A Novel Approach for Book Recommendation Systems

P Devika, R C Jisha and G P Sajeev
Department of Computer Science and Engineering
Amrita School of Engineering, Amritapuri,
Amrita Vishwa Vidyapeetham,
Amrita University, India
E-mail: devuprasannan@gmail.com, jisha@am.amrita.edu, sajeevgp@am.amrita.edu

Abstract— **Recommendation systems are widely used in ecommerce applications. A recommendation system intends to recommend the items or products to a particular user, based on user's interests, other user's preferences, and their ratings. To provide a better recommendation system, it is necessary to generate associations among products. Since e-commerce and social networking sites generates massive data, traditional data mining approaches perform poorly. Also, the pattern mining algorithm such as the traditional Apriori suffers from high latency in scanning the large database for generating association rules. In this paper we propose a novel pattern mining algorithm called as Frequent Pattern Intersect algorithm (FPIntersect algorithm), which overcomes the drawback of Apriori. The proposed method is validated through simulations, and the results are promising.**

Keywords—Book Recommendation, Opinion Mining, FPInterscet, Apriori

I. INTRODUCTION

E-Commerce related industries have shown a rapid growth in recent years [7]. They depend on online shopping websites to advertise their products and services to the consumers/customers. Since a company's development depends on the opinion and feedback given by the customers, these organization will make use of the customers likes and dislikes for a product. All the product features and their feedback will be stored in a large database. However mining such information from a large database about a product from each customer is a difficult task. Therefore better mining algorithms are needed to mine the user interests for a product, which will be useful for developing a recommendation system. Recommendation system is a part of information filtering system that provides a list of recommendations based on the user interests and user's past behavior. Recommendation systems are very common and are applied in various applications. There are several approaches to generate the recommendations they may be collaborative filtering systems, content-based filtering system, data mining techniques, hybrid approach etc. User profile plays an important role in building a personalized recommendation system [14], where the system accepts the user navigation pattern and classifies them according to their interest and behavior [15]. This paper deals with an online book recommendation system where the users can search for books according to his/her choice. For example Flipkart allows the customers to buy the products depending on the feedback of other customers. The feedback will be in the form of ratings or comments that are stored in the database. The database of such websites will contain huge amount of data. Analysis of such huge amount of data in order to gain useful information is a tough job. In this work, user comments and ratings are extracted from the user reviews and the polarity of the user comments is calculated. Based on the polarity of comments and the product ratings, the system will provide score for each product, list of best products will be generated based on the score thereby making the recommendation process easier.

Generally in an online Book recommendation system [11], there is a problem in showing the relationship among books. When the readers/users browse for the purchase of a particular book, the recommender system will also suggests some other books that have been mined from the information database containing other user's interest patterns. This process of mining the data from the database and generating association relationship among them can be processed by pattern mining algorithms. The most commonly used pattern mining algorithm that generates association rules among item is Apriori [12]. However, if we apply the traditional Apriori algorithm to produce common item sets [6], we have to check the entire transactional record always; this will badly disturbs the productivity of the algorithm as well as the system's performance. Another pattern mining algorithm that generates frequent itemsets is FP-Tree algorithm [13]; however the algorithm requires more storage than Apriori and is more expensive.

To solve the above problem we have introduce a modified version of traditional algorithm so that we can reduce the number of scans in the database. The proposed system deals with the online retrieval of books. The system extracts information from the user reviews and provides scores to each book based on the user feedback and ratings. A transactional database is generated that contains group of books purchased by various users at each transaction. Association among books is shown by applying the proposed algorithm which reduces the time taken for execution by minimizing the number of scans. In contrast with the existing systems, our system can

978-1-5090-0612-0/16/$31.00 ©2016 IEEE

reduce the time taken for online retrieval and recommendations, also generates association rules among the items, which make it very easy to use. It will support the new users to decide which product will be better for them and will also provide a reliable system. The rest of the paper is classified as follows. Section II reviews the related work and compares the existing algorithms. Section III explains about the proposed pattern mining algorithm. Section IV, discusses validation of the proposed algorithm with the simulation results. The work is concluded in Section IV with a note on future work

II. RELATED WORKS

The main task of data mining is to collect information from a data set and convert it into a predictable form. The most commonly used data mining methods are discovering frequent itemsets, frequent sequential patterns, frequent sequential rules and frequent association rules. Association rule mining is a process for discovering interesting relationships among items in a large database. This idea is used for mining the frequent products from a database which helps the user to identify the association between various products and can purchase those products without any confusion. Several works have been done related to association rule mining; some of them are described below. Association rule mining is a common research area in data mining. Avadh Kishor Singh et al. [1] proposed a system for association rules to obtain the frequent k-itemsets. They also used attractiveness count that plays an important role in minimizing the size of database. The process reduces large number of unimportant rules and generates new set of rules with catching count. Osmar R Zaane [3] proposed the use of web mining approaches that recommends on-line learning activities appropriate to beginners' access history to improve the route material navigation. It also improves the on-line learning process. Several attempts have made to solve the problem of information encumber on the Internet. Rana Forsati et al. [4] resolves the web page proposal problem, distributed learning automata had used to study the nature of previous users and group of pages appropriate to learned navigational design. S.Rao et al. [2] developed a new method for obtaining association rules that examines the time, number of database scans, memory usage, and the interest of the rules. They discovered a data mining association algorithm that overcomes the disadvantages of traditional Apriori algorithm and based on number of database scans and time. Due to the repeated scanning of database they introduced association rule mining algorithm which is more suitable to mine patterns when database grows. The problem of finding association rules among products in a large database of sales transactions is considered. R.Srikant [5] solves the above problem by applying a combination of two algorithms. Both the algorithms are entirely different from the known algorithms called AprioriHybrid. Based on the size of the transaction and number of products in the database, the AprioriHybrid has an exceptional scale-up property. In traditional Apriori algorithm, the entire items in the transactional database are given equal priority. Jun Yang et al. [8] proposed an improvement in Apriori algorithm, in which all the items has its own features. By implementing this approach it has been proved that the improved Apriori algorithm based on attributes, works well than traditional one in generating recommendations. Due to the extensive implementation of management system, the information develops continuously. As a result the time cost and trouble of people in searching the suitable information also grows. Pijitra Jomsri [9] introduced a recommendation system for library books depending on user records and applies association rules to generate a model.

From the literature survey, it is observed that there are many frequent pattern mining algorithms to generate recommendations and are used to extract useful information from a database but the most commonly used algorithm to mine frequent item sets and to generate association rules among the items is Apriori Algorithm. But the algorithm suffers from a drawback, as the size of dataset increases, the number of scanning process to extract frequent sets also increases thereby increasing the execution time. This will affect the system's performance. To solve the problem of high latency, we introduce a Frequent Pattern Intersect algorithm (FPIntersect) that works better than the traditional algorithm.

III. PROPOSED FRAMEWORK FOR BOOK RECOMMENDATION SYSTEM

The proposed framework consists of Analysis phase and Synthesis phase which is explained in detail. In our work we generate recommendations for books; the proposed algorithm is applied on a list containing several books. Our approach will provide frequent set of books to the new users based on the opinion of existing users. The entire architecture of our work is depicted in Figure 1.

Initially search for a book is made from an online shopping website by giving query Q= <keywords> containing features of the book. The user comments and ratings for each book are extracted from the customer reviews based on the query given. The customer reviews are extracted from an online shopping website. We have used a software package called "beautiful soup" to extract the user comments and ratings for any particular product.

A. Opinion Mining

Opinion mining [10] also known as Sentiment mining, is a method of natural language processing for understanding the interest of a person about a product. This approach of mining opinions from public mostly applies machine learning, artificial intelligence techniques to mine text or phrases for sentiment analysis. The aim of Opinion Mining is to extract opinions from user feedback on products and to present the information in an effective way. Customers usually express their views in the form of review sentences with either single words or phrases. So we need to identify those opinion words

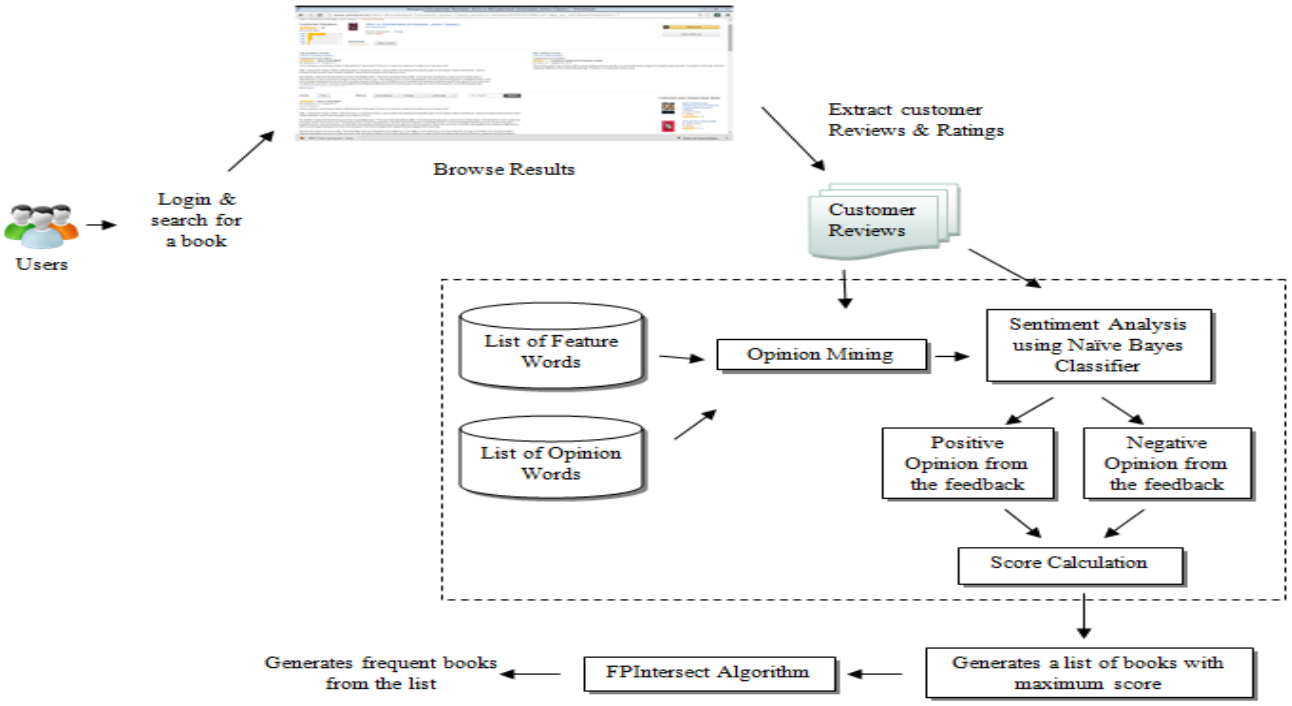

Fig. 1. Proposed framework for Recommendation System

to generate summary. Consider the following review sentence for a book.

”The book was adventurous and had a nice story”
”The Jungle Book is a great tale of friendship and adventure”

In the above sentences, the feature is “adventurous” and opinion word is “nice”. POS tagging is important technique to parse each sentence and produce the part-of-speech tag of each word (whether the word is a noun, adjective, verb, adverb, etc.). Example: The—DT Jungle—NNP Book—NNP is—VBZ a—DT great—JJ tale—NN of—IN friendship—NN and—CC adventure—NN.

After POS tagging is done, we need to extract features that are nouns or phrases of noun. The most commonly occurring words (stop words) such as is, was, to etc. are removed after the POS tagging. This method will help to identify the features and opinions in a better way. Based on the features and opinions of a product, the review comments are grouped into positive and negative classes. More the number of positive opinions, greater will be the demand of the product. This measure of estimating the positiveness and negativeness of user comments is defined as its polarity. This information could be considered to generate recommendations based on the user’s interests towards a product.

B. Analysis of Reviews

Next step is the review extraction process where the user id, user comments and ratings for a product is considered from the user reviews. All details corresponding to a product is stored in a document to perform analysis and retrieves relevant information about user’s interest. Once the user comments are extracted, the comments are classified into either ’positive’ or ’negative’ sentences, this could be considered as an opinion of each user regarding a particular product. A feedback with ’positive’ opinion indicates that the user has liked the book. But, a feedback with ’negative’ opinion represents that the user have a bad experience with the product. This is a process of text classification and there are several approaches for performing opinion mining and sentiment analysis, we have applied a Naive Bayes text classifier that chooses a machine learning approach to classify the documents/text as positive and negative groups.

Naive Bayes Text Classifier

Naive Bayes classifier is used to classify each feedback or review comments as positive and negative. The classifier uses a “Bag of words” which contains adjectives like nice, awesome, bad etc. The adjective words describe the opinion of different users for any particular book or item. The classifier is also used to determine the polarity of review comments as well as the polarity of opinion words. Polarity (Pi) is calculated for user comments, shows how much positive/negative a sentence/text is.

C. Score Calculation

The score for a book is calculated by considering two parameters, user comments and ratings. In order to give a valid score to a book, the user ratings are taken along with the comments. Since user ratings alone may not give the actual opinion of users towards a book and may not be accurate

hence we are also considering the user comments by taking their polarity and by giving more weightage to them and less for user ratings. Average of ratings (Ri) is taken and added along with the calculated polarity of user comments for each book. Based on the score, a list is generated containing books with maximum score along with userIDs who have brought those books. The formula for product score calculation is given as

$$\sum (0.8Pi + 0.2Ri) = \delta i \quad (1)$$

Where Pi is the polarity of user comments and Ri is average of all ratings.

D. FPIntersect algorithm

The database utilized to perform the improved algorithm is a list with 1000 records where each record is a combination of userIDs and list of books having maximum score. The generated list is given as the input; the algorithm first scans the list based on a minimum support (Min Support) which is considered as the threshold value (α). Minimum support shows the minimum number of books being purchased by users or users with userIDs. Based on the support count, candidate itemsets are generated C_k (Table II) and the records with less minimum support are deleted from the list. During the first iteration, the algorithm scans the items in C_k for generating 1-frequent itemsets F_1. From second iteration, the algorithm scans only F_1 for generating 2-frequent itemsets F_2 (Table III and Table IV) by taking the intersection of all userIDs (considering userIDs that are common in the combination of two different books) and removes those userIDs from the list that doesn't satisfy the min support, thus scanning is reduced from n to n-k transactions. Therefore scanning of the entire list is avoided, which reduces the time for frequent itemsets generation. Finally the algorithm generates associations for the combination of frequent itemsets that satisfies the support count which is taken as the output. The above mentioned algorithm is described in algorithm 1.

Algorithm 1: FPIntersect algorithm
Input: Transactional list;
Min_Support: Threshold value
Output: Frequent items/products

Description:
1: Generate items, Min_Support, userIDs
2: For (j: =2; F_{j-1}! = ϕ; j++)
3: Generate C_k from F_{j-1}
4: C_k = Frequent itemsets from F_{j-1}; //j represents the pass number//
5: x = Intersection of userIDs that are common in each item C_k
6: For each value in dict
7: Increment the counter
8: F_j: = items in C_k with minimum support;
9: **Answer** = U_kF_j; //all frequent itemsets satisfying Min_Support//

IV. PERFORMANCE EVALUATION

The performance of our proposed algorithm is analyzed through experiments. We have used TextBlob which is a Python library for processing textual data. It provides a simple API for common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification etc. The textblob directly calls naive bayes classifier for sentence classification and calculates the polarity of user comments using sentiment analysis. Based on the polarity and average of all ratings for a particular product, a list is created with userIDs and list of books with maximum score. The proposed algorithm is applied to the list for generating associations among frequent products. We have developed a python based simulation framework for implementing our algorithm.

A. Dataset

To construct our simulation setup we have extracted the user reviews and ratings for various books from Amazon which is an online shopping using a package called "beautiful soup". Amazon is a famous and commonly used e-commerce site that has more than 22 million visitors and around 70 million users to express their interests and opinions for more than 14 million different books. The quality of a book depends on the comments and ratings of different users. Based on the extracted reviews a list is constructed that consists of more than 1000 records of different books, where each record defines the user IDs and combination of books that they have purchased. Each book in the list is recorded based on the calculated score by considering the user comments and their ratings. We have arranged the records in the list by showing different combination of books purchased by each user with their userIDs in each transaction.

V. RESULTS

Experimental results for analysing the performance of two algorithms are shown in Table I. We compare our proposed algorithm with traditional Apriori with respect to number of scans to complete the process, for transactions n = 9 to n = 1000. We observe that the FPIntersect takes less time for scanning the entire database compared to the traditional Apriori. The result is illustrated using a graph as shown in Figure 2. Both the approach works well when the transactions are less in number but as the data size increases we can see that the traditional approach takes high latency in scanning the large database for generating association rules than the FPIntersect.

Discussion

In this section we discuss the effectiveness of our proposed algorithm with the help of an example. The proposed algorithm is applied on a list containing UserIDs and Books, a

set of transactions along with the purchased products, shown in Table II. We have arranged the records in the list by showing different combination of books purchased by each user with their userIDs in each transaction. Next step is to generate 1-candidate itemsets with each book showing its support count and the book with less minimum support is deleted, shown in Table III. Table IV shows the removal of userIDs along with the books that do not satisfy the minimum support in a particular transaction.

TABLE I. PERFORMANCE OF ARPIORI & FPINTERSECT ALGORITHMS

No.of Transactions	Scans in Apriori	Scans in FPIntersect
9 transaction	40	38
50 transaction	836	789
100 transaction	967	855
500 transaction	4927	4304
1000 transaction	9782	6070

TABLE II. TRANSACTIONS

UserIDs	Books
1	B1,B2,B5
2	B2,B4
3	B2,B4
4	B1,B2,B4
5	B1,B3
6	B2,B3
7	B1,B3
8	B1,B2,B3,B5
9	B1,B2,B3

TABLE III. 1-CANDIDATE GENERATION

Books	Support	
B1	6	
B2	7	
B3	5	
B4	3	
B5	2	deleted

TABLE IV. 1-FREQUENT ITEMSETS GENERATION

Books	Support	UserIDs	
B1	6	1,4,5,7,8,9	
B2	7	1,2,3,4,6,8,9	
B3	5	5,6,7,8,9	
B4	3	2,3,4	
B5	2	1,8	Deleted

TABLE V. 2-FREQUENT ITEMSETS GENERATION

Books	Support	UserIDs	
B1,B2	4	1,4,8,9	
B1,B3	4	5,7,8,9	
B1,B4	1	4	Deleted
B2,B3	3	6,8,9	
B2,B4	3	2,3,4	
B3,B4	0	0	Deleted

The Table V shows the generation of frequent 2-itemset and the books that do not satisfy the support count is deleted along with their userIDs. In this table, the intersection of UserIDs for each combination of books is considered. This method reduces the entire scanning of database as in Apriori algorithm i.e., instead of scanning the entire 9 transactions, the improved algorithm will reduce it to 6 transactions. From the given example we observe that the traditional Apriori scans the entire database repeatedly to generate frequent n-itemsets but the new FPIntersect method reduces that to n-k transactions. The algorithm continues until it generates frequent itemsets that satisfies minimum support and thereby generating association rules. Our proposed algorithm is illustrated in Algorithm 1.

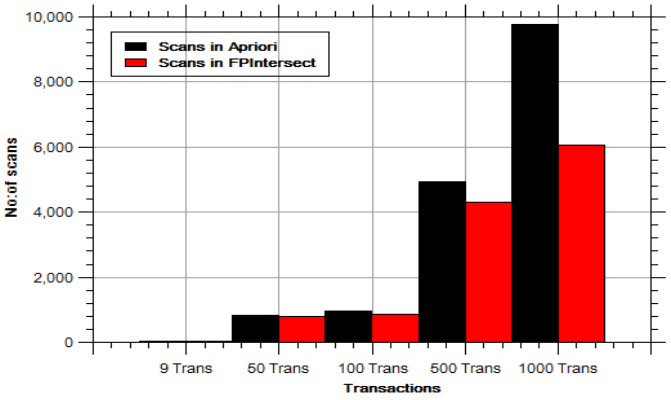

Fig. 2. Performance of Apriori and FPIntersect algorithms

VI. CONCLUSION

This paper has proposed a novel framework for recommendation system by utilizing an FPIntersect algorithm. The system uses the information obtained after analyzing the opinions of each user from the user comments. The user ratings and their comments are extracted from the user reviews. Score is calculated by considering the polarity of user comments and average of ratings, each book is included in the record depending on the score for recommendation. The proposed system overcomes the drawback of Apriori by reducing the number of scans and generates association rules. The work could be enhanced in future by considering some concept generation approaches rather than dealing with keywords to provide recommendations.

REFERENCES

[1] A. K. Singh, A. Kumar, and A. K. Maurya, "Association rule mining for web usage data to improve websites," in Advances in Engineering and Technology Research (ICAETR), 2014 International Conference on. IEEE, 2014, pp. 1–6.

[2] S. Rao and P. Gupta, "Implementing improved algorithm over apriori data mining association rule algorithm 1," 2012.

[3] O. R. Za´ıane, "Building a recommender agent for e-learning systems," in Computers in Education, 2002. Proceedings. International Conference on. IEEE, 2002, pp. 55–59.

[4] R. Forsati, M. R. Meybodi, and A. Rahbar, "An efficient algorithm for web recommendation systems," in 2009 IEEE/ACS International Conference on Computer Systems and Applications. IEEE, 2009, pp. 579–586.

[5] R. Agrawal, R. Srikant et al., "Fast algorithms for mining association rules," in Proc. 20th int. conf. very large data bases, VLDB, vol. 1215, 1994, pp. 487–499.

[6] M. Shweta and D. K. Garg, "Mining efficient association rules through apriori algorithm using attributes and comparative analysis of various association rule algorithms," International Journal of Advanced Research in Computer Science and Software Engineering, vol. 3, no. 6, pp. 306–312, 2013.

[7] P. Nagarnaik and A. Thomas, "Survey on recommendation system methods," in Electronics and Communication Systems (ICECS), 2015 2nd International Conference on. IEEE, 2015, pp. 1496–1501.

[8] K. Hong, H. Jeon, and C. Jeon, "Userprofile-based personalized research paper recommendation system," in Computing and Networking Technology (ICCNT), 2012 8th International Conference on. IEEE, 2012, pp. 134–138.

[9] J. Yang, Z. Li, W. Xiang, and L. Xiao, "An improved apriori algorithm based on features," in Computational Intelligence and Security (CIS), 2013 9th International Conference on. IEEE, 2013, pp. 125–128.

[10] Y. W. Lo and V. Potdar, "A review of opinion mining and sentiment classification framework in social networks," in 2009 3rd IEEE International Conference on Digital Ecosystems and Technologies. Ieee, 2009, pp. 396–401.

[11] P. Jomsri, "Book recommendation system for digital library based on user profiles by using association rule," in Innovative Computing Technology (INTECH), 2014 Fourth International Conference on. IEEE, 2014, pp. 130–134.

[12] M. Al-Maolegi and B. Arkok, "An improved apriori algorithm for association rules," arXiv preprint arXiv:1403.3948, 2014.

[13] F. Coenen, P. Leng, and S. Ahmed, "Data structure for association rule mining: T-trees and p-trees," IEEE Transactions on Knowledge and Data Engineering, vol. 16, no. 6, pp. 774–778, 2004.

[14] P. Ramya and G. Sajeev, "Building web personalization system with time-driven web usage mining," in Proceedings of the Third International Symposium on Women in Computing and Informatics. ACM, 2015, pp. 38–43.

[15] G. Sajeev and P. Ramya, "Effective web personalization system based on time and semantic relatedness," in Advances in Computing, Communications and Informatics (ICACCI), 2016 International Conference on. IEEE, 2016, pp. 1390–1396.