IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 18, Issue 4, Ver. II (Jul.-Aug. 2016), PP 51-55
www.iosrjournals.org

Analysis of Various Recommendation Systems

Sunidhi Sachdeva[1], Namita Tiwari[2]
[1]*(Department Of Computer Science Engineering,MANIT Bhopal,India)*
[2]*(Asstt. Professor,Department Of Computer Science Engineering,MANIT Bhopal,India)*

Abstract: *In today's world the only way to catch and keep user's attention to one's website is to provide them with the recommendation list to match the needs of user. In this paper the most popular and widely used recommendation algorithms to build a recommender system has been surveyed. A recommender system is an intermediate program (or an agent) with a user interface that automatically and shrewdly generates a list of information which fits an individual's needs. Initially recommendation systems were used only in large e-commerce websites but nowadays it's widely used in much other type of websites like websites related to music, video, news and tourism etc. Major recommendation algorithms include the popular user-based and item-based correlation algorithms.*

Keywords: *Recommendation System, Filtering.*

I. Introduction

The explosive growth of internet has led to a whole new source of huge data which can be used in many fields from web personalization to automatically identifying user's next step on the website. Now researchers need to analyze this data in efficient and effective way to make a user's experience better on a website or personalized user surfing experience. A way to do this is to provide what user want as quickly and easily available as possible, and this can be done by giving him a decent recommendation list.

A recommendation system is interactive website software that generates a list of objects, that a user might want to explore, to facilitates and customize their online experience. These recommendation systems are a vital part of most of the website that provide and kind of goods or services. The major benefits of these recommendations are cross-selling, personalization, customer withholding and keeping the customer informed.

A recommendation system is based on what type of filtering technique it has used in its algorithm. Then there is approach, what kind of approach a recommendation system is using is a major factor in its functioning, it can use user-based or item based or hybrid approach. In this paper we will explore all these filtering techniques and approaches to build a recommendation system.

II. Background and Related Work

The very first recommendation system was developed in 1992, as defined by M. deshpande and G. Karypis "A personalized information filtering technology used to either predict whether a particular user will like a particular item (prediction problem) or to identify a set of N items that will be of interest to a certain user." Over the years various other type of recommendation systems have been developed, most of these systems are based on either collaborative filtering or content based filtering.

In collaborative filtering suggests taking user's view on items, i.e. whether they like or dislike a particular item or not, or what he thinks of a particular item. And based on this information predicts things that might be interesting to the user.

In content based filtering results/ recommendations are shown based on the information available in user profile, or taking into account what items user has brought so far. An another less popular approach called rule-based approach proposes to ask a user to answer some questions, and based on the answers of these questions results are tailored.

III. Survey of Recommender Systems

From the very beginning of recommendations, there have been so many types of recommendation system generated based on various types of filtering techniques and algorithms. Some of the algorithms are :

A. ***Traditional collaborative filtering:*** These algorithms use method which is purely based on collaborative filtering. Here we calculate similarity between two users/customers and make recommendations. If two users are similar then we can recommend items to one which is purchase or liked by another one. Now this similarity can be calculated by various methods:

- **Cosine-based Similarity:** In this method we consider two users as two N-dimensional vectors, where N is the number of different items in catalogue, the value of component of vector is positive if the user has reviewed the item positively and negative if the review is negative .And then it calculates similarity between these two vectors with the help of a cosine-based function. The more the value of this cosine-based function the more similar these users are. The function which is commonly used is:

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

But this formula has high complexity so it’s computationally expansive.

B. ***Euclidean and Manhattan Formula:*** As the similarity can be defined as the closeness between two users, this method suggests to use the distance formula to find out the closeness between two users using the following two formulas:

$$d(i,j) = \sqrt{(x_{i1} - x_{j1})^2 + \ldots + (x_{in} - x_{jn})^2}$$
$$d(i,j) = |x_{i1} - x_{j1}| + \ldots + |x_{in} - x_{jn}|$$

This function can only be used when we have numerical values, but when we have categorical values such as color (like red or pink) then we cannot use this function.

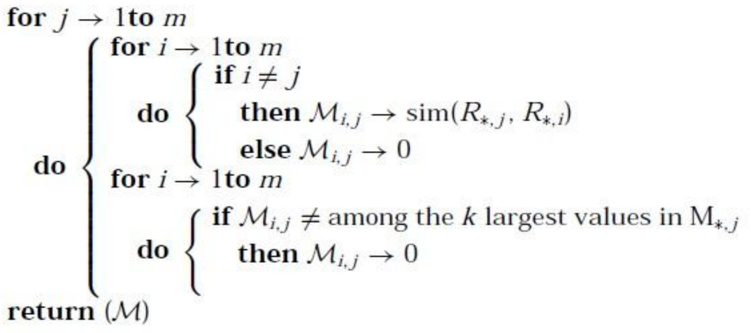
C. ***Conditional Probability Based Similarity:*** Another method is to use conditional probability to find out the similarity between two users. In this method we find the probability of purchase of a particular item i when another item j has already been purchased. Here i is an item that has been purchased by a user B, whereas item j is an item that has been purchased by both user A and B.
sim(i,j)= prob(i|j) * a
here a is a factor dependent on the problem.

D. ***User-based Algorithms:*** These types of algorithms are also known as Cluster based algorithms. In this algorithm all the users are classified into different clusters based on their behavior on the site. These clusters are usually created using unsupervised learning techniques like k-means clustering, or c-means clustering from fuzzy logic. The c-means is shown to give better results as in c-means clustering or fuzzy clustering rather than belonging to a particular cluster completely, it belongs to a cluster to a certain degree. This algorithm can be divided into three steps, first is profiling the user, i.e. identifying to what cluster a user belong based on the items that he has purchased or like or gave review about, second step is giving weight to items that the users in the cluster has purchased to classify which item is more relatable or appropriate to the target user, third step involves selecting the item based on the weight and recommending it to the user[4]. The quality of recommendations depends on the first step i.e. profiling and assigning the user to a cluster. Clustering method is usually faster than collaborative filtering because all the time taking cluster generation process is done offline. And cluster models also have better performance and scalability because a user is only compared with the other users in the clusters not to every user. But this degrades the quality of recommendations.

E. ***Search- based algorithms:*** In these type of algorithms the recommendation problem is treated as a search for related items. Based on the user’s previous purchased and liked item, the algorithms construct a search query which finds suitable items for recommendations. If the user has few purchases or ratings, search-based recommendation algorithms scale and performance well. For users with many purchases its impractical to base a query on all the items[3]. Overall the recommendations produce by these type of algorithm have relatively poor quality, they are either too general or too narrow. These algorithm fails to provide new and interesting and relevant recommendations.

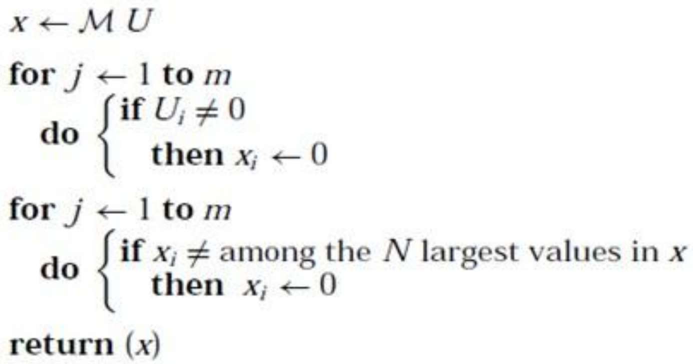
F. ***Item-based algorithms:*** These type of algorithms uses collaborative filtering but on items. It’s based on the fact that a user wants to buy an item based on his previous purchases so by analyzing his historical information, the most similar items that he is likely to purchase can be predicted. These algorithms based on two steps, first step involves building a model based on relation between items, i.e. to what degree a particular item is related to other, second step apply this precomputed model to derive the recommendations for an active user.

- ***Step-1:*** The input to this algorithm is the n × m user–item matrix R and a parameter k that specifies the number of item-to-item similarities that will be stored for each item. The output is the model itself, which is represented by an m × m matrix M such that the jth column stores the k most similar items to item j . In particular, if Mi, j > 0, then the ith item is among the k most similar items of j and the value of Mi, j indicates the degree of similarity between items j and i. The similarity between items can be calculated by any of the methods given above. By using the small value of k the resulted matrix will be sparse and can be stored into main memory but if the value of k is too small then it will contain limited information and thus quality of recommendations will be low.



```
for $j \rightarrow 1$to $m$
do  for $i \rightarrow 1$to $m$
        do  if $i \neq j$
                then $\mathcal{M}_{i,j} \rightarrow \text{sim}(R_{*,j}, R_{*,i})$
                else $\mathcal{M}_{i,j} \rightarrow 0$
    for $i \rightarrow 1$to $m$
        do  if $\mathcal{M}_{i,j} \neq$ among the $k$ largest values in $\text{M}_{*,j}$
                then $\mathcal{M}_{i,j} \rightarrow 0$
return $(\mathcal{M})$
```

- ➢ ***Step*-2:** In second step we apply this model to obtain recommendations. The input of this algorithm is the output of the first step i.e. the m × m matrix M, an m× 1 vector U that stores the items that have already been purchased by the active user, and the number of items to be recommended (N). The active user's information in vector U is encoded by setting Ui = 1 if the user has purchased the ith item and zero otherwise. The output of the algorithm is an m×1 vector x whose nonzero entries correspond to the top-N items that were recommended.



```
$x \leftarrow \mathcal{M}\, U$
for $j \leftarrow 1$ to $m$
    do  if $U_i \neq 0$
            then $x_i \leftarrow 0$
for $j \leftarrow 1$ to $m$
    do  if $x_i \neq$ among the $N$ largest values in $x$
            then $x_i \leftarrow 0$
return $(x)$
```

The weight of these nonzero entries represents a measure of the recommendation strength and the various recommendations can be ordered in non-increasing recommendation strength weight. The complexity of this algorithm depends on the time requires to build the model M and the amount of time needs to generate recommendations by applyling this model. The time required to compute the top recommendations for an active user that has purchased q items is given by O(kq) because we need to access the k most similar items for each one of the items that the user has already purchased and identify the overall N most similar items.

IV. Techniques

Some popular Techniques used in recommendation systems based on the above algorithms:

1. **Association rule mining:** An association rule is a rule to show associations or relations between items. Its applications are in various fields like decision support, selective marketing, medical diagnosis and many other fields that's why this area has attracted a lot of attention in last decade.

Association rule mining is, given a set of transactions, where a transaction is a set of items, finding rules that is in the form of X=>Y, where X and Y are the set of transactions. The meaning of this relationship is presence of X implies the presence of Y. The two factors that define this relationship are support and confidence.

Support is a measure of appearance of a set of items (say X) with the respect of all transaction. Confidence shows the percentage of transactions that contains Y among transactions that contain X. Higher value of confidence shows that there is higher chance of appearance of Y when X is there. This factor helps building recommender system using association mining.

2. **Apriori Algorithm:** The Apriori algorithm To generate frequent itemsets apriori make multiple passes over the database. We use k-itemsets to represent itemsets of size k. In the first pass over the database 1-itemsets are found. For the rest passes as k>1, using (k-1) itemsets, the candidate frequent k-itemsets are generated;

then the actual support count is calculated and at the end of pass k, it founds the candidate k-itemsets who are having supports, above the minimum support as the frequent k-itemsets.

$$\text{Support} = \frac{\text{\#of transaction includes X}}{\text{\#of all transactions}}$$

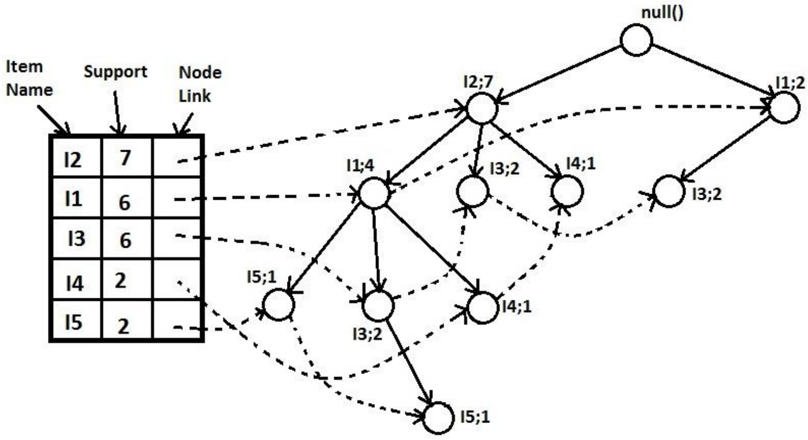
$$\text{Confidence} = \frac{\text{Support(X,Y)}}{\text{Support(Y)}}$$

Candidate Generation: Candidate k-itemsets generation is done by performing two steps: Join and prune. In the join step, two different (k-1) itemsets who share same first (k-2) itemset, are joined together. This is done for every (k-1) itemset until all possible combinations are generated. In the prune step, those combinations which are non-frequent, removed. These two steps are relevant due to the fact that any subset of a frequent itemset must also be frequent.

3. **FP-growth Algorithm:** The FP-Growth algorithm scans the database only two times. In the first scan it calculates the set of frequent items sets and there support and in the second step mining is done using FP-tree. It works in steps, first step involves the construction of the tree and in second step the FP-tree is mined.

Construction of FP-tree: After the first scan of the database, in second scan of the database FP-tree is constructed. As the root of this tree we take a node with label or value as null. In this tree every branch corresponds to one or more transactions.

Every node consists of two parts, first being the item name and second part has the value of item's support count. For every transaction a branch is constructed, and if two transactions have same prefix than suffix of second transaction will be a sub-branch of the first one and the support of the common part will be increased by 1. For maintaining the record of support of items, a table is created and this table also has links to indicate to the corresponding node as it helps when mining of the tree will be done.



I2	7	
I1	6	
I3	6	
I4	2	
I5	2	

Mining of FP-tree: Mining of the tree starts from the item which has the least support count, and goes all the way up from node to root node of the tree and construct its condition pattern base which is a sub database of the set of prefix paths in the FP-tree occurring simultaneously with the suffix pattern. Then we will construct its condition FP-tree and perform mining on this tree.

V. Challenges

1) **Cold-start:** When a user creates his profile his preferences are unknown, so what kind of recommendations should be given to him. This problem is known as cold start problem. Not only users but also items suffer from this problem as when an item is added to the warehouse or website catalogue it has no rating from any user.
2) **Trust:** This issue relate to user review, as a user with bigger history on any website will have more relevant review compare to the one with small history. And it's difficult to determine how relevant a review is to the related product; moreover it's given by a genuine customer or a bogus one.

3) **Scalability:** It becomes a major issue when number of users and items increases. Because as users and item increases in number there is an additional need for more computational power and resources to generate the recommendation list. This also leads to increase in the time that is require to produce the recommendations.
4) **Sparsity:** Sparsity is the problem generated due to the lack of information. In a website catalogue there are comparatively only a few items reviewed or rated by a particular user, and there are also some users that have never reviewed any item. So deciding there flavor or choices is tough.
5) **Privacy:** Privacy has always been a major problem when it comes to internet. Same is applicable when generating recommendations as when there is some personal data of any user there arise a need to protect it from being used in unauthorized manner. For a system to generate useful and accurate recommendations it needs to be provided with sufficient information, this might include some of the user's personal information although many websites claims that they protect user's personal data effectively.

VI. Conclusion

In this paper various filtering techniques and algorithms are represented. The most commonly and widely used algorithms are also explained. These techniques perform good at terms of time and quality but as the internet grow and spread even more we will be needing better techniques in the future.

References

[1] Jiawei Han and Micheline Kamber, *Data Mining:Concepts and Techniques* (Elsevier Inc., 2006).

[2] Deshpande M. and Karypis G., Item-based top-n recommendation algorithms, *ACM Transactions on Information Systems, Vol. 22, No. 1, January 2004, Pages 143–177.*

[3] Greg Linden, Brent Smith, and Jeremy York, Amazon.com Recommendations, *IEEE Internet computing, 7.1, 2003, 76-80.*

[4] David W. Cheungt, Jiawei Hans, Vincent T. Ngtt Ada, W. Fuss Yongjian FuI, A Fast Distributed Algorithm for Mining Association Rules, *Parallel and Distributed Information Systems, 1996, Fourth International Conference on IEEE, 1996, 31-42.*

[5] Weiyang Lin, Association Rule Mining for Collaborative Recommender Systems., *Diss. Worcester Polytechnic Institute, 2000.*

[6] Dhoha Almazro,Ghadeer Shahatah,Lamia Albdulkarim, Mona Kherees,Romy Martinez,William Nzoukou, A Survey Paper on Recommender Systems, *2010.*

[7] Mukta Kohar,Chhavi Rana. Department of computer science and Engg,U.I.E.T, MDU,Rohtak. Survey Paper on Recommendation System.