RESTAURANT RECOMMENDATION SYSTEM USING MACHINE LEARNING

Mr. N. Varatharajan[1], J. Guruprasad[2], K. Mathumitha[2]

[1]*Assistant professor, Department of Information Technology, Info Institute of Engineering, Coimbatore-641107, India.*

[2]*Student, Department of Information Technology, Info Institute of Engineering, Coimbatore-641107, India.*

Abstract: *Recommendation system is widely used for deploying the prediction based on the preferences of users to items. In many places, to find a restaurant in rush hour is a challenging activity to do. It is going to be easy, when someone accesses an application and it recommend the best restaurant then he/she can visit to. In this paper, according to the characteristics of restaurant recommender systems, the restaurant recommendation is done based on improved collaborative and content-based filtering method is proposed to be analyzing the end user's behaviors. The ICCFM implicitly or explicitly considers the influences of individual or similar user preferences and the relationship. Then this experiment is executed on the yelp data set that our application crawls. This recommender system propose a machine learning algorithms to resolve the issues of personalized restaurant selection relying upon yelp data.*

Keywords: *Recommender system; machine learning collaborative filtering; content-based filtering; user inputs and behaviors; user feature.*

1. INTRODUCTION:

Recommendation systems have recently been used in many fields. A recommendation system is a tool that would recommend products to customers based on search history, based on similarities of users with similar patterns, based on ratings and likes that given by the user. Some Realtime recommendation applications are Netflix, YouTube, amazon, Facebook, yelp. Food is the only thing which represent our culture, tradition and values. Values are the places that can be significantly related to varieties of food available there. But, one of the main problems is most of the people are unaware of famous foods and restaurant nearby them. This not only applicable to foreigners or tourists but also suits for the local people. With increaser in restaurant numbers customers often get confused about the best suited restaurant according to their preferences. So for that they need to face hard time to search for the best suitable place and food to eat, especially when they are new to that place. This project is a web based restaurant recommendation system. The primary aim of the application is to suggest users the best food to eat on the given location based on their food preferences.

The primary aim of the application is to suggest users the best food to have on the given food preferences. The application is targeting everyone who wishes to eat food in restaurant. Restaurant recommendation system is works based on collaborative filtering. This application takes the food preferences and ratings into consideration to recommend food to the users. In this collaborative filtering method uses item based and user-based filtering methodologies.

There is a huge impact in social media of having a recommendation system for restaurants. Further it is also important to note that 40% of world population has an internet facility today compared to 1% in 1995. Although there are lots of context-aware restaurant recommenders most of them has focus on location information. Some system uses only content-based filtering whereas some use only collaborative filtering. In this project, it persists the good recommendation because it considers user behaviors and his current location. So we are using both content based and collaborative filtering for make the recommendation more efficient to recommend the restaurant by sorting their behavior.

2. METHODOLOGY:

The proposed Recommendation system has three phases that is Information collection phase, learning/ analyzing phases, prediction or recommendation phases. Information collection phase is the collection information like user details, their preferences, browse histories, ratings and likes. We need to maintain the customer table to store the information. In learning phases, we need to train the recommendation system using machine learning algorithms. It should follow the user behaviors and influences to obtain the preference of the user.

Later in this phase it find out the solution and passes to the recommendation phases. In recommendation phase, recommender system analyze the user inputs and behaviors with the dataset that available in yelp to fetch the suitable restaurant based on user preferences. In this model we are going to use content-based filtering and collaborative filtering.

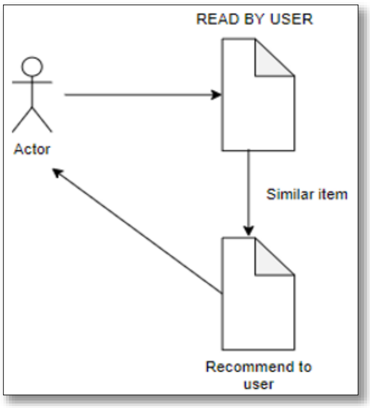
Dataset:

In this recommendation system, the main resources are the yelp dataset that we are going to use in this application for recommend the restaurant. It contains 10,000 No of restaurants with restaurant location, menu item and ratings.

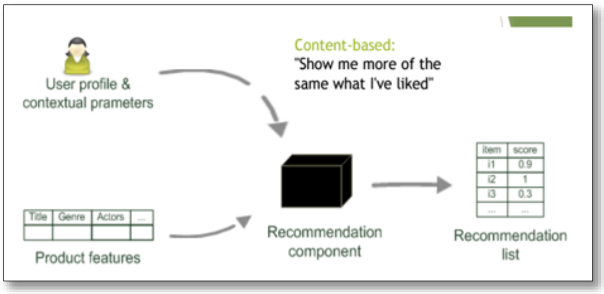
Here we are going to use waterfall approach to develop the application.

Features of Content based filtering:

Content based filtering needs some of basic requirements about the available items and sort of user profile describing what user likes. “Similarity” is computed from item attribute like similarity of food, cuisine, restaurant. This CF algorithm learns user preferences and locate/recommend items that are “similar” to the user preferences.

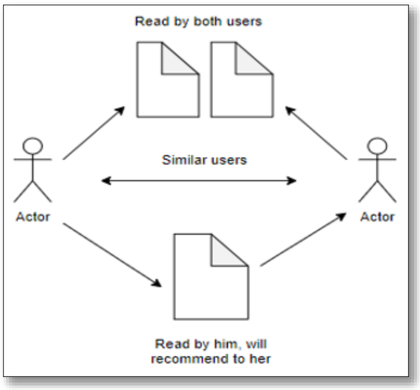


This results the individual users behavior to recommend restaurant.



Features of Collaborative filtering:

Collaborative filtering is a most recently using algorithm for recommendation system. This method uses inputs from the multiple users with similar taste. It has user-based CF and item-based CF. This methodology exploits the users underlying preferences through the analysis of latent features that define the input values.



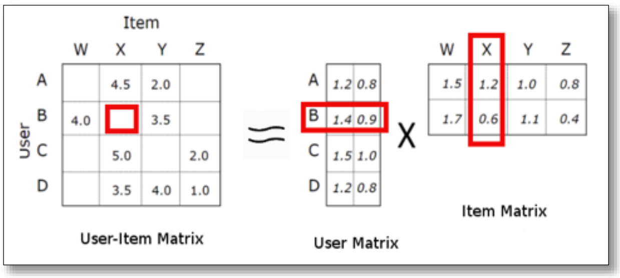
So, the collaborative filtering is mostly used for recommendation system. One of the key techniques that is used in collaborative filtering is matrix factorization.

Algorithm:

Step 1: Calculate the similarity matrix $S(m*m)$ using rating matrix R. Pearson Correlation is adopted as the similarity measure. Each element (u_1, u_2) in S denotes the Pearson Correlation between user u_1 and user u_2.
Step 2: Find *USER_K* nearest neighbors for user u. The larger the correlation, the more similar the two users are.
Step 3: Predict the rating values for items unrated yet by user u. The predictive rating values are obtained by the weighted sum of the items from u*'s USER_K* nearest neighbors.
Step 4: Recommend top *ITEM_K* items to user u by the predictive values.

Matrix Factorization:

Matrix factorization is a decomposition method that reduce a matrix into constituent parts that make it easier to calculate more complex matrix operations. Some of the most successful realizations of latent factor models are based on matrix factorization. In its basic form, matrix factorization characterizes both items and users by vectors of factors inferred from item rating patterns.



Split the matrix M x N to (M x D & D x N) and factorize the user and item matrix. Then multiply the two matrix to form the original matrix.

This matrix factorization is used for predict the users unrated item. It compares the rating with similar users with same taste and fulfills the unrated places. So that it can easily predicts the restaurant based on the factorized rating of the particular user. The idea behind matrix factorization is to represent users and items in a lower dimensional latent space. Since the initial work by Funk in 2006 a multitude of matrix factorization approaches have been proposed for recommender systems.

Funk MF:

The original algorithm proposed by Simon Funk in his blog post factorized the user-item rating matrix as the product of two lower dimensional matrices, the first one has a row for each user, while the second has a column for each item. The row or column associated to a specific user or item is referred to as *latent factors*. Note that, in Funk MF no singular value decomposition is applied, it is a SVD-like machine learning model.

Specifically, the predicted rating user u will give to item i is computed as:

$$\tilde{r}_{ui} = \sum_{f=0}^{nfactors} H_{u,f} W_{f,i}$$

It is possible to tune the expressive power of the model by changing the number of latent factors. It has been demonstrated that a matrix factorization with one latent factor is equivalent to a *most popular* or *top popular* recommender (e.g. recommends the items with the most interactions without any personalization). Increasing the number of latent factor will improve personalization, therefore recommendation quality, until the number of factors becomes too high, at which point the model starts to overfit and the recommendation quality will decrease. A common strategy to avoid overfitting is to add regularization terms to the objective function. Funk MF was developed as a *rating prediction* problem, therefore it uses explicit numerical ratings as user-item interactions.

SVD++:

While Funk MF is able to provide very good recommendation quality, its ability to use only explicit numerical ratings as user-items interactions constitutes a limitation. Modern day recommender systems should exploit all available interactions both explicit (e.g. numerical ratings) and implicit (e.g. likes, purchases, skipped, bookmarked). To this end SVD++ was designed to take into account implicit interactions as well. Compared to Funk MF, SVD++ takes also into account user and item bias.

The predicted rating user *u* will give to item *i* is computed as:

$$\tilde{r}_{ui} = \mu + b_i + b_u + \sum_{f=0}^{nfactors} H_{u,f} W_{f,i}$$

SVD++ has however some disadvantages, with the main drawback being that this method is not *model-based.* This means that if a new user is added, the algorithm is incapable of modeling it unless the whole model is retrained. Even though the system might have gathered some interactions for that new user, its latent factors are not available and therefore no recommendations can be computed. This is an example of a cold-start problem that is the recommender cannot deal efficiently with new users or items and specific strategies should be put in place to handle this disadvantage.

A possible way to address this cold start problem is to modify SVD++ in order for it to become a *model-based* algorithm, therefore allowing to easily manage new items and new users.

As previously mentioned in SVD++ we don't have the latent factors of new users, therefore it is necessary to represent them in a different way. The user's latent factors represent the preference of that user for the corresponding item's latent factors, therefore user's latent factors can be estimated via the past user interactions. If the system is able to gather some interactions for the new user it is possible to estimate its latent factors. Note that this does not entirely solve the cold-start problem, since the recommender still requires some reliable interactions for new users, but at least there is no need to recompute the whole model every time. It has been demonstrated that this formulation is almost equivalent to a SLIM model, which is an item-item model based recommender.

With this formulation, the equivalent item-item recommender would be. Therefore the similarity matrix is symmetric.

$$\tilde{r}_{ui} = \mu + b_i + b_u + \sum_{f=0}^{nfactors} \left(\sum_{j=0}^{nitems} r_{uj} W_{j,f} \right) W_{f,i}$$

Asymmetric SVD

Asymmetric SVD aims at combining the advantages of SVD++ while being a model based algorithm, therefore being able to consider new users with a few ratings without needing to retrain the whole model. As opposed to the model-based SVD here the user latent factor matrix H is replaced by Q, which learns the user's preferences as function of their ratings.

Hybrid:

In recent years many other matrix factorization models have been developed to exploit the ever-increasing amount and variety of available interaction data and use cases. Hybrid matrix factorization algorithms are capable of merging explicit and implicit interactions or both content and collaborative data.

3. LITERARY SURVEY:

Local business review websites such as Yelp and Urban spoon are a very popular destination for a large number of people for deciding on their eat-outs. Being able to recommend local businesses to users is a functionality that would be a very valuable addition to these site's functionality. In this paper I aiming to build a recommendation system for recommend the restaurant for the users with their preferences and nearby location. In this application, we will primarily explore the optimization of machine learning algorithms to predict desired restaurant and to develop features that would help to improve the accuracy of this application.

There are many recommendations existing today, but none of them has used improved collaborative and content based filtering for recommend the restaurant. So, I am going to implement these both the algorithms for recommending the best suited restaurant for the users based on their preferences.

4. RESULTS AND DISCUSSION:

By using this machine learning algorithm, web application will be created to recommend the restaurant based on user preferences. Initially it asks for user login. After that user need to search or find restaurant in the search component. It shows the restaurant with high rating and good menu items in the restaurant. It also helps the user to find the restaurant with nearby location. In this case we achieve our recommendation successfully and provide the suitable recommendation for the user.

5. CONCLUSION:

The main objective of the study is to develop the restaurant recommendation system using machine learning. This is used for the users to predict the suitable restaurant. In this the machine learning promises the recommendation for the user. The content based filtering and collaborative based filtering makes the recommendation more efficient so that the each user can use this application for their easy prediction of restaurant.

Most the case user need the restaurant with their nearby location. We also solving that issue by adding the restaurant location in our dataset. So that our machine learning algorithm easily predicts the restaurant for the customer with their present location.

This machine learning algorithm is done with help of python programming language.

REFERENCES:

1. Anant Gupta; Kuldeep Singh 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI) “Location based personalized restaurant recommendation system for mobile environments”
2. Md. Ahsan Habib; Md. Abdur Rakib; Mohammad Abul Hasan 2016 19th International Conference on Computer and Information Technology (ICCIT). “Location, time, and preference aware restaurant recommendation method”
3. Jun Zeng; Feng Li; Haiyang Liu; Junhao Wen; Sachio Hirokawa 2016 5th IIAI International conference on Advanced Applied Informatics (IIAI-AAI) “A Restaurant Recommender System Based on User Preference and Location in Mobile Environment”
4. Rahul Katarya; Om Prakash Verma 2015 International Conference on Green Computing and Internet of Things (ICGCIoT) “Restaurant recommender system based on psychographic and demographic factors in mobile environment”
5. Anu Taneja; Prashant Gupta; Aayush Garg; Akhil Bansal; Kawal Preet Grewal; Anuja Arora 2016 Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC) “Social graph-based location recommendation using users' behavior: By locating the best route and dining in best restaurant”
6. B. Lee, H. Kim, J. Jung, G. Jo (2006) 430 – 438.“Location-Based Service Context Data for a Restaurant Recommendation”
7. B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, in Proceedings of the 10th international conference on World Wide Web. ACM, 2001, pp. 285– “Item-based collaborative filtering recommendation algorithms,”
8. J. Bao, Y. Zheng, and M. F. Mokbel, in Proceedings of the 20th International Conference on Advances in Geographic Information Systems. ACM, 2012, pp. 199–208.“Locationbased and preference-aware recommendation using sparse geosocial networking data,”
9. Zhichao Chang, Mohammad Shamsul Arefin, Yasuhiko Morimoto 2013 Second IIAI International Conference on Advanced Applied Informatics “Hotel Recommendation Based On Surrounding Environments”
10. Yelp, “www.yelp.com/academicdataset,” Yelp academic dataset, 2016.
11. A. Gandhe, “Restaurant recommendation system,” cs229.stanford.edu, 2015.
12. “A preference-based restaurant recommendation system
13. A. Roy, S. Banerjee, M. Sarkar, A. Darwish, and M. Elhoseny, et al. "Exploring new vista of intelligent collaborative filtering: a restaurant recommendation paradigm", Joumal of Computational Science, 2018 " in press" .
14. W. Cui, P. Wang, X. Chen, D Yi, , and D. Guo, et al. "How to use the social media data in assisting restaurant recommendation", Intem ational Conference on Database Systems for Advanced Applications. Springer, Cham, 2016, pp. 134-141.