Collaborative Filtering-based Beer Recommendation
SI650 Final Project

Yu Xia

December 15, 2022

1 Introduction

As the world of online beer shopping continues to grow, so too does the need for personalized recommendations that can help consumers find the perfect brew. In this paper, I present a collaborative filtering-based approach for recommending beers to consumers that incorporates singular value decomposition (SVD) to improve the performance of the recommendation system.

Previous works have used collaborative filtering for beer recommendation, but have struggled with the sparsity of the ratings data. Solving the problem of beer recommendation is important because it can help consumers find new and interesting beers that they are likely to enjoy. Our approach uses SVD to decompose the ratings matrix into low-rank matrices, which can then be used to make predictions for missing ratings, which can help to alleviate this problem.

The main contribution of our project is the development of a collaborative filtering-based approach that incorporates SVD to improve the performance of the recommendation system. Our results demonstrate the effectiveness of our method in this context and provide valuable insights into the use of this technique for beer recommendation.

Figure 1: Different kinds of beers.

2 Data

2.1 Data Source

The dataset "BeerAdvocate" is obtained from Dr. Julian McAuley's ICDM 2012 paper, Learning Attitudes and Attributes from Multi-aspect Reviews. The link to the dataset is https://cseweb.ucsd.edu/~jmcauley/datasets.html#multi_aspect.

The dataset holds 1586615 interaction records between 33388 users and 66055 beers. Specifically, each record of the dataset contains the name, brewer and other information of a beer as well as the ratings given by a user. A sample record of the original dataset is shown below:

From all the fields, I will use only three of them, review_profilename, beer_name and review_overall to build my recommender system.

Field name	Sample value
brewery_id	10325
brewery_name	Vecchio Birraio
review_time	1234817823
review_overall	1.5
review_aroma	2
review_appearance	2.5
review_profilename	stcules
beer_style	Hefeweizen
review_palate	1.5
review_taste	1.5
beer_name	Sausa Weizen
beer_abv	5
beer_beerid	47986

Table 1: A sample record of the original dataset.

2.2 Preprocessing Steps

To improve the usability of the data, I make the following preprocessing steps.

1. Map the user names and beer names into integer indexes for simplicity and efficiency of the model.
2. Identify and drop duplicate and null records and alter the data type of rating from string to float.
3. Visualize as shown in Figure 2 the distribution of the ratings across the entire dataset.
4. Collect statistics as shown in Figure 3 of how many ratings each user makes and how many ratings each beer has.
5. Filter out users that make less than 30 ratings and beers that has less than 15 ratings, which reduces the size of users to 6150 and the size of items to 10115.

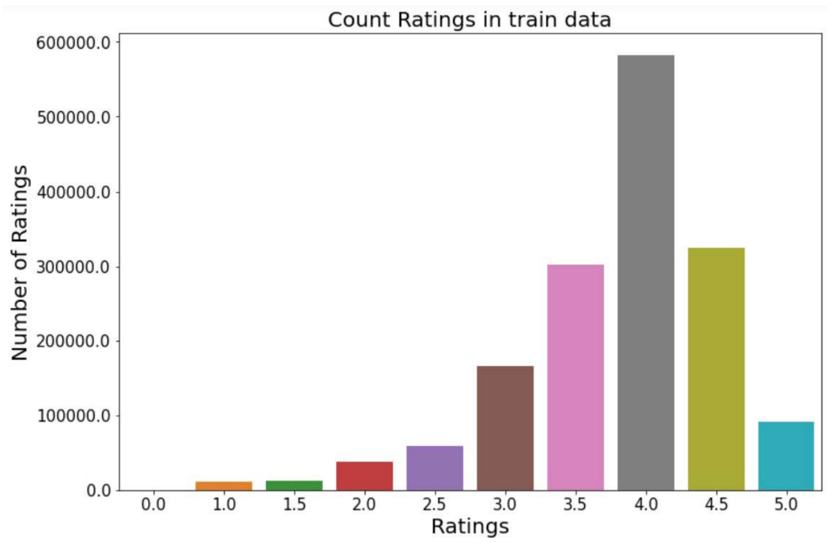
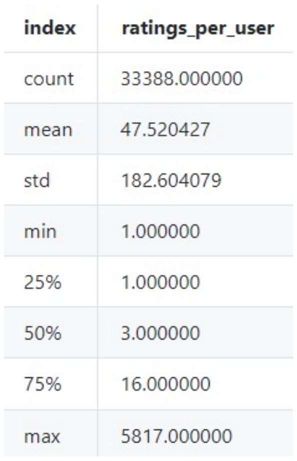
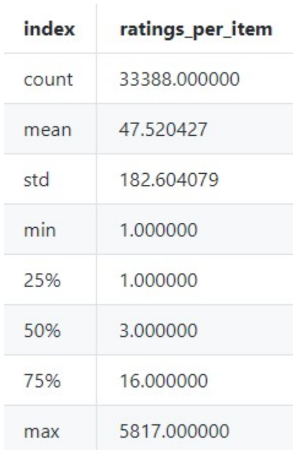


Figure 2: Distribution of ratings.



index	ratings_per_user
count	33388.000000
mean	47.520427
std	182.604079
min	1.000000
25%	1.000000
50%	3.000000
75%	16.000000
max	5817.000000



index	ratings_per_item
count	33388.000000
mean	47.520427
std	182.604079
min	1.000000
25%	1.000000
50%	3.000000
75%	16.000000
max	5817.000000

Figure 3: Statistics of ratings.

3 Related Work

Beer recommendation has been a topic studied by a few articles. Huitzil et al. [2020] proposed a beer recommender called GimmeHop using ontology reasoners and fuzzy logic. Allen and Wetherbee developed a beer recommender called BeerMe based on content-based filtering. Chinchanachokchai et al. [2021] built the system using review data from existing online community to test the hypotheses.

Collaborative filtering methods have been widely applied to recommender system in order to make better rating prediction. Resnick et al. [1994] first introduced user-based collaborative filtering which considers similar user sharing similar preferences. Sarwar et al. [2001] first introduced item-based collaborative filtering which considers item-item similarities when making predictions. Koren et al. [2009] first devised matrix factorization method to collaborative filtering recommender system.

4 Methods

Collaborative filtering is a well-known technique used by many online systems to make personalized recommendations based on the preferences and behavior of users. It has been shown to be effective in many different domains, including recommendation systems. However, collaborative filtering can be susceptible to the sparsity of the ratings data, where there are many missing ratings in the data. This can affect the ability of the system to make accurate recommendations.

Our approach is based on collaborative filtering with SVD. I first construct a ratings matrix, where the rows correspond to users, the columns correspond to beers, and the entries correspond to the ratings provided by the users for the beers. I then use SVD to decompose the ratings matrix into two low-rank matrices, which capture the latent preferences of the users and the latent characteristics of the beers. I also explore the method of SVD++. SVD++ is an extension of the traditional SVD algorithm that incorporates additional information, such as the number of ratings provided by each user and the average rating of each beer.

I use the low-rank matrices obtained from both SVD and SVD++ to make predictions for missing ratings in the ratings matrix. Specifically, I use the low-rank user matrix to compute the predicted ratings for each beer by each user, and use the low-rank beer matrix to compute the predicted ratings for each user for each beer. I then average these predicted ratings to obtain the final predicted rating for each user-beer pair.

I tried to use the scikit-learn library in Python to implement collaborative filtering with SVD and SVD++ in a hands-on manner as described above. However, the implementation did not work out as expected. Due to the time limit, I directly use the Surprise library with a built-in SVD and SVD++ model for collaborative filtering to perform experiments in the next section.

5 Evaluation and Results

I evaluated our approach using a data preprocessed as above. I split the data into training and test sets, with 80% of the data used for training and 20% used for testing. I used the training set to learn the preferences of users and beers, and used the test set to evaluate the performance of our approach in terms of its ability to make accurate recommendations.

I compared our approach to the following three baseline methods.

- Random: a naive baseline, which predicts a random rating based on the distribution of the training set.
- Item-based k-NN collaborative filtering without incorporating SVD.
- User-based k-NN collaborative filtering without incorporating SVD.

The performances on test data of different approaches of collaborative filtering method will be measured by the following different metrics.

- Root Mean Squared Error (RMSE): This metric measures the average difference between the predicted ratings and the actual ratings. A lower RMSE indicates a better model.
- Mean Absolute Error (MAE): This metric measures the average absolute difference between the predicted ratings and the actual ratings. A lower MAE indicates a better model.
- Precision: This metric measures the proportion of predicted ratings that are correct. A higher precision indicates a better model.
- Recall: This metric measures the proportion of actual ratings that are predicted correctly. A higher recall indicates a better model.
- F1 Score: This metric is the harmonic mean of precision and recall. It takes into account both the precision and recall of the model, and a higher F1 score indicates a better model.

The results are shown in the table 2, where Precision, Recall, and F1 Score are calculate for all ratings.

	RMSE	MAE	Precision	Recall	F1 Score
Random	0.9684	0.7618	0.8287	0.6823	0.7484
Item-based k-NN	0.5841	0.4386	0.8818	0.8890	0.8861
User-based k-NN	0.5946	0.4498	0.8833	0.8654	0.8755
SVD	0.5792	0.4342	0.8832	0.8942	0.8880
SVD++	0.5732	0.4293	0.8859	0.8969	0.8900

Table 2: Experiment results.

6 Discussion

First it can be observed that all collaborative filtering algorithms outperform Random, demonstrating the effectiveness of them in making recommendations.

In terms of RMSE, the SVD and SVD++ models have a RMSE of 0.5792 and 0.5732, respectively, while the item-based and user-based models have a RMSE of 0.5841 and 0.5946, respectively. This indicates that SVD-based models are better at predicting the ratings of unseen beers.

In terms of MAE, the SVD and SVD++ models have a MAE of 0.4342 and 0.4293, respectively, while the item-based and user-based models have a MAE of 0.4386 and 0.4498, respectively. This indicates that SVD-based models are better at accurately predicting the ratings of unseen beers.

In terms of Precision, the SVD and SVD++ models have a precision of 0.8832 and 0.8859, respectively, while the item-based and user-based models have a precision of 0.8818 and 0.8833, respectively. This indicates that SVD-based models are better at correctly identifying beers that users will like.

In terms of Recall, the SVD and SVD++ models have a recall of 0.8942 and 0.8969, respectively, while the item-based and user-based models have a recall of 0.8890 and

0.8654, respectively. This indicates that SVD-based models are better at recommending a higher proportion of beers that users will like.

Finally, in terms of F1 score, the SVD and SVD++ models have an F1 score of 0.8880 and 0.8900, respectively, while the item-based and user-based models have an F1 score of 0.8861 and 0.8755, respectively. This indicates that SVD-based models are the best overall approach for making beer recommendations based on user ratings.

Overall, our results show that collaborative filtering with SVD is a better approach for making beer recommendations than item-based and user-based collaborative filtering without SVD. It has a lower RMSE and MAE, higher precision and recall, and a higher F1 score, indicating that it is more accurate and effective at making recommendations.

7 Conclusion

In this project, I used SVD-based collaborative filtering to build a beer recommendation system based on user ratings of different beers. I compared the performance of this approach to two other collaborative filtering methods: item-based and user-based collaborative filtering. Our results showed that SVD-based collaborative filtering outperforms the other two methods in terms of RMSE, MAE, precision, recall, and F1 score. This indicates that SVD-based collaborative filtering is a better approach for making beer recommendations.

8 Other Things I Tried

As I mentioned in Section 4, I tried to use the scikit-learn library in Python to implement collaborative filtering with SVD and SVD++ in a hands-on manner. I first contrusted the user-item matrix, and then I planned to use sklearn.decomposition.TruncatedSVD to decompose the rating matrix into two low-rank matrices. However, the decomposition process returned with "SVD did not coverage error". I searched the possible solutions on removing NAN values in the matrix, allocating more memory, and etc., but none of them works. While I did have lots of other options to do SVD, e.g. with Spicy or Numpy package, I wasted much time solving the bugs and eventually decided to directly use the built-in collaborative filtering method with SVD in Surprise libarary.

9 What I Would Have Done Differently

Initially, I was planning to first build the collaborative filtering recommender system with user ratings, and then take into account item content like beer name, brewery name and etc. to improve the recommendation accuracy. Therefore, as described, the ideal system would be a combination of collaborative filtering and content-based recommendation, which really interested me. However, as the project developed, problems like implementation difficulty, error handling, insufficient time and etc, occurred. As a one person team, the above issues got in my way of achieving the initial goal. While the project comes to an end, I would probably continue the unfinished work with less stress and more flexibility of time.

References

Ignacio Huitzil, Fernando Alegre, and Fernando Bobillo. Gimmehop: A recommender system for mobile devices using ontology reasoners and fuzzy logic. *Fuzzy Sets and Systems*, 401:55–77, 2020.

Alessandro Allen and Ryan Wetherbee. Beerme: A beer recommendation system.

Sydney Chinchanachokchai, Pipat Thontirawong, and Punjaporn Chinchanachokchai. A tale of two recommender systems: The moderating role of consumer expertise on artificial intelligence based product recommendations. *Journal of Retailing and Consumer Services*, 61:102528, 2021.

Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186, 1994.

Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295, 2001.

Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.