International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 05 | May 2020 www.irjet.net p-ISSN: 2395-0072

MULTI-DOMAIN RECOMMENDATION SYSTEM USING HYBRID FILTERING AND SUPPORT VECTOR MACHINE CLASSIFICATION

TK Vasanth[1], C PeriyaKaruppan[2], M PoornaKumar[3]

[4]*Dr Kanchana J S, Dept. of Information Technology, K.L.N. College of Engineering, Tamil Nadu, India*
[1, 2, 3]*Student, Dept. of Information Technology, K.L.N. College of Engineering, Tamil Nadu, India*

Abstract - A Recommender System refers to a system that is capable of predicting the future preference of a set of items for each user, and recommends the top items. One key reason why we need a recommender system in digital society is that people have too many options to use due to the prevalence of an internet. On other hand there has been an emerging growth among the digital content providers who want to engage as many users on their service as possible for the maximum time. This gave birth to the recommender system wherein the content providers recommend users the content according to the users' taste and liking. In this paper we have proposed a recommendation system that works upon hybrid filtering technique which is a mixture of both content-based filtering and collaborative filtering, along with a content classification which uses Support Vector Machine (SVM) classifier to categorize recommended contents. It is capable of recommending a different type of information or a content to a new user as well as the other existing users based on tags given by them. It mines entire databases to collect all the necessary and important information, such as, popularity, attractiveness, and other required attributes, which are required for recommendation and classification. Hybrid filtering technique provides more precise recommendations rather than proceeding either with content – based or collaborative filtering algorithms.

***Key Words*: Recommendation System, SVM Classifier, Collaborative filtering, Content- based filtering, Hybrid filtering**

1. INTRODUCTION

In today's digital world where the internet has become an important part of human life, the users are facing problems of choosing due to the wide variety of collections available. Searching from a motel to good investment options, clothing and other products, there is too much information available over the internet. To help the users cope with this information explosion, many companies have deployed recommendation systems for guiding and helping their users. The research and development in this area of recommendation systems has been going on for quite a long time but the interest still remains high because of the abundance of practical applications and the problem rich domain.

Recommender systems are used for providing personalized recommendations based on the user profile and previous behavior. Most popular Recommender systems such as Amazon, Netflix, and YouTube are widely used in the Internet Industry [1]. It helps the users to find and select items (e.g., books, movies, restaurants) from the wide collection available on the web or in other electronic information sources. Among a large set of items and a description of the user's needs, they present to the user a small set of the items that are well suited to the description. Similarly, a movie recommendation system provides a level of comfort and personalization that helps the user to interact better with the system and watch the movies that best match his needs [2]. Main goal of a recommender system is to suggest to user items that are most likely to meet their interest. But all the users usually do not have a single interest on one domain and all their needs will be span across different application areas. A multi-domain algorithm can able to recommend items in domain B to users with ratings only in domain A. Domain X is referred to as the source domain, domain Y is referred to as the destination domain. For this reason, multi-domain algorithms are attracting more attention because they are able to suggest items that are not necessarily part of the same domain in which the user provided his/her ratings. For example, in order to recommend songs, movies, clothes or books to users that have provided only their musical tastes.

It helps users find compelling content in a large collection. For example, the famous Google Play Store provides millions of apps, while YouTube provides billions of videos. More apps and videos are added every day. Due to this, there comes problem in compelling new content but Yes, one can use search to access content. However, a recommendation engine can display contents that users may not have thought to search for on their own. Mostly available recommendation systems were single domain

oriented such as Amazon can recommend to users particularly on products that they previously bought and not with other information's, whereas Netflix can provide only movie recommendations.

[3] The main purpose of our system is to recommend various contents to the users along different domains based on their viewing history and score or rating that they provide to their previously viewed webpage. This system will also recommend their needful information to specific customers based on the tags they prefer. It makes the system to be more customizable, consistent and also scalable. Since we applied Hybrid filtering technique, The Collaborative filtering and content-based filtering are the prime approaches in providing recommendations to the users. Both of them are best applicable in specific scenarios because of their respective properties and applying support vector machine (SVM) classifier can categorize the user data according to their domain which allows the system to recommend variety of information or content on multiple domains. In this paper a mixed approach has been used such that both the algorithms complement each other thereby improving performance and accuracy to our system.

2. RELATED WORK

As we know, The Recommendation systems are a data-driven tool in which that most of the companies frequently adopt to fulfill their customers' personalization needs (Hinz and Eckert 2010; Ricci, Rokach, and Shapira 2015). That Depends on what customers have already viewed, liked, rated or purchased, these systems can effectively predict what other products that they could be interested in and also delivers instant suggestions. The research in marketing and information systems highlights such recommender systems as important determinants of sales (Bodapati 2008; Fleder and Hosanagar 2009; Pathak et al. 2010). Two typical methods inform these recommendations. First, collaborative filtering identifies customers who are similar in their product rating history and recommends items that one customer likes to similar other customers. The product ratings might be explicitly provided by customers or inferred from their online behavior. Second, content-based filtering identifies the product attributes that a customer likes and recommends products with similar attributes (Ansari, Essegaier, and Kohli 2000). Because each method has shortcomings, companies often combine them to improve the performance of their hybrid recommender systems. Examples include Amazon's "item-to-item collaborative filtering" (Linden, Smith, and York 2017), and the New York Times' collaborative topic modeling (Spangher 2015). Extensive research suggests ways to improve the prediction accuracy of recommendation algorithms using hybrid frameworks (Zhang et al. 2018).

The computationally complex algorithms pose challenges for explaining recommendations to customers. A clear, concise, accurate explanation is crucial, because it promotes customers' trust in the recommender systems (Wang and Benbasat 2007) and acceptance of recommendations (Cramer et al. 2008; Kramer 2007). To the best of our knowledge, no research in marketing has suggested the optimal methods for explaining recommendations. In information systems literature, Tintarev and Masthoff (2015) identify five recommendation explanation types. Two explanations are particularly relevant to our research: collaborative-based and content-based. As their names imply, collaborative-based explanations such as "Customers who bought this item also bought…" rely on recommender systems that adopt collaborative filtering, whereas content-based explanations, such as "Recommended because you said you owned…," involve recommender systems that use content-based filtering. The other explanation types either overlap with the content-based explanation (e.g., cased-based that specifies the items compared by the underlying algorithm) or assume unique inputs (e.g., demographic-based; Tintarev and Masthoff 2015).

Given the issues of scalability and sparsity faced by traditional collaborative filtering algorithms, there have been various approaches proposed to address these issues. For scalability problem, the main idea is to reduce the size of data to be considered during recommendation.

Sarwar et al. [6] has applied clustering to group users with similar rating patterns into clusters, so as to reduce the scope of neighborhood to be considered during recommendation. It's shown to significantly improve online performance while giving comparable prediction accuracy with the traditional algorithm.

Gong and Ye [7] in their work have tried to join user clustering with item-based CF. They apply the K-Means algorithm on the user-item matrix of ratings to obtain clusters as neighborhoods. For a target user X, the closest cluster is identified and item-based CF prediction is then applied within the cluster. SOM clustering has also been experimented as a pre-processing step7 and thereby the similarity and weighted averaging methods only need to be applied on the users in a given cluster. For the issue of sparsity of the User-Item matrix, efforts focus on how to

reduce the sparsity by filling in the vacant cells in the User-Item matrix with likely rating values before the matrix is used for further processing.

Xia et al [8] use a simple technique called average filling, assigning the unrated items of a user with the average value of all his other ratings, and then use the modified data for item based collaborative filtering.

Gong's work [9] employs CBR techniques to fill in vacant ratings. For an active user case, the most similar user cases are retrieved from the case base using Euclidean distance and the missing ratings are estimated based on the ratings from these similar users. There are several optimization techniques proposed to improve the results of collaborative filtering.

Kim and Ahn [10] have proposed a hybrid method combining K-Means clustering with GA optimization. K-Means clustering has the drawback that the clustering results can be sensitive to the initial seed used to partition the dataset. GA is therefore used to select optimal or suboptimal seeds for K-Means clustering. The fitness function for GA is the performance of the clustering algorithm, measured using intra-class inertia which is the average of the distances between the mean and the observations in each cluster. After GA-K Means clustering, the cluster for a target user is identified and the nearest neighbors are found from that cluster.

Zhang et al [11] have explored three new item similarity measures, essentially weighting the original correlation-based similarity formula with a ratio of the number of users that rate both items i and j (N) and the number of users that rate item I or J (M). They have concluded that a simple multiplication of (N/M) with the original similarity measure gives the best result. Item based hybrid similarity13 instead brings in the values of item attribute similarity to adjust the predicted rating of the targeted item. Deviation Adjustment8 is another technique that has been suggested to minimize the error between the actual and predicted user ratings. The CF algorithm's prediction is modified based on user deviation adjustment and item deviation adjustment which are measured from the algorithm's error in predicting the training data.

Other techniques have also been proposed to improve the recommendation quality of item-based CF. Item based CF using UserRank11 [12] tries to use a modification of PageRank algorithm to rank or weight users in the User-Item matrix based on their importance. The weights of the users are then incorporated into the computation of item similarities and item-based CF recommendations.

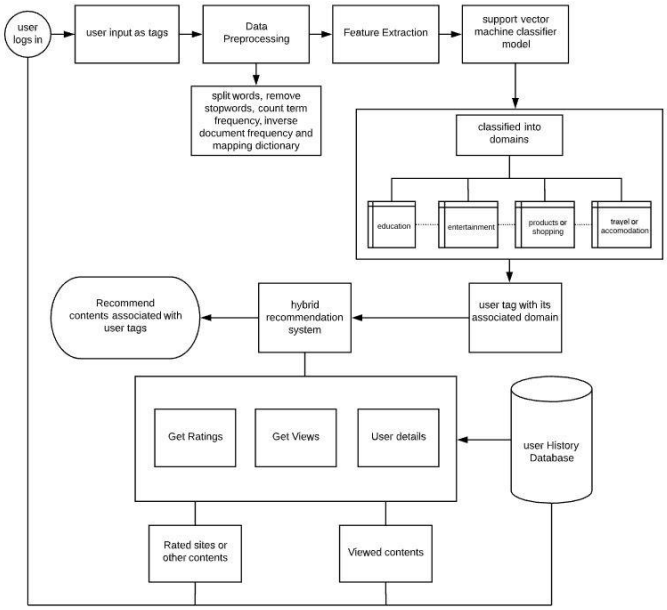


Chart-1: Proposed architecture of Hybrid Recommendation System on multiple domains

2. MODULE DESCRIPTION

2.1 Data mining and Preprocessing

Data mining is a technique for exploration, exploitation and analysis of variety of data's over large volume, which organizes the data in a way more understandable and also well readable. On the other hand, it serves to discover the rules of organization, classification and prediction to assist decision makers in the decision support process. [9]. Data mining involves variety of techniques like classification, clustering which helps organization to get knowledge-based information [10]. Data mining allows also exploiting data to increase the profitability of the institutions namely companies, government etc. and increase the return on investment of their information systems.

Most important process before Data mining is Data preprocessing, which is a most required step in any data mining process. This step involves transforming raw data or primary data into an understandable format for models. All real-world data are often incomplete, inconsistent, and also lacking in certain behaviors or trends, and is likely to contain many errors. Data preprocessing is a proven method of resolving such kind of issues. These processes will help in getting better results through some classification algorithms.
The two techniques that are to be performed during data pre-processing are:

1. **Tokenization**: This is a process which breaks a stream of text up into words, phrases, symbols, or other meaningful elements called as tokens. The list of tokens containing words or other chunks becomes input for further processing. In python, NLTK Library has word_tokenize and sent_tokenize methods that can easily break a stream of text into a list of words or sentences, respectively.

2. **Word Stemming/Lemmatization**: The aim of both processes is the same, to reduce the inflectional forms of each word into a common base or root word. Lemmatization is closely related to stemming. The difference is that a stemmer operates on a single word without knowledge of the context, and therefore cannot discriminate between words which have different meanings depending on part of speech. It can be useful to determine domain vocabularies in domain analysis. However, stemmers are typically easier and simple to implement and runs faster, in some situations the reduced accuracy may not matter for some applications.

2.2 Support Vector Machine (SVM) Classification

Classification is a data analysis approach. It serves to facilitate the study and processing of data with large volumes [11], this approach aims to consolidate data into groups to categorize data. It is a method among the methods used in data mining for the processing, analysis and exploitation of important data. The data is grouped into several classes such a way that the data of the same class are as similar as possible and the classes are the most distinct possible. There are many classification approaches in data mining namely neural networks, Bayesian networks and decision trees, support vector machine etc.

A support vector machine (SVM) is a supervised machine learning method with associated learning algorithms that analyze data used for classification and regression analysis and also uses classification algorithms for two-group classification problems. After giving an SVM model sets of labeled training data for either of two or more categories, they're able to categorize new examples from trained SVM model.

It is mostly used in classification problems. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform SVM classification to find the hyper-plane that differentiates two classes very well.

SVM classification has some benefits compared to others classifiers. They are robust, accurate and very effective even in cases where the number of training samples is small. SVM technique also shows the greater ability to generalize and the greater likelihood of generating good classifiers. In our proposed system, we used to classify the user tags according to their Topic or a Domain and recommend contents to the user by analyzing their previously viewed or rated contents.

Text Classifier Algorithms	Accuracy
Naive Bayes	0.74
Logistic Regression	0.78
Word2vec and Logistic Regression	0.63
Doc2vec and Logistic Regression	0.80
BOW with keras	0.79
Linear Support Vector Machine	**0.84**

Table -1: Various Text Classifier Comparisons

2.3 Proposed Hybrid Approached Recommendation System with SVM Classifier

Recommendation system must be able to make predictions about the interests of users. It must collect a number of data of these users in order to be able to build a profile for each user. There are two different approaches that usually takes place in recommendation system, namely,

Content based approach utilizes a series of discrete characteristics of an item in order to recommend additional items with similar properties.

Collaborative filtering approach builds a model from a user's past behaviors (items previously purchased or selected and/or numerical ratings given to those items) as well as similar decisions made by other users. This model is then used to predict items (or ratings for items) that the user may have an interest in.

Hybrid approach combines the previous two approaches. Most businesses probably use the Hybrid approach in their production recommender systems.

In this section, we will present our proposed recommendation system. Our recommendation system aims to recommend the most appropriate content for

users of different domains. This system helps to build the center of interest for users by facilitating access to content. Most recommendation systems are based on the processing of user profiles only, however, our system designed to take into consideration several aspects when recommending namely:

The consideration of the center of interest if it exists

• The consideration of evaluations on the content and the number of visits by users of the domain as well as the user's reputation who put the contents

• Construction of the center of interest of users using surveys proposed by the platform for users who have not fulfilled or not completed their center of interest to fill it or to evolve it

• Using data collection with dual method, explicit method and implicit method

• Using the Hybrid filter that combines content filtering and collaborative filtering in order to recommend

• Using the data mining and especially the support vector machine in order to provide recommendations and to classify users and content

• The construction of file log and historical data by user in order to be used during the next recommendation Our proposed system is divided into four main parts:

• The first part is for the data collection about the user profile, the content to recommend, the evaluations of the user, the number of visiting a content and on requests sent into the platform.

• The second part is for processing of the information already collected in the previous section and for creating the user model, classification of users who have submitted the contents and for classification of content in order to send the results to the recommendation part

• The third part is for making the similarity between users and content and performing the recommendation

• The fourth part is for creating file log and historical data by user in order to be used during the next recommendation

We have tried to include the recommendation based on the historical data that the user will search in browsers, cookies, and the requests sent on the platform in order to turn our system into a system that is able to recommend for the users who are not registered on the platform.

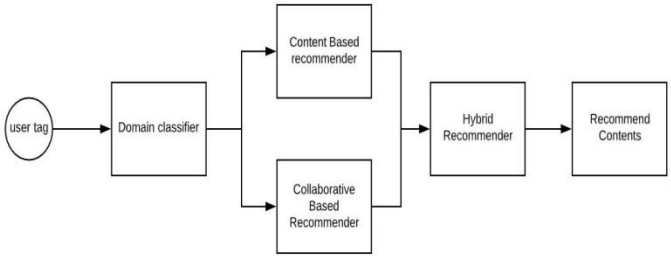


Fig-3: Hybrid recommendation system

3. FUTURE WORK

Our Proposed system consists of multi-domain recommender and it is necessary to maintain accuracy, consistency and also scalability among entire environment. We will try to work with multi-agent systems to manage the modules of our proposed systems to improve the performance in terms of processing and response time because we will use the distributed processing, which reduce the response time. In addition, multi-agent systems will ensure the accuracy of our proposed system in terms of recommendation. Moreover, we will transform our recommendation system into a generic module and also try to make it as an application programmable interface.

4. CONCLUSION

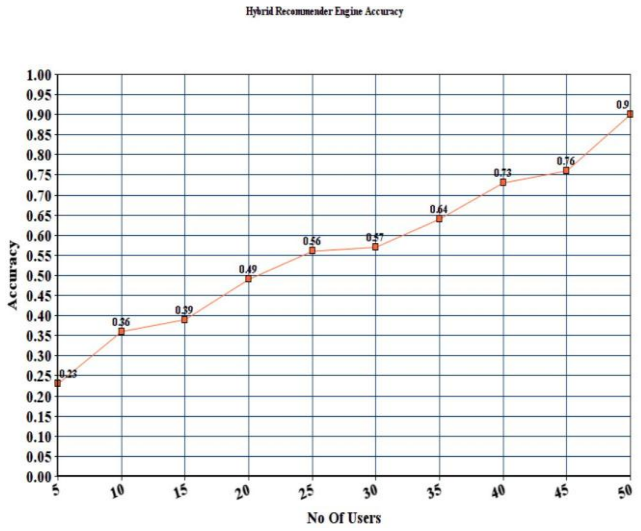
According to the method of the cosine distance, content that has the highest value is the most appropriate content for the user. In the figure, the recommendation is made for the initial user. To evaluate our proposal in terms of accuracy and response time needed to recommend, we conducted many experiments. We started with a matrix of 5 users and 25 content. In each time, we increase the number of users by 5 and the number of contents by 25. To evaluate the performance and accuracy of our proposed system for recommendation against the standard recommendation, we used the following equation:

Accuracy = ITPR / (ITPR + IFPR)

ITPR: Item True Positive Recommendation. Case was positive i.e. the data and the rules were good and the recommendation is also positive.

IFPR: Item False Positive Recommendation. Case was positive i.e. the data and rules and all but unfortunately the recommendation was negative.

At the end, we observed that our proposed system remains beyond the threshold that we set to satisfy the recommendation that is set to 90%, and degrades slowly unlike the standard recommendation that degrades very quickly with time at each increasing in volume of the matrix so that at certain volume, it will no longer satisfy the recommendation close to 90% with an uncertainty of 10% margin.



REFERENCES

[1] R. R. Yager, On ordered weighted averaging aggregation operators in multi-criteria decision making, IEEE Trans. Syst. Man Cybern. 18 (1) (1988) 183–190. doi:10.1109/21.87068.

[2] Resnick P., Iacovou N., Suchak M., Bergstrom P., Riedl J., « GroupLens : an open architecture for collaborative filtering of netnews », CSCW '94 : Proceedings of the 1994 ACM conference on Computer supported cooperative work, ACM, New York, NY, USA, p. 175-186, 1994. https://doi.org/10.1145/192844.192905

[3] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl. GroupLens : Applying Collaborative Filtering to Usenet News. Commun. ACM, 1997

[4] J. Breese, D. Heckerman, and C. Kadie, Empirical analysis of predictive algorithms for collaborative filtering, in Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, 1998

[5] Pazzani M. J., « A Framework for Collaborative, Content-Based and Demographic Filtering », Artif. Intell. Rev., vol. 13, n° 5-6, p. 393-408, 1999. https://doi.org/10.1023/ A:1006544522159

[6] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-Based Collaborative Filtering Recommendation Algorithms. Proceedings of the 10th international conference on World Wide Web, 2001

[7] Sarwar B., Karypis G., Konstan J., Reidl J., « Item-based collaborative filtering recommendation algorithms », WWW '01 : Proceedings of the 10th international conference on World Wide Web, ACM, New York, NY, USA, p. 285-295, 2001. https://doi.org/10.1145/ 371920.372071

[8] R. R. Yager, Fuzzy logic methods in recommender systems, Fuzzy Sets Syst. 136 (2) (2003) 133–149. doi: 10.1016/S0165-0114(02)00223-3.

[9] B. Kitchenham, Procedures for performing systematic reviews, Keele, UK, Keele University, 33, (2004) (2004) 1–26.

[10] B. Kitchenham, S. Charters, Guidelines for performing systematic literature reviews in software engineering, EBSE Technical Report, EBSE 2007-001, Keele University and Durham University Joint Report (2007).
[11] Pazzani M., Billsus D., " Content-Based Recommendation Systems ", p. 325-341, 2007. https://doi.org/10.1007/978-3-540-72079-9_10

[12] Burke R., « Hybrid Web Recommender Systems », in P. Brusilovsky, A. Kobsa, W. Nejdl (eds), The Adaptive Web, vol. 4321 of Lecture Notes in Computer Science, Springer, chapter 12, p. 377-408, 2007. https://doi.org/10.1007/978-3-540-72079-9_12
[13] Z.-K. Zhang, C. Liu, Y.-C. Zhang, T. Zhou, Solving the cold-start problem in recommender systems with social tags, EPL (Europhysics Letters) 92 (2) (2010) 28002. doi:10.1016/j.eswa.2012.03.025.

[14] L. M. de Campos, J. M. Fernndez-Luna, J. F. Huete, M. A. Rueda-Morales, Combining content-based and collaborative recommendations: A hybrid approach based on bayesian networks, International Journal of Approximate Reasoning 51 (7) (2010) 785 – 799. doi:http://dx.doi.org/10.1016/j.ijar.2010.04.001.

[15] M. Ge, C. Delgado-Battenfeld, D. Jannach, Beyond accuracy: Evaluating recommender systems by coverage and serendipity, in: Proceedings of the Fourth ACM Conference on Recommender Systems, RecSys '10, ACM, New York, NY, USA, 2010, pp. 257–260. doi:10.1145/1864708.1864761.

[16] D. Cruzes, T. Dyba, Recommended steps for thematic synthesis in software engineering, in: Empirical Software Engineering and Measurement (ESEM), 2011 International Symposium on, 2011, pp. 275–284. doi:10.1109/ ESEM.2011.36.

[17] X. Amatriain, A. Jaimes*, N. Oliver, J. M. Pujol, Recommender Systems Handbook, Springer US, Boston,

MA, 2011, Ch. Data Mining Methods for Recommender Systems, pp. 39–71. doi:10.1007/ 978-0-387-85820-3_2.

[18] D. Jannach, M. Zanker, M. Ge, M. Groning, E-Commerce and Web Technologies: 13th International Conference, ¨ EC-Web 2012, Vienna, Austria, September 4-5, 2012. Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, Ch. Recommender Systems in Computer Science and Information Systems – A Landscape of Research, pp. 76–87. doi:10.1007/978-3-642-32273-0_7.

[19] B. Lika, K. Kolomvatsos, S. Hadjiefthymiades, Facing the cold start problem in recommender systems, Expert Systems with Applications 41 (4, Part 2) (2014) 2065 – 2073. doi:http://dx.doi.org/10.1016/j. eswa.2013.09.005.

[20]Haruna K, Akmar Ismail M, Damiasih D, Sutopo J, Herawan T (2017) A collaborative approach for a research paper recommender system. PLoS ONE 12(10): e0184516. https://doi.org/10.1371/journal.pone.0184516