A

Project Report

On

Prediction of Yelp Review Star Rating Using User Based Collaborative Filtering

By,

Soumya Medapati

University of Memphis

Computer Science Department

Under the guidance of,

Dr. Deepak Venugopal

Assistant Professor

University of Memphis

Computer Science Department

Abstract

Online reviews and their rating play a very important role in information dissemination and are currently the most influencing factor in user decisions. In this project, we have utilized reviews ratings of yelp data challenge[1] to predict how likely a given user would be interested in particular business. The user-user based collaborative filtering algorithm has been used in generating the predictions. The metrics used to evaluate the obtained results are RMSE and accuracy. We have also analyzed the influence of the number of similar neighbors (K) considered on the root mean square error (RMSE) and accuracy of predicted ratings.

Introduction

Personalization of product information has become one of the most important factors that impact a customer's product selection and satisfaction in today's competitive and challenging market. Personalized service requires firms to understand customers and offer goods or services that meet their needs. Recommendation systems are widely used by e-commerce practitioners and have become an important research topic in information sciences and decision support systems. Recommendation systems are decision aids that analyze customer's prior online behavior and present information on products to match customer's preferences. Through analyzing the customer's purchase history or communicating with them, recommendation systems employ quantitative and qualitative methods to discover the products that best suit the customer. Most of the current recommendation systems recommend products that have a high probability of being purchased. They employ content-based filtering (CBF), collaborative filtering (CF), and other data mining techniques, for example, decision tree, association rule, and semantic approach. In this project, we build our recommendation system based collaborative filtering (CF).

The Dataset from Yelp challenge has been used. Yelp aims to help people and great local businesses, e.g. restaurants. Automated software is currently used to recommend the most helpful and reliable reviews for the Yelp community, based on various measures of quality, reliability, and activity. However, this is not tailored to each customer. Our goal in this project is to apply machine learning to predict a customer's star rating of a restaurant based on his/her reviews, as well as other customers' reviews as shown in Figure 1.

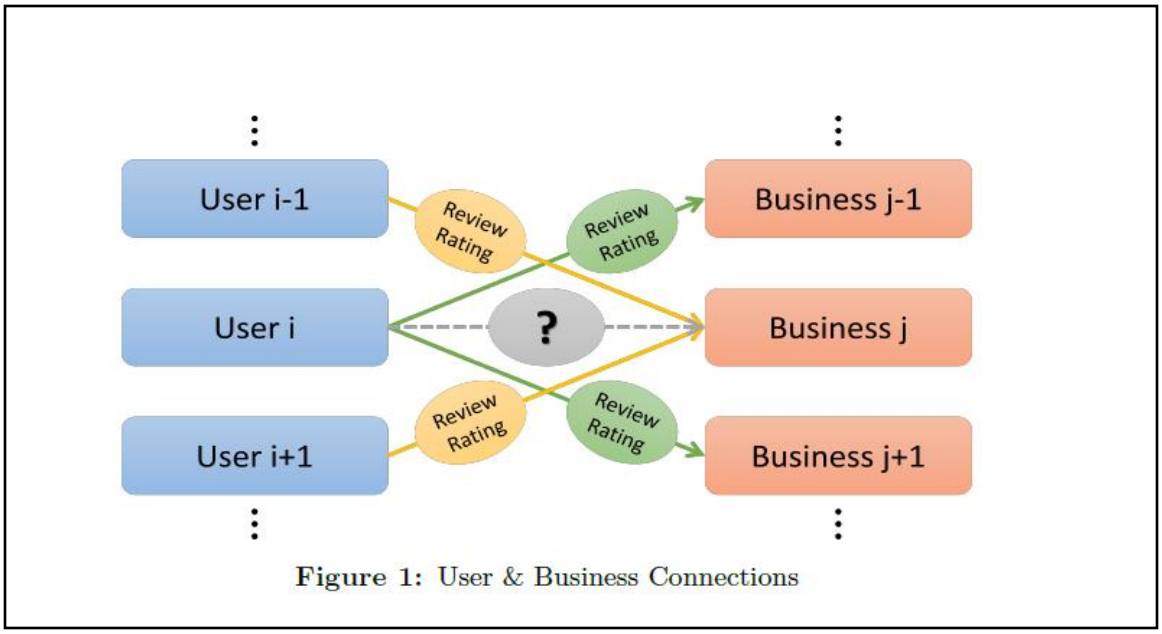


Figure 1: User & Business Connections

Related Work

The GroupLens [5] is one of the earliest implementation of collaborative filtering recommendation system based on ratings. The GroupLens research system provides a pseudonymous collaborative solution for Usenet news and movies. Later on, the item-based and user-based collaborative filtering recommendation systems are proposed [4], which are widely used now by large companies such as Amazon and Dell. We implement the user-based collaborative filtering system in this project.

Dataset description

The dataset considered in this project comes from the Yelp Dataset Challenge (Round 8) [1]. As part of this challenge Yelp releases information about reviews, users and businesses from 4 US cities. The dataset is available for download on Yelp's contest page and contains the following information:

- **4.1M** reviews and **947K** tips by **1M** users for **144K** businesses
- **1.1M** business attributes, e.g., hours, parking availability, ambience.
- Aggregated check-ins over time for each of the **125K** businesses
- **200,000** pictures from the included businesses

The data is quite consistent, with very limited amounts of missing data. It does however, have other weaknesses. For example, Yelp introduced voting on its reviews where Users could vote "Useful", "Funny" or "Cool" on reviews, thus indicating which reviews should be promoted. Since the "useful" voting feature on yelp was only introduced recently, many good reviews may not have been marked useful.

Preprocessing of data

Each file in the dataset composed of a single object type, one json-object per-line. This was converted to the comma separated file (CSV format). We removed all the data we are not interested in to keep the dataset size (2.15 GB) manageable. Thus we were left with only the review data file which had the following structure:

Review{
'type': 'review',
'business_id': (encrypted business id),
'user_id': (encrypted user id),
'stars': (star rating, rounded to half-stars),
'text': (review text),
'date': (date, formatted like '2012-03-14'),
'votes': {(vote type): (count)}, }

Amongst these attributes we have considered user_id, business_id, stars and votes. Due to the extreme sparse nature of data, we used the csr matrix of scipy library to store it. The 8 digit ASCII form business id and user id increased the time required to read the data and perform computations on it. Thus the ASCII ids were mapped to unique integer ids and used as row, column indices for the csr matrix. The 'votes' column had values ranging from 0 to 75000, these were scaled to range of 0 to 1 to obtain the weight of each review.

Methodology/ Algorithm

In this project we have implemented the user based collaborative filtering algorithm [2] in python 3.5 to generate the predictions. The task in collaborative filtering is to predict the utility of items to a particular user (the active user) based on a database of user votes from a sample or population of other users (the user database). It is a memory based algorithm which operates over the entire dataset to obtain predictions. It involves the following steps:

(i) Mean vote computation: The user database consists of a set of votes vi,j corresponding to the vote for user i on item j. If Ii is the set of items on which user i has voted, then we can define the mean vote for user i as:

$$\bar{v}_i = \frac{1}{|I_i|} \sum_{j \in I_i} v_{i,j}$$

(ii) User-User Similarity: Each user in the dataset can be represented as a vector having the ratings he has given for the products he bought. Cosine similarity can then be used to compute how similar the given user is with every other user in the dataset.

$$w_{a,u} = \frac{\sum_{i \in I} (r_{a,i} - \overline{r}_a)(r_{u,i} - \overline{r}_u)}{\sqrt{\sum_{i \in I} (r_{a,i} - \overline{r}_a)^2 \sum_{i \in I} (r_{u,i} - \overline{r}_u)^2}}$$

To make the similarities more meaningful, we have considered the votes along with their ratings. The average vote of active user and user i is computed and scaled to range (0,1). This value is added to w(a,i) to give the final weight of review. Considering votes enables us to get an idea of how useful that particular review is and weight it accordingly. We sorted the list of similarity values for each active user to get k most similar neighbors.

(iii) Predicting the rating: We predict the votes of the active user (indicated with a subscript a) based on some partial information regarding the active user and a set of weights calculated from the user database. We assume that the predicted vote of the active user for item j, Pa,j, is a weighted sum of the votes of the other users:

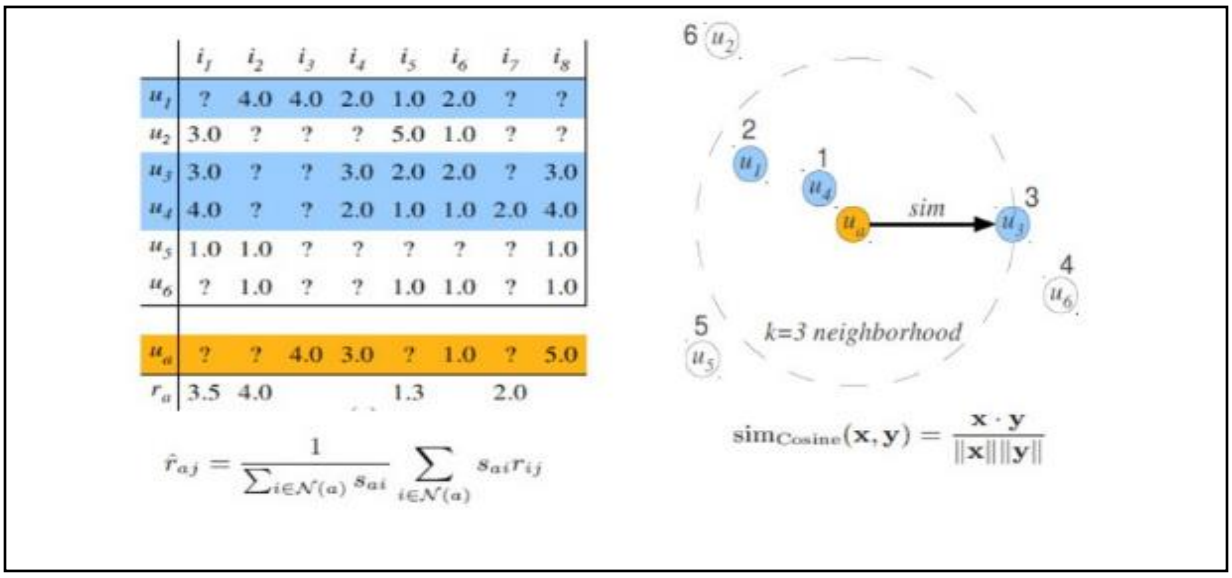
$$p_{ai} = \overline{r_a} + \sum_{u \in K} (r_{ui} - \overline{r_u}) w_{au} / \sum_{u \in K} w_{au}$$

where $\overline{r_u}$ is the average rating for u

w_{au} is similarity weight between a and u

K is the set of nearest neighbors for a based on w_{au}

The User based Collaborative filtering algorithm is illustrated as below in figure 2:



	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8
u_1	?	4.0	4.0	2.0	1.0	2.0	?	?
u_2	3.0	?	?	?	5.0	1.0	?	?
u_3	3.0	?	?	3.0	2.0	2.0	?	3.0
u_4	4.0	?	?	2.0	1.0	1.0	2.0	4.0
u_5	1.0	1.0	?	?	?	?	?	1.0
u_6	?	1.0	?	?	1.0	1.0	?	1.0
u_a	?	?	4.0	3.0	?	1.0	?	5.0
r_a	3.5	4.0			1.3		2.0	

Figure 2

Results and evaluation

The above proposed methodology was run on the entire dataset (500000 tuples) containing users, business and ratings attributes. Root mean square error (RMSE) and accuracy were used as the evaluation metric and were computed as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$$

Where y_i = predicted value of rating $y_{i(bar)}$ = actual value of rating in data N= total no of ratings

Accuracy= (No of times predicted value is equal to actual value)/ N * 100

The values of the root mean square error over 10 folds were recorded. Further we also considered the votes attribute to determine the weight of each review and the new RMSE values were obtained. These are shown below in table 1.

Folds	RMSE (with votes)	RMSE
0	1.169	1.193
1	1.168	1.195
2	1.168	1.209
3	1.168	1.163
4	1.156	1.161
5	1.150	1.153
6	1.171	1.173
7	1.160	1.165
8	1.165	1.168
9	1.175	1.200

Table 1

The average accuracy obtained on the entire dataset was around 60%.

Analysis

(i) The bar graph of Number of neighbors (k) versus RMSE is shown below (Fig 3). It can be observed that when 2 neighbors are taken into account for prediction computation, the RMSE value is 1.20 whereas when 10 neighbors are considered, the RMSE value drops to 1.06. Thus, more the number of neighbors considered the lesser is RMSE value leading to better predictions.

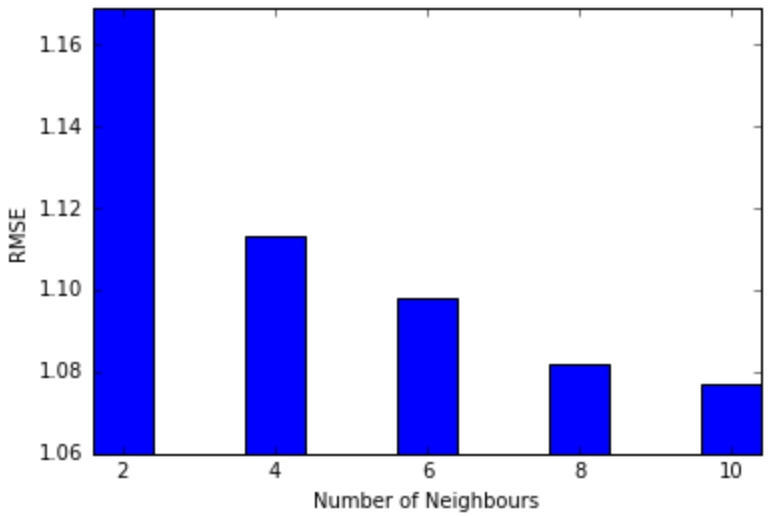


Figure 3

(ii) The bar graph of Number of neighbors (k) versus Accuracy is shown in fig 4. As observed the accuracy obtained when 2 neighbors were considered is 58.98% whereas the accuracy obtained when 10 neighbors is comparatively very less (49%). This shows that, considering more neighbors does not always give maximum accuracy. Some neighbors are irrelevant to active users and considering them in prediction calculation increases the deviation from the actual rating.

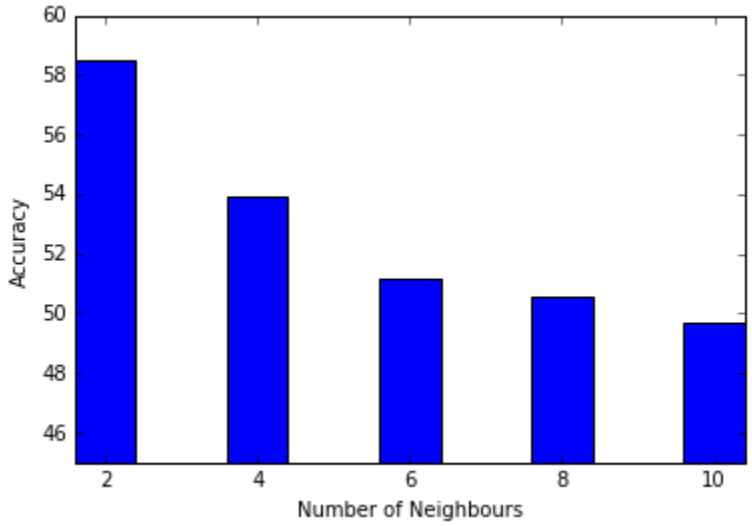


Figure 4

(iii) Plot of Number of data rows versus Running time is shown below. It is observed that running time increases linearly with the number of data rows. For number of rows

less than 100000 running time is around 50seconds while it is 315 seconds for 500000 data rows.

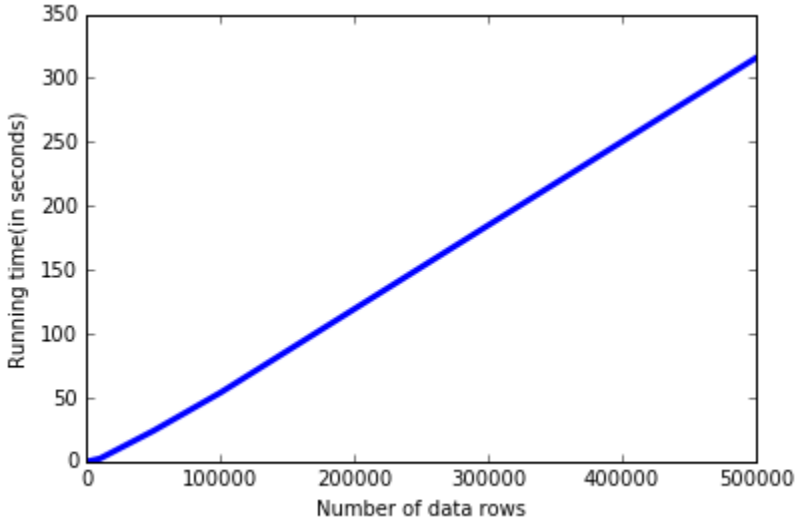


Figure 5

(iv) Graph of percentage of unique users versus RMSE is shown as below. When 20% of users are considered the RMSE value is 1.178. This reduces to 1.068 when all unique users are considered for prediction. Thus the more data we consider better are the predictions we obtain.

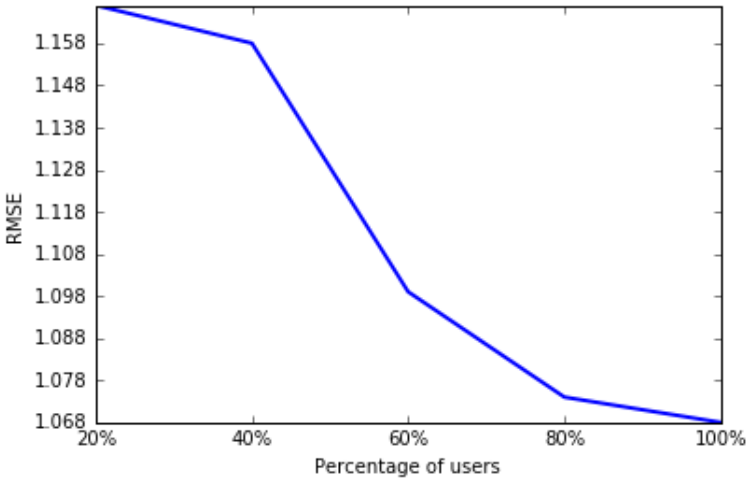


Figure 6

Conclusion

We have efficiently implemented a recommendation system which can predict the ratings for given (user, business) pair based on the existing ratings of other similar users. Analyzing various factors we can conclude that:

i) The greater the number of similar neighbors considered the lesser is the value of RMSE
ii) More number of neighbors considered does not guarantee maximum accuracy of predictions. Thus optimal number of neighbors for max accuracy of predictions differs for each dataset.
iii) The running time of CF algorithm is directly proportional to the number of datarows taken into account.
iv) The higher the percentage of users, the lesser is rmse thus leading to more accurate predictions.

Future Work

This project only considers the "useful" votes and rating to weight the reviews of each user. The predictions can be improved by considering the "review text" [3] as words can better define a customer's satisfactions level as compared to numbers. It can further extended to extract tips from predicted ratings.

References

[1] Dataset: https://www.yelp.com/dataset_challenge

[2] "Empirical Analysis of Predictive Algorithms for Collaborative filtering" by John S.Breese, David Heckerman, Carl Kadie. UAI'98 Proceedings of the fourteenth conference on Uncertainty in artificial intelligence.

[3] "Sentiment Analysis of Yelp's Ratings Based on Text Reviews." Yun Xu, Xinhui Wu, Qinxia Wang, Stanford University.

[4] Guanwen Yao and Lifeng Cai, et al. "User-Based and Item-Based Collaborative Filtering Recommendation Algorithms Design"

[5] Konstan, Joseph A., et al. "GroupLens: applying collaborative filtering to Usenet news." Communications of the ACM 40.3 (1997): 77-87