Efficient Recommendation System Using Decision Tree Classifier and Collaborative Filtering

Sayali D. Jadhav[1], H. P. Channe [2]

[1]*Research Scholar, Dept. of Computer Engineering, PICT, Pune, Maharashtra, India*
[2]*Professor, Dept. of Computer Engineering, PICT, Pune, Maharashtra, India*

---***---

Abstract - *There has been an exponential growth in digital information and there are large number of choices for products and services. So, there is need to filter, prioritize and efficiently deliver relevant information in order to efficiently tackle the problem of information overload. Recommendation systems solve this problem by searching through large volume of dynamically generated information to provide users with personalized contents and services. Recommendation system uses historical data of users' preferences and their purchases to predict items that might interest the users.*

Recommendation systems are mainly dependent on classifier. So, it is important to develop accurate classifier to improve the performance of recommendation system. Generally, recommender systems use KNN classifier but it requires more time for processing large dataset. Decision tree classifiers like C4.5 and C5.0 algorithms have the merits of high accuracy, high classifying speed, strong learning ability and simple construction. In this paper, the decision-tree-based recommendation system framework is proposed. It uses efficient classification algorithm combined with collaborative recommendation approach for book recommendation. This hybrid book recommendation system combines advantages of both decision tree classifier and collaborative filtering. The results of C4.5 and C5.0 decision tree classifiers are compared and book recommendations are given to user by using efficient C5.0 decision tree classifier.

Key Words: Recommendation System, Decision Tree Classifier, C4.5, C5.0, Collaborative Filtering.

1. INTRODUCTION

There has been an exponential growth in the amount of available digital information, electronic origins, and online services in late years. Such a large information overload has created a potential problem of how to handle such a large volume of data efficiently and how to filter and efficiently deliver relevant information to a user. Additionally, information needs to be processed for a user rather than just filtering the right information. This problems highlight a need for information extraction systems that can filter relevant information and can predict the information of users' interest. Such systems are called recommender systems [1]. Recommendation systems apply machine learning and data mining techniques for filtering unseen information and using that it can predict whether a user would like a given resource or not. Large-scale commercial application of the recommendation system can be found in many e-commerce sites such as Amazon, CDNow. Recommender systems are mainly used on the web for recommending products and services to users. Many e-commerce sites have such systems. Such systems provides two main functions. They help users in dealing with the information overload by giving them recommendations of products, services etc. Secondly, they help businesses make more profits, i.e., by selling more products. Recommender systems are mainly dependent on classifier. So, it important to develop accurate classifier [2]. There are different classification techniques like K-Nearest Neighbors, Naive Bayes classifier, Support Vector Machine and Decision tree algorithms. Amongst all, decision tree classifiers are easy to build and relatively fast classifiers. They produce much accurate result than other classifier in less time. Decision tree classifiers like C4.5 and C5.0 algorithms have the merits of high accuracy, high classifying speed, strong learning ability and simple construction. So, in this paper, efficient decision tree classifier is combined with collaborative filtering recommendation approach.

2. RELATED WORK

To date there has been a tremendous growth in the development of recommender sites. The number of people using the recommender systems are increasing exponentially day by day which makes it very important for these systems to generate recommendations that are close to the items of users' interest. Historically, recommender systems are categorized into collaborative filtering, content-based or hybrid systems [3], where content-based recommender systems recommend items based on the content information of the items. It uses the textual information of an item, under the assumption that users will like similar items to the ones they liked before. Collaborative filtering recommender systems [4] recommend items by taking into account the taste (in terms of preferences of items) of users, under the assumption that users will be interested in items that users similar to them have rated highly and hybrid combine or unify, user and content oriented approaches and have shown to outperform their two-mode counterparts in many scenarios. To improve the performance of recommender system, various classification approaches have been used for recommender systems. In [5], the authors have used linear classifier in a model-based recommender system. In [1], authors have proposed unique generalized switching hybrid recommendation algorithms that combine machine learning classifiers with the collaborative filtering recommender systems.

Collaborating filtering recommender systems are based on the assumption that people who agreed in the past, will agree in the future too. In [6], authors have proposed a unique switching hybrid recommendation approach by combining a Naive Bayes classification approach with the collaborative filtering recommendation approach. Experimental results on two different data sets, showed that the proposed algorithm provides scalability and provide better performance in terms of accuracy and coverage than other algorithms while at the same time it also eliminates some recorded problems with the recommender systems.

Collaborative filtering can be classified into two subcategories: memory-based (user based) CF and model based (item based) CF. Memory-based approaches make a prediction by taking into account the entire collection of previous rated items by a user, examples include GroupLens recommender systems [7]. The advantage of these algorithms is the quick incorporation of the most recent information, but the disadvantage is that the search for neighbors in large databases is slow [8]. In order to avoid this inconvenience, model-based CF algorithms have been proposed. There are great variety of data mining algorithms that can be applied in model-based CF. Neural networks were the first of this kind of method [8]. In an example of the Amazon's recommender engine [9], authors have used model based Item-to-Item Collaborative Filtering algorithm. Their algorithm's online computation scales independently of the number of customers and number of items in the product catalog and produces recommendations in realtime, scales to massive data sets and generates high quality recommendations. But these systems suffer from scalability, data sparsity, over specialization, and cold-start problems resulting in poor quality recommendations and reduced coverage. To achieve higher performance and overcome the drawbacks of traditional recommendation techniques, a hybrid recommendation technique that combines the best features of two recommendation techniques into one hybrid technique has been proposed [10]. It is used in an attempt to avoid cold-start, sparseness and/or scalability problems. In this paper, to improve performance of recommender system, decision tree classifier is trained on content information and then combined with collaborative filtering approach. Use of decision tree classifier also reduces search time of finding neighbors.

3. DECISION TREE CLASSIFIER

3.1 C4.5

C4.5 [11] is a decision tree based classification algorithm developed by Ross Quinlan. C4.5 is an extension of Quinlan's earlier ID3 algorithm. As the decision trees generated by C4.5 can be used for classification, C4.5 is often referred to as a statistical classifier. C4.5 algorithm uses information gain as splitting criteria [11]. It can handle data with categorical as well as numerical values. To handle continuous values it generates threshold and then divides attributes with values more than the threshold value and values equal to or less than the threshold value. C4.5 algorithm can easily handle missing values but missing attribute values are not utilized in gain calculations by C4.5.

Algorithm:

1. Let T be the training sample. C_j is class label and j = 1, 2,......., Nclass. Let freq (C_j, T) stand for the number of samples in T that belong to class C_j (out of N possible classes), and |T| denotes the number of samples in the training set T.
2. Check for base cases.
3. Find the best split attribute for splitting that provides maximum information gain.
4. It uses two measures to find best split.

Entropy: It is used to measure impurity i.e. to calculate the homogeneity of a sample. Then the entropy of the set T is calculated as:

$$info(T) = -\sum_{j=1}^{NClass} \frac{Freq(C_j,T)}{|T|} \times log_2 \frac{Freq(C_j,T)}{|T|}$$

Information Gain: Information gain tells us how important a given attribute is.

$$gain = info(T) - \sum_{i=1}^{s} \frac{|T_i|}{|T|} \times info(T_i)$$

5. Best splitting attribute is the one which provides maximum information gain.
6. Using the attribute that provides maximum information gain, decision tree is generated.
7. And same steps are recursively applied to each impure node of tree. C4.5 algorithm then stops when all nodes are pure.

Base cases are as follows:

1. All the examples from the training set belong to the same class (a tree leaf labeled with that class is returned).
2. The training set is empty (returns a tree leaf called failure).
3. The attribute list is empty (returns a leaf labeled with the most frequent class or the disjunction of all the classes).

3.2 C5.0

C5.0 algorithm is an extension of C4.5 algorithm. C5.0 [12] is the classification algorithm which is generally used for big data set. C5.0 has better efficiency and memory utilization than C4.5. Overfitting problem of C4.5 is solved by the C5.0. As a result, the results generated by C5.0 classifier are more accurate. In C5.0, the sample subsets that don't have remarkable contribution to the model will be rejected. So, C5.0 algorithm generates considerably smaller decision tree than C4.5. It can also easily handle missing attribute from data set. In this paper, C5.0 algorithm uses information gain ratio as splitting criteria.

All other steps in C5.0 algorithm are same as C4.5.

Gain ratio is calculated as follows.
Gain(A)= gain

$$gain = info\,(T) - \sum_{i=1}^{s} \frac{|T_i|}{|T|} \times info(T_i)$$

SplitInfo(A)= Split(T)

$$Split\,(T) = -\sum_{i=1}^{s} \left(\frac{|\mathrm{Ti}|}{|T|}\right) * \log_2\left(\frac{|\mathrm{Ti}|}{|T|}\right)$$

The gain ratio is defined as

$$Gain\ Ratio\,(A) = \frac{Gain\,(A)}{Split\ Info\,(A)}$$

The attribute with the maximum gain ratio is selected as the splitting attribute.

But if the value of SplitInfo(A)=0, the gain ratio fails.

Note that as the split information approaches 0, the ratio becomes unstable. A constraint is added to avoid this, whereby the information gain of the test selected must be large-at least as great as the average gain over all tests examined i.e. selected gain value should be greater than or equal to average gain of all tests examined. And lastly, pruning is applied on generated decision tree to minimize the classification error.

3.3 C5.0 improvements from C4.5 algorithm

- Speed - C5.0 is significantly faster than C4.5.
- Memory usage - C5.0 is more memory efficient than C4.5.
- Accuracy: The C5.0 rulesets have noticeably lower error rates on unseen cases. Sometimes the C4.5 and C5.0 rulesets have the same predictive accuracy, but the C5.0 ruleset is smaller.
- Smaller decision trees - C5.0 gets similar results to C4.5 with considerably smaller decision trees.
- Support for boosting - Boosting improves the trees and gives them more accuracy.

4. COLLABORATIVE FILTERING

Collaborative filtering (CF) is a popular recommendation algorithm that bases its predictions and recommendations on the ratings or behavior of other users in the system [13]. Collaborative filtering is also referred to as social filtering as it filters information by using the recommendations of other people. Collaborative filtering recommender systems recommend items by identifying other users with similar taste and use their opinions for recommendation. Collaborative filtering explores techniques for matching people with similar interests and making recommendations on this basis.

The workflow of a collaborative filtering approach in this system is:

1. A user expresses his or her preferences by rating books of the system. These ratings can be viewed as an approximate representation of the user's interest in the corresponding domain.

2. The system matches this user's ratings against other users' and finds the people with most "similar" tastes.

3. Similarity between users is calculated using Pearson Correlation formula as below.

Let, a, b : Users for which the coefficient is being calculated.

P: Set of books, rated by both a and b.

$r_{a,p}$ and $r_{b,p}$ are individual ratings from a and b for p.

$\bar{r_a}$ and $\bar{r_b}$ are average ratings for user a and b.

$$Sim(a,b) = \frac{\sum_{p\in P}(r_{a,p} - \bar{r}_a)(r_{b,p} - \bar{r}_b)}{\sqrt{\sum_{p\in P}(r_{a,p} - \bar{r}_a)^2}\sqrt{\sum_{p\in P}(r_{b,p} - \bar{r}_b)^2}}$$

4. With similar users, the system recommends books that the similar users have rated highly but not yet being rated by this user (presumably the absence of rating is often considered as the unfamiliarity of an book.)

5. Similarly, item-item similarity is also computed using Pearson Correlation i.e. similarity between books rated by user and others books in system is calculated.

6. And lastly, the most similar books are given as recommendations to the target user.

5. PROPOSED SYSTEM

In this proposed system, a collaborative filtering recommendation method is combined with the efficient decision tree classifier to improve the performance of recommendation system. As results of recommendation systems are mainly dependent on classifier, so it is important to develop accurate classifier. In this proposed system, C4.5 and C5.0 classifiers are applied to training database and the results of classifiers are compared and efficient classifier model is then combined with collaborative filtering recommendation approach and the recommendations are given to the user. This is all shown in following figure 1.

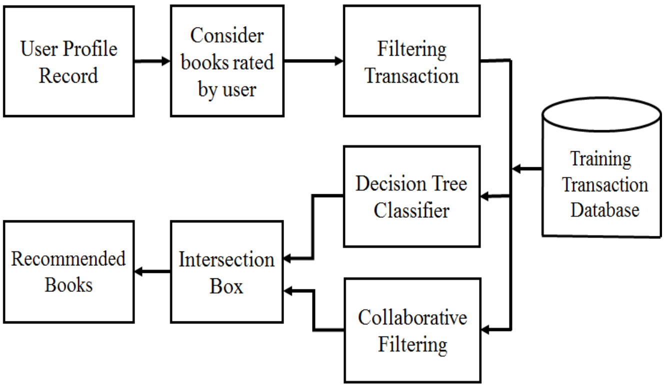

Fig -1: Proposed Framework

Purpose of this book recommendation system is to recommend books to the buyer that suits their interest. This recommendation system works offline and stores recommendations in the buyer's web profile. This system has following steps:
1. First after login / register, the user profile record is given as input to the system.
2. Find out the books that the user has bought earlier from the user's profile.
3. Find out the ratings given by user to that books if there is any found book in the step 2.
4. Perform filtering of transactions found in step 2 and 3, to find out the books that are much similar to the books that the user has bought earlier.
5. Apply decision tree classifier on all the transactions from database to find out the books that are much similar to the books that the user has bought earlier based on the books overview content from the user's past history record.
6. Perform collaborative filtering on user profile record to find out the other users that are much similar to the target user and then find the other books that are similar to books the user has bought earlier based on the ratings given by user to different books from the user's past history record.
7. At this stage, we have two result lists, one from decision tree classifier and other from collaborative filtering recommendation approach.
8. Then in intersection box, we combine the two results into one by considering books with maximum confidence values. This step is actually more refinement of the recommendations generated by the step 5 and 6.
9. Arrange the intersection result in the descending order of recommendations.
10. Outcome of the step 9 is the final recommendations for the user. All these steps are performed when the user is offline and the results are stored in the user's web profile. When the user comes online next time the list of recommended books is given to the target user.

6. RESULTS

This section provides the performance and accuracy results of C4.5 and C5.0 classifiers for book recommendation system. Comparison between C4.5 and C5.0 Classifiers is done by using following strategies:

1. Accuracy

Accuracy is calculated as:

$$Accuracy = \frac{No.of\ books\ recommended\ by\ classifier}{No.of\ books\ with\ user\ ratings}$$

Table -1: Results of accuracy of classifier

Size of Dataset	C4.5	C5.0
10 instances	50%	66.67%
103 instances	68.88%	68.88%
262 instances	93.87%	93.87%
500 instances	94%	94%
1000 instances	98.27%	98.27%

2. Time for execution

Table -2: Results of time taken by classifier for recommendation

Size of Dataset	Time of C4.5	Time of C5.0
10 instances	367 msec	43 msec
103 instances	3328 msec	421 msec
262 instances	8699 msec	560 msec
500 instances	16665 msec	644 msec
1000 instances	27993 msec	785 msec

6.1 PERFORMACE RESULTS WITH GRAPHS

Above accuracy and execution time values are plotted in these graphs.

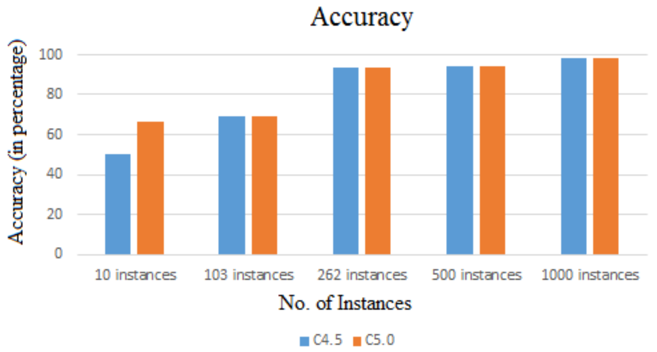

Chart -1: Results of accuracy of classifiers for recommendation

This chart -1 graph gives accuracy details of each algorithm. No. of instances are nothing but the No. of transactions in database. This graph shows that C5.0 algorithm has more accuracy in all cases.

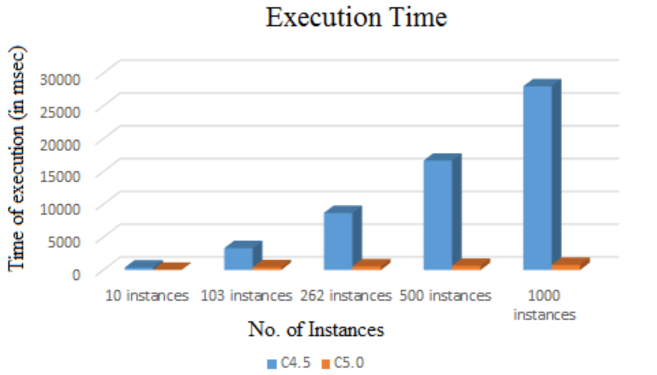

Chart -2: Results of execution times of classifiers for recommendation

This chart -2 graph gives execution time details of each algorithm. No. of instances are nothing but the No. of transactions in database. This graph shows that C5.0 algorithm requires very less execution time than C4.5 in all cases.

7. CONCLUSION

In order to meet the requirements of efficient handling of large volume of data, recommendation systems are used to deliver meaningful recommendations to a collection of users for items or products. The performance of recommendation system is mainly dependent on classifier. So, in this proposed system, collaborative filtering recommendation method is combined with the efficient C5.0 decision tree classifier. Comparative study and analysis between two decision tree algorithms C4.5 & C5.0 have shown that C5.0 algorithm provides more accurate results for book recommendation in less time. Use of Pearson Correlation similarity measure also provides more accurate results. So, the system generates high quality recommendations for the user. This approach outperforms others in terms of accuracy, time and coverage.

REFERENCES

[1] Mustansar Ali Ghazanfar and A. P. Bennett, "Building Switching Hybrid Recommender System Using Machine Learning Classifiers and Collaborative Filtering", IAENG International Journal of Computer Science, 19 August 2010.

[2] Zhi Qiao, Peng Zhang, Yanan Cao, Chuan Zhou and Li Guo, "Improving Collaborative Recommendation via Location-based User-Item Subgroup", 14th International Conference on Computational Science, Vol. 29, 2014.

[3] M. Balabanovic and Y. Shoham, "Content-Based, Collaborative Recommendation", Communications of the ACM, Vol. 40, No. 3, pp. 66-72, 1997.

[4] D. Goldberg, D. Nichols, B. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry", Communications of the ACM, Vol. 35, No. 12, pp. 70, 1992.

[5] Tong Zhang and Vijay S. Iyengar, "Recommender Systems Using Linear Classifiers", Journal of Machine Learning Research 2, 2002.

[6] Mustansar Ali Ghazanfar and Adam Prugel-Bennett, "An Improved Switching Hybrid Recommender System Using Naive Bayes Classifier and Collaborative Filtering", International MultiConferernce of Engineers Computer Scientists, Vol. 1, 2010.

[7] F.O. Isinkaye, Y.O. Folajimi and B.A. Ojokoh, "Recommendation systems: Principles, methods and evaluation", Egyptian Informatics Journal, 2015.

[8] Maria N.Moreno and Saddys Segrera,, "Web mining based framework for solving usual problems in recommender systems: A case study for movies' recommendation", Neurocomputing Elsevier Journal, 2015.

[9] Greg Linden, Brent Smith and Jeremy York, "Amazon.com Recommendations, Item-to-Item Collaborative Filtering", IEEE Internet Computing, 2003.

[10] Jie Lu, Dianshuang Wu, Mingsong Mao, Wei Wang and Guangquan Zhang, "Recommender system application developments: A survey", Decision Support Systems Elsevier Journal, 2015.

[11] Salvatore Ruggieri, "Efficient C4.5", IEEE transaction on knowledge and data engineering, Vol. 14, N0. 2 March/April 2012.

[12] A. S. Galathiya, A. P. Ganatra and C. K. Bhensdadia, "Improved Decision Tree Induction Algorithm with Feature Selection, Cross Validation, Model Complexity and Reduced Error Pruning", International Journal of Computer Science and Information Technologies, Vol. 3, No. 2, 2012.

[13] Chee Seng Chong, Tianyou Zhang, Kee Khoon Lee and Bu-Sung Lee, "Collaborative Analytics with Genetic Programming for Workflow Recommendation", IEEE International Conference on Systems, Man, and Cybernetics, 2013.

BIOGRAPHIES

Sayali D. Jadhav received B.E. degree in Computer Engineering from Vidya Pratishthan's College of Engineering, Baramati, Pune and currently pursuing M.E. degree in Computer Engineering from Pune Institute of Computer Technology, Pune. Her research interest is in Data Mining.

Prof. H. P. Channe is an Assistant Professor in Computer Engineering Department at Pune Institute of Computer Technology, Pune. Her research area includes Distributed Systems, Cloud Computing and Security.