Collaborative Filtering Restaurant Recommendation Engine with Skyline Queries

1st Vi Nguyen
Department of Computational and Data Sciences
George Mason University
Fairfax, Virginia, USA
vnguye65@gmu.edu

2nd Ron Mahabir
Department of Computational and Data Sciences
George Mason University
Fairfax, Virginia, USA
rmahabir@gmu.edu

3rd Olga Gkountouna
Department of Computational and Data Sciences
George Mason University
Fairfax, Virginia, USA
ogkounto@gmu.edu

***Abstract*—Recommendation systems are some of the most heavily studied topics in the field of data science. Given today's competitive marketplace, the search for more contextual and personalized user experiences has propelled to the forefront in data science research in recent years. Prior studies have extensively improved filtering algorithms used in these recommendation engines to more accurately recommend the most relevant items. This paper applies collaborative filtering and non-negative matrix factorization in a case of a restaurant recommendation engine to make predictions based on similar users' known preferences. We introduce the concept of skyline query processing as an objective ranking function in a 2-dimensional space, distance and rating. Running a skyline query on a large number of recommendations returns only those restaurants representing a good combination of both criteria, highest rating and shortest distance. Overall, we present a general framework for building a restaurant recommendation engine that can be served as a road-map for future research.**

***Index Terms*—Recommendation engine, Non-negative Matrix Factorization, Skyline Queries, Restaurant Recommendation System, Pareto efficiency**

I. Introduction

With the rapid transition to digital interaction, consumers are presented with huge volumes of information every day. While offering a wider range of options has been shown to have a positive impact on drawing in new customers, over-abundance of information can lead to confusion and inability to make quality decisions. Consumers only value opportunities that are perceived as significantly different from one another. Being presented with heavy loads of items that share similar characteristics might deter them from committing to a decision due to fears of making the wrong choice [12]. This condition is known as information overload which occurs when consumers are overwhelmed with large amounts of alternatives that exceed their cognitive capacity and in turn create decision dilemmas [4]. Furthermore, Bettman, Luce, and Payne argue that consumer preferences are not definitive but rather constructive and malleable; as in the process of prioritization, consumers adjust their likings to justify trade-offs between similar alternatives. In other words, preferences fluctuate as the user explores different options [1]. There have also been studies to further validate that information overload can have negative impacts on consumer choice and uncertainty [6].

Recommendation engines help solve the information overload problem by effectively searching through the option space and assigning ranks to each item by its relevance to the user. A shorter list of items can help consumers make better and more quality decisions, given that the list is personalized to the user's needs and preferences. The demand for improved customer experience has inspired the data science community to develop algorithms to better solve this problem with machine learning. The algorithm works by dynamically searching through all items and assigning ranks to each recommendation as a function of the user's input preferences. This paper provides a general framework for a typical restaurant recommender engine. The applications of such recommender engine are wide-ranging as it is useful to both consumers and businesses. From a consumer perspective, the recommender provides a personalized list of restaurants that best fit the user's preferences which helps narrow down the number of options to only those most relevant. Looking from a business angle, the recommender provides a platform to bridge the gap between consumers and businesses. Restaurants owners can use this application as a way to increase their visibility, reaching a wider audience and advertising their own restaurant features against competitors. The paper is organized into sections as follows: Section II outlines prior research in collaborative filtering, Section III defines the methodology to build the recommendation model; Section IV evaluates the results of such model, and Section V discusses the short-comings and how the model can be improved in future research.

II. Background

Prior research has extensively explored diverse approaches to collaborative filtering in recommendation engines. One of the most popular implementations of such filtering technique is in a movie recommender. The ultimate goal of this recommender is to answer the question: Given user X's rating history, can we predict his/her future ratings and derive a list of recommendations based on those predictions? Netflix has been known to place their movie recommendation algorithms at the core of their products. Their aim is to provide members with great suggestions at the top of page to reduce the time they spend looking for content to watch [9]. In 2009, Netflix held an open competition awarding $1 million to the best collaborate filtering recommendation engine. Since then, there has been a great growing interest in the field, leading to improvements in existing algorithmic approaches and advancements in new methodologies to tackle the problem. Bei Cui reintroduces the application of a KNN algorithm in collaborative filtering that relies heavily on the concept of user similarity. New user registration information is matched against old users using KNN to predict ratings [3]. Alternatively, Katarya R. and Verma O. P. propose the use of K-Means Clustering to group users based on similarity and Cuckoo search optimization algorithm to find the best predictions [7]. Postmus S. provides a comprehensive comparison of three different methods of collaborative filtering techniques: item-based, user-based, and singular value decomposition and concludes that the singular value decomposition model provides the best results [10]. These studies have added to the established knowledge about the structure of recommendation engines and provided new horizons for further exploration into collaborative filtering.

III. Methodology

This section describes the methodology used to build a typical recommendation engine. Figure 1 depicts a flowchart of each step in the process. We first use collaborative filtering to select items based on interactions collected from similar users. Next, we use Non-negative Matrix Factorization (NMF) to decompose the user-to-restaurant relationship matrix to two matrices that express hidden associations between users/restaurants and latent features. Finally, skyline queries are applied to rank the recommendations based on their relevance to the user.

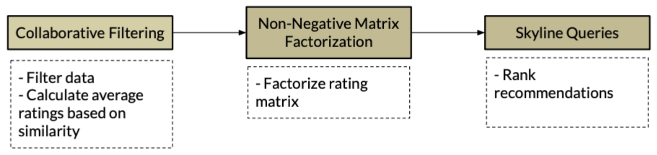
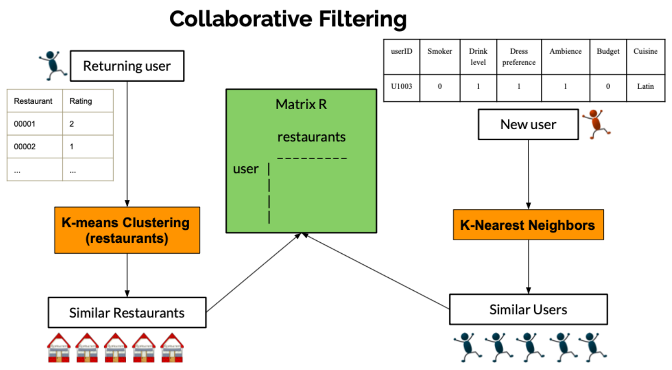

Collaborative Filtering
- Filter data
- Calculate average ratings based on similarity
Non-Negative Matrix Factorization
- Factorize rating matrix
Skyline Queries
- Rank recommendations

Fig. 1. Flowchart

A. Filtering Methods

Collaborative filtering is a popular algorithm used in most recommendation engines. The fundamental concept of this algorithm lies upon the assumption that users with similar preferences share the same opinion about an item. In other words, the user is defined by the group in which their attributes and rating history are most aligned with. As a result, collaborative filtering relies heavily on user similarity to make predictions.



Restaurant	Rating
00001	2
00002	1
...	...

userID	Smoker	Drink level	Dress preference	Ambience	Budget	Cuisine
U1003	0	1	1	1	0	Latin

Fig. 2. Two methods of collaborative filtering for returning users and new users with no rating history

In this paper, we explore two different methods of collaborative filtering for grouping items by similarity (Figure 2). The user-base is split into two buckets, new users with no rating history and returning users with actual ratings. In the case of a new user, a K-Nearest Neighbors algorithm (KNN) is used to detect similarity across users in the database. This is also known as user-to-user collaborative filtering where the new user is cross-referenced against every user in the database to find a group of users that share the most similar characteristics. We first filter the *Users* data to those having similar favorite types of cuisine. KNN is then applied on this smaller set of data to ensure that the algorithm is only selecting from groups of users that have the similar tastes. Table I shows the five select attributes that are used in the KNN algorithm to assign users to their appropriate group. A minimum of five users with ratings is required to ensure that we have sufficient input to inform the model. New user's ratings are calculated by taking the average ratings of the restaurants that k-nearest-neighbors have rated.

TABLE I
User attributes

Attribute	Type	Values	Code
Smoker	Binary	Yes/No	1/0
Drinker	Binary	Yes/No	1/0
Dress Preference	Binary	Professional/Casual	1/0
Enjoys Company	Binary	Yes/No	1/0
Budget	Ordinal	Low/Medium/High	0/0.5/1

In the case of a returning user, K-Means Clustering algorithm is applied to the restaurants. This is often referred to as item-to-item collaborative filtering. The goal is to find restaurants with similar features to the ones already rated highly by the user. Similarly, ratings on new restaurants are calculated by taking the average ratings of similar restaurants that the user has visited.

B. Non-Negative Matrix Factorization

Data on user-to-restaurant relationship approximated by Collaborative Filtering is organized into a matrix R in which each cell represents the degree to which a user prefers a restaurant on a scale from 0 to 2. Matrix R has a sparsity of 6% consisting of mostly null values at positions where there has been no recorded user-to-restaurant relationship. Non-negative matrix Factorization (NMF) attempts to fill in those missing values based on hidden underlying factors present in the nature of the existing ratings. Null values are replaced with 0's to fulfill the requirements of NMF function from the Scikit-learn library. To account for this change, 1 was added to the real-valued ratings, shifting the scale one unit up. Table II shows the coding dictionary and Table III shows the shift.

TABLE II
RATING DICTIONARY

Null	0	1	2
No Rating	Bad	OK	Good

TABLE III
SHIFT

0	1	2	3
No Rating	Bad	OK	Good

The NMF method from the Scikit-learn library performs iterative computations to find matrices, W and H, such that their dot product produces a matrix as close to matrix R as possible. Matrix W is a user-by-k matrix where k is the number of latent features and matrix H is restaurant-by-k. The computation is driven by a minimization of a cost function. The cost is calculated by comparing the product of W and H to the matrix R. The final W and H are those matrices that produce a close approximation of matrix R with the lowest cost. The cost function is exclusive of the 0 inputs in matrix R which represent the rating predictions that need filling in when H and W are multiplied back together. The product of W and H contains the complete user-to-restaurant relationship. Each row in the matrix represents the predicted ratings of each user for each restaurant. These predictions can then be sorted to retrieve the items with the highest ratings. Figure 3 depicts this process to fill in the missing values in matrix R.

C. Skyline Queries

We use skyline queries as an objective ranking function to order the recommendations based on their relevance. There are two important criteria that consumers often consider when choosing a restaurant, distance and rating. Skyline queries help in the evaluation of both criteria to determine where each recommendation should be placed relative to its alternatives. The goal is to answer the question: what are the best nearby restaurants? In applying skyline queries, we aim at highlighting restaurants that exhibit a good combination of both criteria the user is interested in. The recommended items at the top of the list are those that represent maximization of the predicted

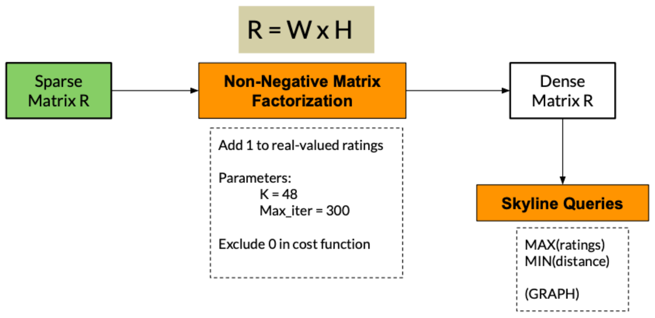


Fig. 3. NMF and Skyline Queries

ratings and minimization of the distances. Figure 4 shows the restaurants that are in the optimum set for a particular user after running skyline queries on all recommendations.

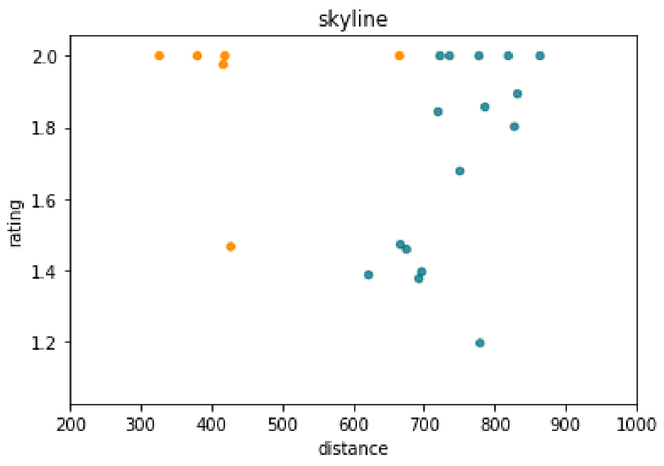


Fig. 4. Skyline detected from recommendations

IV. Results

In this section, we evaluate the performance of the methodology described in Section III applied on a restaurant recommendation engine. Figure 5 shows the dataset, consisting of 1161 ratings from 138 users on 130 restaurants in Mexico. The ratings are integers valued from 0 to 2 (See Figure II for coding dictionary).

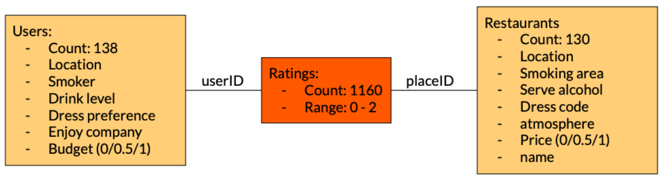


Fig. 5. Data Schema

For a randomly selected returning user U1003 with rating history shown in Table IV , an NMF model with 48 latent features returns the top 10 recommended restaurants that share the most similarity to those already rated highly by this user (Figure V). The decision of setting 48 as the number of latent features in NMF is informed by a grid search to minimize the fit residuals between actual matrix R and W x H approximation. The mean squared error of the recommendations for this user is 0.8.

TABLE IV
USER U1003 RATING HISTORY

userID	placeID	rating
U1003	135041	0
U1003	132755	2
U1003	132723	2
U1003	132825	2
U1003	135075	2
U1003	135079	2
U1003	132862	1
U1003	132937	2

TABLE V
TOP 10 RECOMMENDATIONS FOR RETURNING USER U1003

placeID	name	distance	rating
132858	Hamburguesas Valle Dorado	93.776000	2.0
132877	sirloin stockade	122.979148	2.0
132854	Sirlone	125.945912	2.0
135069	Abondance Restaurante Bar	129.678569	2.0
132851	KFC	132.162188	2.0
135108	Potzocalli	132.695695	2.0
132846	el lechon potosino	144.997232	2.0
132870	Tortas y hamburguesas el gordo	163.645134	2.0
132847	don burguers	166.395474	2.0
132866	Chaires	166.657773	2.0

For a new user U1012 with registration information given in Table VI, the described model returns the 10 recommendations ordered using Skyline processing which can be found in Figure VII. The MSE for this particular random new user is 0.25.

TABLE VI
NEW USER U1012 ATTRIBUTES (SEE TABLE II FOR COMPLETE DICTIONARY

Attribute	Value
userID	U1012
Location	18.81, -99.24
Smoker	0
Drink level	0.5
Dress preference	1
Enjoys Company	1
Budget	0.5
Cuisine	Latin American

We use 4 metrics to measure the accuracy: Mean Average Recall at k (MAR@K), Precision, Personalization, and Mean Squared Error (MSE) (See Table VIII). The model achieves a 0.228 in MAR@K at k = 20 where k is the number of recommendations for returning users with rating history. This indicates that the restaurants rated highly by the user are in the first 20 recommendations only 22.8% of the time. For new users with no rating history, the model recommends good restaurants 21.83% of the time. To test whether the model is able to personalize recommendations based on user preferences, we check for similarities across all recommendations. The system recommends 57.09% different restaurants to returning users and 66.84% to new users.

TABLE VII
TOP 10 RECOMMENDATIONS FOR NEW USER U1012

placeID	name	distance	rating
132854	Sirlone	125.945912	2.000000
132858	Hamburguesas Valle Dorado	93.776000	1.331826
132877	sirloin stockade	122.979148	1.508919
135069	Abondance Restaurante Bar	129.678569	1.303193
132851	KFC	132.162188	1.919605
135108	Potzocalli	132.695695	2.000000
135057	El Herradero Restaurante and Bar	175.215598	2.000000
132846	el lechon potosino	144.997232	1.664171
132866	Chaires	166.657773	1.735800
135033	Restaurant El Muladar de Calzada	217.826616	2.000000

TABLE VIII
EVALUATION METRICS

Metric	Returning users	New users
MAR@K	0.2288	0.2183
Precision	0.0177	0.0278
Personalization	0.5709	0.6684
MSE	0.5956	0.9422

V. CONCLUSION AND FUTURE WORK

The proposed restaurant recommendation system provides a general framework for the purposes of building recommendation engines using collaborative filtering. Improvements are still needed to fine-tune the performance of the model and thus provide more relevant and accurate recommendations. This goes to prove that the field of recommendation engines still warrant more attention and research despite endless efforts to enhance existing models and develop new methodologies. For future research, parameters used in the model can be further fine-tuned to optimize the outputs of each step in the process. The number of latent features in NMF affects the final matrices, W and H, which in turn determine the final predictions. Changing this key parameter to a more optimal value can increase the accuracy when W and H are multiplied together in the reconstruction of matrix R. Other important parameters are: the number of iterations and cost tolerance. Increasing the number of iterations in the calculation of W and H can have major impacts on the accuracy. The methodology is applied on a small sized dataset consisting of only 130 items and 138 users which might have limited the model's learning capabilities. More data is needed to help better inform the model of user preferences and rating history. As future work, we plan to improve the personalization score by extending our analysis to new datasets that include more restaurants with diverse sets of attributes.

ACKNOWLEDGMENT

Thank you Dr. Ron Mahabir and Dr. Olga Gkountouna (Computational and Data Sciences, George Mason University) for providing much needed guidance and assistance to this project,

References

[1] Bettman, J. R., Luce, M. F., Payne, J. W. (1998). Constructive Consumer Choice Processes. Journal of Consumer Research, 25, 187-217.

[2] Bonhard P., Sasse M. A. (2006). "Knowing me, knowing you" – Using Profiles and Social Networkin to Improve Recommender Systems. BT Technology Journal, 24, 3, 84-98.

[3] Cui, B. B. (2017). Design and Implementation of Movie Recommendation System Based on Knn Collaborative Filtering Algorithm. ITM Web of Conferences. 12, 8.

[4] Goswami S. (2015). Analyzing Effects of Information Overload on Decision Quality in an Online Environment. Journal of Management Research, 15, 231-245.

[5] Hose K. (2016, July 6). Skyline Queries. Datenbank Spektrum, 16, 247-251.

[6] Jacoby J., Jaccard J. J., Currim I., Kuss A., Ansari A., Troutman T. (1994). Tracing the Impact of Item-by-Item Information Assessing on Uncertainty Reductio. Journal of Consumer Research, 21, 291-303.

[7] Katarya R., Verma O. P. (2017). An effective collaborative movie recommender system with cuckoo search. Egyptian Informatics Journal. 18, 2, 105-112.

[8] Longo C. (2018, November 22). Evaluation Metrics for Recommendation Engines. Towards Data Science. Retrieved from https://towardsdatascience.com/evaluation-metrics-for-recommender-systems-df56c6611093.

[9] Netflix Research. Recommendations – Figuring out how to bring unique joy to each member. Retrieved from https://research.netflix.com/research-area/recommendations.

[10] Postmus S. (2018). Recommender System Techniques Applied to Netflix Movies Sata. Research Paper Business Analytics.

[11] Raval N., Khedkar V. (2019). A Review Paper on Collaborative Filtering Based Movie Recommendation System. International Journal of Scientific Technology Research, 8, 12, 2507-2512.

[12] Salokar, Bill. (2018, January 20). Consumers want choice – just not too many. The Insights Associations. Retrieved from https://www.insightsassociation.org/article/consumers-want-choices-just-not-too-many.

[13] Wu M. C., Garg D., Bhandary U. (2018). Movie Recommendation System Using Collaborative Filtering. 2018 IEEE 9th International Conference on Software Engineering and Service Science, 11-15, doi: 10.1109/ICSESS.2018.8663822.