Where should I go for Breakfast/Lunch/Dinner?

A recommendation system based on advanced collaborative filtering algorithm with restaurants similarity

Zhonghui Xie
University of Southern California
zhonghux@usc.edu

Yueqi Zhu
University of Southern California
yueqizhu@usc.edu

Xinyi Shen
University of Southern California
xinyishe@usc.edu

Yilun Wang
University of Southern California
yilunwan@usc.edu

ABSTRACT

Recommendation is a crucial part of user experience. And for many e-commerce companies, designing an effective recommendation system is the foundation of improving customer retention rate. A key factor of implementing recommendation systems is to comprehend how decision-making process works in real world. Collaborative filtering is one of the most common approaches to model this process and be used for recommendation.

In this paper, we will introduce a new collaborative filtering method to recommend restaurants to a user. Our method is an advanced version of traditional user-based CF which also benefited from item-based CF. To evaluate the system we built, we separate the dataset by time, extracted filtered data after a given date and utilized these data to verify the achieved results.

KEYWORDS

Geographical convenience, recommendation system, user based collaborative filtering, rating, similarity

1 INTRODUCTION

Recommendation system has become a top topic for many industry-leading companies. And collaborative filtering is a well-used method for recommendation. However, data sparsity which is a key disadvantage of user-based collaborative filtering often results in unreliable similarity information. To solve this problem, we propose an algorithm that applies item similarity to the calculation of user similarity.

In addition, geographical and temporal factors will also heavily influence a user's actual decision-making. Thus, these factors are also included in our work.

The rest of the paper is organized as follows. First, we introduce the data we use in section 2. Then the details of the implementation present in section 3. In section 4, the results are analyzed and evaluated, and a concise discussion on the result is provided.

2 DATASET

The goal is to recommend restaurants for a user based on predicting this user's rating on candidate restaurants. To achieve that, we need users' ratings on these restaurants.

Considering the size of the whole dataset and run-time performance, we decide to filter restaurants from a selected area as our target dataset. At first, we combine yelp review dataset and user dataset as well as sort these data by location. After checking the order of this list, we choose the second-ranked Las Vegas as the candidate sample set. However, due to the data's massiveness and irregular distribution, we further narrow down the scope and selected a neighborhood (i.e., Strip) as our experimental subject.

Then we filter out every review which was not make about a restaurant in Strip area. And we sort the left data based on the date on which each review was made. For each input user, we select all this user's reviews and chose a temporal demarcation point based on these reviews. Finally, all review data made

before this point in the Strip area would be used as training dataset.

Also, Considering the serving timeliness of restaurants, we also process the data to provide suitable meal type information of recommended restaurants to the target user. The details are shown in in Figure 1.

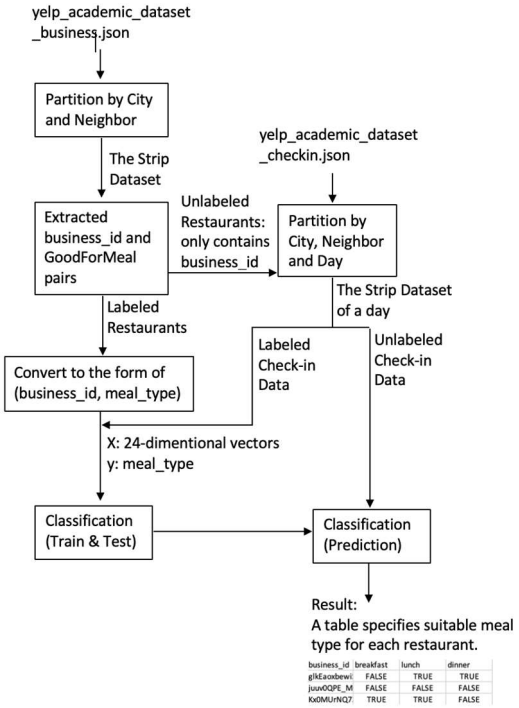

Figure 1: the data processing flowchart of predicting suitable meal for each restaurant.

3 METHODOLOGY

3.1 Time Category

For every business in a specific day, we create a 24-dimensional vector to represent the number of customers checking in per hour. Applying PCA to these check-in vectors shows there is a strong relation between customers' check-in activities and restaurant's suitable meal types shown as Figure 2(a). Depending on the GoodForMeal attribute in business dataset, we generate a set of labeled data points to train and test the classification model. After trying different linear classifiers including Lasso and Bayesian Ridge with 73 percent accurate, the finial result shown as Figure 2(b) is produced by OneVsRestClassifier using Linear SVC and achieves 78 percent accurate (the method of calculating accuracy is shown as equation [1]). Finally, restaurants that lacks GoodForMeal attribute can be labeled according to check-in data through this model.

$$eq[1]\quad accuracy = \frac{the\ number\ of\ correct\ prediction\ for\ each\ meal\ type}{3 * number\ of\ businesses\ in\ testing\ data\ set}$$

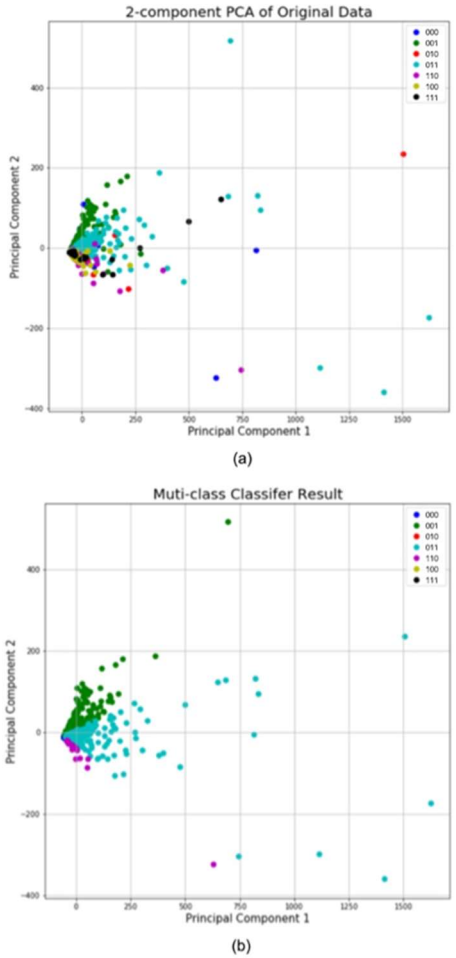

Figure 2:(a) PCA result of training data set. (b) PCA result of testing data with predicted labels. The rightest bit represents dinner, the most left bit represents breakfast, the bit in the middle represent lunch. Bits are set to 1 if the corresponding meal type in GoodForMeal attribute is true.

3.2 User Similarity

We use Pearson correlation to compute user similarity.

$w_{u,v}$, the similarity of user u and v is:

$$eq[2]\quad w_{u,v} = \frac{\sum_{i \in I}(r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I}(r_{u,i} - \bar{r}_u)^2}\sqrt{\sum_{i \in I}(r_{v,i} - \bar{r}_v)^2}}$$

$r_{u,i}$ is the rating of business i by user u.
I is the set of businesses rated by both user u and v.
$\bar{r}_u$ is the average rating of user u over all businesses in I.

3.3 Business Similarity

We use Jaccard similarity to compute business similarity.
The similarity of business a and b is:

$$eq[3] \quad SIM(a,b) = \frac{|C_a \cap C_b|}{|C_a \cup C_b|}$$

C denotes the category of corresponding business.

3.4 Rating Prediction

Let x be the inputted user ID, and y be the inputted business ID. Given a (x, y) pair, apply User-Based Collaborative Filtering (CF) using Review_Rating as training dataset to compute a predicted rating for the pair. In addition to traditional User-Based CF, in our CF algorithm, we also use the most similar business of y as a factor in our computation.
By selecting all ratings by x and for y, we compute user average rating $\bar{x}$ and business average rating $\bar{y}$ from Review_Rating. If cannot found any rating by x or cannot found any rating for y, $\bar{x}$ or $\bar{y}$ will be *3* correspondingly.
Then select all other users U that been to y. For all users in the list U, use the method in section 3.2 to find each users similarity with x, that is $w_{(u,x)}$. If all users in U has 0 similarity to x, then return $(x, y, \bar{x})$ as predicted rating. Else, find average rating $\bar{u}$ for each $u \in U$, and use equation [4] and equation [5] to calculate a numerator and denominator. Then user equation [6] to calculate direction predicted rating.

$$eq[4] \quad DirectN = \sum_{u \in U} (r_{u,y} - \bar{u}) * w_{(u,x)}$$

$$eq[5] \quad DirectD = \sum_{u \in U} |w_{(u,x)}|$$

$$eq[6] \quad DirectPredicted = \bar{x} + \frac{DirectN}{DirectD}$$

So far, the described method is traditional User-Based CF. Next, we will use similar business as a supporting factor in our calculation of final predicted rating. First find the most similar business z to business y using the method in section 3.3. Then find all other users been to z, let the list be T. Compute the similarities between t $\in T$ and x, $w_{(t,x)}$. If $T = \emptyset$ or all $w_{(t,x)} = 0$, then will return $(x, y, DirectPredicted)$, as we are not using z in our calculation. Else we will continue to calculate a supporting predicted rating from z. Because this calculation is not about y, so we need use a weight, $SIM(y, z)$ from section 3.3, to control the contribution of business z to our final predicted rating result. And the equation to calculate a numerator and denominator is as below equation [7] and equation [8]:

$$eq[7] \quad Support = \sum_{t \in T} (r_{t,z} - \bar{t}) * w_{(t,x)} * SIM(y,z)$$

$$eq[8] \quad SupportD = \sum_{t \in T} |w_{(t,x)} * SIM(y,z)|$$

Using this equation, we can combine the numerator and denominator from both direct prediction and similar business prediction as below equation [9]:

$$eq[9] \quad FinalRating = \bar{x} + \frac{DirectN + SupportN}{DirectD + SupportD}$$

This calculation will be our final prediction of given (x, y) *p*air. And the return $(x, y, FinalRating)$ as final predicted rating.

3.5 Recommendation to Input User

To decide which restaurant to recommend to users, we find all restaurants that input user x had not visited, Let the list be B. For every pair $(x, b) : b \in B$, calculate the predicted rating using method in section 3.3. Filter B to only contain resturants that have higher than x's average rating and return those restaurants.
To recommend restaurant based on time of the day, we further filter B with the results from section 3.1. For each given pair (x, d), for $d \in \{'breakfast', 'lunch', 'dinner'\}$, filter the output list with the inputted choice of meal kind.

4 RESULT & DISCUSSION

There are two type of results from our work. One is basic recommending restaurants to users, and another is recommending restaurants that are good at the time of day preferred by user.

4.1 Basic Restaurants Recommendation

Input: userid: qQecSd0lynfB4g-LPa9JCw

Output: CSV file of (userid, business, predicted rating)

qQecSd0lynfB4g-LPa9JCw	Yp9w4nhUowBU_IS_StFXbQ	4.92857
qQecSd0lynfB4g-LPa9JCw	0q_BHpxbikVtPRRLRu-U0g	4.82924
qQecSd0lynfB4g-LPa9JCw	nDKcQkh8vZkY_u1kyww44g	4.73077
qQecSd0lynfB4g-LPa9JCw	pmoIMK8zGWvKsVOPDBYfzg	4.73077
qQecSd0lynfB4g-LPa9JCw	-Ylpy3VyRWwubf9dysuwjQ	4.73077
qQecSd0lynfB4g-LPa9JCw	ntfDRwVlUb3nmWdMdPjq0Q	4.5
qQecSd0lynfB4g-LPa9JCw	G58YATMKnn-M-RUDWg3lxw	4.5
qQecSd0lynfB4g-LPa9JCw	ruWTngdiC68O9la27hvvHw	4.44593
qQecSd0lynfB4g-LPa9JCw	aKEEQqL1UFMieilnylI1gw	4.3651
qQecSd0lynfB4g-LPa9JCw	-3zffZUHoY8bQjGfPSoBKQ	4.36017
qQecSd0lynfB4g-LPa9JCw	qmh6zxtJ8C8-YiUPv7ySlw	4.30973
qQecSd0lynfB4g-LPa9JCw	oJZNHz5UUVUgrZwVBVlpYw	4.3
qQecSd0lynfB4g-LPa9JCw	lQ9VmIWvM6v-rONPkYX8aQ	4.26065
qQecSd0lynfB4g-LPa9JCw	pKk7jCFIm96qDdk0laVT2w	4.25983
qQecSd0lynfB4g-LPa9JCw	_j2EtQtgLuXGRBfbM5YwZA	4.25

Figure 3: A capture of the first 15 recommended business to user ID qQecSd0lynfB4g-LPa9JCw

For the inputted user ID, we have a result of recommending 55 restaurants. By using testing data to verify our output, we can see that out of the 55 restaurants, the user did go to 6 of our recommended results.

```
Ratio:
Number of restaurants user visited 6 / Total number of restaurants recommended 55
```

Figure 4: A capture of the ratio of correct prediction in all recommended restaurants from the output file for user ID qQecSd0lynfB4g-LPa9JCw

4.2 Time Based Restaurants Recommendation

Input:(userid, time): (qQecSd0lynfB4g-LPa9JCw, 'lunch')

Output: CSV file of

(userid, business, predicted rating, breakfast, lunch, dinner)

(See figure 5 for output result)

For the output result, it is hard to verify the correctness because we do not know what time of the day the user visited the restaurants. But we can still verify whether the user visited or not. *(see figure 6 for output result*)

business_id	rating	breakfast	lunch	dinner
-Ylpy3VyRWwubf9dysuwjQ	4.730769	False	True	True
oJZNHz5UUVUgrZwVBVlpYw	4.300000	False	True	True
_j2EtQtgLuXGRBfbM5YwZA	4.250000	False	True	True
Ou8pYS24azDWG0ru_vUcqg	4.245935	True	True	True
Cni2l-VKG_pdospJ6xliXQ	4.236111	False	True	True
ujHiaprwCQ5ewziu0Vi9rw	4.139029	False	True	True
OVTZNSkSfbl3gVB9XQIJfw	4.137143	False	True	True
MnYGGxWpMyQ7oncUPjbEmw	4.000000	False	True	False
RycZOiohghoI0Ssg2Qqgqw	3.947619	True	True	False
tjYHsz4ydS6GuBSv-uifQA	3.928571	False	True	True
d_L-rfS1vT3JMzgCUGtiow	3.750000	False	True	True
4JNXUYY8wbaaDmk3BPzlWw	3.644777	False	True	True
2weQS-RnoOBhb1KsHKyoSQ	3.595336	False	True	True
ZCQa7CJxZ-53Zxd_pobWug	3.575797	False	True	False
ZkGDCVKSdf8m76cnnalL-A	3.504026	False	True	False

Figure 5: A capture of the 15 recommended business to user ID qQecSd0lynfB4g-LPa9JCw for lunch time

```
Ratio:
Number of restaurants user visited 3 / Total number of restaurants recommended 15
```

Figure 6: A capture of the ratio of correct prediction in lunch recommended restaurants from the output file for user ID qQecSd0lynfB4g-LPa9JCw

4.3 Discussion of Other Findings

For the outputs of predicting rating, many predicted ratings are the average rating of the inputted user ID. This is because some of these users who also visited that business have no similarity with the input user. Therefore, we choose to recommend restaurants that have predicted rating higher than user ID's average rating.

5 APPENDIXES

GitHub Link

https://github.com/sxy1412/INF553-Recomandation-System

Individual Contribution

Zhonghui Xie – Predict rating

Yueqi Zhu – Build similarity table, verify predict result

Yilun Wang – Data pre-processing

Xinyi Shen – Time category