrugger'. The training data set is used. Ten selected predict-label pairs is given in Table IV as an example. The fourth column, N_{sample}, stands for how many item ratings are used to get this prediction. MSE of prediction for this user is 0.004 using the training data set.

TABLE IV

An Example of Rating prediction

Beer ID	Predict	Label	N_{sample}
5368	0.78	0.75	1
51	0.67	0.7	3
53	0.73	0.75	4
52	0.64	0.65	3
2358	0.63	0.65	1
1477	0.60	0.70	2
1100	0.63	0.75	1
144045	0.72	0.70	1
102834	0.52	0.6	1
2530	0.65	0.65	1

In order to get the MSE of this recommendation system on the data set, we applies rating prediction to all users and calculate MSE for all users. Because the training data set is large ($900k$), the computation time is too long. Instead we use the test set ($100k$) to get MSE over all users. We study MSE with different choice of the number of nearest neighbors (N_{nn}), as shown in Figure 4.

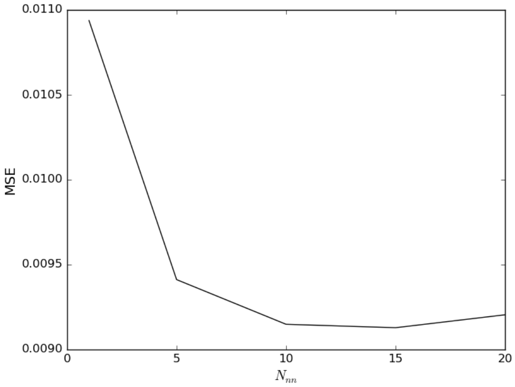

Fig. 4. Performance of collaborate filtering with different values of N_{nn}.

C. Discussion

By comparing the performance, we can see random forest regression outperforms collaborative filtering. One explanation could be that the regression use four features in the review text that could directly impact the overall rating while the collaborative filtering predict the rating only based on similar users without obtaining the information directly from a particular user or text.

VI. Conclusions

In this report, we use various regression models and collaborative filtering models to predict the beer overall ratings from a pre-processed data set 'RateBeer' [3]. The regression models take multi-aspect rating from review text as features while the collaborative filtering predict user-item rating based on the similarity check. By comparing these two models we found regression models outperforms collaborative filtering due to its extraction of the features that could directly impact the overall rating. Among the regression models we found random forest regressor has best performance compared to linear regression and SVM regression in either performance or efficiency. By using validation set to tune the model parameters we can achieve the best $MSE = 0.00514$.

References

[1] Narendra Gupta, Giuseppe Di Fabbrizio and Patrick Haffner, Capturing the stars: predicting ratings for service and product reviews(Published Conference Proceedings style), in HLT Workshops, 2010.

[2] Popescu, Ana-Maria and Oren Etzioni, Extracting product features and opinions from reviews(Published Conference Proceedings style), in Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)., 2010.

[3] RateBeer, http://snap.stanford.edu/data/Ratebeer.txt.gz

[4] Julian McAuley, Jure Leskovec, Dan Jurafsky, Learning Attitudes and Attributes from Multi-Aspect Reviews(Published Conference Proceedings style), in Data Mining (ICDM), 2012 IEEE 12th International Conference on, pages 10201025. IEEE, 2012.

[5] J. J. McAuley and J. Leskovec, From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews(Published Conference Proceedings style), in Proceedings of the 22nd international conference on World Wide Web, pages 897908. International World Wide Web Conferences Steering Committee, 2013.

[6] L. Qu, G. Ifrim, and G. Weikum, The bag-of-opinions method for review rating prediction from sparse text patterns(Published Conference Proceedings style), in Proceedings of the 23rd International Conference on Computational Linguistics, pages 913921. Association for Computational Linguistics, 2010.