Product Recommendation System using SVD and Machine Learning techniques

Joanne Lim Jo En
School of Computing
Asia Pacific University of Technology and Innovation (APU)
Kuala Lumpur, Malaysia
tp065117@mail.apu.edu.my

Chong Kae Li
School of Computing
Asia Pacific University of Technology and Innovation (APU)
Kuala Lumpur, Malaysia
tp065335@mail.apu.edu.my

Davina Lim Zhe Yen
School of Computing
Asia Pacific University of Technology and Innovation (APU)
Kuala Lumpur, Malaysia
tp065310@mail.apu.edu.my

Chan Vern Jian
School of Computing
Asia Pacific University of Technology and Innovation (APU)
Kuala Lumpur, Malaysia
tp069357@mail.apu.edu.my

Zailan Arabee Bin Abdul Salam
School of Computing Computer Science & Software Engineering
Asia Pacific University of Technology and Innovation (APU)
Kuala Lumpur, Malaysia
zailan@apu.edu.my

Abstract*—Recommendation system is commonly used in many areas. Machine learning is the new trend in the field of analysis data. In this study, recommendation system is also applied to machine learning, the AI method implemented in this recommendation system are Collaborative Filtering and Content-based Filtering. The demonstration of the result is present in this study too. The articles reference a few papers, Crop Plantation Recommendation using Feature, Product Based Recommendation System Using Machine Learning Techniques, Product recommendation based on shared customer's behaviour, recommender systems enhancement using deep reinforcement learning.

***Keywords*— E-commerce, Recommendation System, Machine Learning, Similarity-based Collaborative Filtering, Product-based Filtering**

I. Introduction

In this day and age, a colossal amount of information is available at our fingertips. This abundance of information, however, has its drawbacks as it can lead to information overload and an increased difficulty in making decisions due to the sheer complexity and number of options a person can choose from. To help alleviate this issue, the recommendation or recommender system was born. However, the recommendation system itself faces a whole new host of problems. One of the issues faced by this system is the cold start problem. The cold start problem occurs when there is not enough information available on a user or item to make an accurate recommendation. To resolve this problem various approaches can be applied such as content-based filtering, collaborative filtering, and hybrid filtering (Shuhrat et al., 2021). The purpose of this research is to compare several different approaches used to tackle the cold start problem as well as the performance of the collaborative filtering system when the SVD matrix's latent features, k, is manipulated.

II. Literature Review

A. *Similar Projects*

There are a few other research related to the study topic that we are discussing in this paper. Findings from these research papers can help in giving a better view of Machine Learning techniques to come out as efficient and good recommendation system.

The research done by Rodrigues & Ferreira (2016) was generating a product recommendation based on customer segmentation. They used hybrid approaches by combining both content-based and collaborative recommendations. Two sources of information used are clustering groups of customers with similar interests according to their purchasing patterns and using association rule generation to extract the best products to recommend. A RFM (Recency, Frequency and Monetary) model characterizes a customer based on its behavior and is used as segmenting variables to observe the customer's attitudes on the products. Clustering method groups object based on similarity function and partitional algorithm, k-means is run several times with different random starts and selects segmentation with lowest total Within Sum of Squares (WSS) and categorized the clusters into R, F, M parameters. Association rule mining here analyzes random data and uses it as reference for decision making. High confidence ensures good predictability of the rule, X -> Y. Hybrid recommendation will come out N highest ranked (top sellers) to selected as top-N recommended products. Some measures such as precision, recall and F1 metric are used in the experiment. The hybrid recommendation algorithm used is found useful in improving average sales without decreasing its recommendation accuracy, meaning it yields recommendations to higher quality.

According to Kanagala (2020), we can determine that they generated user recommendations using machine learning and the KNN classification method. They measured the model's effectiveness using RMSE (Root Mean Square Error). Collaborative filtering and content-based filtering are the two main filtering approaches on which these recommendation systems are often built. Filtering that is based on content examines the material. When a consumer looks for one thing from the cluster, they create a relationship or cluster between the collection of items and show the relevant items to the customer. User behavior serves as the basis for collaborative filtering. In order to decompose the original matrix and retrieve the latent factor, they also used

SVD (Single Value Decomposition), which aids in the generation of recommendations.

B. Methodology/ Approach

This part will discuss the overview of previous research and studies that used Machine Learning (ML) algorithm as an approach to solve the cold start problem.

According to Shuhrat et al. (2021) has reviewed different studies on using ML algorithms with deep Reinforcement Learning (RL) approaches for studying recommender systems. Their aim is to find out solutions of the existing ML techniques for non-RL based and RL based recommendations, but the study's focus is more towards RL based approaches. They found out that DRR and DEERS frameworks took short considerations and provide long-term benefits can satisfy the limitation of changing user preferences. Based on their analysis, by applying DRL framework it could achieve for dynamic processed recommender systems, DEERS framework will consider negative user feedbacks to get accurate results and trust-based recommendation using DQN RL approach which system calculates value of expectation on recommended item according to users' trust. In short, the limitations of the ML techniques could be solved using different deep RL approaches and frameworks.

According to Attaluri et al. (2020), we can determine that ML algorithms is used to predict a binary classification output. They used a) Random Forest b) Logistic Regression & c) Artificial Neural Network (ANN) to predict if a crop is profitable to the farmer or not. This study uses data-driven machine learning (ML) models to analyze historical data from 2009 to 2015. The dataset consists of 23 variables and 2049 observations across seven years from 2009 to 2015. The model performance is evaluated using metrics like Accuracy, AUC, Sensitivity, Specificity, and Confusion matrix. The results in this journal show that there is a class imbalance in the data, with profitable crop entries being less compared to non-profitable crops. In conclusion, this result give the reasonable accurate predictions can be made if a crop is profitable or not.

C. Conclusion/ Recommendation

Similarity-based collaborative filtering and product-based filtering have solved many problems in analyzing the user behavior and creating recommendations for the user. To get a better solution for recommend content based on a user's common interests, the value of k parameter in SVD matrix is used to give more accurate recommendation.

In this part, SVD will be used to generate the predictions of k parameter. The discussion and results of this experiment will be done in this paper for reference of the parameter in Collaborative filtering.

III. Materials and Methods

A. Machine Learning (ML) Algorithm

- Similarity-based Collaborative Filtering
- Product-based Filtering
- Singular Value Decomposition (SVD)

Similarity-based collaborative filtering and product-based filtering are the techniques used in ML algorithm to analyze user behavior and generate relevant recommendations. The source code of the algorithms was not created by researchers. The source code used in this paper belongs to Vaibhav67979 in GitHub. Researchers have done to show the effect to the result by modifying the parameters. The parameter in this paper is the k parameter from SVD matrix.

Similarity-based collaborative filtering recommends products based on a user's common interests. For a given user, product-based filtering recommends the top 5 products. The product rating matrix is dimensionally reduced by SVD to get 50 potential features. SVD is used to calculate the anticipated ratings for each user. The anticipated ratings are determined by multiplying the U, sigma, and Vt matrices.

B. Software Requirements

- Python
- Google Colab

The source code is a python-based source code and researchers have run the code by using Google Colab. If user used Google Colab to run the code, can just modify the k parameter in the SVD matrix to see the changes.

C. Purpose

This study was designed to investigate how collaborative filtering and product-based filtering may be used to predict ratings. The goal of this research is to ascertain the factors that influence how these systems have an impact. The current recommendation algorithm heavily relies on observable elements like behavioral outcomes and past purchases. The system then develops a collaborative filtering model using customer reviews and purchase data, which offers product suggestions based on user ratings and past purchase behavior of returning customers. The performance of e-commerce recommendation systems is one area where future study will focus.

D. Parameter

- k parameter from SVD matrix

The parameter that can be modified to see the changes of accuracy and performance for this recommendation system is the k parameter. This parameter is changeable and not fixed, therefore it could be modified depending on how much latent features is set to be used for predictions. According to the original source code by Vaibhav67979 in GitHub, the k parameter is set as "k = 50 latent features" in predicting the rating for the dataset products. In this paper, different values will be assigned to this k parameter to test and generate results to show how it affects predicted results by recommendation system.

IV. Results and Discussions

The AI method will be implemented in this recommendation system to test out the method in this section to find the recommend product based on each method, similarity-based collaborative filtering, and product-based collaborative filtering. The findings of each method will be user similarity and product rating prediction.

A. *Discussion on Implementation*

1) Experimental Setup with Google colab

To set up the experiment, the dataset 'ratings_Electronics.csv' which contains the rating of amazon's electronic products was loaded into google collab. Libraries such as numpy, pandas, and matplotlib.pyplot were imported. Columns were given the names 'user_id', 'prod_id', 'rating', and 'timestamp'. The timestamp column was dropped as it was not relevant to the experiment. Next the dataset was pre-processed by subsetting the data to only include users who had 50 or more ratings to make the dataset easier to work with. An interaction matrix, final_ratings_matrix, between products and users based on their ratings was created.

The user based collaborative filtering Cosine similarity was used to find the similarity between user index and user interaction. Users were appended to their corresponding similarity scores. Then the user was extracted with similarity score. The original user is removed, and the similarity score and only similar users are kept. To make the actual recommendation, a function was used to get similar users, find products the user has interacted with and find products rated by the similar users.

The model based collaborative filtering using singular value decomposition (SVD) was used to deal with the data sparsity. A csr matrix was made of the final_ratings_matrix

2) Implementation Challenges

The implementation challenges faced were minimal. Exploration of the dataset revealed that there was only one null value in the dataset which was in the rating column. The sparsity of the dataset in terms of reviews per user proved to be challenge but was handled by the subsetting mentioned above.

B. *Results*

1) Collaborative Filtering: User Behavoir

The experiment result for this recommendation system for each method will be explained through graphic of the result. First, clustering technique is used for finding the similar user for user 3. The Figure shows the user behavior that is closely aligns with user 3, the ranking is from left to right, 320 user will have the most highest similarity score with user 3. The Figure shows the top 10 of the similar users based on the similarity score. These findings can help the system to understand the user behavior of user 3.

```
[320, 12, 793, 261, 156, 1493, 1250, 567, 753, 1360]
```

Fig. 1 Similar user behavior as user 3.

```
[array([[0.05662371]]),
 array([[0.05549645]]),
 array([[0.05098326]]),
 array([[0.05024185]]),
 array([[0.05003874]]),
 array([[0.04930111]]),
 array([[0.04889354]]),
 array([[0.04672744]]),
 array([[0.04637283]]),
 array([[0.04492668]])]
```

Fig. 2 Similar score of top 10 similar user.

After finding the similar user behavior user, it will show recommendations of their product to the user 3 based on the similarity collaborate filtering. Figure unveils 5 recommended products for user 3 which user 3 has not interacted with it. The similarity score range is -1 until 1, the result of the similarity score from user 320 is only 0.05662371, this can be considered as low. Although, user 3 will not be likely to buy products from this recommendation but will get chances to have user 3 to interact with them.

```
['B002V88HFE', 'B003O1UYHG', 'B000ENRQ3M', 'B0015ZIS8K', 'B0000A0AEM']
```

Fig. 3 Recommend product id.

2) Product based Collaborative Filtering: Singular Value Decomposition

The result of Model based Collaborative Filtering is based on predicting the user prediction of the product which is not interacting with the product and recommend the highest rating from the predicting result. The k parameter is used separately, k=10, k=50 and k=100. To compare the result of RMSE at the end of this section.

When k = 50 (default),

```
Below are the recommended products for user(user_id = 22):

Recommended Products
11078    4.205389
40248    3.436487
40249    3.343402
37043    3.173292
45181    2.700111
```

Fig. 4 Recommended product based on predicted rating score.

The result for recommendation item for user 22 is presented in Fig. 4, the highest rating score is 4.205389 out of 5. This product is more likely toward user 22 preference.

To evaluate the prediction rating, it can compare with the actual mean of the rating and see the difference between. The result of RMSE is stated out as 0.013679389779858, it means it has accurate rating predictions.

prod_id	Avg_actual_ratings	Avg_predicted_ratings
0594451647	0.003247	0.003360
0594481813	0.001948	0.005729
0970407998	0.003247	0.008566
0972683275	0.012338	0.035330
1400501466	0.012987	0.006966

Fig. 5 Comparison of actual ratings and predicted ratings.

```
RMSE SVD Model = 0.013679389779858
```

Fig. 6 Result of RMSE SVD.

When k = 10 (less than default),

```
RMSE SVD Model = 0.005540195270197609
```

Fig. 7 Result of RMSE SVD.

When k = 100 (more than default),

```
RMSE SVD Model = 0.020216530134588375
```

Fig. 8 Result of RMSE SVD.

From observing the evaluation result from different k latent values, which are 10, 50 and 100. The RMSE results show in form of gradually increase when k is increasing. Analysis is made from this evaluation analysis, increasing the k value will not increase the accuracy of the prediction, because it will cause overfitting for the training model. Although the k=10 gets the lowest RMSE but it is not suitable for the dataset, rating_Electronic.csv, due to the size of the data frame. For this problem, it is better to find a reasonable acceptable range of RMSE for recommending product or use multiple evaluation method, such as F1 score, mean absolute error, etc.

V. Conclusion

In conclusion, recommendation systems are an asset to decision making in the modern world. Recommendation systems face a challenge in the cold start problem which is when there is not enough information on new users to properly recommend an item. Through the usage of collaborative filtering, users can slowly build more accurate recommendations for themselves through their interactions with products and similarity with other users. Collaborative filtering can utilize SVD to deal with the sparsity of the data and reduce the dimensionality of its user-interaction matrix. Increasing the value of k can help improve the accuracy of the model but only within a certain range. Therefore, it is recommended to use a value of k that is neither too low or too high.

VI. Acknowledgements

The authors from this journal paper would like to thank user Vaibhav67979 who have shared their source code online at GitHub in which it allows us to implement in our research study. Another grateful credits for all journal papers' authors that have provided their ambitious research papers to enhance our knowledge towards the various approaches that can be taken by a recommendation system. The utmost gratitude to the group mates involved in this journal paper. Without their invaluable contribution, this paper could not have become a reality.

References

Attaluri, S.,S, Batcha, N.,K & Mafas, R. (2020). Crop Plantation Recommendation using Feature Extraction and Machine Learning Techniques. *Journal of Applied Technology and Innovation*, 4(4), 1. pp1 -4. https://jati.sites.apiit.edu.my/files/2020/11/Crop-Plantation-Recommendation-using-Feature-Extraction-and-Machine-Learning-Techniques.pdf [accessed Sept. 13, 2023]

Kanagala, M (2020). Product based recommendation system using Machine Learning techniques. *The California State University*. pp 32. https://scholarworks.calstate.edu/downloads/1n79h8686 [accessed Sept 12, 2023]

P, V. (2023, August 28). Ecommerce-product-recommendation-system. GitHub. https://github.com/Vaibhav67979/Ecommerce-product-recommendation-system [accessed June 30, 2023]

Rodrigues, F & Ferreira, B (2016). Product recommendation based on shared customer behaviour. *Procedia Computer Science*, 100, pp 136-146. https://doi.org/10.1016/j.procs.2016.09.133. [accessed Sept. 13, 2023]

Shuhrat, B, Ramachandran, C.R & Salam, Z.A.A (2021). Recommender systems enhancement using deep reinforcement learning. *Journal of Applied Technology and Innovation*, 5(4), pp 38-43.https://dif7uuh3zqcps.cloudfront.net/wp-content/uploads/sites/11/2021/10/10121349/Volume5_Issue4_Paper9_2021.pdf [accessed Sept. 13, 2023]