Content-Aware Collaborative Filtering for Yelp Restaurant Recommendation

Arun Babu
Stony Brook University
arbabu@cs.stonybrook.edu

Rahool Arun Paliwal
Stony Brook University
rpaliwal@cs.stonybrook.edu

Syamsankar Kottukkal Sureshbabu
Stony Brook University
skottukkalsu@cs.stonybrook.edu

ABSTRACT

Yelp online reviews are invaluable source of information for users to choose where to visit or what to eat among numerous available options. But due to overwhelming number of reviews, it is almost impossible for users to go through all reviews and find the information they are looking for. To solve this problem, we create a recommender system. Recommender systems typically produce a list of recommendations in one of the two ways - through collaborative or content-based filtering. Collaborative filtering approaches build a model from a user's past behaviour (items previously purchased or selected and/or numerical ratings given to those items) as well as similar decisions made by other users; then use that model to predict items (or ratings for items) that the user may have an interest in. Content-based filtering approaches utilize a series of discrete characteristics of an item in order to recommend additional items with similar properties. In this paper, we propose a hybrid recommender system which uses a novel approach of combining content-based technique with collaborative filtering. We show that as compared to naive baseline and neighbourhood model, our method provides predictions more relevant to user's taste with similar performance to the neighbourhood model.

Keywords

Collaborative Filtering; Content Based Filtering; Recommendation System

1. INTRODUCTION

Yelp has quickly emerged as the most popular review site which has resulted in Yelp receiving massive amount of user and business data. The data varies from people's preferences and personalities, to reviews, ratings, and general information provided by the community about any business. The consumers in the world today have a vast number of choices to make for even the simplest of services. In spite of all this information being readily available, it is often hard for people to make these choices while relying on just the raw data provided by Yelp. The format in which this data is presented to a regular Yelp user is not optimal and doesn't help them make a quick, informed decision. Therefore, majority of the users feel overwhelmed with the raw data presented to them by Yelp and would instead prefer some means of getting organized, digestible data to make a quick, informed choice.

This problem is made easier for users by recommendation systems which utilize their personal preferences along with preferences of similar users to suggest potentially best choices for them. Recommender systems have become an area of active research. They are especially important in the e-commerce industry since they help increase revenues and improve customer experience/satisfaction. Popular examples of recommender systems are Facebook suggesting friendships to users, media applications such as NetFlix suggesting movies/tv shows to watch, Amazon suggesting products to buy and so on.

We aim to build a recommendation system that will enable us to make sophisticated restaurant recommendations for Yelp users by applying learning algorithms to develop a predictive model of customers' restaurant ratings.

We begin by providing a brief explanation of the dataset we used while creating our recommendation system. We follow this with a relevant exploratory analysis of data. We discuss the performance metrics we used to evaluate our results. We provide an explanation of our baseline algorithm and its performance, the algorithms we implemented, and the processes we used during our development. Finally, we conclude with comparing results obtained from the different approaches and a discussion of future work that could be explored.

2. DATASET AND TOOLS

Yelp.com is a website where users can review local business on a 5 star scale where 5 is for the best and 1 is for the worst. A user can also write a review text which gives an account of his/her experience. The website contains reviews about several different types of businesses including restaurants, shops, nightlife, and beauty spas. Our primary dataset is the Yelp Dataset Challenge data http://www.yelp.com/datasetchallenge that contains actual business, user, and users' review data along with checkin information of users and the tips users suggest for different businesses. We have constrained this project to Yelp businesses with *Restaurant* as a category.

We used Google BigQuery for filtering the data and quick exploratory analysis. Owing to the nature of the Yelp dataset, MongoDB was used to store and query data in JSON format. Python libraries NumPY, Pandas were used in development of different algorithms. MapReduce was used to process computationally intensive task of finding similarities.

Table 1: Data Statistics

Type of Statistics	Value
Number of reviews	89235
Average number of reviews per user	2.57
Median number of reviews per user	1
Maximum number of reviews per user	147

3. PRIOR WORK

Through the review of recommender system literature, we see that collaborative filtering (CF) has been the most popular way to implement recommendation system. Interestingly, CF techniques require no domain knowledge and thus we can avoid the need for extensive data collection. In addition, relying directly on user behavior allows uncovering complex and unexpected patterns that would be difficult or impossible to profile using known data attributes.

The two more successful approaches to collaborative filtering are latent factor models and neighborhood models.

Neighborhood methods compute the relationships between users or between items. In an item oriented approach, the preference of a user to an item is evaluated based on the ratings of similar items by the given user. Thus in a way, these methods transform users to the item space so that we no longer need to compare users to items, but instead directly compare items to items.

In latent factor models such as Singular Value Decomposition (SVD), an alternative approach is used to transform both items and users to the same latent factor space, thus we can compare them directly. The latent space tries to explain ratings by inferring factors about the product and user from user feedback. For instance in the case of movies as products, dimensions such as genre, amount of action, family friendly along with other less well defined dimensions are measured by the latent factor model.

Latent factor models are usually more effective at estimating the overall structure that relates to almost all items at the cost of detecting strong associations among a small set of closely related items. On the other hand, neighbourhood models does a very good job of detecting such localized relationships, but often ignore the vast majority of ratings by a user.

4. EXPLORATORY ANALYSIS

Table (1) shows the statistics for the dataset of *restaurants* in Yelp dataset.

The main problem with the Yelp dataset is its sparsity - the median review count for user is *one*. Over half of the users in the dataset have written just a single review and if we try to predict a rating for these users we do not have any other preference history to make an accurate prediction. The sparsity problem has a major negative impact on the effectiveness of the collaborative filtering approach.

As is evident from figure (1) and figure (2), both the count of reviews per person and count of reviews per business follow power law distribution, and slope is considerably steeper in the case of user-review relationship. In both of these cases, persons and the businesses having very low number of reviews is very high and it falls very steeply that the number of persons/businesses having higher number of reviews is very low.

Yet another interesting relationship was found between

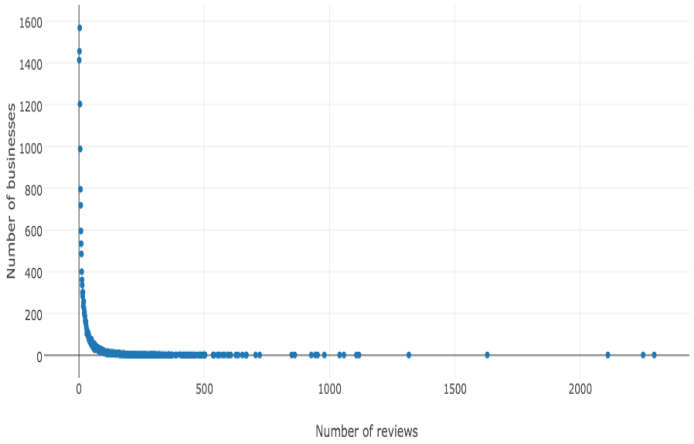

Figure 1: Distribution of reviews per business

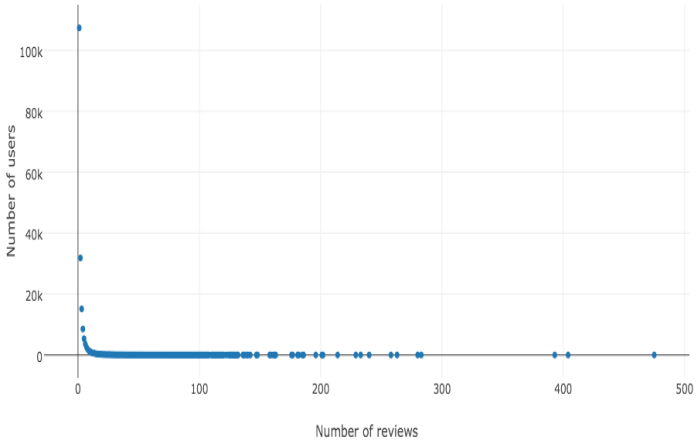

Figure 2: Distribution of reviews per user

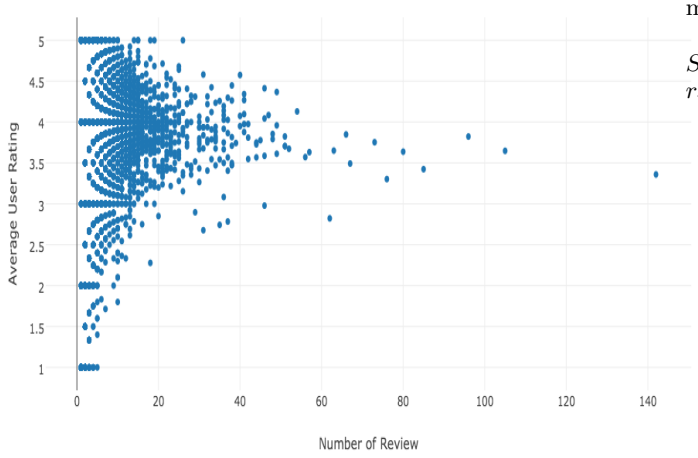

Figure 3: Distribution of avg user rating vs number of reviews by user

the number of reviews of a user and his average rating. Figure (3) shows that users who have written large number of reviews, their average rating tends to settle around in the range of 3.5 to 4 stars.

Because of sparsity, finding similarity between the preferences of two users is extremely hard since both of them may have written very few reviews and chances of common reviews between these two is thus very low. Also, even if there are common reviews, as the total number of reviews is less, the correlation measures might be unreliable.

5. EVALUATION METRIC

To evaluate a recommender system, we need metrics that can measure how close the estimates are to the actual preferences. For testing a particular prediction value, we need to compare them with the actual preference value of that user. But *actual preference* value doesn't exist, because nobody knows for certain how they might like some new items in the future. So we can simulate this by setting aside a small part of the real data set as test data. These test preferences are absent in the training data. Then the recommender is asked to estimate preferences for the missing test data, and estimates are compared to the actual values.

5.1 Metrics for Evaluating the Recommender

After considerable research on the type of evaluation metrics to be used for evaluating the recommender systems, we chose the root mean square error (RMSE) and the mean absolute error (MAE). As the Microsoft Research paper [1] suggests, both these metrics are highly accurate for measuring the effectiveness of a recommender system.

MAE is the absolute value of the difference between the predicted value and the actual value. It tells us how big of an error we can expect from the prediction on average. In MAE, we may understate the impact of big, but infrequent, errors. To adjust for such large, rare errors, we calculate the Root Mean Square Error. By squaring the errors before we calculate their mean, and then taking the square root of the mean, we arrive at a measure of the size of the error that gives more weight to the large but infrequent error than the mean.

For a user u and item/restaurant i from our training set S of data (where we know what the actual ratings r_{ui} are), r_{ui} is the actual ratings and $\widehat{r}_{ui}$ is the predicted rating.

$$\text{RMSE} = \sqrt{\frac{1}{|S|}\sum_{(u,i)\epsilon S}(\widehat{r}_{ui} - r_{uf})^2} \tag{1}$$

$$\text{MAE} = \sqrt{\frac{1}{|S|}\sum_{(u,i)\epsilon S}|\widehat{r}_{ui} - r_{uf}|} \tag{2}$$

6. NAIVE BASELINE

For our baseline, we adopted a similar but slightly modified approach to [2]. The baseline estimate $\widehat{r}_{ui}^{baseline}$ for an unknown rating r_{ui} for user u and item/restaurant i:

$$\widehat{r}_{ui}^{baseline} = \bar{r} + (\bar{r}_u - \bar{r}) + (\bar{r}_i - \bar{r}) \tag{3}$$

where $\bar{r}_u$ is average of all of a user $u's$ ratings and $\bar{r}_i$ is average of all ratings for a restaurant i. $\bar{r}$ is the average rating over all reviews.

The last two terms account for the difference in the rating which is created by user bias and business bias. The results were obtained when the entire set was partitioned into training set and testing set. The results are discussed in Results and Discussion section of the paper.

7. COLLABORATIVE FILTERING USING K NEAREST NEIGHBOURS

In this algorithm we predict the rating of a particular user based on the ratings of restaurants which were rated similarly as this restaurant.

7.1 Collaborative Filtering

In this project we are using a user-based collaborative filtering approach. Initially we calculate and store the similarity metric between two users using Pearson coefficient. To predict the rating of a new item, we compute the weighted average of user $u's$ neighbours' ratings of item i where weights are directly proportional to the similarity. Here $\bar{r}_u$ is the average rating given by user u. $sim(u,j)$ is the similarity measure between user u and user j. k is a normalizing factor so that the absolute values of the similarity metrics sum to 1. $\widehat{r}_{u,i}$ is the predicted rating given by user u to item i and $r_{u,j}$ is the actual rating given by user u to item i. This can be formally expressed as:

$$\widehat{r}_{u,i} = \bar{r}_u + k\sum_{j=1}^{n} sim(u,j)(r_{j,i} - \bar{r}_j) \tag{4}$$

where $\bar{r}_u = \frac{1}{I_u}\sum_{j\epsilon I_u} r_{u,j}$

7.2 Similarity of Restaurants and User Rating Tendency

In this approach *similarity between restaurants* are calculated by comparing the ratings given by the common reviewers for both these restaurants. As it has been discussed before Yelp has a very sparse dataset and many restaurants have no common reviewers. In these case we assume the similarity of two restaurants to be 0 and we handle it at a later stage.

While analysing the reviews of a user, here we are also taking into consideration the *rating tendency* of different types of users. While some users tend to give high ratings for the restaurants, some cranky users tend to give low ratings. Hence in this case comparatively low rating for these curmudgeon users are equivalent to comparatively high ratings from generous users.

$$r' = r - \bar{r}_u \tag{5}$$

Here r' is the normalized rating, r is the actual rating and $\bar{r}_u$ is the average rating of the user.

Once we have the rating given by common users, Pearson correlation coefficient is used to calculate the similarity between two businesses.

7.3 Pearson Coefficient

Pearson Coefficient is a measure of the linear correlation between two variables X and Y, giving a value between +1 and -1 inclusive, where 1 is total positive correlation, 0 is no correlation, and -1 is total negative correlation. Pearson correlation coefficient is the covariance of the two variables divided by the product of their standard deviations.

$$sim(x,y) = \frac{\sum_{i \epsilon I_{xy}} (r_{x,i} - \bar{r}_x)(r_{y,i} - \bar{r}_y)}{\sqrt{\sum_{i \epsilon I_{xy}} (r_{x,i} - \bar{r}_x)^2 \sum_{i \epsilon I_{xy}} (r_{y,i} - \bar{r}_y)^2}} \tag{6}$$

Here $\bar{r}_x$ is the average rating given by user x. $\bar{r}_y$ is the average rating given by user y. $\bar{r}_{x,i}$ is the actual rating given by user x to item i. $\bar{r}_{y,i}$ is the actual rating given by user y to item i. I_{xy} is the set of items rated by both users x and y.

7.4 Regularization of similarity metric

When the ratings of two businesses are compared using Pearson similarity, if the number of common reviewers of these two businesses are low, it will lead to large coefficient. Hence to penalize the similarity metric where the number of common reviewers is small, we apply a regularization parameter. We take the regularization parameter(reg) as value 2.

$$S_{m,j} = \frac{N_{common} \times \rho_{mj}}{N_{common} + reg} \tag{7}$$

Here, N_{common} is the number of common reviewers and ρ_{mj} is the Pearson coefficient between business m and j and reg is the regularization parameter.

Using our newly computed similarity metrics, we can modify our original baseline estimate by pulling in information from the user u's neighbourhood of the restaurant m, and predict r_{um} as:

$$\widehat{r}_{um} = \widehat{r}_{um}^{baseline} + \frac{\sum_{j \epsilon S^k(m)} S_{mj} \times (r_{u,j} - \widehat{r}_{um}^{baseline})}{\sum_{j \epsilon S^k(m)} S_{mj}} \tag{8}$$

where $S^k(m)$ is the k neighbour items of item m which have been rated by user u.

8. MODIFIED COLLABORATIVE FILTERING USING K NEAREST NEIGHBOURS AND JACCARD SIMILARITY

Using the previous approach, when the businesses most similar to a particular business are found, we are taking into consideration only the ratings of the common reviewers. It doesn't take into consideration the preferences of the user.

In this methodology, we are trying to give more importance to the businesses which are more closely related to the interests of the user and thus influence the recommendation for that user. We compare the *categories* field of the business, with the *categories* field of different businesses the user has rated using Jaccard Similarity.

After the nearest neighbours of a particular restaurant are calculated based on the similarity of ratings, we further prioritize the restaurants using the new Jaccard similarity metric which indicates the user's preference for a particular *category*. Thus adding this new feature, we are able to come up with better recommendations, which are more aware of the user's interest/taste.

8.1 Jaccard Similarity

The Jaccard Similarity Coefficient is a statistic used for comparing the similarity and diversity of sample set. The Jaccard coefficient measures similarity between finite sample sets and is defined as the size of the intersection divided by the size of the union of the sample set.

If C_{m1} and C_{m2} are categories of two restaurants m_1 and m_2, then the Jaccard Similarity between them is given by:

$$\left| \frac{C_{m1} \cap C_{m2}}{C_{m1} \cup C_{m2}} \right| \tag{9}$$

9. RESULTS AND DISCUSSION

In this section, we discuss and analyze our results and findings of our model on Yelp Dataset.

Dataset was divided into training and testing set. Total number of reviews taken for analysis were 89235. Around 20% of the data was separated into testing set. Training set had 71388 reviews and the testing set had 17847 reviews.

9.1 Naive Baseline

The results of baseline model are good. Owing to sparse nature of the dataset, often even incorporating complex models bring in little improvement compared to the performance of the baseline model. The RMSE and MAE values for Naive Baseline are shown in Table (2).

9.2 Collaborative Filtering using K nearest Neighbours

Simple neighbourhood model performs slightly better than the baseline model. We tried varying the number of neighbours considered during the K Nearest neighbours. When

Table 2: Naive Baseline Results

Error Metric	Error Value
Mean Absolute Error	1.0424064380505162
Root Mean Square Error	1.5233449027672543

Table 3: Collaborative Filtering using K nearest Neighbours

K-value	RMSE	MAE
3	1.03614	0.70324
5	1.16483	0.71394
7	1.22369	0.72407
10	1.32369	0.73292

the value of k was increased it was found that the error also increased. These results are intuitively aligned to the assumption that as we take more neighbours into consideration, we add neighbours which are less and less similar, thus increasing the error. The results are shown in Table (3). Figures (4) to (7) show the distribution of Actual Rating vs Predicted Rating for different values of K.

9.3 Modified Collaborative Filtering using K nearest Neighbours and Jaccard Similarity

In the modified neighbourhood model, the similarity of a restaurant to a user is also taken into consideration. Results obtained are shown in Table (4).

As it can be seen from the results in table (4) and figure (8), even though the performance of the modified neighbourhood model is very slightly worse than the simple neighbourhood model, it takes into consideration the preference of the user for a particular kind of restaurant and thus it is able to provide more intelligent and relevant recommendations. The set of categories rated by a user are indeed very representative of his/her taste and interest.

There were some observations which were evident across all these models. Deciding the k value for the neighbourhood model is a case of variance-bias trade-off. Generally when we include more neighbours, we are taking into consideration more ratings but in this case it leads more error. Hence we need to decide upon a k which suits us the best.

Also across all models and all value of k, it was found that the difference between mean predicted value and the actual predicted value is *higher towards the lower end of the ratings spectrum*. This could mean that the ratings become less predictable when a restaurant is rated low.

Yelp dataset has its own set of unique challenges. User to restaurant relation is very sparse. Even though the number of users are very high, the median number of ratings per user is 1.

Despite these challenges, we were able to come up with a reasonable recommendation system for Yelp users. On an absolute scale, with an RMSE of **1.09038**, most predictions

Table 4: Modified CF using K nearest Neighbours and Jaccard Similarity

K-value	RMSE	MAE
3	1.0903839768178607	0.70655222095358516

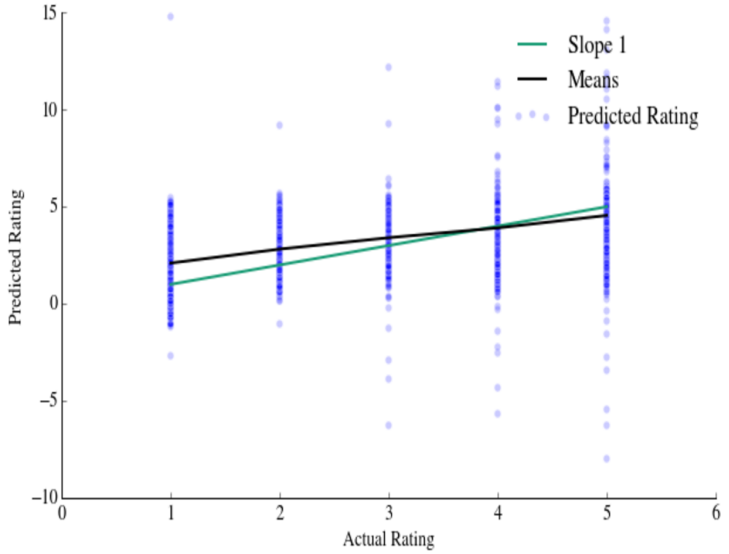

Figure 4: CF with K=3: Actual Rating vs Predicted Rating

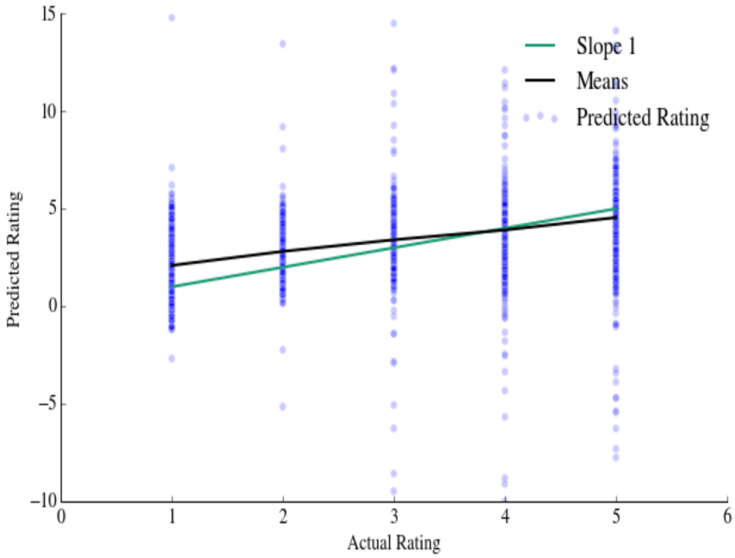

Figure 5: CF with K=5: Actual Rating vs Predicted Rating

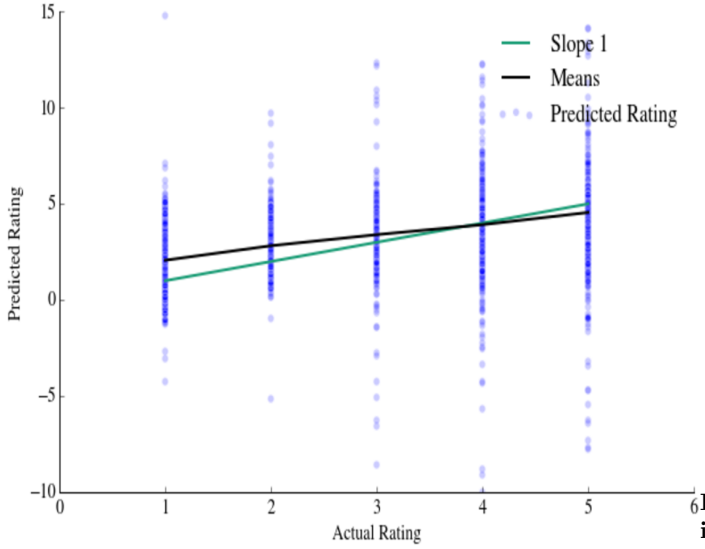

Figure 6: CF with K=7: Actual Rating vs Predicted Rating

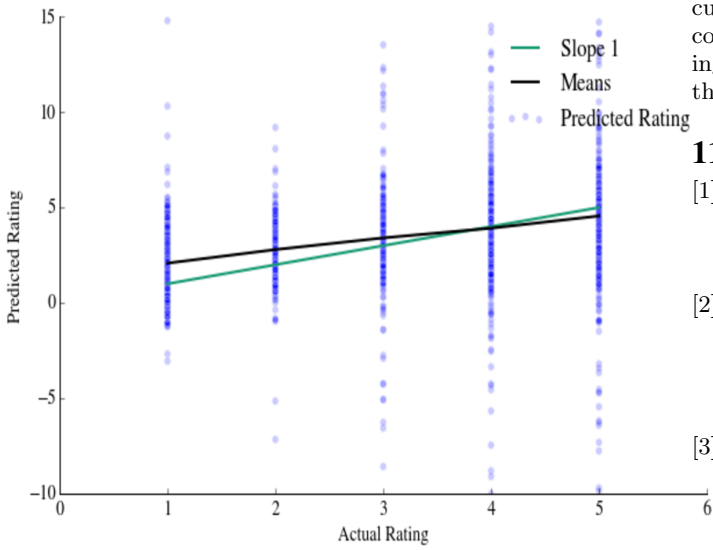

Figure 7: CF with K=10: Actual Rating vs Predicted Rating

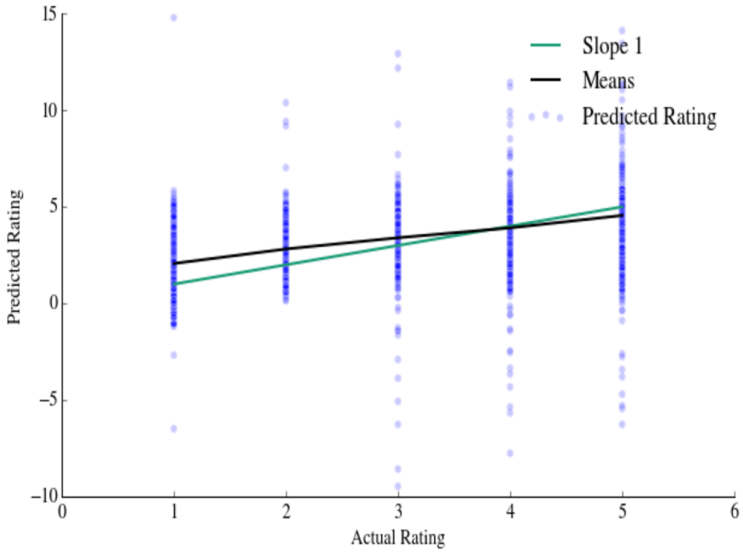

Figure 8: Modified CF with K=3 with Jaccard Similarity: Actual Rating vs Predicted Rating

are correct, rounded to the nearest star. The hybrid approach we implemented has an advantage over the baseline, even though the RMSE is only slightly better, because it takes into consideration the userâĂŹs preference for a particular type of restaurant.

10. CONCLUSIONS

Collaborative Filtering with K nearest neighbours gives reasonably good predictions. However, the recommendations might not take into consideration the type of food the customer is most interested in. We use a novel approach to combine the content-based filtering with collaborative filtering to arrive at more relevant recommendations with almost the same accuracy.

11. REFERENCES

[1] Guy Shani and Asela Gunawardana, *Evaluating Recommendation Systems,* http://research.microsoft.com/pubs/115396/evaluationmetrics.tr.pdf

[2] Robert M. Bell and Yehuda Koren, *Scalable Collaborative Filtering with Jointly Derived Neighborhood Interpolation Weights,* http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4470228&tag=1

[3] Michael D. Ekstrand, John T. Riedl and Joseph A. Konstan, *Collaborative Filtering Recommender Systems,* http://files.grouplens.org/papers/FnT%20CF%20Recsys%20Survey.pdf

[4] Vladimir Nikulin, *Hybrid Recommender System for Prediction of the Yelp Users Preferences,* http://link.springer.com/chapter/10.1007%2F978-3-319-08976-8_7