Predicting user rating for Yelp businesses leveraging user similarity

Kritika Singh
kritika@eng.ucsd.edu

Abstract

Users visit a Yelp business, such as a restaurant, based on its overall rating and often based on other factors like location, hours of location, type/cuisine or other attributes such as free Wifi. In addition to this, users gain useful insight for a Yelp business based on its top reviews and highlights. However, the average rating that a business has, or the top reviews/feedback as per certain users may not necessarily provide a user the right perspective that he/she seeks while selecting the most suitable Yelp business to visit. For example, in the case of restaurants, different people have different tastes and a high rating by person A might be due to a feature that person B does not appreciate (such as the spice content of food). Thus, even though a Chinese restaurant may have high ratings primarily because of several American raters, users with a different preference, say from the middle east may not find it palatable. A rating which takes into account the feedback of $similar$ users is expected to help people in making the right selection and enhance their experience. With this as motivation, we have utilized an additive Collaborative Filtering model to predict the rating by a user for a business, and have combined that with estimated similarity between users. We show that as compared to only using average ratings by a user, or the average ratings for a business, our method provides more accurate predictions as evidenced by low Mean Squared Errors in predicted ratings on test data.

1 Introduction

Our problem is similar to that of the Netflix prize which was about building a model that will predict the rating given by a user to a movie. These ratings were to help Netflix in making personalized movie recommendations to their users. We have worked on the Yelp dataset [1] with the aim of predicting the rating a user would give to a food business. This again will help in recommendation and search ranking. Since the Yelp dataset has more information about a business and a user than the Netflix dataset, we have tried to utilize that for rating prediction. This information is derived from the *categories* that a business is labeled with, and the *reviews* a user has written in the past. Using this information, we can find which users are similar to a particular user and give more accurate predictions. This is expected to improve the user experience as the recommendations are now closer to a user's taste/interests.

2 Related Work

Recommender systems broadly fall under two types. Content based recommender systems model the preferences of users and recommend other content based on its similarity to the content that a user views or likes. For example, in a content based music recommendation system [4], each song is manually assigned attributes or $genes$. If a user has shown interest in songs having dominant amount of acoustic guitar, similar songs will be recommended to the user. As compared to this, Collaborative filtering (CF) based recommender systems are based on the assumption that users that are similar are likely to like similar content or give similar ratings to items. For example, Amazon provides users item-recommendations based on what other $similar$ users purchased, or what users in similar shopping sessions liked. In this report, we have studied a CF based approach for estimating the rating of a business as per a user.

The input to a CF based recommender system is an incomplete matrix of ratings where each row corresponds to a user, and each column corresponds to an item (or in the context of Yelp, a business) [5]. If a user has not rated an item, the corresponding matrix entry is missing. The output of the CF based recommender system is a predicted rating for each missing entry in the matrix. There are two main general approaches for CF: the neighborhood based [2], and model based [3]. The former infers $similar$ users and predicts missing ratings by an aggregate of ratings of such users, and the latter learns a low complexity representation of the complete ratings matrix and thus predicts the missing values. Among the latter works, many algorithms have been proposed to apply and extend matrix factorization for CF problem where the input matrix is

incomplete. Unfortunately, such methods are often based on Expectation-Maximization [6] and are typically slow for large CF problems. In this study, we have defined and used a hybrid approach that utilizes the benefits of the nearest-neighbor and model based approaches. For the model based approach, we have used an additive model where the rating of (user, business) pair can be obtained as a weighted linear sum of - (1) the average rating given by the user to all businesses rated by the user, and (2) the average rating obtained by the business by all users. In order to determine a neighborhood of similar users, we utilize the common businesses and the common categories rated by the two users. For each common business, we look at the rating given by both the users. Details of our approach are provided in Section 4. In order to evaluate the model, we measure the Mean Squared Error as has been used in Netflix prize. Section 5 provides performance comparison with relevant baselines and Section 6 discusses the results and our approach.

First in Section 3 we explore the data and provide useful insights to give the reader a better context of available information and their relationships.

3 Exploratory analysis

We have constrained this project to Yelp businesses with categories Restaurants, Food and Bars. Figure below shows the overlap and relative proportions among these categories of businesses.

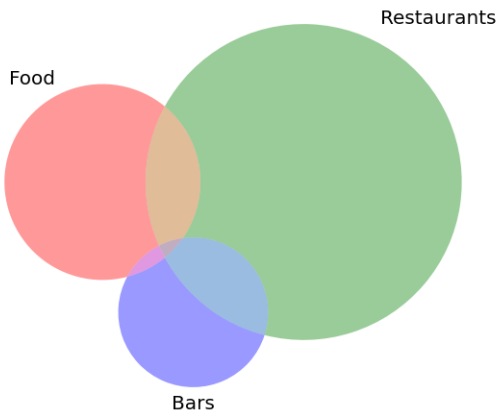


Figure 1: Businesses with categories Restaurants, Food and Bars.

In order to analyze what makes a food related business popular, we look at a variety of factors. These include the average rating (in terms of number of stars), review count, the categories a business belongs to, and several others. The complete list of attributes associated with a business is given in Table (a).

city
review count
name
neighborhoods
type
business id
full address
hours
state
longitude
stars
latitude
attributes
open
categories

(a) Businesses

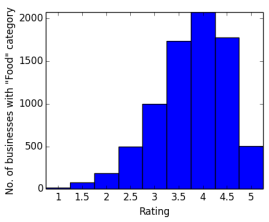
yelping since
votes
user id
name
elite
type
compliments
fans
average stars
review count
friends

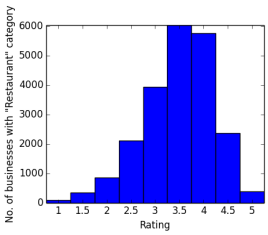
(b) Users

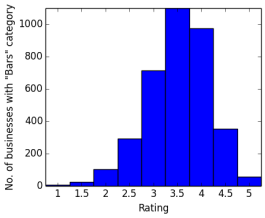
votes
user id
review id
text
business id
stars
date
type

(c) Reviews

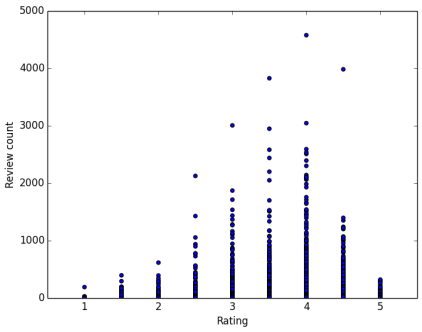
Figures below show how the distribution of business ratings vary with the business category.





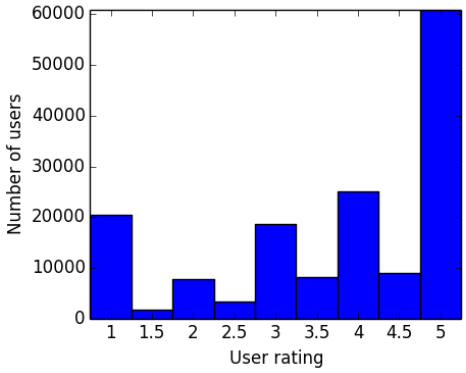


Based on the above figures a similar trend can be noted: 3.5 and 4 are the most common ratings that a food business receives from users. A very low percentage of restaurants and bars have a rating of 5.
Next, we look at how how star rating correlates with review count, which can be considered another measure of the popularity of the business.

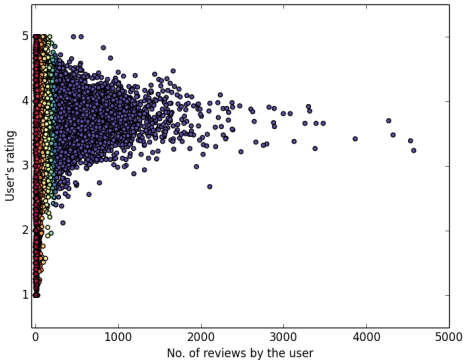


One striking observation from the above graph is that a business can have a very low(1) or a very high(5) rating only if the number of reviews is small. This makes intuitive sense for businesses with a very high rating as no business can be perfect in the eyes of a large sample of population. Another reason could be a lot of fake 1-star reviews to a new business from a competitor and a lot of fake 5-star reviews to a new business from the owner himself.

We then analyze the data of Yelp users. Table (b) lists the user attributes present in the given data.

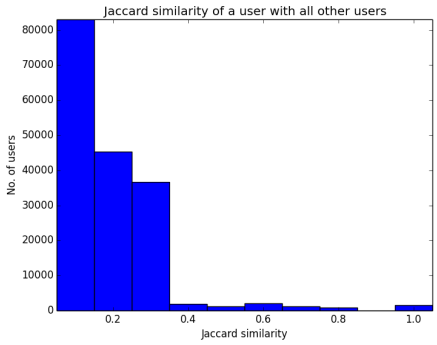


It appears that, in general, users are very trusting of other users as a considerable percentage of users have very high rating (as seen in the last figure). An explanation for this could be that Yelp ranks a user's friends reviews higher than other people and it is less likely that a person will give their friend a low rating.



A very interesting observation from the above graph is that users who have written a large number of reviews cannot not have a low rating. This might be due to them becoming good at reviewing with practice. As observed for businesses, users who have written a lot of reviews never have a rating of 5. This is because they have written so many reviews not all of which can be perfect.

One of the metrics we used to quantify user similarity between two users was the Jaccard similarity between the categories of food places rated by them. This showed that there is a considerable number of users with a high Jaccard similarity value w.r.t to a particular user. The graph below shows the Jaccard similarity between a randomly picked user with all other users.



4 Algorithm

4.1 User representation

The first step was to identify the attributes which characterize a user. The features we found useful to profile a user are:

- their yelp age in months
- their review count

- their average stars
- the number of years they have been elite
- their count of friends
- compliments users have received on their profile
- information extracted from the businesses they have rated and reviewed

Some of the above user attributes have been discussed in the exploratory analysis. An example of a feature vector before processing is given below.

```
{yelping_since: '2014-03',
review_count: 3,
average_stars: 4.33,
businesses: [
('bid1', 5, ['Steakhouses', 'Seafood'])
('bid2', 3, ['Mexican', 'American'])
],
reviews: [
"Beautiful atmosphere. wonderful service.",
"Great place"
],
elite: [],
num_friends: 0,
funny: 1,
cute: 0,
plain: 0,
writer: 2,
fans: 2,
note: 0,
photos: 0,
hot: 1,
profile: 0,
more: 0,
cool: 0}
```

4.2 User similarity

We have represented a user in vector notation based on the attributes discussed above. Once each user has a feature vector, there are multiple ways of finding similarity between two users. These include inverse of Euclidean distance, Pearson Correlation Coefficient, Cosine Similarity among several others. We used different metrics for different kinds of attributes. These are given below.

- **Cosine similarity** - To find similarity using the numerical quantities (compliments, stars, review count, etc), we used the cosine similarity metric, $cos(v_1, v_2)$, where v_1 and v_2 are vectors constructed out of numerical features.
- **Difference in rating** - This is the averaged difference in rating given by two users to the same businesses. It is given by $DR = -\frac{\sum_{i \in I} abs(rating1[i] - rating2[i])}{|I| * 4}$ where I is the set of common businesses rated by them.
- **Jaccard similarity** - To utilize the information in the categories a business is labeled with, we found all the categories both the users have rated and found the Jaccard similarity of the two sets. If $C1$ and $C2$ are the sets of categories rated by user 1 and user 2, the similarity is given by $\frac{|C1 \cap C2|}{|C1 \cup C2|}$.

The overall similarity is given by -
$sim(u1, u2) = cos(v_1, v_2) + DR + \frac{|C1 \cap C2|}{|C1 \cup C2|}$

4.3 Baselines

We chose two baselines to obtain the rating of a Yelp business by a user. These are used to provide performance comparisons with the proposed method.

- **Average User**: Predicting a rating equal to the average rating by a user.
- **Average Business**: Predicting a rating equal to the average rating of a business.

4.4 Final model

Our model is based on a simple linear combination of the baseline model predictions and the user similarity prediction. There is a weight associated with each of these components. To find the user similarity prediction by a user u for a business b, the first step is to identify top K users similar to u who have rated the business b and then average their ratings for b. The rating function can be written as -

$$rating(u, b) = w_1\alpha + w_2\theta + w_3\eta$$

Here, α is the average rating given by the user u to all businesses but b, θ is the average of rating of the business b by all users but u, and η is the average rating given by top K users similar to u to business b. While the calculation of α and θ are straight forward, to find η for (u, b) tuple, the top K most similar users to u are picked, who have also rated the business b. We have empirically set K to be 5. Since typically a user has high similarity with very less number of other users, a small value for K is justified. w_1, w_2, and w_3 sum to 1 and can be learned by hyper parameter search on the validation set with least mean squared error (RMSE) as the objective. We have used RMSE between the true rating by a user to a business and the predicted rating for the test data as the evaluation metric for our approach.

5 Results

The subset of the dataset we used consists of 50k users and all the businesses rated by them. 70% of the ratings by each user were used for training and validation, the rest of testing.

5.1 Hyperparameter search

The training did not have any parameters but hyperparameters which were learned using grid search in a 2D space on the validation set. $w1, w2$, and $w3$ were varied with the constraints $0 \leq w_i \leq 1$ and $w_1 + w_2 + w_3 = 1$. Hyperparameter K which is the number of similar users to be used while calculating η was set to 5 using a similar approach.

The set $(w_1, w_2, w_3) = (0.01, 0.43, 0.56)$ gave the best performance on the validation set. The obtained weights show that the importance of α or the average rating given by a user to all businesses is not important probably due to the high variation in ratings given by a single person. We also see the term η got the highest weight showing that it is indeed true that similar users rate similarly.

These weights were used for the final evaluation on the test set reported in the next section.

5.2 Comparison

The results obtained on the test set are shown in the table below.

Method	RMSE
Baseline 1 (Average user)	1.1629
Baseline 2 (Average business)	1.0228
Before using category based similarity	1.115
After adding category based similarity	0.8737

RMSE comparison with baseline

Several interesting and useful observations can be derived from the table above. Overall we were able to do better than the baseline. Earlier when Jaccard similarity (Section 4.2) between sets of categories rated by two users was not used to compute the similarity between them, the obtained RMSE could beat only one of the two baselines. Adding it later improved the performance by a lot and could beat both the baselines by a significant amount. This shows that the set of categories rated by a user are indeed very representative of his/her taste and interest.

6 Discussion and Conclusion

The proposed approach of combining the user similarity in a simplistic additive CF framework is seen to outperform only using average user rating or the average rating that the business received. In addition, we see that computing user similarity on the basis of categories of businesses rated by user (as calculated by Jaccard similarity in Section 4.2) leads to better performance of the proposed approach than only similarity calculated based on numerical user attributes (Cosine similarity) and Difference in rating as outlined in Section 4.2. This shows that the set of categories are quite indicative of the similarity between two users. The hyperparameter search in the space of weights on the validation set shows that rating calculated using user similarity is indeed the most important of all the factors followed by the average rating of the business being rated.

Even though the results are better than those obtained using the baseline methods, a limitation of this approach is that it doesn't scale well. If there are n users in the system, their similarity with the remaining $n - 1$ needs to be computed which is a $O(n^2)$ computation. Hence, even though this report validates the importance of user similarity and provides initial results on which sources are useful to compute the similarity, further investigation is needed to improve the performance and efficiency of the model.

In the future, review text can also be looked at along with the existing factors to compute user similarity. It is expected to give more fine grained insight into the kind of things that matter to a user and are responsible for the ratings they give.

References

[1] Yelp dataset challenge. 2014.

[2] R. M. Bell and Y. Koren. Improved neighborhood-based collaborative filtering. 2007.

[3] E. Frank and M. Hall. Additive regression applied to a large-scale collaborative filtering problem. In *AI 2008: Advances in Artificial Intelligence*, pages 435–446. Springer, 2008.

[4] M. J. Pazzani and D. Billsus. Content-based recommendation systems. In *The adaptive web*, pages 325–341. Springer, 2007.

[5] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295. ACM, 2001.

[6] S. Zhang, W. Wang, J. Ford, and F. Makedon. Learning from incomplete ratings using non-negative matrix factorization. In *SDM*, volume 6, pages 548–552. SIAM, 2006.