Volume 5, Issue 10, October-2015 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Recommender Systems for e-commerce Application using Collaborative Filtering Approach

[1]Mohammad Naveed Farooqui, [2]Shaik Mulla Almas

[1](M.Tech) –CSE, Vasireddy Venkatadri Institute of Technology (VVIT), Andhra Pradesh, India

[2]Assistant Professor, Dept of IT, Vasireddy Venkatadri Institute of Technology (VVIT), Andhra Pradesh, India

Abstract: Collaborative filtering (CF) is an important and popular technology for recommender systems. 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. However, current CF methods suffer from such problems as data sparsity, recommendation inaccuracy and big-error in predictions with lack of user privacy. There are three common approaches to solving the recommendation problem: traditional collaborative filtering, cluster models, and search-based methods and a novel algorithm to recommend items to users based on a hybrid method. First we use clustering to form the user clusters based on the similarity of users. We have taken users watching 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. In ordered to provide the security we used onion routing algorithms.

Keywords: Collaborative filtering, data sparsity, association rule mining, onion routing algorithm.

I. INTRODUCTION

Recommendation systems found their application in the field of e-commerce and internet where items suggest to a group of user on the basis of their requirement based on their area of interest. A recommendation system is an information filtering system that built a model from the characteristic of an item according to the rating or prediction, given by a user to an item. Recommendation system has an important component in social media sites (such as Amazon, IMDB, Movie Lens), social sites giants such as Amazon have been greatly gained from the capability of their recommenders in accurately delivering the correct item to the correct user [17]. Collaborative filtering (CF) is an important and popular technology for recommender system. CF methods are classified into user based CF and item-based CF. The basic idea of user-based CF approach is to find out a set of users who have similar favor patterns or interest to a given user and the basic idea of item-based CF approach is to find out a set of items having highest correlation with the given item. In reality, people may like to group items into categories, and for each category there is a corresponding group of people who like items in the category [18]. Cognitive psychologists find that objects (items) have different typicality degrees in categories in real life [19], [20], [21]. But these collaborative filtering methods have facing some problems.

II. SYSTEM ANALYSIS

Collaborative filtering (CF) is an important and popular technology for recommender systems. However, current CF methods suffer from such problems as

- Data Sparsity.
- Recommendation accuracy
- Scalability
- Security

Data Sparsity: The data sparsity problem is the problem of having too few ratings and hence it is difficult to find out correlations between users and items. (1) It occurs when the available data are insufficient for identifying similar users or items. It is a major issue that limits the quality of CF recommendations. (2)

Recommendation accuracy: People require recommender systems to predict users‟ preferences or ratings as accurately as possible. An example of a recommender system is flipkart.com, a site where users can enter a title of a recent book they have read and enjoyed to see recommended books that they are likely to also enjoy. One approach to the design of recommender systems that has seen wide use is collaborative filtering. Collaborative filtering techniques are based on gathering and analyzing a huge amount of information on users‟ behaviors, activities or preferences and getting predicted on what users will like based on their similarity to other users. A main advantage of the collaborative filtering approach is that it does not dependent on machine analyzable content and therefore it is capable of accurately recommending lots of complex items.

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.

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%.

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

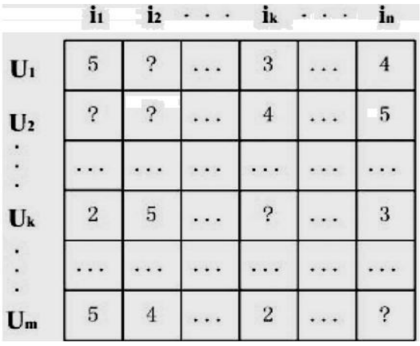
There could be any number of items present in the body and in the head of a rule. A user could also specify some rule constraints, for example, he/she might only be interested in finding rules containing certain items. The traditional association rule mining problem definition is: given a set of transactions, 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 and itemset. The support of an itemsets is the percentage of transactions that contain this itemsets among all transactions. An itemsets is frequent if its support is greater than the user-specified minimum support. The problem of discovering association rules could be decomposed into two sub problems.

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
```

III. EXISTING SYSTEM

In existing approach the user's preferences at low level is only captured which leads to inaccurate results. Difficulty to find correlations between users and items when very few ratings are given and it limits the quality of collaborative filtering recommendations. User based and item based collaborative filtering is not accurate to pose on the available data. Item and user groups are not correlated which makes inaccurate data recommends for users.



	i_1	i_2	. . .	i_k	. . .	i_n
U_1	5	?	. . .	3	. . .	4
U_2	?	?	. . .	4	. . .	5
.	. . .	. . .	. . .	. . .	. . .	. . .
U_k	2	5	. . .	?	. . .	3
.	. . .	. . .	. . .	. . .	. . .	. . .
U_m	5	4	. . .	2	. . .	?

Fig1: user rating matrix in traditional Collaborative Filtering.

Disadvantages of existing system:

- It is difficult to find out correlations between users and items.
- It occurs when the available data are insufficient for identifying similar users or items.

IV. PROPOSED SYSTEM

In this paper we have collaborative filtering using clustering. At first all items are grouped as several groups, next we form a user group corresponding to each item group, at last we build user typicality matrix and measure users similarities based on users. The neighbor's selection by measuring user's similarity based on user typicality in user groups can be done by using the collaborative filtering recommendation. Proposed system reduces the number of big error predictions, improves accuracy of predictions and works with sparse training data sets.

TYPICALITY BASED COLLABORATIVEFILTERING

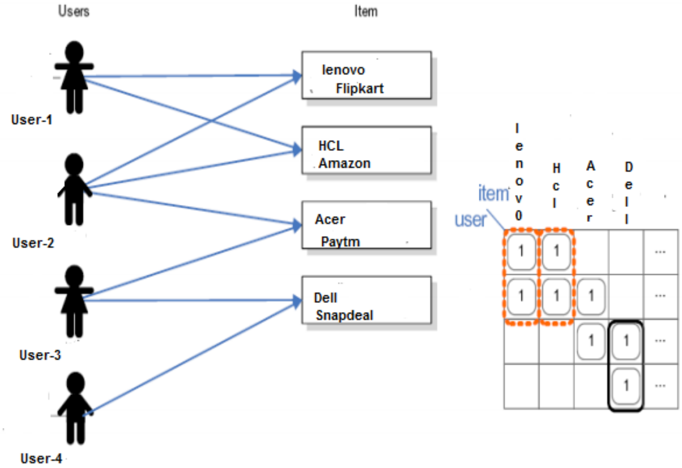


Fig2: The relations among users, user groups and item groups.

There are a set U of users, and a set O of items. Items can be clustered into several item groups and an item group is intuitively a set of similar items. For example, movies can be clustered into action movies, war movies, and so on. Each movie belongs to different movie groups to different degrees. The choice of clustering method is application domain dependent.

V. RECOMMENDER SYSTEMS

There have been many works on recommender systems and most of these works focus on developing new methods of recommending items to user's .The objective of recommender systems is to assist users to find out items which they would be interested in. Items canbe of any type, such as products like basic phones, android phones, windows phone. Currently, recommendation methods are mainly classified into collaborative filtering (CF), content based (CB), and hybrid methods. For the reason that we are focusing on proposing a new CF method, we will introduce the related works about CF methods in more details.

Recommendation Algorithms

Most recommendation algorithms start by finding a set of customers whose purchased and rated items overlap the user's purchased and rated items.2 The algorithm aggregates items from these similar customers, eliminates items the user has already purchased or rated, and recommends the remaining items to the user. Two popular versions of these algorithms are collaborative filtering and cluster models. Other algorithms including search-based methods and our own item-to-item collaborative filtering focus on finding similar items, not similar customers. For each of the user's purchased and rated items, the algorithm attempts to find similar items. It then aggregates the similar items and recommends them.

Traditional Collaborative filtering

A traditional collaborative filtering algorithm represents a customer as an N-dimensional vector of items, where N is the number of distinct catalog items. The components of the vector are positive for purchased or positively rated items and negative for negatively rated items.

The algorithm generates recommendations based on a few customers who are most similar to the user. It can measure the similarity of two customers, A and B, in various ways; a common method is to measure the cosine of the angle between the two vectors:

$$similarity(\vec{A},\vec{B}) = \cos(\vec{A},\vec{B}) = \frac{\vec{A} \bullet \vec{B}}{\|\vec{A}\| * \|\vec{B}\|}$$

The algorithm can select recommendations from the similar customers' items using various methods as well, a common technique is to rank each item according to how many similar customers purchased it. Using collaborative filtering to generate recommendations is computationally expensive. It is O(MN) in the worst case, where M is the number of customers and N is the number of product catalog items, since it examines M customers and up to N items for each customer. However, because the average customer vector is extremely sparse, the algorithm's performance tends to be closer to O(M + N). Scanning every customer is approximately O(M), not O(MN), because almost all customer vectors contain a small number of items, regardless of the size of the catalog. But there are a few customers who have purchased or rated a significant percentage of the catalog, requiring O(N) processing time. Thus, the final performance of the algorithm is approximately O(M + N). Even so, for very large data sets such as 10 million or more customers and 1 million or more catalog items the algorithm encounters severe performance and scaling issues.

It is possible to partially address these scaling issues by reducing the data size. We can reduce M by randomly sampling the customers or discarding customers with few purchases, and reduce N by discarding very popular or unpopular items. It is also possible to reduce the number of items examined by a small, constant factor by partitioning the item space based on product category or subject classification. Dimensionality reduction techniques such as clustering and principal component analysis can reduce M or N by a large factor.

Unfortunately, all these methods also reduce recommendation quality in several ways. First, if the algorithm examines only a small customer sample, the selected customers will be less similar to the user. Second, item-space partitioning restricts recommendations to a specific product or subject area. Third, if the algorithm discards the most popular or unpopular items, they will never appear as recommendations, and customers who have purchased only those items will not get recommendations. Dimensionality reduction techniques applied to the item space tend to have the same effect by eliminating low-frequency items. Dimensionality reduction applied to the customer space effectively group's similar customers into clusters; as we now describe, such clustering can also degrade recommendation quality.

Cluster Models

To find customers who are similar to the user, cluster models divide the customer base into many segments and treat the task as a classification problem. The algorithm's goal is to assign the user to the segment containing the most similar customers. It then uses the purchases and ratings of the customers in the segment to generate recommendations. The segments typically are created using a clustering or other unsupervised learning algorithm, although some applications use manually determined segments. Using a similarity metric, a clustering algorithm groups the most similar customers together to form clusters or segments. Because optimal clustering over large data sets is impractical, most applications use various forms of greedy cluster generation. These algorithms typically start with an initial set of segments, which often contain one randomly selected customer each. They then repeatedly match customers to the existing segments, usually with some provision for creating new or merging existing segments.

For very large data sets especially those with high dimensionality sampling or dimensionality reduction is also necessary. Once the algorithm generates the segments, it computes the user's similarity to vectors that summarize each segment, then chooses the segment with the strongest similarity and classifies the user accordingly. Some algorithms classify users into multiple segments and describe the strength of each relationship. Cluster models have better online scalability and performance than collaborative filtering because they compare the user to a controlled number of segments rather than the entire customer base. The complex and expensive clustering computation is run offline. However, recommendation quality is low. Cluster models group numerous customers together in a segment, match a user to a segment, and then consider all customers in the segment similar customers for the purpose of making recommendations. Because the similar customers that the cluster models find are not the most similar customers, the recommendations they produce are less relevant. It is possible to improve quality by using numerous fine-grained segments, but then online user–segment classification becomes almost as expensive as finding similar customers using collaborative filtering.

Search-Based Methods

Search- or content-based methods treat the recommendations problem as a search for related items. Given the user's purchased and rated items, the algorithm constructs a search query to find other popular items by the same author, artist, or director, or with similar keywords or subjects.

Content-Based Recommender Systems The descriptions of items are analyzed to identify interesting items for users in CB recommender systems. Based on the items a user has rated, a CB recommender learns a profile of user's interests or preferences. According to a user's interest profile, the items which are similar to the ones that the user has preferred or rated highly in the past will be recommended to the user. For CB recommender systems, it is important to learn users' profiles. Various learning approaches have been applied to construct profiles of users.

Collaborative Filtering

For the reason that CF methods do not require well-structured .There are two kinds of CF methods, namely User-based CF approach and item-based CF approach. User-based CF approach first finds out a set of nearest "neighbors" (similar users) for each user, who share similar favorites or interests. Then, the rating of a user on an unrated item is predicted based on the ratings given by the user's "neighbors" on the item.

Hybrid Recommender Systems

Several recommender systems and use a hybrid approach by combining collaborative and content based methods, so as to help avoid some limitations of content-based and collaborative systems. A novel hybrid approach is to implement collaborative and CB methods separately, and then combines their predictions by a combining function, such as a linear combination of ratings or a voting scheme or other metrics. Melville et al. use a CB method to augment the rating matrix and then use a CF method for recommendation. Some hybrid recommender systems combine item-based CF and user-based CF. For example; Ma et al. propose an effective missing data prediction (EMDP) by combining item-based CF and user-based CF.

Scalability

To improve the scalability, it's possible to increase the number of segments, but this makes the online user–segment classification expensive. Search-based models build keyword, category, and author indexes offline, but fail to provide recommendations with interesting, targeted titles. They also scale poorly for customers with numerous purchases and ratings. The key to item-to-item collaborative filtering scalability and performance is that it creates the expensive similar-items table offline. The algorithm's online component looking up similar items for the user's purchases and ratings scales independently of the catalog size or the total number of customers; it is dependent only on how many titles the user has purchased or rated. Thus, the algorithm is fast even for extremely large data sets. Because the algorithm recommends highly correlated similar items, recommendation quality is excellent. Unlike traditional collaborative filtering, the algorithm also performs well with limited user data, producing high-quality recommendations based on as few as two or three items.

Security: Security is required for protecting user privacy while sharing user data among multiple users. In this regards onion routing algorithm will be applied at dataset level , when a user register with the database, soon after login registered data will be encrypted at dataset level and generated encrypted data stored in to another layer of the dataset. Further encrypted data will be sent along with user reviews for providing user ratings on data items finally our system becomes more secured.

Advantages:

- It generally improves the accuracy of predictions when compared with previous recommendation methods.
- It works well even with sparse training data sets, especially in data sets with sparse ratings for each item.
- It can reduce the number of big-error predictions.
- It is more efficient and secured than the compared methods.
- User-based techniques hold allowing CF-based– algorithms to scale to large data sets and at the same time produce high-quality recommendations. The user will get more precise (accurate) and– optimum recommendation.

VI. CONCLUSION

In this paper we investigate on Collaborative filtering (CF). Which is an important and popular technology for recommender systems with various challenging issues like Collaborative filtering (CF) is an important and popular technology for recommender systems. However, current CF methods suffer from such problems as Data Sparsity, Recommendation accuracy, Scalability and Security. In order to address the above issues in this paper we have proposed a novel algorithm to recommend items to users based on a hybrid method. First we use clustering to form the user clusters based on the similarity of users. We have taken users watching 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. In ordered to provide the security we used onion routing algorithms.

REFERENCES

[1] Z. Huang, H. Chen, and D. Zeng, "Applying Associative Retrieval Techniques to Alleviate the Sparsity Problem in Collaborative Filtering," ACM Trans. Information Systems, vol. 22, no. 1, pp. 116- 142, 2004.

[2] G. Adomavicius and A. Tuzhilin, "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," IEEE Trans. Knowledge and Data Eng., vol. 17, no. 6, pp. 734-749, June 2005.

[3] K.M. Galotti, Cognitive Psychology In and Out of the Laboratory, third ed. Wadsworth, 2004.

[4] G.L. Murphy, The Big Book of Concepts. MIT Press, 2002.

[5] L.W. Barsalou, Cognitive Psychology: An Overview for Cognitive Scientists. Lawrence Erlbaum Assoc., 1992.

[6] S. Schiffer and S. Steele, Cognition and Representation. Westview Press, 1988.

[7] D.L. Medin and E.E. Smith, "Concepts and Concept Formation," Ann. Rev. of Psychology, vol. 35, pp. 113-138, 1984.

[8] W. Vanpaemel, G. Storms, and B. Ons, "A Varying Abstraction Model for Categorization," Proc. Cognitive Science Conf. (CogSci'05), pp. 2277-2282, 2005.

[9] Tekur Vijetha, M.Sri lakshmi, Dr.S.Prem Kumar ," Survey on Collaborative Filtering and Content-Based Recommending " International Journal Of Computer Engineering In Research Trends. Volume 2, Issue 9, September 2015, Pp 594-599, ISSN (Online): 2349-7084. www.ijcert.org.