Yelp Recommendation System Using Advanced Collaborative Filtering

Chee Hoon Ha
Stanford Univerisy
cheehoon@stanford.edu

1. INTRODUCTION

Thanks to the advancement in technology, we live in a world where everything runs faster than ever. With the easy access to computers and smartphones, people now can achieve anything they desire faster than ever. However, at the same time, the expectation for rapid, accurate, comfortable, and convenient services are rising. And to help those people who can't stop seeking for restaurants with delicious foods and pleasurable services, Yelp provides a perfect recommendation service.

Yelp aggregates review data from its users and rank restaurants based on them. It does a fantastic job of suggesting appetizing restaurants. However, there seems to be room for improvements. The problem is Yelp provides same rank for everyone. Especially in a diversified country like United States, every people have different taste for food. Some people like Mexican food while others like Asian food. Some people care about taste only while others care about decor and services. Yelp is not addressing this problem right now, but if we can build a system that can identify a user's preferences and provide customized rankings for each individual users, people will benefit more from the service.

Our work will focus solely on predicting the latent rating value that a user would have given to a certain restaurant, which then can be used to rank all the restaurants including those that have not been rated by the user. First part of our paper will provide a brief explanation of the dataset that will be used throughout the paper. We then follow this with an explanation of baseline algorithm and other possibly better algorithms.

2. PROBLEM FORMULATION

We will try to predict star ratings of all restaurants for each users. In particular, assume that there are n users and m restaurants. Our goal is that given a set of training examples, define the matrix $A \in R^{n*m}$ as following:

A_{nm} =	{ r_{nm},	user n's rating on restaurant m	If user has given a rate
	{ ?,	undefined	If user has not given a rate

Our work will focus on predicting all elements with question marks in matrix A, which can have an integer value in between 1 and 5. To test the performance of algorithms, we will use RMSE (root mean squared error) with k-fold cross validation (10 fold).

$$\text{RMSE} = \sqrt{(1 / |S|) * \sum_{(n,m) \in S} (r_{mn} - \check{r}_{mn})^2}$$

3. Dataset

Yelp provides a portion of its data through Yelp Dataset Challenge event. The dataset includes 42,153 businesses, 252,898 users, and 1,125,458 reviews, which include star rating in the range of 1 to 5 and user's opinions in text. This dataset includes business other than restaurants, which is not what we want. In addition, only a handful of users have written reviews on more than 15 restaurants, which we believe is the minimum number of reviews required for accurately predicting a user's preferences. And for restaurants too, only a handful of them have more than 20 reviews, which we thought was the necessary number of reviews for restaurants. After all the trimming, we reduced our dataset size to >4,000 restaurants, >25,000 users, and >100,000 reviews.

4. Algorithms

In this paper, we use several collaborative filtering (CF) methods to figure out each "questions marks" in the matrix defined above. First, we will use basic CF as our baseline. Then using K-Nearest Neighbors and clusterings and SVD, we will show how each algorithm performs compared to the others.

4.1. Baseline

For our baseline, we used the similar baseline method implemented by Yehuda Koren[1].

$$b_{ur} = \mu + b_u + b_r$$

Here, μ is the average rating of all restaurants by all users, b_u is the difference between μ and the user u's average star ratings, and b_r is the difference between μ and the restaurant r's average star ratings, and b_{ur} is the predicted star rating from user u to restaurant r. Specifically, in the case where average ratings of all the reviews in our dataset is 3.6, user u's average rating is 4.1, and restaurant r's average rating is 3.2, then $b_{ur} = 3.6 + 0.5 - 0.4 = 3.7$. After repeat this process until it converges, this will normalize the widely noticed tendency of some user giving higher rating than others and some restaurants getting higher ratings than others.

This mean predictor is the possibly simplest type of predictor that can be calculated rapidly provided that there is sufficient amount of data. However, it has a poor RMSE value of 1.4742, making it a valid candidate for our baseline.

4.2. K NEAREST NEIGHBORS

People with analogous preferences tends to act and think in similar fashion. For Yelp, if it is possible to figure out n users who has similar preferences as user u, predicting user u's ratings on unrated restaurant, by utilizing information from the n users, is possible.

In particular, consider the case where we want to predict how user u would rate on restaurant r. We first find n users who has rated restaurant r and determine how similar the traits of user u and that of n users are by comparing other restaurants that both u and members of n have rated. Then use the Euclidean

distance to calculate the similarity between the two users, and use them to predict the possible ratings that user u would give to restaurant r.

$$s_{uj} = \sum_{r \in R} (r_{ur} - r_{jr})^2 \; / \; |R|$$

Here, R is the list of restaurants where both u and j have written reviews and s is a vector representing the similarity between user u and user j. The smaller the value is, the closer their preferences are. Thus, we could use this vector as weights to predict the user's future ratings by

$$\acute{r}_{ur} = b_{ur} + \sum_{i}^{n} (r_{ir} - \mu - b_i - b_r) * s_{ui} \quad \text{where } s_{ui} \textit{ is not undefined}$$

After learning all $\acute{r}_{ur}$, we get RMSE score of 1.5274, which is worse than our baseline model. The reason that this model produces such low RMSE score is that there simply is not enough data to form a well-validate vectors. Yelp is a type of service where there are lots of businesses, but just a few reviewers. This sparseness made it very difficult to accurately predict with this model because two different users barely write reviews to same restaurant, resulting in worse RMSE.

4.3. K NEAREST NEIGHBORS WITH CLUSTERING (TYPE OF FOOD)

As seen in the previous model, there was this problem of sparseness. It is very unlikely that two users write review on a same restaurant, resulting in an unreliable outcome. To solve this problem, this model will form clusters among restaurants and treat each of the clusters as if they were an individual restaurant. Then we can apply collaborative filtering with a more abundant vector information.

For this model, we assume that the most influencing factor when rating and choosing a restaurant is the type of food. For instance, if restaurant a and b are both italian restaurant, they would have been treated as same restaurant i.e. they are in the same cluster.

After filtering and also manually categorizing restaurants that have similar type of food, we came up with tree structure that defined all the food types like Figure 1 shows. We experimented this model with the identical predictor used in the previous model with only the leaf node of the tree we structured. The obtained RMSE was 1.3091.

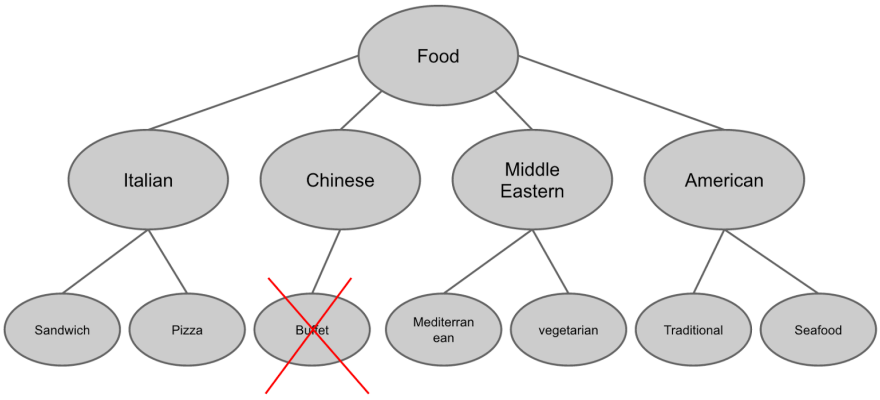

Figure1. Types of food categorized by Yelp and our group.

Red X shows that we have manually taken out that node because it was the only type in chinese food.

4.4. K NEAREST NEIGHBORS WITH CLUSTERING (TYPE OF FOOD AND STYLE)

For this model, we made an assumption that when people choose restaurant, they first remove all the type of food that they do not enjoy eating and just choose one of the remainings depending on their mood instead of directly choosing most desirable food type. This says that the type of food may not be the most critical factor in choosing restaurant, and maybe the price, services, decor, or combination of those are the more important factor in choosing them.

For example, when people are discussing about what to eat, there can be several types of food that one is willing to eat, but one might choose not to go to a restaurant with too classic, trendy or hipster style even if the food is delicious. Thus, taking whether the restaurant is romantic, touristy, hipster, classy, trendy, or casual into consideration seems to be a reasonable choice.

But this diversification process brings us back to the sparseness problem again. Thus, for this model, we are going to use the Figure 2, which has more general categories of food (Italian or American instead of Pizza or Sandwich), to form clusters, and experimented this model with the identical predictor used in the previous model and obtained RMSE of 1.1235.

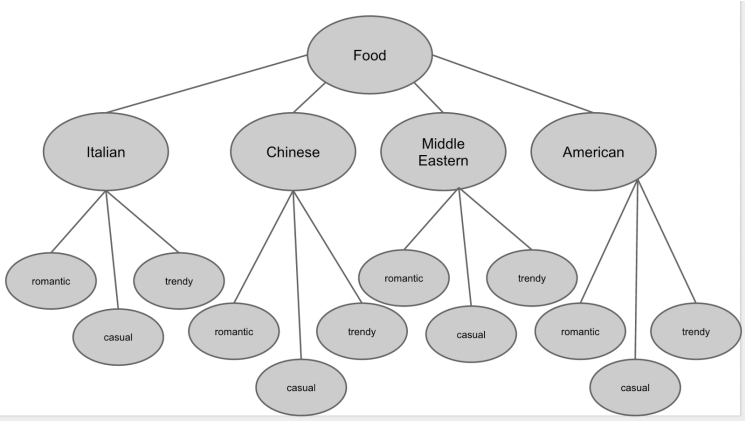

Figure 2. Categorize by type of food and style of restaurant.

4.5. Singular Value Decomposition (SVD)

Up until now, we have focused only on k nearest users and restaurants to solve the problem of sparseness. However, there is another approach to this problem that is based on sparse matrix SVD. This model approaches the problem by assuming that there is some factors that majorly contribute in determining ratings. Thus, projecting user and restaurant relationship into a lower dimensional feature space. This feature space is approximated by

$$argmin_{ui,rj} \sum_{(i,j)\in R} (A_{ij} - \mu - u_i{}^T r_j)^2 + \lambda(||u_i||^2 + ||r_j||^2)$$

In order to solve this problem, we alternate solving u and r using least square until it converges as shown by Yunhong Zhou, Dennis Wilkinson, Robert Schreiber and Rong Pan[2].

5. VALIDATION AND RESULT

Algorithm	RMSE
Baseline	1.4742
k nearest neighbors	1.5274
k nearest neighbors with restaurant clustering (Type of food)	1.3091
k nearest neighbors with restaurant clustering (Type of food and style)	1.1235
SVD	1.3822 (Based on result, this model seems to be incomplete and need update)

In this experiment, we used k-fold cross validation to evaluate the RMSE value. As the result shows, k nearest neighbors with restaurant clustering (Type of food and style) performs best out of 5 algorithms.

6. FUTURE WORK

For the future work, we will first implement the combination of all the models that have been used in this paper. Then, we will be focusing on more advances CF using bipartite graph and text classification that can reduce the problem of biased users and restaurant. We will rate the restaurants solely based on the text as it has been demonstrated on Jack Linshi's paper [3]. After that, we will try to merge all the algorithms to get the best performance.

REFERENCES

[1] Yehuda Koren, Collaborative Filtering with Temporal Dynamics <http://sydney.edu.au/engineering/it/~josiah/lemma/kdd-fp074-koren.pdf>

[2] Yunhong Zhou, Dennis Wilkinson, Robert Schreiber and Rong Pan, Large-scale Parallel Collaborative Filtering for the Netflix Prize <http://www.hpl.hp.com/personal/Robert_Schreiber/papers/2008%20AAIM%20Netflix/netflix_aaim08(submitted).pdf>

[3] Jack Linshi, Personalizing Yelp Star Ratings: a Semantic Topic Modeling Approach <http://www.yelp.com/html/pdf/YelpDatasetChallengeWinner_PersonalizingRatings.pdf>