International Journal of Engineering

Journal Homepage: www.ije.ir

TACHNICAL NOTE

Using a Data Mining Tool and FP-growth Algorithm Application for Extraction of the Rules in Two Different Dataset

E. Hashemzadeh, H. Hamidi*

Department of Industrial Engineering, K. N. Toosi University of Technology, Tehran, Iran

PAPER INFO

Paper history:
Received 13 May 2016
Received in revised form 02 June 2016
Accepted 02 June 2016

Keywords:
Recommender Systems
Collaborative Filtering
Association Rules
Support
Confidence

ABSTRACT

In this paper, we want to improve association rules in order to be used in recommenders. Recommender systems present a method to create personalized offers. One of the most important types of recommender systems is the collaborative filtering that deals with data mining in user information and offering them the appropriate item. Among the data mining methods, finding frequent item sets and creating association rules are included in dataset. In this method, through separating the data of more active users, those who are interested in more items, we make sample from the training set and continue finding the association rules on the selected sample. Therefore, while the training set gets smaller, the production speed of rules increases. At the same time, we will show that the quality of the produced rules has been improved. Among the advantages of the proposed method, it can be referred to its simplicity and rapid implementation. Moreover, through a sampling from training set, the speed of association rules will be increased.

***doi**: 10.5829/idosi.ije.2016.29.06c.08*

1. INTRODUCTION

The increase of products and service decisions to select the most appropriate product is difficult for users; recommender systems by studying the users' interests can offer useful products and services to the users. These systems play an important role in finding the customers' interests; therefore, they are widely considered in e-commerce businesses such as online purchase and sale [1]. In recent decades, using recommenders is very common in commercial areas, and following the commercial popularity of these systems, many scientific researches have been implemented. Researchers try to develop and improve the personalized suggestions to the people. The meaning of personalization is "to provide topics and services that are suitable for people based on their behavior and preferences". Since the recommender systems investigate the interests and preferences of individuals, they can offer special products and services to them. They also can offer new products to the people so they may love it. It should be mentioned that analyzing the product content similarity with the favorite product and removing the unrelated items, the recommenders can identify more useful items and offer them to the users [2]. Generally, so many algorithms are presented to create the recommendation that we can classify them in different categorizations. Including the most important used algorithms in these systems is collaborative filtering, content-based, knowledge-based and also hybrid ones. In addition, with the development of web 2.0, a new generation of recommenders is presented that we will discuss them in brief. The collaborative filtering is one of the most practical methods in the recommenders that offer people the appropriate products using the data mining in the people rating information. Including the data mining methods is the mining association rules that in this paper we want to use it to find the available rules in the users' rating information, and using more useful users' data we want to create more suitable rules. In fact, in this method, through separating the data of more active users (those who are interested in more items), we make sample from the training set and continue finding the association rules on the selected set. Therefore, while

*Corresponding Author's Email: *h_hamidi@kntu.ac.ir* (H. Hamidi) the training set gets smaller, the production speed of rules increases. At the same time, we will show that the quality of the produced rules has been improved.

For this purpose, in section 2 we will talk about the types of recommenders and their features. In section 3, we will explain data mining and association rules and in sections 4 and 5, we will implement the proposed method and review the results and finally we will reach a conclusion.

2. TYPES OF RECOMMENDER SYSTEMS

In various articles, different categories are considered for recommender systems. According to the traditional classification, the recommender systems are in four categories of collaborative filtering, content-based, knowledge-based and hybrid. In addition, today new types of recommenders are introduced that may not be fitted in recent categories. Some of them are: social-based, tag-based and context-aware recommenders. In the following section the different recommendation types are briefly explained.

2. 1. Collaborative Filtering Recommenders

The purpose of collaborative recommender systems is to find similar users with the user that we are going to recommend something to him. For this purpose, we should investigate interests and preferences of the other like-minded users. Collaborative filtering techniques are highly functional in various fields such as electronic education, multimedia and digital libraries. It is obvious that the interests and views of users can be explicitly or implicitly extracted. The most important problem of collaborative filtering is that new items that are added will not be offered until they would be rated or chosen by the users. This problem is called “cold start”. Another problem of collaborative filtering technique is “data sparseness”. It means that when the number of required rates is low compared to the number of items, the quality of suggestions will be affected [2]. Despite these problems, majority of studies in recommender area use collaborative filtering method to make recommendations [3].

In filtering techniques there are different criteria to find the similar users. The criteria is defined in Equations (1), (2) and (3) [2].

$$\mathrm{sim}(x,y)=\frac{\sum_{i=1}^{m} r_{x,i} r_{y,i}}{\sum_{i=1}^{m} r_{x,i}^{2} \sum_{i=1}^{m} r_{y,i}^{2}} \quad (1)$$

$$\mathrm{sim}(x,y)=\frac{\sum_{i=1}^{m} (r_{x,i}-\bar{r}_{x})(r_{y,i}-\bar{r}_{y})}{\sum_{i=1}^{m} (r_{x,i}-\bar{r}_{x})^{2} \sum_{i=1}^{m} (r_{y,i}-\bar{r}_{y})^{2}} \quad (2)$$

$$\mathrm{sim}(x,y)=\frac{1}{m}\sum_{i=1}^{m} (r_{x,i}-r_{y,i})^{2} \quad (3)$$

where $r_{x,i}$ and $r_{y,i}$ are the rate that the users have given to the item i; these functions will calculate the similarity between two users.

2. 2. Content-based Recommenders

In a content-based recommender system, the items are proposed based on their characteristics and properties. In these systems instead of considering the similarities among users, the features and special characteristics of the items will be considered as a measure of recommendations. In fact, it is assumed that an active user may prefer the items to be suggested to him that he has previously mentioned them [2].

In content-based method, users are limited to items that they have previously scored or showed their interest to them; because, in this method the like-minded users comments is not considered with the active user comments for the items recommendations. This is a fundamental weakness for content-based method [2]. One of the most popular algorithm used in this field is term frequency–inverse document frequency (TF-IDF).

Assume that N is the total number of texts that can be offered to the user, and the keyword k_j is available in n_i texts. We also can assume that $f_{i,j}$ is the number of k_i keyword repetition in the d_j text. In this case, the frequency of the k_i keyword in the d_j text is defined as (4):

$$TF_{i,j}=\frac{f_{i,j}}{\max_{x} f_{z,j}} \quad (4)$$

$\max_z f_{z,j}$ is the maximum number of $f_{z,j}$ that is repetitions for all the k_z keywords in the text d_j.

Usually they use inverse frequency (IDF_i) in the union with term frequency ($TF_{i,j}$), the inverse frequency for the keyword k_i is as:

$$IDF_i=\log\frac{N}{n_i} \quad (5)$$

The TF-IDF weight for the k_i keyword in the d_j text is as follows:

$$w_{i,j}=TF_{i,j}\times IDF_i \quad (6)$$

We can present a vector display of the d_j text content [4]:

$$\mathrm{content}(d_j)=(w_{1j},\ldots,w_{kj}) \quad (7)$$

2. 3. Knowledge-based Recommenders These systems rely on the products' features and according to these features they offer some items to the user that would be in accordance with his needs and interests. The user can express his preferences by determining the features of the required item [5].

There is no exact border between content-based and knowledge-based systems and many researchers have considered the content-based methods as a subset knowledge-based methods. But we can say that content-based systems use the described information of the item; while, knowledge-based systems use an additional knowledge information such as utility function to create a recommendation [6].

2. 4. Recommenders and New Generation of Web

The development of web 2 and increased use of social networks has led to the emergence of new types of recommender systems. To improve recommendation in online communities we can use trust-aware recommender systems. Studies have shown that use of reliable information improves the accuracy of recommender systems. In these systems we are faced with cases such as recommendation strategies, implicit trust and new friends' recommendation. Also by creating the possibility of annotation of resources on the web, folksonomy (folk taxonomies) is created that to improve the recommendation it provides valuable information for the recommender systems. These systems are known as tag-based recommender systems. Tagged information is suitable for recommender systems because collaborative filtering method have failed in combined and heterogeneous domains; while, the tagged information create high-quality recommendations for the systems with heterogeneous domains [6].

The move towards web 3.0 and the Internet of Things, context-aware information are used from various devices and sensors [7]. Such information can be used in recommender systems so a new generation of systems that is called context-aware recommender system can be formed. These systems use the information such as time, location and wireless sensor networks. This information can be used explicitly, implicitly, through data mining or by a combination of methods. For example, the use of geographical data in mobile devices has developed. This information is used in the recommenders that are aware of the geographical location so with respect to the location of the users some offers will be provided to them [7].

2. 5. Hybrid Recommenders The hybrid recommender systems combine two or more recommendation methods to deal with the problems of each method.

3. ASSOCIATION RULE MINING

To handle the vast amounts of data that are rapidly generated, exploratory data analysis methods are being increasingly popular [8]. So nowadays various data mining techniques are emergenced to extract the knowledge contained in this data.The task of data mining is to discover similar patterns, unfamiliar events, or relationship between data through analyzing data [9]. Among these methods are the data clustering, decision trees generation, association rules, and so on.

One of the most important data mining techniques is the association rules mining which consists of two stages: in the first stage using the minimum support the frequent item sets are generated, and in the second stage using the minimum confidence the association rules are extracted.

Association rules indicate mutual relationships and dependencies between large set of data items. Finding such rules can be used in different areas and have different applications. For example, the discovery of association relations between massive volumes of business transactions can be used in fraud detection, in the field of medicine and also data mining of the information about the method of using the web by the users [10]. They also can be effective in catalog design, marketing and other business decisions processes.

The most current example in relation with the discovery of association rules is "market basket analysis". In this process according to the various items that customers put in their cart, their purchasing habits and behaviors are analyzed. For example, based on this idea when a user buys the x product, with c% possibility he will also buy the y product. The purpose in this process is to automatically find the rules that are expressed as a conditional sentence. Also, for the acceptability of the rules some criteria are proposed that will be explained later.

3. 1. Basic Concepts The basic assumptions are as follows: we consider the $I = I_1, I_2, I_3, \ldots$ as the items set and D as the dataset that contains transactions so each transaction includes a set of items, each transaction is a subset of I ($T \subseteq I$). If A is a set of items we say that *T* transaction includes A if and only if $A \subseteq T$. An association rule is a statement that is as $A \Rightarrow B$ while if $A \subseteq I$, $B \subseteq I$, $A \cap B = \varphi$ (at first the rules were considered only as the form of $A \Rightarrow I_j$). To asset the value and acceptability of the association rules criteria we introduce two important parameters: support and confidence. The $A \Rightarrow B$ rule in the D transaction sets has the s value of support, if s percent of D transactions include $A \cup B$ and this rule has the confidence value of c, if c percent of the transactions that include A include B too, or:

$$\sup(A \rightarrow B) = \frac{|A \cup B|}{n} \tag{8}$$

$$\text{conf}\,(A \rightarrow B) = P(B\,|\,A) = \frac{|A \cup B|}{|A|} \tag{9}$$

In Equation (8), n is the number of transactions.

The rules that have minimum support (min-sup) or minimum confidence (min-conf) are called the strong association rules and the purpose of every algorithm is to generate such rules.

Including the association rules, generation algorithms are Appriori and FP-Growth. In most of the algorithms and also Appriori, finding the association rules is a two-step process. At the first stage, the item sets that satisfy min-sup, a predetermined minimum support, are identified, and at the second stage these item sets are used to generate the association rules. The overall performance of mining association rules depends primarily on the first step. The second step is easy [11, 12]. After the identification of the important set of items, the association rules can be easily extracted from them. The FP-Growth performance is different; because this algorithm only deals with the generation of frequent item sets. Therefore, it is necessary to generate association rules after its implementation. In order to generate association rules, we apply the useable FP-Growth algorithm through the RapidMiner.

4. PROPOSED METHOD AND EXPERIMENTS

In this paper we want to examine the association rules of two datasets and also find a way to improve the rules. For this purpose, we try to improve the quality of generated rules by separating the better information among the used data. Using more and better rules for the recommenders increase their efficiency in the items prediction.

In order to implement the project we need to install RapidMiner Studio[2] and Microsoft Visual Studio[3] software. Also we need a dataset to apply the operation of the association rules generation on them. For this purpose, we have used two datasets that the description of the association rules generation is given to them. After the installation of RapidMiner, we have designed a process that is shown in Figure 1. In this process, at first the data is prepared and then the FP-Growth algorithm is applied on them. Later, the operation of the association rules generation is implemented and the quality of the rules is evaluated.

4. 1. The Generation of Association Rules for MovieLense Dataset

1) Description of dataset: MovieLense[4] dataset version 1.0 is published in May 2011 by GroupLense[5] research group. In this dataset the users have both tagging and rating information. MovieLense includes the movie information including actors, producer country, director, movie locations, the tags assigned to the movies, and the number of times the tags were assigned to each movie, and the user information including their ratings, time of rating, movies' tags, and the time tagging. Each of these data is placed in a separate file. Since our goal is to generate the association rules with regard to the user ratings, we will only use the files that are related to user ratings.

Our used information include: 2113 users, 10197 movies, and 855598 ratings. The average number of rating for each user is 404.921 and for each movie is 84.637. The movies are rated by the users from 0 to 5.

2) Data preparation and implementation of project: In order to implement the project, we have used two files in the dataset that one contains a list of movies and other one contains each user ratings for different movies. The files of the studied movies include ID, name and other information related to the movie. It should be noted that data review and generation of rules occurs only based on the movies ID and the movies names are not used. Therefore, only the first column that contains the movies ID will be extracted from the file. The next used file is related to the ratings that gave to the movies. In this file, only the first three columns are extracted that first one is userID, the second one is the rated movieID, and the third one includes the related ratings.

In order to study the favorite movies of each user it is necessary to determine a threshold for ratings. If the rate that the user has given is more than the threshold, the considered item is named as the user favorite movie, and if the rate is lower than the threshold the movie is not the user favorite. The reason for this thresholding and separating the favorite movies of the user is that the association rules should be generated on the favorite items or products purchased by the user and the items that are not the users' favorite won't be useful.

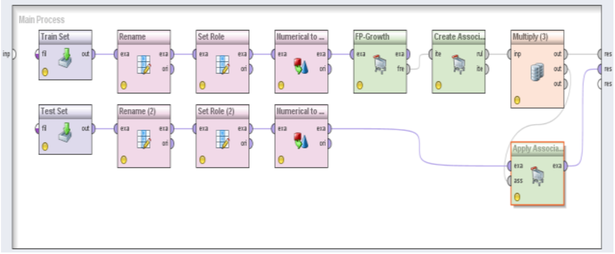


Figure 1. Process to generate the association rules in RapidMiner

[2] www.rapidminer.com

[3] www.visualstudio.com

[4] www.grouplens.org/datasets/movielens

[5] www.grouplense.org

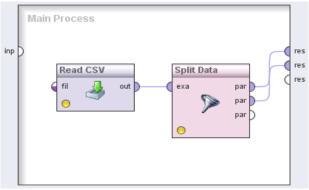


Figure 2. The separation process of train set and test set by RapidMiner

In order to separate the favorite items we have written a code in C++. According to this code the items that their corresponding rating is at least 4 are written in a separate file as the user favorite items. Thus, we have obtained a file with the format of csv as the output that includes 376106 lines. Each line includes the user ID and the ID of one of his favorite movies. In order to generate association rules and evaluate the efficiency of the generated rules we need two datasets:

- train set to generate rules
- test set to examine the usefulness of the generated rules

So we should convert the data that include users' ratings into two sets. For this purpose we use RapidMiner. We draw a process in the software (Figure 2) and consider 50% of the data as train set and 50% as the test set.

To generate association rules by RapidMiner the data should have two values so we should make the data in the form of matrixes that the first line includes the items ID and the second column includes the users ID, and we fill the rest of the cells based on the user interests with one and zero. Using a code with the C++ language we have created such layout for the data.

In this article we intend to improve the association rules resulted from the train set, for this purpose try to separate the best users' information from the train set. Users' separation is done in a way that the users who have at least expressed their interest to different items at the average number of all favorites will be selected. In fact, these users are considered as more active users that have rated more items and using their information would be more useful. The separation of active users was implemented through coding. The purpose of separating the most active users is to generate better and more accurate rules because the train set matrix will be filled better and better rules will be extracted. The results will be shown. It should be mentioned that, we place the train set at various stages for each dataset without any change, so the comparison of the results would be possible.

For MovieLense dataset at first we place the whole users' information (2113 users) in the train set, and then we will turn them to a suitable form to be entered to the software. With regards to the mean of the whole users' favorites, we only select the users that have at least rated 89 items. In this case 747 active users are selected so proper train set are provided to reenter the software and mining the association rules. We will apply Figure 1 process on the obtained datasets. The results will be explained in the next section.

4. 2. The Generation of Association Rules for Jester Dataset

1) Description of dataset: Jester dataset has three versions that we have used the Dataset 1 which is also used in many other articles. The information of this dataset is accessible in 3 Excel files. These data include more than 1.4 million rates for 100 different jokes that have been collected from April 1999 to May 2003. These datasets include the ratings of 73421 users.

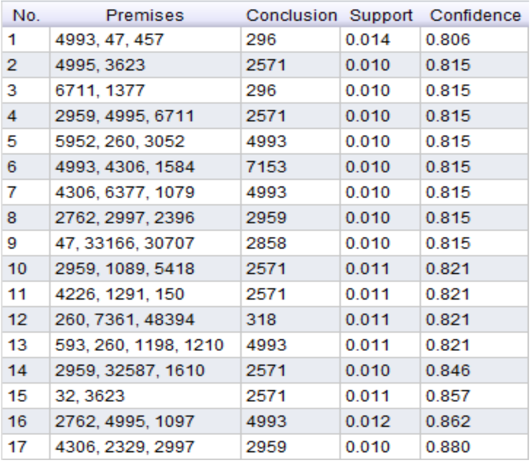
2) Data preparation and implementation of project: This time like before, by thresholding for the rates we should find the favorite jokes of each user. The maximum possible rate is 10. In order to determine users' interests, we have used coding and by putting the threshold of 7.5 for the ratings, we identified the favorite jokes for the users**.**

As usual in order to generate the train set and test set we have used a process in RapidMiner as shown in Figure 2. Randomly we have considered half of the data as train set and the other half as test set. Then, using the relevant codes we have turned the data to the suitable matrix form to generate the rules. At first, we consider the data of all the users. At the next stage we only separate the information of the users who have liked at least 3 jokes (the mean of the whole number of jokes that were liked by the users). Therefore, we have placed the interest of 23144 users in the train set. The resulted files of each time of each code implementation are separately given to the software as in Figure 1. The results will be explained in the next section.

In order to clarify the issue, we propose an example for the method. We assume that we have transactions according to Table 1. In this set, we have five users that are interested in 20 items. Since the mean of items that are interesting for the user's equals 4, we only select those users' data as the sample that at least are interested in 4 items. Therefore, the transactional information of A2, A3 and A5 are selected as the sample of the dataset. Now, using FP-Growth algorithm we try to find association rules existing in the selected sample. In fact, A1 and A4 are ignored.

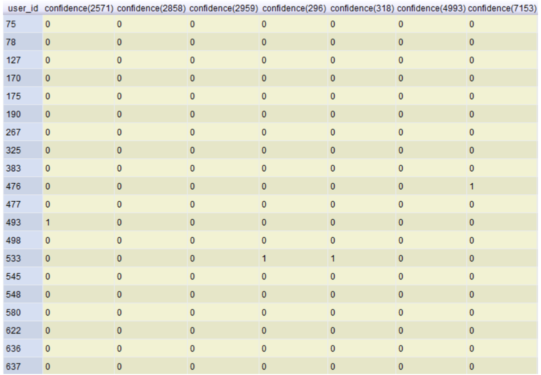
TABLE 1. Example of train set

Users	Items
A1	I1, I2, I7
A2	I3, I8, I10, I6
A3	I4, I3, I10, I8, I2, I5
A4	I8, I2
A5	I4, I5, I9, I1, I6



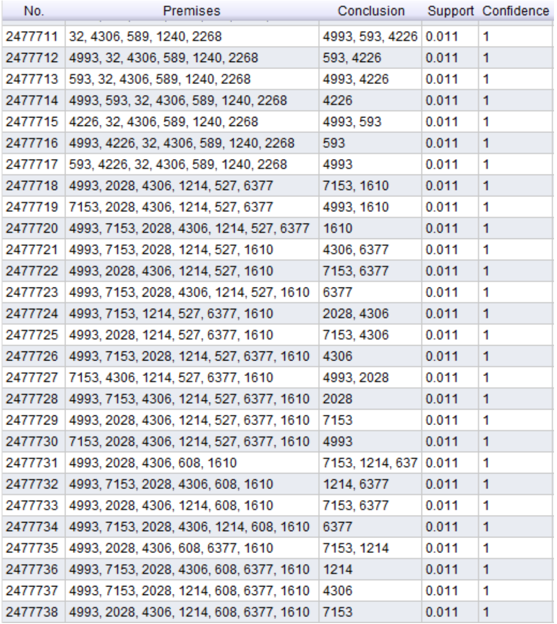
No.	Premises	Conclusion	Support	Confidence
1	4993, 47, 457	296	0.014	0.806
2	4995, 3623	2571	0.010	0.815
3	6711, 1377	296	0.010	0.815
4	2959, 4995, 6711	2571	0.010	0.815
5	5952, 260, 3052	4993	0.010	0.815
6	4993, 4306, 1584	7153	0.010	0.815
7	4306, 6377, 1079	4993	0.010	0.815
8	2762, 2997, 2396	2959	0.010	0.815
9	47, 33166, 30707	2858	0.010	0.815
10	2959, 1089, 5418	2571	0.011	0.821
11	4226, 1291, 150	2571	0.011	0.821
12	260, 7361, 48394	318	0.011	0.821
13	593, 260, 1198, 1210	4993	0.011	0.821
14	2959, 32587, 1610	2571	0.010	0.846
15	32, 3623	2571	0.011	0.857
16	2762, 4995, 1097	4993	0.012	0.862
17	4306, 2329, 2997	2959	0.010	0.880

Figure 3. The generated rules for MovieLense dataset (for all the 2113 users)



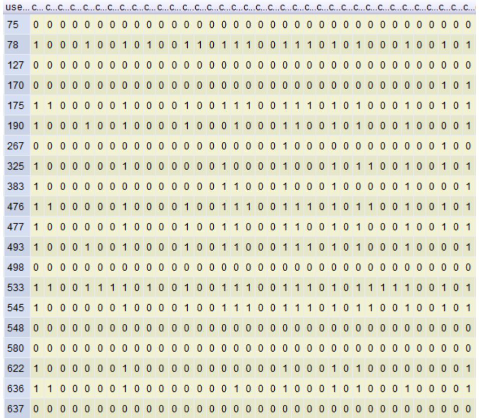
user_id	confidence(2571)	confidence(2858)	confidence(2959)	confidence(296)	confidence(318)	confidence(4993)	confidence(7153)
75	0	0	0	0	0	0	0
78	0	0	0	0	0	0	0
127	0	0	0	0	0	0	0
170	0	0	0	0	0	0	0
175	0	0	0	0	0	0	0
190	0	0	0	0	0	0	0
267	0	0	0	0	0	0	0
325	0	0	0	0	0	0	0
383	0	0	0	0	0	0	0
476	0	0	0	0	0	0	1
477	0	0	0	0	0	0	0
493	1	0	0	0	0	0	0
498	0	0	0	0	0	0	0
533	0	0	0	1	1	0	0
545	0	0	0	0	0	0	0
548	0	0	0	0	0	0	0
580	0	0	0	0	0	0	0
622	0	0	0	0	0	0	0
636	0	0	0	0	0	0	0
637	0	0	0	0	0	0	0

Figure 4. An example of rules' confidence after applying them on the test set in MovieLense (for all the 2113 users)



No.	Premises	Conclusion	Support	Confidence
2477711	32, 4306, 589, 1240, 2268	4993, 593, 4226	0.011	1
2477712	4993, 32, 4306, 589, 1240, 2268	593, 4226	0.011	1
2477713	593, 32, 4306, 589, 1240, 2268	4993, 4226	0.011	1
2477714	4993, 593, 32, 4306, 589, 1240, 2268	4226	0.011	1
2477715	4226, 32, 4306, 589, 1240, 2268	4993, 593	0.011	1
2477716	4993, 4226, 32, 4306, 589, 1240, 2268	593	0.011	1
2477717	593, 4226, 32, 4306, 589, 1240, 2268	4993	0.011	1
2477718	4993, 2028, 4306, 1214, 527, 6377	7153, 1610	0.011	1
2477719	7153, 2028, 4306, 1214, 527, 6377	4993, 1610	0.011	1
2477720	4993, 7153, 2028, 4306, 1214, 527, 6377	1610	0.011	1
2477721	4993, 7153, 2028, 1214, 527, 1610	4306, 6377	0.011	1
2477722	4993, 2028, 4306, 1214, 527, 1610	7153, 6377	0.011	1
2477723	4993, 7153, 2028, 4306, 1214, 527, 1610	6377	0.011	1
2477724	4993, 7153, 1214, 527, 6377, 1610	2028, 4306	0.011	1
2477725	4993, 2028, 1214, 527, 6377, 1610	7153, 4306	0.011	1
2477726	4993, 7153, 2028, 1214, 527, 6377, 1610	4306	0.011	1
2477727	7153, 4306, 1214, 527, 6377, 1610	4993, 2028	0.011	1
2477728	4993, 7153, 4306, 1214, 527, 6377, 1610	2028	0.011	1
2477729	4993, 2028, 4306, 1214, 527, 6377, 1610	7153	0.011	1
2477730	7153, 2028, 4306, 1214, 527, 6377, 1610	4993	0.011	1
2477731	4993, 2028, 4306, 608, 1610	7153, 1214, 637	0.011	1
2477732	4993, 7153, 2028, 4306, 608, 1610	1214, 6377	0.011	1
2477733	4993, 2028, 4306, 1214, 608, 1610	7153, 6377	0.011	1
2477734	4993, 7153, 2028, 4306, 1214, 608, 1610	6377	0.011	1
2477735	4993, 2028, 4306, 608, 6377, 1610	7153, 1214	0.011	1
2477736	4993, 7153, 2028, 4306, 608, 6377, 1610	1214	0.011	1
2477737	4993, 7153, 2028, 1214, 608, 6377, 1610	4306	0.011	1
2477738	4993, 2028, 4306, 1214, 608, 6377, 1610	7153	0.011	1

Figure 5. The generated rules for MovieLense dataset (for 747 users)



use...	c...	c...	c...	c...	c...	c...	c...	c...	c...	c...	c...	c...	c...	c...	c...	c...	c...	c...	c...	c...	c...	c...	c...	c...	c...	c...	c...	c...	c...	c...	c...	c...	c...	c...	c...	c...
75	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
78	1	0	0	0	1	0	0	1	0	1	0	0	1	1	0	1	1	1	0	0	1	1	1	0	1	0	1	0	0	0	1	0	0	1	0	1
127	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
170	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
175	1	1	0	0	0	0	0	1	0	0	0	0	1	0	0	1	1	1	0	0	1	1	1	0	1	0	1	0	0	0	1	0	0	1	0	1
190	1	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	1	1	0	0	1	0	1	0	0	0	1	0	0	0	0	1
267	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
325	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1	0	1	1	0	0	1	0	0	1	0	1
383	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1
476	1	1	0	0	0	0	0	1	0	0	0	0	1	0	0	1	1	1	0	0	1	1	1	0	1	0	1	1	0	0	1	0	0	1	0	1
477	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	1	0	0	0	1	1	0	0	1	0	1	0	0	0	1	0	0	1	0	1
493	1	0	0	0	1	0	0	1	0	0	0	0	1	0	0	1	1	0	0	0	1	1	1	0	1	0	1	0	0	0	1	0	0	0	0	1
498	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
533	1	1	0	0	1	1	1	1	0	1	0	0	1	0	0	1	1	1	0	0	1	1	1	0	1	0	1	1	1	1	1	0	0	1	0	1
545	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	1	1	0	0	1	1	1	0	1	0	1	1	0	0	1	0	0	1	0	1
548	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
580	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
622	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	1
636	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	1	0	0	0	1	0	0	0	0	1
637	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 6. An example of rules' confidence after applying them on the test set in MovieLense (for 747 users)

5. RESULTS

In order to assess the extracted rules we have used the support and confidence criteria based on Equations (8) and (9).

5. 1. The Results of MovieLense Dataset

We have extracted the available rules on the users' data and also all the available items of MovieLense dataset with the help of RapidMiner. By placing the value of 0.01 for the rules' support and 0.8 for the rules' confidence, 17 rules are obtained that are shown in Figure 3.

As we can see, 17 rules are generated that all have the support value of 0.01 and confidence of 0.8.

Although the obtained rules do not have enough support value, they are acceptable in terms of confidence. It should be mentioned that, in the generated rules the importance of confidence criterion is more than support criterion. So we are trying to improve the rules' confidence.

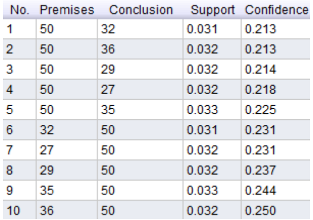
An example of the above rules' confidence after applying them on the test set is shown in Figure 4.

Investigating the results of the applied rules on test set, we can see that in 350 cases the generated rules can be applied on the test set. We will mine the association rules for data of 747 users without changing the support and confidence values. Now we can see that compared to the previous status the generated rules have greatly increased and have reached 2477738 rules. Due to the high number of rules, we only show the rules with higher confidence value in Figure 5.

As we can see, the confidence of many rules is 1 (284521 rules has the confidence value of 1) that shows the generation of suitable rules. As we can see, with more condensation of the test set matrixes, the generated data were more improved. Applying the rules on the test set in 347928 cases, we have found the generated rules applicable on the data (Figure 6). It shows the efficiency of the generated rules.

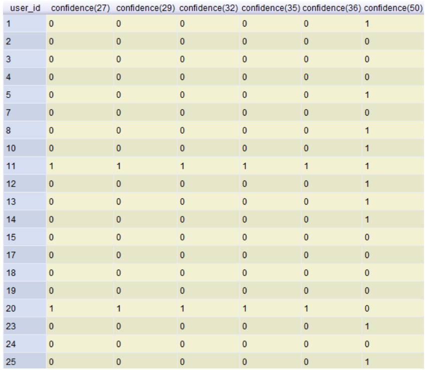
TABLE 2. The comparison of obtained results from the generation of association rules on the MovieLense dataset

	The number of generated association rules with 0.01 support and 0.8 confidence	**The mean of generated rules' confidence**	**The number of cases on which the test set can be applied**
The generated rules on 2113 users' information	17	0.826	350
The generated rules on 747 number of the most active users	2477738	0.856	347928



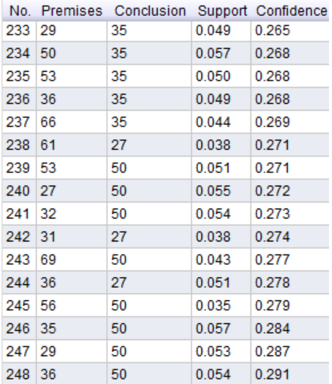
No.	Premises	Conclusion	Support	Confidence
1	50	32	0.031	0.213
2	50	36	0.032	0.213
3	50	29	0.032	0.214
4	50	27	0.032	0.218
5	50	35	0.033	0.225
6	32	50	0.031	0.231
7	27	50	0.032	0.231
8	29	50	0.032	0.237
9	35	50	0.033	0.244
10	36	50	0.032	0.250

Figure 7. The generated rules for Jester dataset (for all the 73421 users)



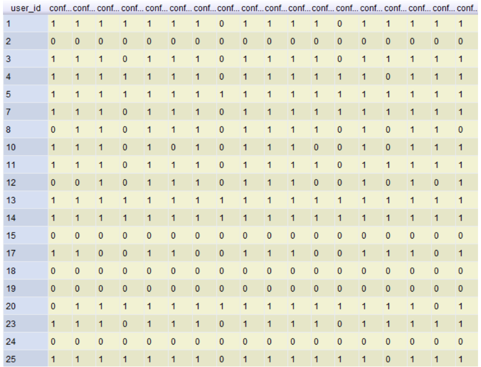
user_id	confidence(27)	confidence(29)	confidence(32)	confidence(35)	confidence(36)	confidence(50)
1	0	0	0	0	0	1
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	1
7	0	0	0	0	0	0
8	0	0	0	0	0	1
10	0	0	0	0	0	1
11	1	1	1	1	1	1
12	0	0	0	0	0	1
13	0	0	0	0	0	1
14	0	0	0	0	0	1
15	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	0	0
19	0	0	0	0	0	0
20	1	1	1	1	1	0
23	0	0	0	0	0	1
24	0	0	0	0	0	0
25	0	0	0	0	0	1

Figure 8. An example of rules' confidence after applying them on the test set in Jester (for 73421 users)



No.	Premises	Conclusion	Support	Confidence
233	29	35	0.049	0.265
234	50	35	0.057	0.268
235	53	35	0.050	0.268
236	36	35	0.049	0.268
237	66	35	0.044	0.269
238	61	27	0.038	0.271
239	53	50	0.051	0.271
240	27	50	0.055	0.272
241	32	50	0.054	0.273
242	31	27	0.038	0.274
243	69	50	0.043	0.277
244	36	27	0.051	0.278
245	56	50	0.035	0.279
246	35	50	0.057	0.284
247	29	50	0.053	0.287
248	36	50	0.054	0.291

Figure 9. The generated rules for Jester dataset (for 23144 users)



user_id	conf...	conf...	conf...	conf...	conf...	conf...	conf...	conf...	conf...	conf...	conf...	conf...	conf...	conf...	conf...	conf...	conf...	conf...
1	1	1	1	1	1	1	1	0	1	1	1	1	0	1	1	1	1	1
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	1	1	0	1	1	1	0	1	1	1	1	0	1	1	1	1	1
4	1	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1	1	1
5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
7	1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1
8	0	1	1	0	1	1	1	0	1	1	1	1	0	1	0	1	1	0
10	1	1	1	0	1	0	1	0	1	1	1	0	0	1	0	1	1	1
11	1	1	1	0	1	1	1	0	1	1	1	1	0	1	1	1	1	1
12	0	0	1	0	1	1	1	0	1	1	1	0	0	1	0	1	0	1
13	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
14	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	1	1	0	0	1	1	0	0	1	1	1	0	0	1	1	1	0	1
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
23	1	1	1	0	1	1	1	0	1	1	1	1	0	1	1	1	1	1
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	1	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1	1	1

Figure 10. An example of rules' confidence after applying them on the test set in Jester (for 23144 users)

Table 2 shows the comparison of obtained results from the generation of association rules on the MovieLense dataset.

Through comparing the available values in the table, we observe that the number of produced rules are significantly increased. On the other hand, the accuracy of rules has not been decreased. Also, the implementation of these rules on the test set has indicated acceptable performance.

5. 2. The Results of Jester Dataset

The extracted rules from the whole data of Jester dataset for the minimum value of support 0.03 and confidence of 0.2 are shown in Figure 7.

Therefore, 10 association rules are generated that have 0.03 support, and the confidence value of these rules is 0.25. We can see that the resulted rules do not have enough confidence and the number of generated rules is low. An example of rules' confidence after applying them on the test set is shown in Figure 8.

Studying the results of applying rules on the test set in 48725 cases we have found 10 generated rules applicable. The low confidence value of the generated rules indicates the inefficiency of the extracted rules. Also, by screening the data we have tried to create better rules. After applying the process on 23144 most active users we have achieved 248 rules. Due to the high volume of these rules we just show a part of these rules in Figure 9.

As we can see, the support and confidence values had not that much improvement compared to the previous status (the support has improved to 0.05 and confidence has improved to 0.29) but the number of rules has increased from 10 to 248 rules. Investigating the results of applied rules on the test set (Figure 10) we have obtained the number of rules that can be applied on the test set and we also have compared them to their previous status.

TABLE 3. Comparison of the obtained results from the generation of association rules on the Jester dataset

	The number of generated association rules with 0.03 support and 0.2 confidence	**The mean of generated rules' confidence**	**The number of cases on which the test set can be applied**
The generated rules on 73421 users' information	10	0.227	48725
The generated rules on 23144 number of the most active users	248	0.235	464400

In 464400 cases the achieved rules in the process were applied in the test set and this amount was higher than the previous status. Table 3 shows the comparison of obtained results from the generation of association rules on the Jester dataset.

As we observed, through sampling from training sets, we could create better rules. In this method, due to smaller training set, finding the association rules can be done with higher speed and at the same time, the quality of the rules improves.

6. CONCLUSION

We applied the generation stages of association rules twice on each dataset, and we achieved a more acceptable result by filtering the information and separating the better data. The results improved due to the fact that we selected the users that we had more information about them and we made the data table denser. We also tried to include more information on the forwarding table to the RapidMiner. Therefore, by increasing the data and making the rating more comprehensive for all the users we can help to produce more and better association rules. Finally, we found that our proposed system is effective for the MovieLense dataset. The obtained results were not that much suitable for Jester dataset but with our proposed methods we tried to improve the quantity and quality of the rules. These results indicate that the effectiveness of the system greatly depends on the input data and the applied dataset. In addition, if the user rates more number of the items the system efficiency will be more increased. The obtained results of these rules can be applied in the recommender systems. Among the advantages of the proposed method, it can be referred to its simplicity and rapid implementation. Moreover, through a sampling from training set, the speed of association rules will be increased.

7. REFERENCE

1. Chandak, M., Girase, S. and Mukhopadhyay, D., "Introducing hybrid technique for optimization of book recommender system", ***Procedia Computer Science***, Vol. 45, (2015), 23-31.

2. Kardan, A. A. and Ebrahimi, M., "A novel approach to hybrid recommendation systems based on association rules mining for content recommendation in asynchronous discussion groups", ***Information Sciences***, Vol. 219, (2013), 93-110.

3. Salehi, M., "Latent feature based recommender system for learning materials using genetic algorithm", ***Information Systems & Telecommunication***, Vol. 2 , No. 3, (2014).

4. Adomavicius, G. and Tuzhilin, A., "Recommendation technologies: Survey of current methods and possible extensions", ***Information Systems Working Papers Series,*** (2003).

5. Al-Shamri, M. Y. H. and Bharadwaj, K. K., "Fuzzy-genetic approach to recommender systems based on a novel hybrid user model", ***Expert Systems with Applications***, Vol. 35, No. 3, (2008), 1386-1399.

6. Jannach, D., Zanker, M., Felfernig, A. and Friedrich, G., "Recommender systems: An introduction, Cambridge University Press, (2010).

7. Bobadilla, J., Ortega, F., Hernando, A. and Gutierrez, A., "Recommender systems survey", ***Knowledge-based Systems***, Vol. 46, (2013), 109-132.

8. Shaeiri, Z. and Ghaderi, R., "Modification of the fast global k-means using a fuzzy relation with application in microarray data analysis", ***International Journal of Engineering-Transactions C: Aspects***, Vol. 25, No. 4, (2012), 283-292.

9. Darvishi, A. and Hassanpour, H., "A geometric view of similarity measures in data mining", ***International Journal of Engineering-Transactions C: Aspects***, Vol. 28, No. 12, (2015), 1728-1737.

10. Chen, M. S., Park, J. S. and Yu, P. S., "Data mining for path traversal patterns in a web environment", in Distributed Computing Systems, 1996., Proceedings of the 16th International Conference on, IEEE, (1996), 385-392.

11. Rupayla, P. and Patidar, K., "A comprehensive survey of frequent item set mining methods", ***International Journal of Emerging Technology and Advanced Engineering***, Vol. 4, (2014), 2250-2459.

12. Vennila, G., Shalini, N. S. and Manikandan, M., "Navie bayes intrusion classification system for voip network using honeypot (research note)", ***International Journal of Engineering-Transactions A: Basics***, Vol. 28, No. 1, (2014), 44-51.

Using a Data Mining Tool and FP-growth Algorithm Application for Extraction of the Rules in Two Different Dataset

TACHNICAL NOTE

E. Hashemzadeh, H. Hamidi

Department of Industrial Engineering, K. N. Toosi University of Technology, Tehran, Iran

PAPER INFO

Paper history:
Received 13 May 2016
Received in revised form 02 June 2016
Accepted 02 June 2016

Keywords:
Recommender Systems
Collaborative Filtering
Association Rules
Support
Confidence

چکیده

در این مقاله قصد داریم قوانین انجمنی را به منظور استفاده در توصیه‌گرها بهبود دهیم. سیستم‌های توصیه‌گر روشی برای ایجاد پیشنهادات شخصی‌سازی‌شده ارائه می‌دهند. یکی از مهم‌ترین انواع توصیه‌گرها تصفیه‌ی همکارانه است که به داده‌کاوی در اطلاعات کاربران و پیشنهاد آیتم‌های مناسب به آن‌ها می‌پردازد. از جمله روش‌های داده‌کاوی، یافتن مجموعه آیتم‌های مکرر و قوانین انجمنی موجود در مجموعه داده است. در این روش، با جداسازی اطلاعات کاربران فعال‌تر، کاربرانی که به آیتم‌های بیشتری ابراز علاقمندی کرده‌اند، از مجموعه‌ی آموزشی نمونه‌برداری می‌کنیم و یافتن قوانین انجمنی را روی نمونه‌ی انتخابی ادامه می‌دهیم. بدین ترتیب با کوچک شدن مجموعه‌ی آموزشی، سرعت تولید قوانین انجمنی بیشتر خواهد شد. در عین حال نشان خواهیم داد کیفیت قوانین تولید شده نیز بهبود خواهد یافت. از مزایای روش پیشنهادی ساده و سریع بودن اجرای روش است، به علاوه با انجام نمونه بردای از مجموعه‌ی آموزشی، سرعت یافتن قوانین انجمنی افزایش خواهد یافت.

doi*: 10.5829/idosi.ije.2016.29.06c.08*