Survey on Product Recommendation System Using Hybrid Collaborative Filtering

Miss. Sadiya Arshad Khan[1] Miss. Akanksha Pramod Kole[2] Miss. Misba Sajid Inamdar[3]
Mr. Atish Ashok Kurade[4] Dr. Prof. Deepali Avinash Nikam[5]
[1,2,3,4,5]Department of Computer Science and Engineering
[1,2,3,4,5]Dr. J. J. Magdum College of Engineering, Jaysingpur, India

Abstract— In today's world, people with their trend to shop their daily needs in e-commerce sites and here the product recommendation takes a major role in every e-commerce site to overcome their failures. It is one kind of marketing process by which we can advertise for many products and make the customers feel comfort while purchasing into the sites. Recommender systems or recommendation systems are a subset of information filtering system that used to anticipate the 'evaluation' or 'preference' that user would feed to an item. In recent years E-commerce applications are widely using a Recommender system. Generally the most popular Ecommerce sites are probably music, news, books, research articles, and products. Product recommendation helps to satisfy customers as one of the useful applications of electronic commerce. Recommending the right products to right customers enhances the customer's utility and firm profitability. Different types of customers have different interests, so we should first segment customers into groups and recommend the right product to users. The recommendations are based on processing product information received from consumers as part of their input.

Keywords: Recommendation System, Hybrid Collaborative Filtering, Content Based Filtering

I. Introduction

A. *Introduction*

A recommender system, or a recommendation system is a subclass of information filtering system that seeks to predict the "rating" or "preference" a user would give to an item. Recommender systems are technologies that assist businesses to implement one-to-one marketing strategies. Recommender systems rely on customer purchase history to determine customer preferences and to identify products that customers may purchase. Supporting product recommendation services can strengthen the relationship between the buyer and seller and thus increase profit. Web page recommendations based on the bookmarks of the user virtual neighbors. Content-based filtering is another approach different from collaborative filtering. Conventional content-based filtering provides recommendations by matching customer profiles with content's features. Taxonomy of recommender systems in E-commerce, and elucidated how they can provide personalization to establish customer loyalty. Generally, recommender systems provide several merits, including increasing the probability of cross-selling; establishing customer loyalty and fulfilling customer needs by discovering products in which they may be interested in purchasing. Various recommendation methods have been proposed for recommender systems. Collaborative filtering (CF) has been successfully used in various applications. The CF method utilizes preference ratings given by various customers to determine recommendations to a target customer based on the opinions of other similar customers. A typical CF method employs the K-nearest neighbors approach to derive top-N recommendations.

B. *Existing System*

There has been a lot of work done in this field. For example, one very popular algorithm is Collaborative Filtering. One type of collaborative filtering is user-based collaborative filtering, which starts by finding a set of customers who have purchased and rated similar items with the target of users purchasing history. The algorithm aggregates items from these similar customers, and uses the ratings from other similar users to predict the ratings from this user. Another type of collaborative filtering is item-based collaborative filtering, which was first brought up by Amazon and focuses on finding similar items instead of similar 1 customers. For each of the users purchased and rated items, the algorithm attempts to find similar items. It then aggregates these similar items and recommends them. There are also other algorithms that try to exploit graph structures to predict links or ratings. Random walks algorithms could be used in predicting links in complex graphs in a very efficient manner. And also, if we model the user and product graph as a bipartite graph, then it is also feasible to use a Bipartite Projection algorithm to calculate the relevance between two customers. So the predicted rating is essentially based on the other relevant customers' ratings. In later sections of this paper, we will introduce three models and algorithms which are derived from the prior work mentioned above with application-specific improvements.

C. *Limitations of Existing system*

There are multiple problems faced by present recommender engines.[1][8] But the major problems that are creating a huge failure in real world use of these systems areas given below:

1) Data Sparsity: In practice, many commercial recommender systems are based on large datasets. As a result, the user-item matrix used for collaborative filtering could be extremely large and sparse, which brings about the challenges in the performance of the recommendation.
2) Scalability: As the numbers of users and items grow, traditional CF algorithms will suffer serious scalability problems. For example, with tens of millions of customers O(M) and millions of items O(N), a CF algorithm with the complexity of η is already too large.
3) Cold Start Problem: The cold start problem is a typical problem in recommendation systems. This refers to the situation when a new user or item just enters the system. Three kinds of cold start problems are: new user problem, new item problem and new system problem. In such cases, it is really very difficult to provide the

recommendation as in case of a new user, there is less information about the user that is available and also for a new item, no ratings are usually available.

II. Proposed System

A. *Hybrid Recommendation*

Combining collaborative and content-based recommendations can be more effective. Hybrid approaches can be implemented by making content-based and collaborative-based predictions separately and then combining them. Further, by adding content-based capabilities to a collaborative-based approach and vice versa; or by unifying the approaches into one model.Several studies focused on comparing the performance of the hybrid with the pure collaborative and content-based methods and demonstrate that hybrid methods can provide more accurate recommendations than pure approaches. Such methods can be used to overcome the common problems in recommendation systems such as cold start and the data paucity problem.Netflix is a good example of the use of hybrid recommender systems. The website makes recommendations by comparing the watching and searching habits of similar users (i.e., collaborative filtering) as well as by offering movies that share characteristics with films that a user has rated highly (content-based filtering).

B. *System Architecture*

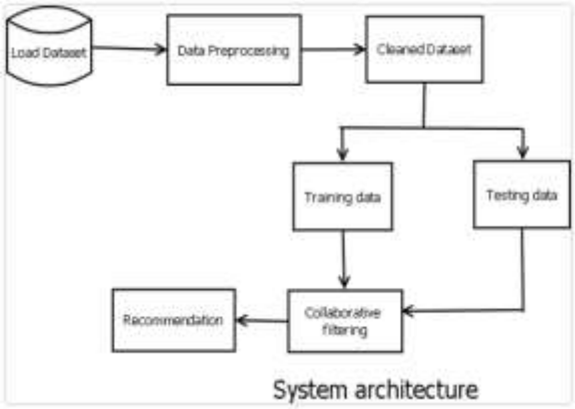

Fig. 1: System architecture for Hybrid RS

1) *To Perform Hybrid Recommendation System:*

Load the dataset. Perform Model-based collaborative filtering to recommend items to users based on previous purchases of a user and similar categories for the same product provided by other users. Make use of a utility matrix which consists of all possible user-item preferences details represented as a matrix. Decomposing the Matrix. Performing Correlation for all items with the item purchased by this customer based on items purchased by other customer's people who bought the same product. Item to item based recommendation system based on product description. Feature extraction from product descriptions. Visualizing product clusters in a subset of data. Recommendation of product based on the current product selected by the user. To recommend related products based on, frequently bought together.

III. Implementation

A. *Tools*

1) *KNN Algorithm for Collaborative Filtering:*

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique. The K-NN algorithm assumes the similarity between the new case/data and available cases and puts the new case into the category that is most similar to the available categories. K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suited category by using K- NN algorithm. The K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems. K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data.It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.[6][3]

$$\text{Euclidean Distance between } A_1 \text{ and } B_2 = \sqrt{(X_2-X_1)^2+(Y_2-Y_1)^2}$$

2) *Item-based collaborative filtering*

It is an algorithm for making recommendations. In the algorithm, the similarities between different items in the dataset are calculated by using one of a number of similarity measures, and then these similarity values are used to predict ratings for user-item pairs not present in the dataset.[2][6]

3) *Data Cleansing Techniques*

1) Remove Irrelevant Values

The first and foremost thing you should do is remove useless pieces of data from your system. Any useless or irrelevant data is the one you don't need. It might not fit the context of your issue. You might only have to measure the average age of your sales staff. Then their email address wouldn't be required. Another example is you might be checking to see how many customers you contacted in a month. In this case, you wouldn't need the data of people you reached in a prior month.However, before you remove a particular piece of data, make sure that it is irrelevant because you might need it to check its correlated values later on (for checking the consistency). And if you can get a second opinion from a more experienced expert before removing data, feel free to do so. You wouldn't want to delete some values and regret the decision later on. But once you're assured that the data is irrelevant, get rid of it.

2) Get Rid of Duplicate Values

Duplicates are similar to useless values – You don't need them. They only increase the amount of data you have and waste your time. You can get rid of them with simple searches. Duplicate values could be present in your system for several reasons. Maybe you combined the data of multiple sources. Or, perhaps the person submitting the data repeated a value mistakenly. Some users clicked twice on 'enter' when they were filling an online form. You should remove the duplicates as soon as you find them.

3) Convert Data Types

Data types should be uniform across your dataset. A string can't be numeric nor can a numeric be a boolean. There are several things you should keep in mind when it comes to converting data types:Keep numeric values as numerics. Check whether a numeric is a string or not. If you entered it as a string, it would be incorrect.If you can't convert a specific data value, you should enter 'NA value' or something of this sort. Make sure you add a warning as well to show that this particular value is wrong.

4) Take Care of Missing Values

There would always be a piece of missing data. You can't avoid it. So you should know how to handle them to keep your data clean and free from errors. A particular column in your dataset may have too many missing values. In that case, it would be wise to get rid of the entire column because it doesn't have enough data to work with.Ignoring missing values can be a significant mistake because they will contaminate your data, and you won't get accurate results. There are multiple ways to deal with missing values.[3]

4) *Content-Based Filtering:*

CBF is based on the assumption that people who liked items with certain attributes in the past, will like the same kind of items in the future as well. It makes use of item features to compare the item with user profiles and provide recommendations. Recommendation quality is limited by the selected features of the recommended items. Same as CF, CBF suffer from the cold-start problem.[6]

5) *Knowledge-Based Filtering:*

KBF uses knowledge about users and items to reason about what items meet the users' requirements, and generate recommendations accordingly. A special type of KBFs are constraint-based RSs which are capable of recommending complex items that are rarely bought (i.e. cars or houses) and manifest important constraints for the user (price) [7]. It is not possible to successfully use CF or CBF in this domain of items as few user-system interaction data are available (people rarely buy houses).

6) *Association rules:*

Association rule mining tries to discover valuable relations (association rules) in large databases of data. These associations are in the form X => Y, where X and Y are sets of items. The association that is above a minimum level of support with an acceptable level of confidence can be used to derive certain conclusions. In recommender systems these conclusions are of the form ”X likes Y” where X is a user to whom the system can recommend item Y[3][2]. In information collected from a discussion group is mined and association rules are used to form the user similarity neighborhood. Word Sense Disambiguation is also used to select the appropriate semantically related concept from posts which are then recommended to the appropriate users of the forum. This hybrid meliorates different problems such as cold-start, data sparsity and scalability. In classification based on association methods is applied to build a RS in the domain of tourism. The system is more resistant to cold-start and sparsity problems. To overcome cold-start, the authors propose a procedure for finding similar items by association rules. Their algorithm considers the user-item matrix as a transaction database where the user Id is the transactional Id. They find the support of each item and keep items with support greater than a threshold. Afterwards, they calculate the confidence of remaining rules and rule scores by which they find the most similar item to any of the items.[1]

B. *Modules*

1) *Data Preprocessing Module*

a) Efficiency Check :

Having clean data (free from wrong and inconsistent values) can help you in performing your analysis a lot faster. You'd save a considerable amount of time by doing this task beforehand. When you clean your data before using it, you'd be able to avoid multiple errors. If you use data containing false values, your results won't be accurate. And the chances are, you would have to redo the entire task again, which can cause a lot of waste of time. If you choose to clean your data before using it, you can generate results faster and avoid redoing the entire task again.

b) Error Margin

When you don't use accurate data for analysis, you will surely make mistakes. Suppose, you've gotten a lot of effort and time into analyzing a specific group of datasets. You are very eager to show the results to your superior, but in the meeting, your superior points out a few mistakes and the situation gets kind of embarrassing and painful. Wouldn't you want to avoid such mistakes from happening? Not only do they cause embarrassment, but they also waste resources. Data cleansing helps you in that regard full stop. It is a widespread practice, and you should learn the methods used to clean data. Using a simple algorithm with clean data is way better than using an advanced one with unclean data.

c) Load Database:

Load database sets the status of the database to “offline.” No one can use the database while its status is “offline.” The “offline” status prevents users from accessing and changing the database during a load sequence. A database loaded by load database remains inaccessible until an online database is issued. Data preprocessing is a data mining technique which is used to transform the raw data in a useful and efficient format.

d) Cleaned Dataset:

Data cleaning is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset. When combining multiple data sources, there are many opportunities for data to be duplicated or mislabeled.

2) *Training and Testing Module*

a) Training Dataset:

The sample of data used to fit the model. The actual dataset that we use to train the model. The model sees and learns from this data. Validation Dataset: The sample of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyperparameters. The evaluation becomes more biased as skill on the validation dataset is incorporated into the model configuration. The validation set is used to evaluate a given model, but this is for frequent evaluation. We, as machine learning engineers, use this data to fine-tune the model hyperparameters. Hence the model occasionally sees this data, but never does it “Learn” from this. We use the validation set results, and update higher level hyperparameters. So the validation set affects a model, but only indirectly. The validation set is also known as the Dev set or the Development set. This makes sense since this dataset helps during the “development” stage of the model.

b) Test Dataset:

The sample of data used to provide an unbiased evaluation of a final model fit on the training dataset. The Test dataset provides the gold standard used to evaluate the model. It is only used once a model is completely trained (using the train and validation sets). The test set is generally what is used to evaluate competing models. Many times the validation set is used as the test set, but it is not good practice. The test set is generally well curated. It contains carefully sampled data that spans the various classes that the model would face, when used in the real world.

3) *Collaborative Filtering Module*

Collaborative filtering (CF) is widely used in recommendation systems. Traditional collaborative filtering (CF) algorithms face two major challenges: data sparsity and scalability. In this study, we propose a hybrid method based on item based CF trying to achieve a more personalized product recommendation for a user while addressing some of these challenges. Case Based Reasoning (CBR) combined with average filling is used to handle the sparsity of data set, while Self-Organizing Map (SOM) optimized with Genetic Algorithm (GA) performs user clustering in large datasets to reduce the scope for item-based CF.[4] Collaborative Filtering is the most common technique used when it comes to building intelligent recommender systems that can learn to give better recommendations as more information about users is collected. To build a system that can automatically recommend items to users based on the preferences of other users, the first step is to find similar users or items. The second step is to predict the ratings of the items that are not yet rated by a user The proposed method shows encouraging results when evaluated and compared with the traditional item based CF algorithm. We have designed a CF hybrid method that merges recommendations provided by different CF approaches based on a multi-class classification algorithm. This classification is performed based on the user rating behavior.[2]

a) Sparsity Removal:

To fill the vacant cells in the matrix, we employ CBR followed by average filling. Since we assume that similar users have similar ratings for the same items, CBR is an intuitive method to be used here with its assumption that similar cases require similar solutions. The ratings given by individual users form the cases and the similarity between two users is determined by how similar their rating patterns are. The similarity measure used here is Euclidean distance, given by following equation :

$$\boldsymbol{sim(X,Y)} = \sqrt{\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \quad (\boldsymbol{x_i} - \boldsymbol{y_i})(\boldsymbol{x_i} - \boldsymbol{y_i})}$$

where xi and yi are the values of the ith feature of the input case and that of the existing case respectively. Only the items rated by both users are used for similarity calculation. For a target user, a sorted list of top K similar users is returned based on the similarity scores between this user and other users.[9]

4) *Recommendation System Module*

The recommendation system is designed in 3 parts based on the users:

a) Recommendation system part I:

Popularity based is a great strategy to manage the new customers with the most popular products sold on a website and is very useful to start a recommendation engine.

b) Recommendation system part II:

Recommend items to users based on purchase history and similarity of ratings provided by other users who bought items to that of a particular customer.A model based collaborative filtering technique is closed here as it helps in predicting products for a particular user by identifying patterns based on preferences from multiple user data.

c) Utility Matrix:

A utility matrix consists of all possible user-item preferences (ratings) details represented as a matrix. The utility matrix is sparse as none of the users would buy all the items in the list, hence, most of the values are unknown.

d) Recommendation system part III:

For a website without any user-item purchase history, a search engine based recommendation system can be designed for users. The product recommendations can be based on textual clustering analysis given in product description. This done for the new items designated on the site have no previous ratings or reviews. For which we use hybrid collaborative filtering by implementing item-based collaborative filtering and user-based collaborative filtering.

IV. Conclusion

Recommendation systems help users discover items they might not have found by themselves and promote sales to potential customers, which provide an effective form of targeted marketing by creating a personalized shopping experience for each customer. Lots of companies have such systems, especially for e-commerce companies like Amazon.com, an effective product recommendation system is very essential to their businesses. In this paper, based on the research on some existing models and algorithms, we design three new recommendation systems, Item Similarity, Bipartite Projection and Spanning Tree. They can be used to predict the rating for a product that a customer has never reviewed, based on the data of all other users and their ratings in the system. To examine and compare their effectiveness, we implement these three algorithms and test them on some existing datasets. In our experiments, we found that, in terms of effectiveness measured with mean squared error (MSE), for all users, Item Similarity has the best result, then followed by Spinning Tree, and Bipartite Projection is the worst. For new users, Spinning Tree has the best result, followed by Item Similarity, and Bipartite Projection cannot even generate results because of lack of data. For old users, Bipartite Projection has the best result, followed by Item Similarity, and Spinning Tree is the worst. In terms of computational performance, Bipartite Projection is the fastest algorithm that gives results within fraction of seconds, while Item Similarity can be very computationally expensive[5]. In the future, we plan to improve the effectiveness and performance by exploring a hybrid system which will apply different algorithms on different user segments. One concrete thought is to use Spinning Tree on new users and use Bipartite Projection on old users. And we also need to experiment on different criteria to decide whether a user is a new user or an old user, and then choose the criterion that has the best result. We also would like to study how we could control or tweak the outputs of recommendation systems based on application-specific requirements. For example, the company might want to avoid recommending some very popular items to distribute the traffic to other products, or the company would like to promote some newly listed products. In general, it is a promising direction to build recommendation systems that can adapt to more granular and 8 flexible application-specific requirements.

REFERENCES

[1] Erion Cano, Maurizio Morisio, “Hybrid Recommender Systems: A Systematic Literature Review” January 2015.

[2] D. -R. Liu and Y. -Y Shih, “Hybrid approaches to product recommendation based on customer lifetime value and purchase preferences,” Journal of Systems and Software, Vol. 77, pp. 181–191, August 2005

[3] Xavier Amatrian, Alejandro Jaimes, Nuria Oliver, and Josep M. Pujol.,”Data Mining Methods for Recommender Systems “

[4] D. -R. Liu and Y. -Y Shih, “Product recommendation approaches: Collaborative filtering via customer lifetime value and customer demands,” Journal of Systems and Software, Vol. 35, pp. 350–360, August 2008.

[5] Deepali Kale, “Use of Data Mining Method For Secure Privacy in Second Networking Sites ``International Journal of Information Technology & Management,Vol-XII, ISSN 2249-4510, Feb-2017.

[6] Robin Burke, Alexander Felfernig, Mehmat E. Goker,”Recommender Systems: An Overview”.

[7] Zeshan Fayyaz, Mahsa Ebrahimian, Dina Nawara, Ahmed Ibrahim, Rasha Kashef,”Recommendation Systems: Algorithms, Challenges, Metrics, and Business Opportunities” Applied Sciences, November 2020.

[8] Kumar, Nitin & Fan, Zhenzhen. (2015). Hybrid User-Item Based Collaborative Filtering. Procedia Computer Science. 60. 1453-1461. 10.1016/j.procs.2015.08.222.