CS229 Project Final Report

A Personalized Recommendation System for Yelp Users

Name	SUNetID
Yinuo Yao	yaoyinuo
Fangmingyu Yang	clyang
Xin Niu	xinniu

Introduction

Yelp, a platform that consists of various business information, allows the users to provide star ratings and text reviews for businesses they visited before. This provides insights for other potential users. Particularly for restaurants, most users choose based on two different considerations, the overall ratings and the reviews posted by other users. However, there is a severe problem associated with this approach. The current rating system only provides an average value without considering any personalized information of the individual user. Thus, the efficacy of the rating system is diminished severely. It is not uncommon for the user to think of a restaurant as overrated or underrated after visiting. The underlying cause of this problem is the inability to provide a personalized rating.

The objective of this project is to construct a more sophisticated model that provides a personalized star ratings of the restaurants based on more features such as the similarity among different users.

Data Preprocessing

The dataset comes from Yelp Data Set challenge (https://www.yelp.com/dataset_challenge). There are three json dataset files used, which are Yelp_business.json, Yelp_review.json and Yelp_tip.json. Table 1 listed all the features that each data set file contains. Each data set file was processed separately to extract important information. The highlighted features are removed because they are not very relevant to the construction of the model. In this project, the business type we focused on was restaurants located in Arizona.

Table 1 Summary of features in each dataset

Yelp_business.json	Yelp_review.json	Yelp_tip.json
Business _ID	Business_ID	User_ID
Full Address	User_ID	Text
Hours	Type	Likes
Open	Votes	Date
Latitude	Date	Type
Categories	Text	Business_ID
Review Count	Review_ID	
Name	Stars	
Neighborhoods		
Longitude		
State		
Stars		
City		
Attributes		

Methodology

1. Input and output

The search is initiated by a user input of either a restaurant type, such as “Mexican Restaurant”, or a specific restaurant name like “YAYOI”. Then based on the input, we will output the related restaurants, each with a personalized star rating calculated based on other information extracted.

2. General Outline of Approach

To provide a more accurate rating as we envisioned, we need to be able to extract the group of users or restaurants relevant to the query we receive. We will describe the idea as follows, using the model with user-user similarity as an example. Other models follow the same principle, albeit with their corresponding modifications, mainly regarding the similarity metrics.

If we would like to provide a rating that takes the user's taste into account, the natural idea is to gather ratings provided by users similar to the user we are considering. However, in our dataset, there is no characteristics of the user related to their taste, due to privacy concerns. Thus, we cannot directly derive a group of users with similar preferences as the user being considered. Nevertheless, as we could imagine, similar users would likely rate restaurants in a similar fashion. For example, if user A and user B has very similar tastes and they have both been to restaurant R, then we can expect their ratings of R to be close to each other. This can be generalized to a particular restaurant category as well. Thus, given a query about a user and a restaurant that belongs to a set of categories, if we can derive a group of users that rated the set of categories similarly to this particular user, then we can aggregate their ratings of the restaurant in question and provide a rating for the user in the query. Given a query about user U and business B, in pseudocode format, we can describe the algorithm as follows.

Algorithm 1: Determining Personalized Rating

```
1   Personalized Rating (U, B)
2   {
3       GET categories of B as C
4       CALCULATE similarity (U, C)
5         FOR each User in category Ci
6             CALCULATE similarity (U, User)
7             IF User rated B
8                 ADD User to Pool
9         CALCULATE PredictedRating of B by Pool
10      CALCULATE PredictedRating of B by (U, C)
11      RETURN PredictedRating
11  }
```

As shown in the algorithm above, this will give a rating based on the ratings of users similar to the user we are considering for the particular restaurant. Some modifications to the models in our project revolve around the step to derive users similar to the one in the query. The algorithm above uses the simplest definition, which just considers the ratings. However, there could be more nuances to be explored both in the ratings and the review text. By exploring those, we can define similarities between users more accurately, which will result in better ratings provided.

3. Similarity Metric (User-User)

User-User similarity metric measures the similarity between two users on a specific restaurant type such as "Mexican" or "Chinese". The calculation is based on the heuristic method and the weighted Jaccard similarity, which defines a metric $0 \leq sim(u,i) \leq 1$. Intuitively, this similarity metric will sum the overlapped restaurants with same ratings from two different users. As a point for illustration, both user i and user j have visited 10 restaurants in common. For each user, we compute the mean rating for a particular restaurant since it is possible that a user has been to a restaurant more than one time. Say that out of the 10 restaurants, they provided same ratings for 3 restaurants, then their similarity is $\frac{3}{10}$. Therefore, based on the above similarity metric, two users' similarities will increase if they provide same rating to same restaurant more often.

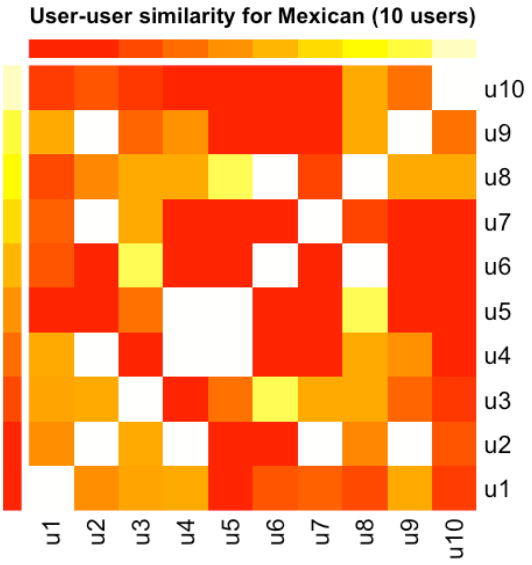

Figure 1 User-User Similarity Metric for Mexican Food (10 Users)

For Category C_p,

$$m_{C_p} = \sum_{i=1}^{m} 1\{B_i \in C_p\}$$

$$sim(u_i, u_j) = \frac{\sum_{k=1}^{m_{C_p}} 1\{\bar{s}_i^{(k)} = \bar{s}_j^{(k)} \wedge v_i^{(k)} = v_j^{(k)}\}}{\sum_{k=1}^{m_{C_p}} 1\{v_i^{(k)} = v_j^{(k)}\}}$$

where m is the number of all the unique business ID drawn from the business data set, m_{C_p} is the number

of business ID which belongs to the category C_p, B_i is the i[th] business ID. $v_i^{(k)}$ is 1 if user i has visited business k, $\bar{s}_i^{(k)}$ is the average star ratings on business ID k from user i. $\sum_{P=1}^{N_c} m_{C_p} \geq m$ since one business ID can have multiple labels, where N_c is the total number of categories associated with the restaurant business type. Figure 1 shows the pairwise similarity among ten users for Mexican while the details are shown in Figure 2 in appendix.

4. Similarity Metric (User-Category)

As compared to User-User similarity metric, the User-Category similarity metric measures the similarity between an individual and a category. This is a way to explicitly model the user's preference in a particular category. The calculation was based on the weighted Jaccard similarity as below. To be specific, we sum the number of restaurants belonging to category C_p as a fraction of the total number of restaurants user i has visited.

$$sim(u_i, C_p) = \frac{\sum_{k=1}^{N_R^{(i)}} 1\{B_k \in C_p\}}{N_R^{(i)}}$$

where $N_R^{(i)}$ is the total number of review from user i. The matrix is visualized in Figure 3 where the white block denotes the strongest similarity.

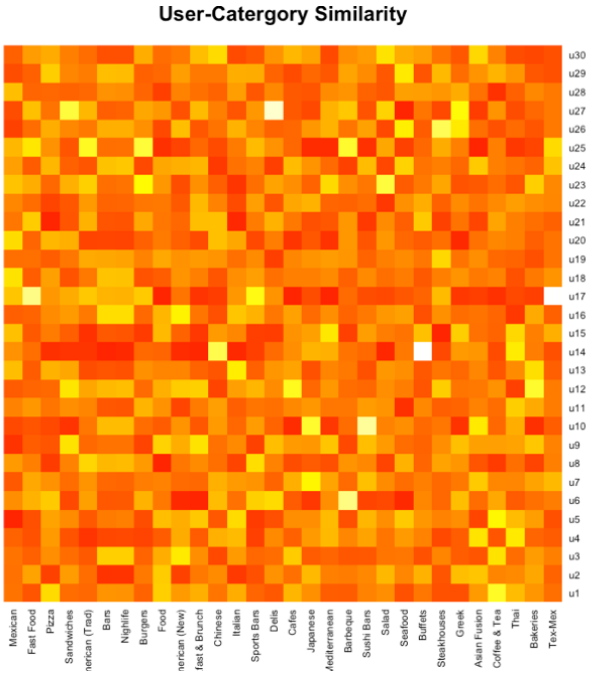

Figure 3. User-Category Similarity Metric (30 users and 30 categories)

5. Algorithm

With our similarity matrix between users and users as well as our similarity matrix between users and categories, we are ready to put together our algorithm for presenting a personalized rating for any restaurant for any user.

Given a user u and a restaurant b, we first get the set of categories of b as C. For each of the category in C, we create a matrix with each row representing a restaurant in that category and each column representing a user. Each entry of the matrix is the rating of a restaurant by a user. For each of these matrices, we first determine the similarity between u and the rest of the users, which comes from our similarity metric between users and users. To calculate the predicted rating for b by u, we first center the ratings for b by each of the other users around the average rating for all the restaurants in this category by each of the other users. The reason for this adjustment is that each user may have different scales for their ratings. By centering these ratings, we remove their individual variances. Then we aggregate the centered ratings using the similarity between users and users as weights. Adding to this aggregate rating the average rating for all the restaurants in this category by u, we have our predicted rating for b by u under this category. Furthermore, we can again aggregate the predicted ratings using the similarity between users and categories as weights. This produces the final predicted rating for b by u. It can be written in the following form (given user j and restaurant k):

$$d_j = s_j^{(k)} - \frac{\sum_{l \neq k}^{m_{C_p}} v_j^{(l)} s_j^{(l)}}{\sum_{l \neq k}^{m_{C_p}} v_j^{(l)}}$$

$$\widehat{s_{C_p}}_i^{(k)} = \frac{\sum_j sim(u_i, u_j) d_j}{\sum_j sim(u_i, u_j)} + \frac{\sum_{l \neq k}^{m_{C_p}} v_i^{(l)} s_i^{(l)}}{\sum_{l \neq k}^{m_{C_p}} v_i^{(l)}}$$

$$\hat{s}_i^{(k)} = \frac{\sum_{C_p} sim(u_i, C_p) \widehat{s_{C_p}}_i^{(k)}}{\sum_{C_p} sim(u_i, C_p)}$$

The idea behind this algorithm is to predict a rating for b by u that is as close as we can estimate to the real value.

How to insure a great prediction?

First, in order to avoid the bias we could have introduced by taking those ratings at their face value, we centered the ratings of b by other users according to their respective average ratings of restaurants. Second, we weigh the ratings of b by other users using their similarity metrics with u to guarantee that the opinions from users similar to u are valued higher. Last but not least, by calculating a predicted rating under each category that b belongs to and aggregating them together using similarity metrics between u and all those categories, we try to make the prediction more accurate since a restaurant can and often belongs to multiple categories. **However, not only do we care about the prediction, but we try to make inference of the model why leads to the outcome.** We will discuss it in detail in the following.

6. Results

After we developed our algorithm, we tested it on many users and restaurants. The algorithm itself will produce real values for the rating. But since rating in Yelp is always an integer, we rounded the results from 1 to5. We define the error as follows.

$$error = \frac{\sum |predicted - actual|}{\sum actual}$$

Figure 4 shows an example of how users rated a particular restaurant.

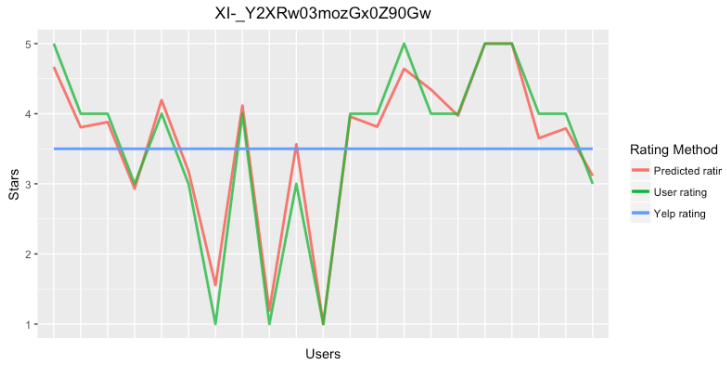

Figure 2 Comparison of Predicted Rating, User Rating & Yelp Rating for all the users that have visited one business

The X-axis is users and the Y-axis shows ratings. As indicated in the legends, the blue line is the Yelp rating. This is the rating everyone would see for a particular restaurant on Yelp, which means it's the same for everyone. Thus, it's a horizontal line. The green line shows what each user actually rated after they went to this restaurant. The orange line shows our prediction through the algorithm. As can be seen from the figure, the orange line closely follows the green line, which means that our prediction is a decent predicator of the users' rating. Compared to the Yelp rating, which is the same for everyone, our prediction is a huge improvement. If our prediction results are shown to the users on Yelp instead of the current Yelp ratings, we believe it would be much more helpful for them to determine if they want to go to this restaurant or not.

Figure 5 is how a particular user rated for different restaurants.

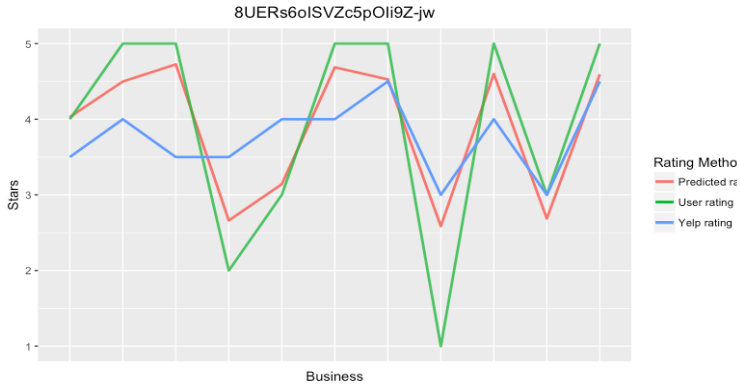

Figure 3 Comparison of Predicted Rating, User Rating & Yelp Rating for all the business that one User has visited

The blue line shows ratings for different restaurants by all the users. The green line shows ratings for different restaurants by the particular user. The orange line is our prediction. As can be seen from the figure, our prediction is closer to the actual ratings by the particular user, meaning that our prediction is a better predictor for the user's rating than the average rating by all the users. Table 2 below shows detailed error rate for five restaurants.

Table 2 Error Rate of Prediction Produced by Algorithm

Restaurant Yelp ID	Error Rate
If2DUhmWIvlu2JFc_rR-Bw	0.0950
yExYqENb4F6qH6kJxTOaSQ	0.1028
LXhL5X3edNRy7epku6UAEw	0.0883
XI-_Y2XRw03mozGx0Z90GW	0.0265
kR5i58Pcse1FD9tk-yYLlA	0.0807

Yelp ID was used instead of name because it was not given. All the restaurants we tested and tallied are summarized in Table 3 in the appendix. The average test error rate for 30 restaurants was approximately **0.0652**. We can arrive at the conclusion that our algorithm makes the predicted rating more personalized and more relevant by the fact that we are valuing ratings from different users differently based on their similarity with the user we are considering.

7. Conclusion

As shown throughout this project, we indeed improve the current rating system in the way that is more relevant to the users. We first developed two similarity metrics to measure similarity between users and users as well as between users and categories. We then designed an algorithm to calculate a predicted rating for a specific user using these similarity metrics by utilizing other users' rating of the restaurant of interest. This method is more trustworthy than the overall average rating system because it values more the opinion of users that are similar to the one we are predicting for.

Concerns for personalized ratings

From the perspective of user, it is indeed more helpful and enjoyable to use personalized rating than the current same-for-all, but it may cause some **personalized ads** as well. This might not be a problem to some people but it is for some users.

From the perspective of Yelp, it could benefit from the more accurate recommendation but the challenge here is that the data is updated nearly second by second which require a **great capacity of computation** and also need to determine how frequently they would to update the model.

Why didn't compare with other personalized methods?

We could achieve lower **prediction** error on training set without knowing the true model behind the data. However, there is a high variance on dataset and this is not our goal for the project. **Inference** is much more valuable to us because knowing exactly why people rate like this is the root of **user decision and business action**, which are the meaning of similarity metric we defined. Besides, the **calculation workload** of our algorithm is not intense thanks to using the idea of distance. A more complex model might achieve lower error but at an extremely high cost of computation. As a result, we don't think simply developing another prediction model without any justification of its mechanism would help us understand the performance of our model.

8. Future Work

An obvious extension is to study the text part of the review in addition to the stars. Even if two users rate a restaurant the same, their reviews could be vastly different, which show their respective characteristics. Algorithms such as **Latent Dirichlet Allocation** can be used. We could even analyze the user's review text to **extract preferences** and analyze review text of a restaurant to **extract features**. If the preferences match the features, then the rating should be high. Another possibility, which is beyond the scope of predicting a rating, is to **rank** the results in a better way. For example, if a user values price the most, then among the results returned, we should list the cheapest restaurant first. These additions will make the recommendation more useful.

All our code could be accessed in Github: https://github.com/FangmingyuYang/Yelp-challenge.git

Appendix

Table 3 Test error rate of 30 different restaurants randomly selected from the Yelp database

Restaurant	Error Rate
Sample 1	0.0276
Sample 2	0.0846
Sample 3	0.1371
Sample 4	0.0823
Sample 5	0.0694
Sample 6	0.1317
Sample 7	0.0950
Sample 8	0.0934
Sample 9	0.0438
Sample 10	0.1381
Sample 11	0.0765
Sample 12	0.0795
Sample 13	0.0586
Sample 14	0.0489
Sample 15	0.0445
Sample 16	0.0646
Sample 17	0.0709
Sample 18	0.0754
Sample 19	0.0276
Sample 20	0.0679
Sample 21	0.0655
Sample 22	0.0262
Sample 23	0.1123
Sample 24	0.0498
Sample 25	0.0959
Sample 26	0.0340
Sample 27	0.0585
Sample 28	0.0223
Sample 29	0.0751
Sample 30	0.0255

Figure 2. User-User Similarity Metric for 12 different categories