A NOVEL ASSOCIATION RULE MINING AND CLUSTERING BASED HYBRID METHOD FOR MUSIC RECOMMENDATION SYSTEM

M. Sunitha Reddy[1], T. Adilakshmi[2], V. Swathi[3]

[1]*Dept. of CSE, Vasavi College of Engineering Ibrahimbagh, Hyderabad-31, AP, India*
[2]*Dept. of CSE, Vasavi College of Engineering Ibrahimbagh, Hyderabad-31, AP, India*
[3]*Dept. of CSE, Vasavi College of Engineering Ibrahimbagh, Hyderabad-31, AP, India*

Abstract

Recommender systems have been proven to be valuable means for web online users to cope with the information overload and have become one of the most powerful and popular tools in electronic commerce. In this paper we have proposed a novel algorithm to recommend items to users based on an hybrid method. First we use clustering to form the user clusters based on the similarity of users. We have taken users listening history for similarity calculation. Second we are going to find the items which are strongly associated with each other by using association rule mining. Finally we will be using these strong association rules in recommendation of items.

Index Terms- *Recommender system, clustering, Association Rule Mining, hybrid method*

----------***----------

1. INTRODUCTION

Recommendation systems get the information from the user's listening history and recommend the items which will be interesting for them [2] i.e which fits their listening pattern. Recommendation systems can be broadly divided into two categories Collaborative Filtering methods and Content based methods. Collaborative Filtering (CF) methods are based on the user item matrix. They can be viewed as item-based CF techniques and user-based CF techniques. Content based recommendation system uses content of the items for recommendation.

The major challenges faced by recommender systems are Data Sparsity, Cold-Start Problem [3] and Non-transitive Association.

Data Sparsity means the lack availability of user ratings for items. The number of items will be much more when compared to number of users. Each user will rate only very few items.

Cold-Start Problem means unable to recommend new items or new users. As the new item is added, traditional CF methods cannot provide any recommendations as the model has not learnt enough knowledge about the new item or new user.

Non-transitive association indicates the inability of finding association among the items which are not rated by the same users.

In this paper we have proposed a novel hybrid algorithm for recommendation based on clustering and association rule mining. Clustering is used to form the user clusters based on the similarity. Once the similar users form a cluster we use these clusters to find items strongly associated with other. This information is used while recommending items to new test users.

The remaining paper is organized as follows. Section 2 discuss about the basics of Association rule mining in recommendation system. Section 3 describes about the proposed system. Section 4 briefs about conclusion and future scope. References are mentioned in section 5.

2. RELATED WORK

Any recommendation system involve 3 phases[1]: data preparation and transformation, pattern discovery, and recommendation. First two phases can be performed offline whereas the third phase has to be performed online. The pattern discovery phase may include the discovery of association rules, sequential navigational patterns, clustering of users or sessions, and clustering of page views or products [4]. The recommendation system considers the active user session in conjunction with the discovered patterns to provide personalized content. The personalized content can take the form of recommended links or products, or targeted advertisements tailored to the user's perceived preferences as determined by the matching usage patterns [7].

2.1 Data Preparation & Pattern Discovery

The first step in recommendation system is data preparation. In this step user data is transformed into transactional database.

Second step is pattern discovery from the transactional databases. Association rule mining is used to identify the relationship between users and items. Given a set of transactions, where each transaction is a set of items, an association rule is a rule of the form X ⇒ Y, where X and Y are sets of items. The meaning of this rule is that the presence of X in a transaction implies the presence of Y in the same transaction. X and Y are respectively called the body and the head of the rule. Each rule has two measures: confidence and support. The confidence of the rule is the percentage of transactions that contain Y among transactions that contain X; The support of the rule is the percentage of transactions that contain both X and Y among all transactions in the input data set. In other words, the confidence of a rule measures the degree of the correlation between itemsets, while the support of a rule measures the significance of the correlation between itemsets.

To consider an example, assume we have a database of transactions as listed in Table 2.1, for association rule "{A} ⇒ {C}", the confidence of the rule is 66%, and the support of the rule is 50%.

Table 2.1: Sample Transactions

Transaction Id	Purchased Items
1	{A, B, C}
2	{A, D}
3	{A, C}
4	{B, E,}

There could be any number of items present in the body and in the head of a rule. A user could also specify some rule constraints, for example, he/she might only be interested in finding rules containing certain items.

The traditional association rule mining problem definition is: given a set of transactions, where each transaction is a set of items, and a user-specified minimum support and minimum confidence, the problem of mining association rules is to find all association rules that are above the user-specified minimum support and minimum confidence [1] . We call a set of items an itemset. The support of an itemset is the percentage of transactions that contain this itemset among all transactions. An itemset is frequent if its support is greater than the user-specified minimum support. The problem of discovering association rules could be decomposed into two subproblems:

2.1.1 Find all frequent itemsets.
2.1.2 Generate association rules from frequent itemset

For example, if {a, b, c, d} and {a, b} are frequent itemsets, then compute the ratio:

Confidence =support{a,b,c,d}/support{a,b}

If confidence is not less than the user-specified minimum confidence, then "{a,b} ⇒ {c, d}" is one desired association rule. This rule satisfies the minimum support constraint because {a, b, c, d} is a frequent itemset.

Apriori Algorithm is the most commonly used algorithm for this purpose. The Apriori algorithm generates frequent itemsets by making multiple passes over the transaction data. We use k-itemsets to denote itemsets of size k. The first pass finds the frequent 1-itemsets. For pass k > 1, it generates the candidate frequent k-itemsets using the frequent (k-1)-itemsets; then it scans all transactions to count the actual supports of the candidate k-itemsets; at the end of pass k, it collects the candidate k-itemsets whose supports are above the minimum support as the frequent k-itemsets.

The Apriori algorithm consists of two steps
1. Join Step: Candidate k-itemsets are generated by performing join operations on frequent (k-1)-itemsets. In the join step, two different frequent (k-1)-itemsets which share the first k-2 items are joined together to generate a candidate frequent k-itemset.

2. Prune Step: In the prune step, we delete all candidate k-itemsets which have non-frequent (k-1)-subset.

These two steps are correct due to the fact that any subset of a frequent itemset must also be frequent.

A transaction data set can be seen as a relational database table with Boolean valued attributes that correspond to items and records that correspond to transactions. The value of an attribute for a given record is "1" if the corresponding item is present in the corresponding transaction, "0" otherwise.

Table 2.2: Sample Transactions with Boolean Valued Attributes

Transaction ID	A	B	C	D	E
1	1	1	1	1	0
2	1	1	0	1	1
3	0	0	0	1	1
4	0	1	0	0	1

Transactional database is used to find frequent itemsets which satisfy minimum support threshold. These frequent itemsets are used in finding strong association rules. The association

rules which satisfy the minimum confidence threshold are called as strong association rules.

For example {a,b,c,d} is a frequent itemset , we can form strong association rule such as {a}-> {b,c,d} if it satisfy the minimum confidence threshold.

2.2 Recommendation Phase

This phase uses the strong association rules to perform recommendations. For example {b,c,d}-> {a} is a strong association rule and the user has already listened to items b,c,d then according to the above rule the item a is also recommended to the user.

Algorithm agg_hierarchical_clustering()
Input: User-Item Matrix
Output: User Clusters
Method:
begin

1. Consider each user vector I_1, I_2,..I_k where k is the number of distinct items rated by all users
2. Initialize threshold_cutoff value
3. Consider the first user and put in C_1
4. For all remaining users repeat the steps from 4 to 8
5. Find the similarity of the $user_i$ with all the clusters formed
6. Put the $user_i$ in the cluster with more similarity
7. If the $user_i$ is not in the threshold value of anu cluster
8. Create a new cluster

end

Fig 1 Pseudocode for agglomerative hierarchical clustering

Algorithm Extended FP-tree()
Input: User clusters
Output: recommendations to a new unknown user
Method:
begin

1. consider the user clusters C_1, C_2,….C_m as input
2. for each test user repeat the steps through 3 to 8
3. map the test user to the user cluster C_i to which he/she is most similar
4. use the cluster C_i to form transactional database
5. generate frequent itemsets by using gen-frequent-itemset()
6. use extended FP-tree to represent frequent itemsets
7. traverse the extended FP-tree for the test user in DFS order
8. recommend the items on the path which the test user has not already listened

end

Fig 2 Pseudocode for extended FP-tree recommendation Algorithm

3. PROPOSED ALGORITHM

This section describes the algorithm proposed in Fig 2.Input to this algorithm is user clusters formed based collaborative filtering. First step is to convert the mapped cluster to transactional database. Second step is to find frequent itemsets from the transactional database. Next strong association rules are mined to form the items which are strongly associated with each other. Finally these associations among items are used in recommendation.

3.1 User Based Clusters

In this step we are going to use collaborative filtering technique to form the user clusters. Users implicit feedback is taken into consideration to form user-item matrix which is the basis of CF method. Each user is represented as a vector of items (i_1,i_2,……i_k) where k is the number of items. We use cosine similarity to form clusters by using threshold based Agglomerative hierarchical clustering technique as shown in Fig1.

We initially start with first user in the first cluster and then from second user onwards put the user in the same cluster if he/she is within the threshold value. Otherwise create a new cluster. This process will be repeated for all remaining users to form clusters.

TID	Items	TID	Items
1	a,b	11	a,b
2	a,b,e	12	a,c,f
3	c,d	13	a,b,e
4	e,f	14	b,e,f,g
5	c,d	15	c,d
6	a,c	16	a,b
7	a,b	17	c,d
8	e,f	18	a,c
9	c,d,g	19	a,b
10	a,b	20	c,d,f

Fig 3 Transaction database for 20 users for 5 items

3.2 Forming Transactional Database for each User Cluster

Each cluster contains a set of users who are similar to each other and dissimilar to the users in the other clusters. In this step we consider each cluster and form transactional database for that cluster.

For example cluster C_1 contains the users u_1, u_2,……u_m. We consider each user and the items which have been rated by them to form the one record in transactional database as given in Fig 3.

Example 1: Consider the transactional database of 20 transactions shown in Fig 2. The set of items

I = {a; b; c; d; e; f; g}. The set of items `a' and `b',

i.e., {a; b} is an itemset. It is a 2-itemset. For simplicity, we write this itemset as “ab". It occurs in

8 transactions (tids of 1;2;7;10;11;13;16 and 19)

Therefore, the support count of ”ab," i.e., f(ab) = 8

The support of ab is 0.4. if the user specified minimum support is 0.3 then itemset ab is a 2-frequent item as it satisfies minimum support.

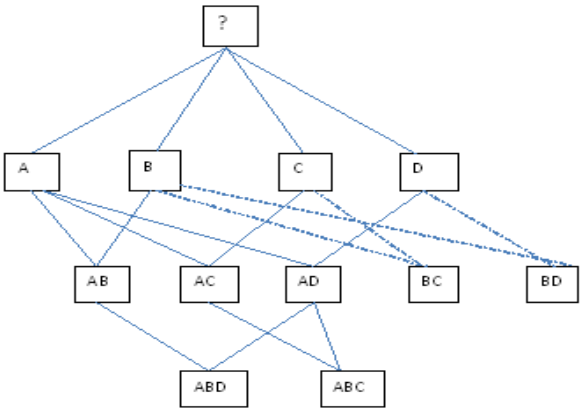

Fig 4 Extended FP-tree

3.3 Using extended fp-tree to form frequent items

Transactional database is used to find frequent itemsets. We proposed an algorithm Extended FP-tree() to find frequent itemsets. The first step in this algorithm is to scan the transactional database to find frequent 1-itemset. Next we use frequent 1-itemset to generate 2-frequent itemset by using the prefixes. Suppose if {a} if infrequent itemset then there is no need to find the 2-frequent itemsets starting with {a}. This method is continued till we find the maximal frequent itemset. In Fig1. we can see that when the itemsets “bc” and “bd” are infrequent they are not further explored to find 3- frequent itemsets. This reduces the number of candidate itemsets as compared to Aproiri algorithm.

3.4 Use Frequent Itemsets to Form Association Rules and Recommend the Items

Frequent itemsets are stored in the form of extended FP-tree. From the frequent itemsets we are going to form strong association rules by traversing the tree when it is required. i.e association rules are formed only when they are required.

Example 2: let the frequent itemsets discovered from the transactional databases of user be {{A : 10};{B : 20};{C : 30};{D : 35};{AB : 5};{AC : 3};{AD : 5};{ABC : 3};{ACD : 2}} . Consider the user who has listened the items {CD}, then recommend item {A} to the user by traversing the extended FP-tree shown in Fig 1. in the DFS order.

	Actual – True	Actual- False
Predicted- True	True Positives (TP)	False Positives (FP)
Predicted- False	False Negatives (FN)	True Negatives (TN)

Fig 5 Confusion Matrix

3.5 Evaluating the Performance of Recommender System

Recommendation systems are evaluated by using the measures like Precision, Recall and F-measure which is combination of Precision and Recall.

Precision and Recall [6] are the most commonly used measures in information retrieval systems. Precision indicates accuracy of recommendation system and recall measures the extent to which recommender system can recommend the items which are interesting to the user.

Precision and Recall are defined by using the confusion matrix shown in the Fig2.

Precision is a measure of exactness, determines the fraction of relevant items retrieved out of all items retrieved. Recall is a measure of completeness, determines the fraction of relevant items retrieved out of all relevant items. F-measure is the measure which stabilizes the changes in Precision and Recall.

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F - measure = 2 * \frac{(Precision * Recall)}{(Precision + Recall)}$$

This experiment we are going to implement on the user log dataset obtained from Last.fm which is a popular internet radio. Last.fm [8] allows users to listen the songs based on the

artist, album or title. Each user activity is maintained in a user log file. The fields in each record of log file are given below.

User_000004 2009-04-09T12:49:50Z 078a9376-3c04-4280-b7d7-b20e158f345d A Perfect Circle 5ca13249-26da-47bd-bba7-80c2efebe9cd People Are People

Fig 6 User Record tuple in the dataset

User id (User_000004) – Since the data is captured anonymously, we assigned each user, a user-id of the format user_000004.

Date–Time (2009-04-09T12:49:50Z) – Time of activity is recorded which will be used in our algorithm to determine the session in which it will belong.

Album Id (078a9376-3c04-4280-b7d720e158f345d) – A unique identifier is Attributed to each Album.

Album name (A Perfect Circle) – An album to which that song belongs to.

Trackid (5ca13249-26da-47bd-bba7-80c2efebe9cd) – A unique identifier is attributed to each track / song.

Track name (People are People) – The songs which the user listened to.

We are going to consider the history of 50 users over a period of 3 yrs into consideration to form user clusters. Before forming the clusters the first step in any Data mining task is pre-processing the data. After filtering the data user-item matrix is obtained from the cleaned data.

User-item matrix is the basis for any collaborative clustering method. Threshold based hierarchical clustering is used to form user clusters.

Association rule mining is applied on the user clusters to improve the performance of recommendation system. Association rules are used while performing the recommendations. The items are recommended to a user only when they are strongly associated with the items which the user has already listened.

4. CONCLUSIONS

In this paper we have proposed a novel algorithm for recommendation system which will form the user clusters based on the similarity as the first step. Second step is to use these user clusters to form transactional database for a user. From the transactional databases strong association rules are determined by using frequent itemsets.

Hybrid recommendation system which combines clustering and association rule mining can be used to address one of the most challenging issue of recommendation systems- Cold-start Problem.

REFERENCES

[1]. Agrawal, R., Imielinski, T., Swami, A. Data Mining: A performance Perspective. IEEE Trans. Knowledge and Data Engineering, vol, 5,6, 1993a, pp. 914-925.

[2]. A.B. Devale, Dr. R. V. Kulkarni Applications of Data Mining Techniques in Life Insurance. International Journal of Data Mining and Knowledge Management Process Vol.2, No.4 July 2012.

[3]. Maria N. Moreno, Saddys Segrera, Vivian F Lopez, Maria Dolores Munoz and Angel Luis Sanchez, Mining Semantic Data for Solving First-rater and Cold-start Problems in Recommender system ACM IDEAS 11 2011, September 21-23

[4]. Lee, CH., Kim, Y.H., Rhee, P.K. 2001. Web personalization expert with combining collaborative filtering and association rule Mining Technique. Expert System with Applications 21. 131-137

[5]. Sarwar, B., Karypis, G., Konstan, J., Riedl, J. 2001. Item-based Collaborative Filtering Recommendation Algorithm. Proceedings of the tenth International World Wide Web Conference, 285-295

[6]. Adomavicius, G. & Tuzhilin, A. (2005), Toward the next generation recommender systems : a survey of the state of the art and possible extensions, IEEE transactions on knowledge discovery and data mining, 17(6),734-749.

[7]. Schafer, J.B., Konstant, J.A. and Riedl, J. 2001. E-Commerce Recommendation Applications. Data Mining and knowledge Discovery, 5, 115-153.

[8]. Last.fm