

Data Mining Techniques and Preference Learning in Recommender Systems

Bandar Mohammed^{1,2}, Malek Mouhoub¹, Eisa Alanazi^{1,2} & Samira Sadaoui¹

¹ Department of Computer Science, University of Regina, Regina, Saskatchewan, Canada

² Sponsored by Ministry of Higher Education, Saudi Arabia

Correspondence: Bandar Mohammed, Department of Computer Science, University of Regina, Regina, SK., S4S 0A2, Canada. Tel: 1-306-585-4700. E-mail: mohammeb@uregina.ca; mouhoubm@uregina.ca; alanazie@gmail.com; sadaouis@uregina.ca

Received: July 8, 2013 Accepted: August 29, 2013 Online Published: September 11, 2013

doi:10.5539/cis.v6n4p88

URL: <http://dx.doi.org/10.5539/cis.v6n4p88>

Abstract

The importance of implementing recommender systems has significantly increased during the last decade. The majority of available recommender systems do not offer clients the ability to make selections based on their choices or desires. This has motivated the development of a web based recommender system in order to recommend products to users and customers. The new system is an extension of an online application previously developed for online shopping under constraints and preferences. In this work, the system is enhanced by introducing a learning component to learn user preferences and suggests products based on them. More precisely, the new component learns from other customers' preferences and makes a set of recommendations using data mining techniques including classification, association rules and cluster analysis techniques. The results of experimental tests, conducted to evaluate the performance of this component when compiling a list of recommendations, are very promising.

Keywords: preference learning, recommender system, artificial intelligence, E-commerce, data mining

1. Introduction

Designing an appropriate recommender system, to meet the business needs of clients is the first and foremost consideration of this research. A recommender system for online shopping, based on preference learning, is a potential tool for business development and marketing. In this paper, an online shopping system is extended and based on preference elicitation (Alanazi, Mouhoub, & Mohammed, 2012; Mouhoub, Mohammed, & Alanazi, 2012), to recommend products based on customer suggestions. Recommender systems have significantly increased in the past decade. Preference learning in a recommender system is considered one of the most popular and significant techniques from Information Filtering (Eaton & Wagstaff, 2006; Gemmis, Iaquinta, Lops, Musto, Narducci, & Semeraro, 2009). Information filtering assists in the removal of insignificant information and content that does not need to be stored in a customer profile. When a recommender system is applied, for instance, to learn the interests of users (Eaton & Wagstaff, 2006; Gemmis et al., 2009), it will study and learn some of the user's behavioural aspects in order to generate and recommend a list of products (Eaton & Wagstaff, 2006; Gemmis et al., 2009). Learning the user's preferences is one technique to discover the best outcomes to recommend items (Eaton & Wagstaff, 2006; Gemmis et al., 2009).

Currently, it is important for clients to be assisted with their choices due to the exponential increase in existing data (Gemmis et al., 2009). Adaptive tools, algorithms, and user profiles (Gemmis et al., 2009) are the three most significant components for designing and managing personalized recommendations. The popular recommender systems approaches are Content-Based, Collaborative, Demographic, Knowledge-Based and Hybrid (Suguna & Sharmila, 2013; Tran, Phung, & Venkatesh, 2012; Gemmis et al., 2009). There are many techniques for learning user profiles including probabilistic approaches, neural networks, decision trees and association rules (Deshpande & Karypis, 2004; Gemmis et al., 2009).

The idea of preference learning is easy to understand but challenging to implement. A line of investigation is presented as follows: "Can we learn, and know the preferences of users especially when there are missing data"? Also, "Are there any application platforms or recommender system for online shopping based on learning

preferences”? According to the current position in the field, the answer is no. In addition, there do not appear to be any recommender systems representing both numeric (quantitative) and ordinal (qualitative) preferences.

Preference interactive systems include recommender systems; For instance, Amazon.com and Netflix which recommend products based on a client’s similarities to other clients or viewed products and conversational applications which interact with the client with a simple dialogue to execute a task (Alanazi, Mouhoub, & Mohammed, 2012). However, Amazon.com and Netflix cannot handle the constraints and different forms of preferences such as numeric (quantitative) and ordinal (qualitative) preferences. As well, their learning preference components are limited in comparison to the learning preferences components, constraints and forms of preferences in the present research. Also, they ask clients questions that may not help them state their preferences. Table 1 lists some interactive systems along with their parameters for handling constraints and forms of preferences in comparison to our online shopping system.

Table 1. Comparison in terms of types of information

Application	Quantitative	Qualitative	Mixed	Constraints
Amazon	Yes	No	No	No
Netflix	Yes	No	No	No
SmartClient	Yes	No	No	No
Teaching Salesman	Yes	No	No	No
Project Execution Assistant	Yes	No	No	No
Online Shopping System	Yes	Yes	Yes	Yes

The new learning component added to the present online shopping system will learn some of the customers’ preferences by applying data mining association rules and using the AprioriTid algorithm in order to provide and recommend products to users. The main objective is to recommend products to customers using two different methods. The first method learns some of the user’s preferences and recommends products based on the preferences. The second method recommends products based on the user’s preferences by applying the AprioriTid algorithm and data mining association rules. The system generates general recommendations for all customers with the AprioriTid algorithm which reviews all transactions to correlate frequent itemsets between customers.

The following article is organized as follows. A brief introduction and description of preference learning is presented in the following section. The data mining techniques are discussed with examples in section 3. The description and methodology of the proposed system are respectively presented in sections 4 and 5. In section 6, the results of the experiments are presented to evaluate the performance of the recommender component. Section 6 summarizes the research contributions and suggests some future directions.

2. Preference Learning

The work on preference learning is mostly focused on rankings over individual items and pairwise preferences (Furnkranz & Hallermeier, 2010). It has gained significant attention in the area of artificial intelligence, especially in the last few years. It involves knowledge and learning from explanations that expose information and knowledge about the preferences of a person or group of people (Furnkranz & Hallermeier, 2010). Processing knowledge representing preferences permits anyone to state desires in a declarative method to have qualitative and quantitative forms of reasoning. It deals with exceptions and variations in a flexible manner. This leads to the creation and implementation of models to learn the preferences of users, and might be utilized for prediction of preferences (Furnkranz & Hallermeier, 2010). The idea of preference learning is very easy to understand but hard to manage and implement. Indeed, every problem of learning preferences depends on the number of dimensions used to show the set of potential preferences (Eaton & Wagstaff, 2006; Gemmis et al., 2009). For instance, when the user would like to purchase a laptop or desktop computer, and the only option considered is the user’s budget, the recommendations provided to the user are going to be easy to provide since there is just one dimension. If there are more options, such as the size of the memory or the speed of the processor, the problem becomes harder in terms of managing user preferences in an effective way. A method is needed to consider all the options and generate a list of recommendations to the user. Feedback is required from the user in order to gain information about user preferences to model the preferences by utilizing a particular

representation. The information can be processed with machine learning techniques in order to learn user preferences for use in the recommendation procedure (Eaton & Wagstaff, 2006; Gemmis et al., 2009).

3. Data Mining Techniques

Data mining is the field of extracting valuable information and knowledge from large amounts of data stored in databases. It is the process of finding out formerly unknown, useful and valuable patterns from a large amount of data stored in a database (Kaur & Aggarwal, 2010; Tan, Steinbach, & Kumar, 2005; Han & Kamber, 2006). Database mining deals with the data stored in a database administration scheme/system. The tools and techniques for data mining identify business trends which may occur in the future. It also answers many questions of businesses with regard to time consumption for decision making (Kaur & Aggarwal, 2010). There are two significant reasons why data mining has attracted and gained a lot of attention in the last few years (Kaur & Aggarwal, 2010; Tan, Steinbach, & Kumar, 2005). It has the capability to store and collect a large amount of data while this storage quickly increases every day. As a result of improvements in processing power, there is the potential to store a large amount of relevant data which can be processed anytime. The most significant reason is the need to transform data into useful and valuable knowledge and information (Kaur & Aggarwal, 2010; Tan, Steinbach, & Kumar, 2005; Han & Kamber, 2006). Data mining examines databases in order to discover hidden patterns and valuable information that sometimes experts may not observe as it occurs outside their expectations (Kaur & Aggarwal, 2010; Tan, Steinbach, & Kumar, 2005). The discovered patterns are accessible to the user and could be stored as new information in the information database (Kaur & Aggarwal, 2010; Han & Kamber, 2006; Han, Kamber, & Pei, 2011).

3.1 Data Mining Association Rules

Association rule mining (Kaur & Aggarwal, 2010; Tan, Steinbach, & Kumar, 2005; Han & Kamber, 2006) is a data mining task for finding and discovering hidden associations between items in a transaction. It is a well-known technique to find and discover interesting and attractive relationships between variables and items in large databases (Kaur & Aggarwal, 2010; Han & Kamber, 2006). This method relies on the extraction of an association rule with algorithmic techniques such as the FP-tree, Apriori and AprioriTid algorithms to obtain and generate the appropriate association rules between items in a transaction (Kaur & Aggarwal, 2010; Han & Kamber, 2006; Han, Kamber, & Pei, 2011). More precisely, it is based on association rule evolution by utilizing different measures such as support and confidence factors. Support (s) defines how frequently a rule is appropriate, and applicable to a particular data set, whereas confidence (c) defines how often items in set B appear in transactions containing set A (Tan, Steinbach, & Kumar, 2005). The next two equations are the formal definitions for support (s) and confidence (c) (Tan, Steinbach, & Kumar, 2005):

$$s(A \rightarrow B) = \sigma(A \cup B)/T \quad (1)$$

$$c(A \rightarrow B) = \sigma(A \cup B)/\sigma(A) \quad (2)$$

where s is a support and c is a confidence. A and B are sets, and T is a transaction. σ is the support and confidence count and $\sigma(A)$ is the union count of A:

$$\sigma(A) = |\{t_i \mid A \text{ is a subset of } t_i \text{ and } t_i \in T\}| \quad (3)$$

Association rules are utilized in several areas, such as “medical diagnosis and research, website navigation analysis, churn analysis and prevention, market basket analysis, and retail data analysis” (Kaur & Aggarwal, 2010; Han & Kamber, 2006; Han, Kamber, & Pei, 2011). A classic example is the market basket analysis where retailers identify and analyze what customers would like or prefer to purchase to find an association between items that customers have purchased. Retailers can identify frequent items between customers to aid and assist them in order to plan diverse item placement, advertising and inventory administration (Kaur & Aggarwal, 2010; Tan, Steinbach, & Kumar, 2005; Han & Kamber, 2006). There are many algorithmic techniques used for association rule mining. The most popular are the Apriori, AprioriTid, Partition, FP growth, and Eclat algorithms (Hipp, Guntzer, & Nakhaeizadeh, 2000).

3.1.1 Example of Data Mining Association Rules

Figure 1 shows an example of 10 transactions with 6 itemsets (Dell, Apple, Samsung, Sony, LG, Toshiba). In this example, the method for computing and calculating the support (s), and confidence (c) from 10 transactions with 6 itemsets is shown. As mentioned in section 3.1, support (s) defines how frequently a rule is appropriate and applicable to a particular data set, whereas confidence (c) defines how often items in Y appear in transactions which contain X (Tan, Steinbach, & Kumar, 2005). For example, $\{Apple, Samsung\} \rightarrow \{LG\}$, where $X = \{Apple, Samsung\}$ AND $Y = \{LG\}$. The number of times the subsets $\{LG, Apple\}$ and the subset $\{Samsung\}$ appear in 10 transactions is determined to calculate the support (s) for the following rule: $\{Apple, Samsung\} \rightarrow$

{LG}. The number of times $X \cup Y$ {Apple, Samsung, LG} appear together in the transactions table is divided by T which is the total number of transactions (10) in the example.

$$S = (X \cup Y) / (|T|) = ((Apple, Samsung, LG)) / (|T|) = 3/10 = 0.3 \quad (4)$$

Once the support is calculated, the confidence (c) is calculated for the following rule: {Apple, Samsung} \rightarrow {LG}. The number of times $X \cup Y$ {Apple, Samsung, LG} appear together in the transactions table divided by the number of time the itemsets of X {Apple, Samsung} appear together in the transactions table.

$$C = (X \cup Y) / X = ((Apple, Samsung, LG)) / ((Apple, Samsung)) = 3/3 = 1 \quad (5)$$

Table 2. 10 Transactions with 6 itemsets

TID	Items
1	Apple, Samsung, LG, Sony
2	Dell, Toshiba, Apple
3	LG, Apple, Samsung
4	Apple, Sony
5	Dell, Sony, Toshiba
6	Apple, LG, Sony
7	Samsung, Toshiba
8	Dell, Samsung
9	LG, Apple, Samsung, Toshiba
10	Sony, Samsung, Dell, Toshiba

3.1.2 Apriori Algorithm

The Apriori algorithm (Kaur & Aggarwal, 2010; Han, Kamber, & Pei, 2011) is one of the most significant and classic algorithmic techniques for learning data mining association rules. The main purpose of the Apriori algorithm is to discover associations between different data sets. The Apriori algorithm is implemented for databases that have transactions such as a list of products or items that customers or buyers have purchased from any store or supermarket (Wu, Lu, Pan, Xu, & Jiang, 2009). The main problem with the Apriori algorithm is that it exchanges information with the database in order to compute the number of transactions for each item to occur, lowering its efficiency (Li, He, & Lei, 2005; Wu, Lu, Pan, Xu, & Jiang, 2009). The best solution to this issue is to reduce the number of itemsets in the transactions, and establish a better way to exchange information with the database in order to compute the number of transactions for each item to occur (Li, He, & Lei, 2005; Wu, Lu, Pan, Xu, & Jiang, 2009). The pseudo code, which implements the Apriori algorithm, to generate frequent itemsets (Tan, Steinbach, & Kumar, 2005) is presented below. C_k denotes the set of candidate K – itemsets and F_k indicates the set of frequent K – itemsets. The algorithm makes a single pass over the dataset to define the support for each item. Once the first step is completed, the set of all frequent 1 – itemsets and F_1 will be recognized. Then the algorithm will iterate in order to produce, and create, a new candidate K – itemsets using the frequent $(K - 1)$ – itemsets found in the previous step.

Algorithm

```

1:  $k = 1$ .
2:  $F_k = \{ i \mid i \in I \wedge \sigma(\{i\}) \geq N \times \text{minsup} \}$ .    {Find all frequent 1-itemsets}
3: repeat
4:    $k = k + 1$ .
5:    $C_k = \text{apriori-gen}(F_{k-1})$ .    {Generate candidate itemsets}
6:   for each transaction  $t \in T$  do
7:      $C_t = \text{subset}(C_k, t)$ .    {Identify all candidates that belong to  $t$ }
8:     for each candidate itemset  $c \in C_t$  do
9:        $\sigma(c) = \sigma(c) + 1$ .    {Increment support count}
10:    end for
11:  end for
12:   $F_k = \{ c \mid c \in C_k \wedge \sigma(c) \geq N \times \text{minsup} \}$ .    {Extract the frequent  $k$ -itemsets}
13: until  $F_k = \emptyset$ 
14: Result =  $\bigcup F_k$ .
```

In the following example, the frequent itemsets will be found using the aforementioned algorithm. The following steps, including the list of transactions, are visualized in Figure 1. An item or itemset is frequent if it occurs at least twice. This is the general rule in this example.

Step 1: Calculate the number of transactions for each item to occur.

Step 2: Remove all the items that occur less than twice (the items are identified in red in Step 1 of Figure 2).

Step 3: Generate all the possible pairs of itemsets in Step 2.

Step 4: Calculate how many times each pair occurs in the list of transactions. For instance, {A, L} occur together in {A, L, SO} and {A, SA, L, D}, so they occur twice in this example.

Step 5: Remove all the item pairs appearing less than twice (identified in red in Step 4).

Step 6: Produce a set of three distinct items from the table in step 5.

Step 7: From step 6, calculate the number of transactions for {SA, L, D}. Therefore, the set of three items that are frequently together are {SA, L, D}.

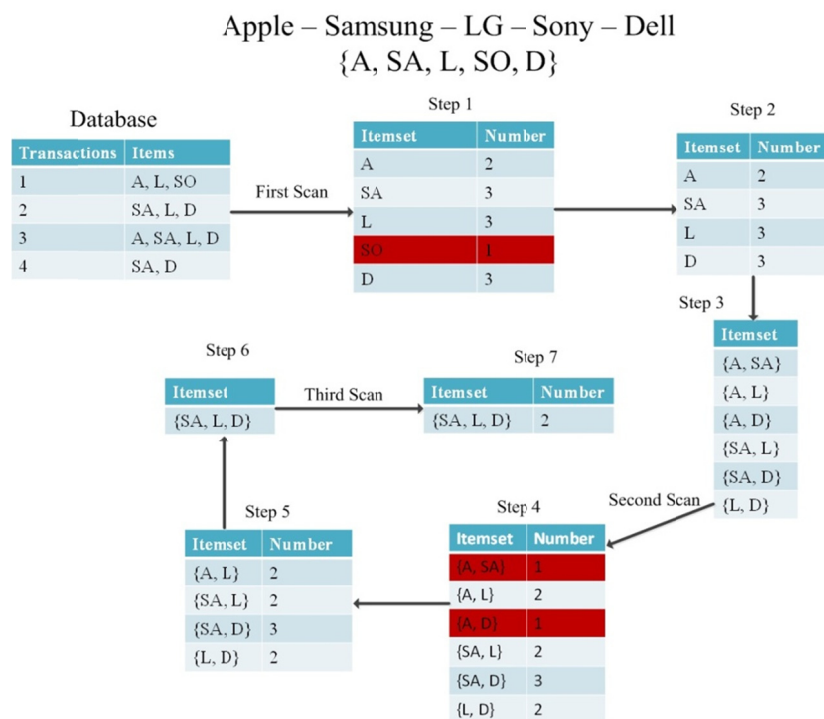


Figure 1. Example of the Apriori algorithm

3.1.3 AprioriTid Algorithm

The AprioriTid algorithm improves upon the classic Apriori algorithm for learning data mining association rules. The main objective of the AprioriTid algorithm is to discover associations among different sets of data (Han, Pei, & Yin, 2000; Wu, Lu, Pan, Xu, & Jiang, 2009). A component is added to improve the efficiency of the AprioriTid algorithm. A database is not used to compute the support or the number of transactions for each item to occur and it is only used once in the first pass (Kaur & Aggarwal, 2010; Wu, Lu, Pan, Xu, & Jiang, 2009; Agrawal & Srikant, 1994). Instead of using the database every time, it utilizes storage \bar{C}_k . This storage is used instead of the database to compute the support each time (Wu, Lu, Pan, Xu, & Jiang, 2009; Li-jian, Li-chao, & shuang-ying, 2003). \bar{C}_k is {TID, set of Itemsets}, where TID is the transaction number associated with the item set or candidate. C_k is the set of itemsets, and L_k is the set of large K-itemsets, and it has two columns (itemsets and support count). Overall, \bar{C}_k provides an advantage for the in the AprioriTid algorithm as it is a more efficient technique to avoid scanning the database (Hipp, Guntzer, & Nakhaeizadeh, 2000; Li-jian, Li-chao, & shuang-ying, 2003). However, the disadvantage of the AprioriTid algorithm is that a similar item will appear in several candidate itemsets for the transaction in storage \bar{C}_k and it will be repeatedly stored, increasing the range of query data (Hipp, Guntzer, & Nakhaeizadeh, 2000; Li-jian, Li-chao, & shuang-ying, 2003). The

AprioriTid algorithm is implemented via the following steps (see also Figure 2):

An item or itemset is frequent if it occurs at least twice. This is the general rule in this example.

Step 1: Create and build a set of {TID, set of Itemsets} for the first item set $\overline{C_1}$

Step2: Define the large first itemsets in L_1 with a minimum support = 2 (less than two occurrences) with two columns (itemsets and support count).

Step 3: Initiate the creation of pairs from the first items (Apple). For instance, {Apple, Samsung}, {Apple, LG}, {Apple, Dell} and repeat the operation for the second item (Samsung). Produce the pairs and get the outcomes in C_2 .

Step 4: Check the itemsets in C_2 with storage $\overline{C_1}$ in order to define $\overline{C_2}$ {TID, set of Itemsets} for the second round or scan.

Step 5: Start the second round or scan in $\overline{C_2}$ to calculate how often each pair occurs together to define the large second itemsets in L_2 with minimum support = 2 (less than two occurrences) with two columns (itemsets and support count).

Step 6: Define C_3 and produce a set of three items from L_2 in order to define $\overline{C_3}$ {TID, set of Itemsets} for the third round or scan. Add one more rule, which is two pairs with a similar first Alphabet. Therefore, {Samsung, LG} and {Samsung, Dell} have the same first Alphabet which produces {Samsung, LG, Dell}.

Step 7: The third round or scan is started in $\overline{C_3}$ to calculate the number of transactions for {Samsung, LG, Dell} in order to define the large second itemsets in L_3 with minimum support = 2 (less than two occurrences) with two columns (itemsets and support count). Therefore, the set of three items that are frequently together are {Samsung, LG, Dell}.

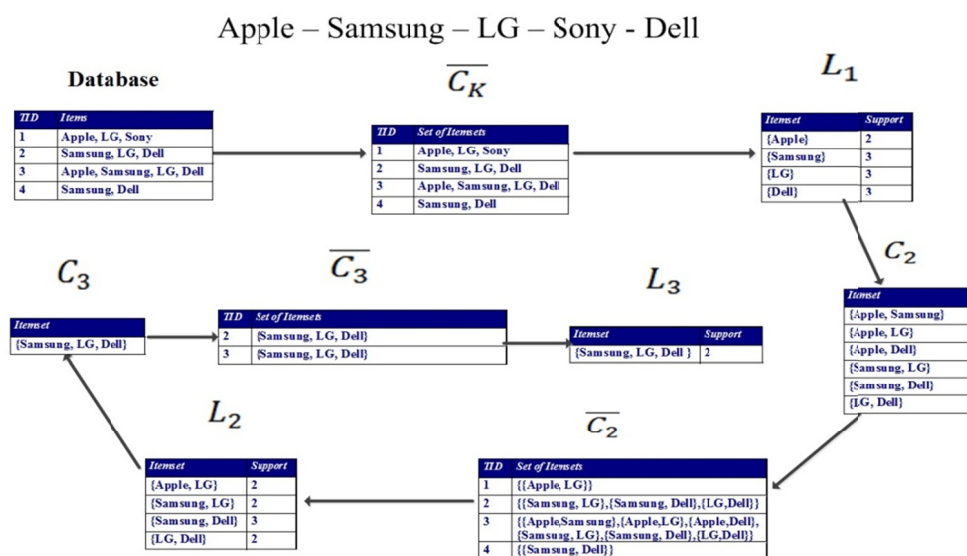


Figure 2. Example of the AprioriTid algorithm

3.2 Data Mining Classification Technique

The general definition of a classification technique is the operation of assigning items to single or many predefined types (Tan, Steinbach, & Kumar, 2005). Also, it is a general problem which includes many applications (Tan, Steinbach, & Kumar, 2005). There are many methods and techniques for solving a classification problem (Tan, Steinbach, & Kumar, 2005). A classification method, also called a classifier, is a methodical technique for constructing classification models from an input data set (Tan, Steinbach, & Kumar, 2005). Examples of such techniques include "Rough set, support vector machines, rule-based classifiers, naive Bayes classifiers, decision tree classifiers, and neural networks" (Tan, Steinbach, & Kumar, 2005). Every technique for solving a classification problem needs to employ a learning algorithm in order to identify a model which can best fit the association between the class label for inserting data and the attribute set (Tan, Steinbach, & Kumar, 2005). The main objective of the learning algorithm is to construct models with sufficient capability to

generalize (Tan, Steinbach, & Kumar, 2005).

3.2.1 Example of Classification Technique

A data mining classification technique is applied to the recommender system in order to classify the users as groups based on their budgets. Seven users have stated their budgets in the column Budget of Figure 4. The users are grouped according to their budgets, and once the recommender system groups the users, it recommends products and items based on their budgets.

Table 3. Information and budgets for 7 users

TID	First Name	Last Name	Email	Budget
1	Allan	Cooper	a@a.com	>500
2	John	Smith	j@j.com	>1500
3	Zhang	Ming	z@z.com	>1000
4	Eisa	Ayed	e@e.com	>1500
5	Yao	Hua	y@y.com	>500
6	Jonathan	William	w@w.com	>1000
7	Bandar	Ghalib	b@b.com	>500

3.3 Data Mining Cluster Analysis Technique

The cluster analysis technique (Tan, Steinbach, & Kumar, 2005) splits data into groups or clusters which are useful or expressive. The cluster analysis technique groups data items according to information which has been found in the data that define the items and their relationships (Tan, Steinbach, & Kumar, 2005). The main goal of the cluster analysis technique is to separate the clustering (Tan, Steinbach, & Kumar, 2005). Nevertheless, there are some cases where the cluster analysis technique is only useful for obtaining a result for other purposes; for instance, data summarization (Tan, Steinbach, & Kumar, 2005). Cluster analysis techniques have played a significant part in a wide variety of fields including machine learning, information retrieval, pattern recognition, statistics, psychology, biology, data mining, and social science (Tan, Steinbach, & Kumar, 2005). There are several techniques for cluster analysis data mining including K-means and Agglomerative Hierarchical Clustering (Tan, Steinbach, & Kumar, 2005).

3.3.1 Example of Cluster Analysis Technique

A cluster analysis technique can be applied to a recommender system to recommend products based on similarities between users. The users are clustered when they have similar products in their wish list or if the users purchased many similar, or the same, products. Then, the recommender system will recommend products to the users who have purchased similar products. For example, assume users A and B have purchased three similar products from Dell, and user A has purchased two more products from Apple. User B has similarities user A because they purchased three similar products from Dell, and in this case, the recommender system will recommend two products from Apple to user B. The recommender system does the same if users have similar products in their basket. The recommender system uses a basic K-means algorithm to execute the cluster analysis technique. The following algorithm describes the basic K-means algorithm (Tan, Steinbach, & Kumar, 2005).

Algorithm

- 1: Select K points as initial centroids.
 - 2: repeat
 - 3: Form K clusters by assigning each point to its closest centroid.
 - 4: Recompute the centroid of each cluster.
 - 5: until Centroids do not change.
-

4. System Description

Our proposed recommender system has been implemented using the following tools and environment.

- Programming languages: JavaScript, ASP.NET, C#, and HTML.
- Tools and environments: Microsoft Visual Studio 2010 and 2012 “ASP.NET - C#”.
- Database system: SQL Server Database 2012.
- Windows operating system supporting .NET Runtime Version ASP.NET 4, and PHP Version (PHP 5.2, PHP 5.3).

As illustrated in Figure 3, the online shopping system is designed with three layers.

The first layer is a graphic user interface where the user can register or create a new account in order to sign in and use the system’s features. Figures 9 and 12 belong to this layer. This layer also shows the results of the user queries and additional information. The user can choose the representation or elicitation of preferences or learning of preferences as the system learns the user’s preferences (Figure 6 provides a screenshot of the graphic user interface).

The second layer is a business layer where the user can interact or deal with two types or pages. The first page is the Quantitative and Qualitative Mixed page, and the second page is the Data Mining Task for learning of preferences.

The third layer is a database layer where the user can save data and import and export them into a database. A web server associated with the domain on Godaddy Web Hosting was reserved for the online system. The features for web hosting 150 GB of space, unlimited websites and bandwidth, 500 email Accounts, and SQL Server 2005 and MySQL databases.

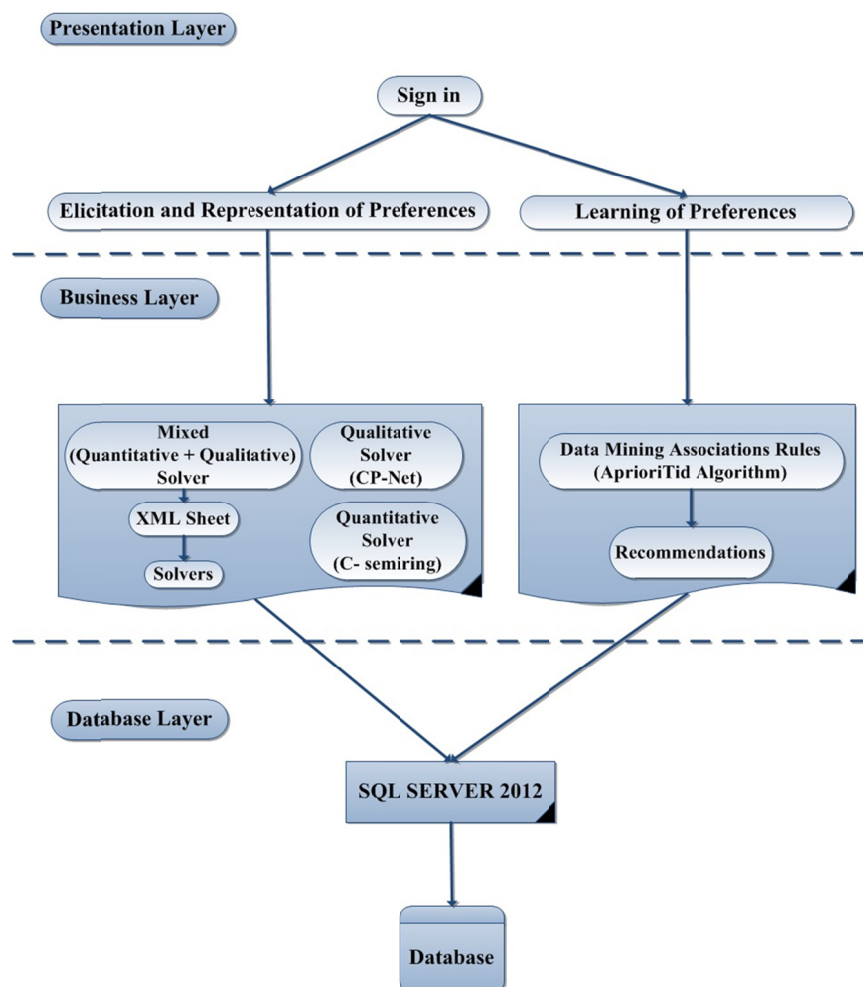


Figure 3. Software architecture of the online shopping system

5. System Methodology

An online shopping system has been proposed to take into account user preferences when looking for an optimal solution (Alanazi, Mouhoub, & Mohammed, 2012; Mouhoub, Mohammed, & Alanazi, 2012). The objective is to decide which configuration (i.e. combination of items) is the most preferred according to a set of preferences while respecting some requirements provided by the user. Requirements and preferences are represented through the Semiring-based Constraint Satisfaction Problem (SCSP) and CP-nets, respectively.

The previous work has been extended to investigate the applicability of learning some user preferences. A new component is proposed to learn preferences and suggest products for users according to their interests. For the time being, only one attribute (brand) is considered for new users. It is assumed there is a set of existing user profiles. Then the AprioriTid algorithm is applied to generate the frequent itemsets or products. Once the user is registered and an account is created, the user can choose his or her preferred brands. For example, the user may show interest in one of the following brands: Apple, Samsung, LG, Dell, Toshiba and Sony. The system learns the user's interests in order to recommend new products. It is assumed the user is able to specify the preferences or brands as shown in the graphic user interface illustrated in Figure 4, and the minimum support is equal to 20%.

It is also assumed there are four transactions or registered users, and that the users have chosen their brands as shown in Table 4. Figure 5 shows the itemsets represented with a lattice, and the minimum support is 20%. Each node has its support number in order to compare it with the minimum support of 20% to reduce the size of the tree to find the frequent itemsets. The interests for the first transaction are {Apple, LG, Sony}, and the interests for the second transaction are {Samsung, LG, Dell}. Once the system learns the user preferences, it will apply the AprioriTid algorithm and generate the frequent itemsets in order to suggest products to all users who share common interests as shown in Figure 6. The system will generate a specific page called "customer's recommendation" to each user in order to suggest products based on their preferences as shown in Figure 6.

Figure 4. Preference learning page

Table 4. 4 Transactions with 5 itemsets

Database	
TID	Items
1	Apple, LG, Sony
2	Samsung, LG, Dell
3	Apple, Samsung, LG, Dell
4	Samsung, Dell

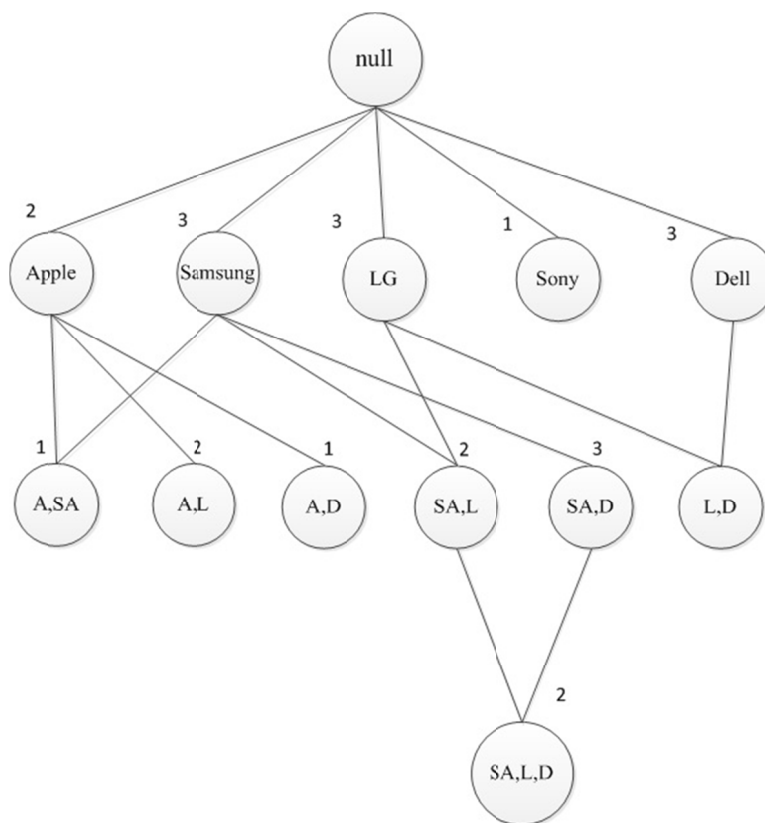


Figure 5. Reduce the size of the tree

A Proposed Recommender System under Association Rules and Preference Learning

Main Menu

- ☐ Home Page
- ☐ About us
- ☐ Contact us
- ☐ Supervisor
- ☐ Admin
- ☐ My Search
- ☐ My Interests
- ☐ My Recommendations

Welcome! Bandar [Logout](#) Most Common Recommendations [LG](#) [Samsung](#) [Dell](#)

My Recommendations

Products

ProductId	BrandName	ProdTypeName	ProductName	Specifications	
1	Apple	Laptop	MacBook Pro	New Version	Select
2	Apple	Laptop	MacAir	New	Select
3	SAMSUNG	TV	SmartTV	55 Inch	Select

Id	Product Name	Component Name	
1	MacBook Pro	Memory	Select
2	MacBook Pro	HardDrive	Select
3	MacBook Pro	Screen	Select

Id	Component Name	Type Name	Price
5	Screen	15" Inch	150

33 Visitors
17 Aug 2012 - 5 Dec 2012

ClustrMaps® Click to see

Copyright © 2012/2013 - Bandar Ghalib Mohammed

Figure 6. The user's recommendation page

6. Experimentation

The average time required to recommend new items to users is presented in this section. Two experiments have been conducted with two different datasets. The first experiment is based on synthetic data generated below. The second experiment is based on a dataset imported from the UCI KDD Archive (UCI Knowledge Discovery in Databases Archive - <http://kdd.ics.uci.edu/>). The experiments were performed on the online store website in order to determine the average time needed to obtain the list of recommendations (frequent itemsets). The experiments were completed via an abstract solver containing two mini solvers for CSP, CP-net (Alanazi, Mouhoub, & Mohammed, 2012) and AprioriTid. The system is implemented with the C# programming language in a Microsoft Visual Studio 2010 environment operating on a Windows 7 operating system. The experiments were conducted on an Intel Core i7 PC with a 2.7 GHz CPU and 4 GB of RAM.

The synthetic data experiments were performed on a dataset with 6 k-itemsets {Apple, Samsung, Dell, Sony, Toshiba and LG} and a varying number of transactions, from 5 to 40. It is assumed the threshold or minimum support is equal to 30% where the itemset is frequent if it is chosen at least 3 times. Figure 7 shows the time needed (in milliseconds) to find the frequent itemsets with a varying number of transactions. Experimental tests conducted on random datasets show the running time of the AprioriTid algorithm increases when the minimum support has a low value and decreases when its value is high.

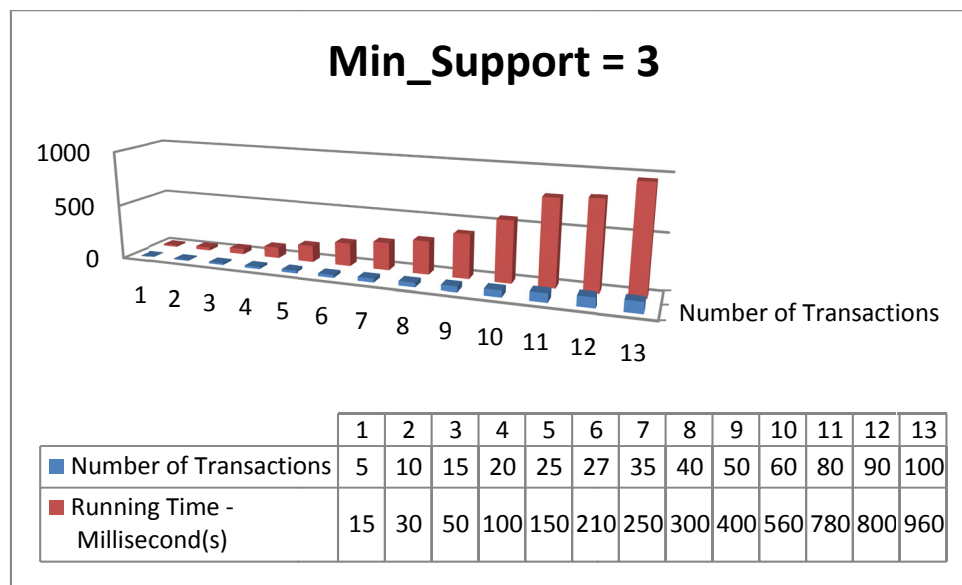


Figure 7. Running time needed to generate frequency

The second experiment was performed with different transactions from the UCI KDD archive in order to obtain the frequent itemsets. Figure 8 represents a sample dataset with 1199 transactions tested with the AprioriTid algorithm in order to obtain the frequent itemsets. The number of features in the data is 257 but Figure 9 displays additional features. Figure 10 presents four charts for the dataset with varying minimum supports (5, 7, 10 and 15) along with the time required in seconds, to find the frequent itemsets. The running time increases when the minimum support has a lower value and decreases when the minimum support is higher.

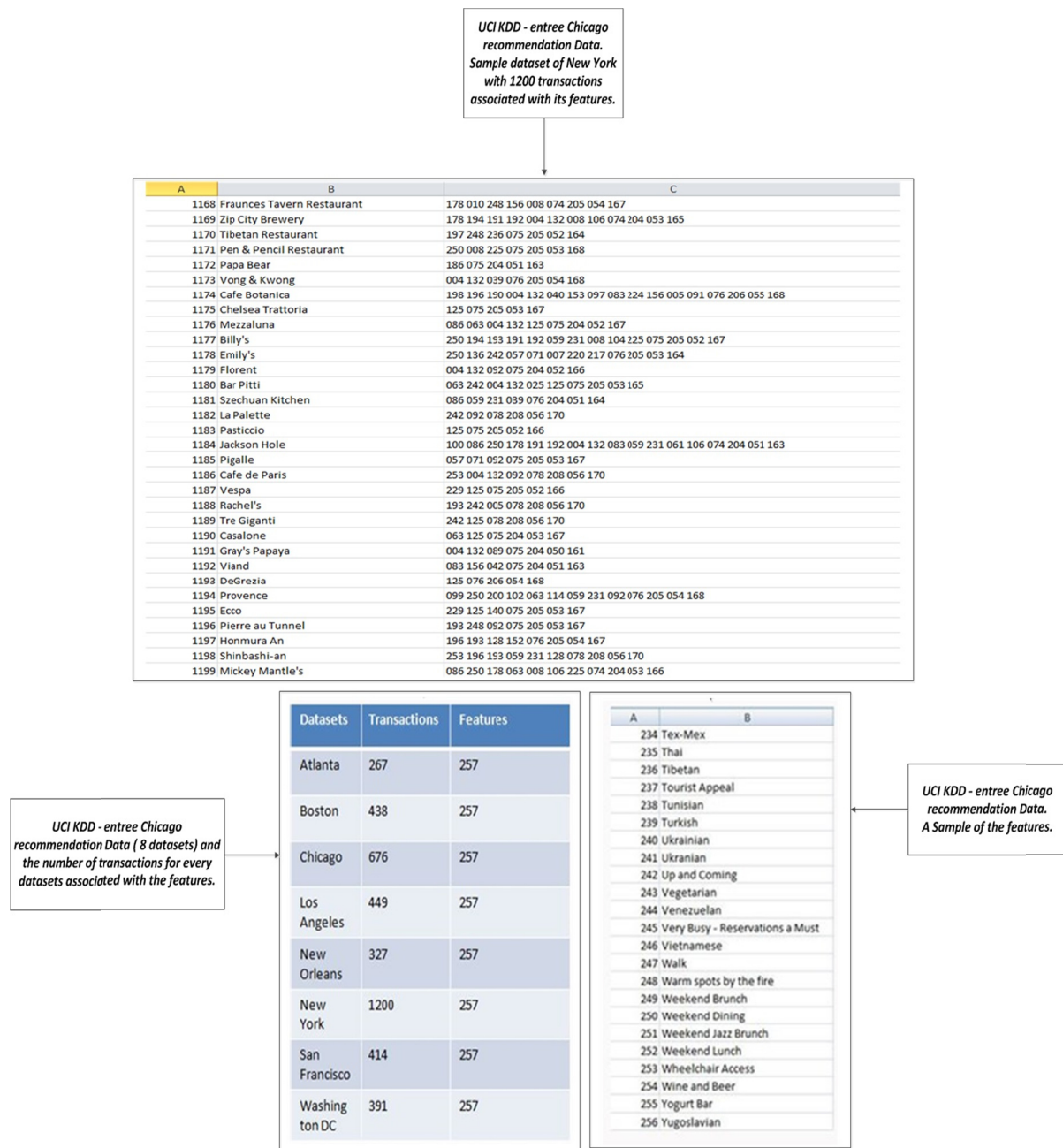


Figure 8. UCI KDD datasets, and sample of dataset and sample of features

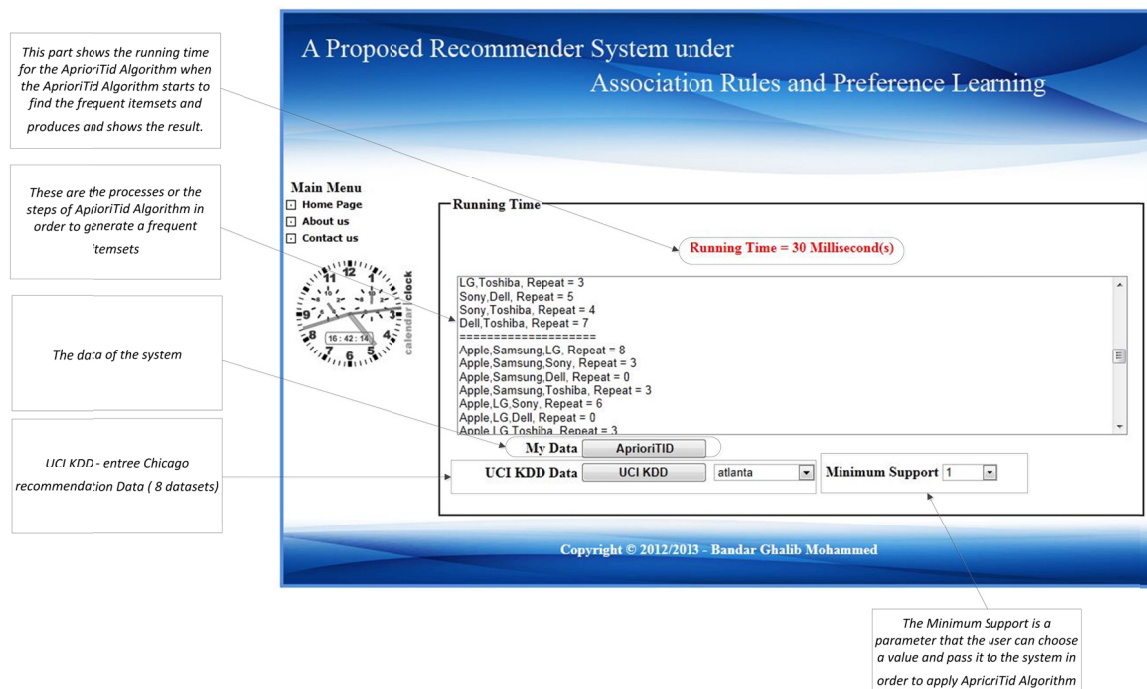


Figure 9. Running time should be inside the circle in the GUI

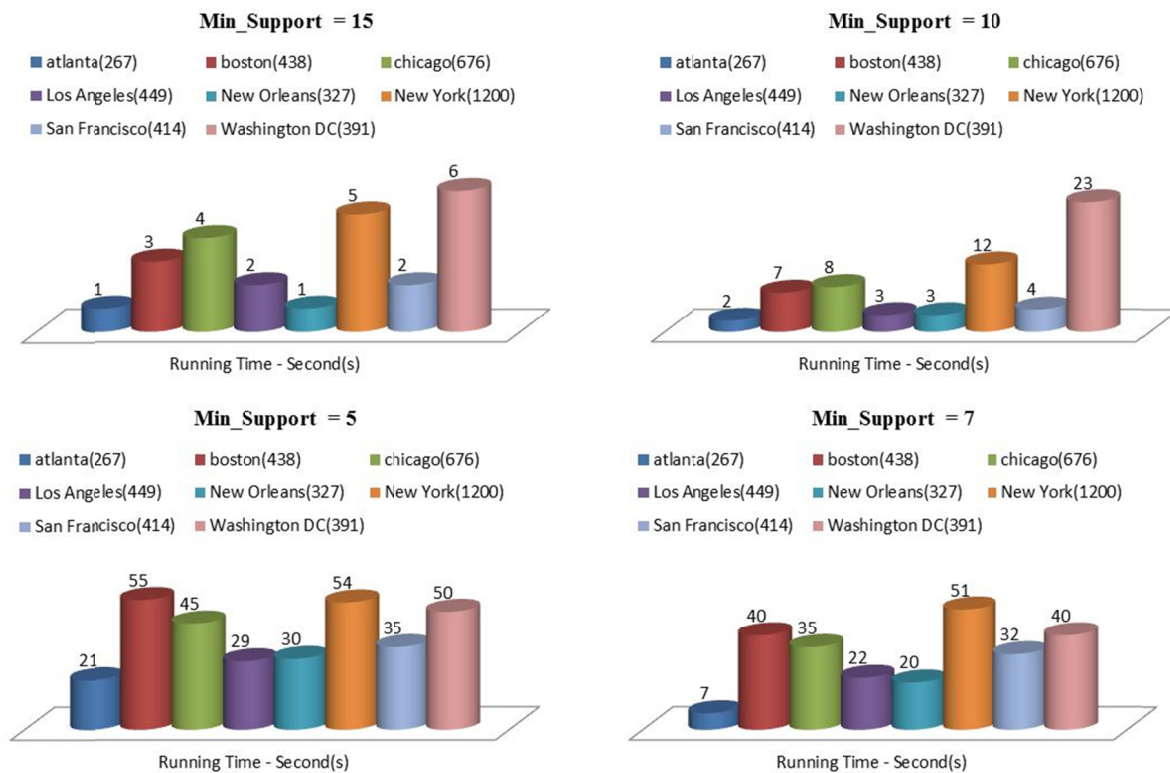


Figure 10. Running time for UCI KDD datasets with threshold: 5, 7, 10, and 15

7. Conclusion and Future Work

A recommender system based on user preferences and constraints has been proposed. The AprioriTid algorithm has been used to find frequent itemsets in order to recommend products to users. Experimental tests conducted on random datasets show the running time of the AprioriTid algorithm increases when the minimum support has

a low value and decreases when the value is high. Additional solutions and algorithmic techniques can be used in this system in order to recommend products, and make it easier for users. New techniques for preferences and constraints can be implemented and tested on the system to see if the latter can handle more complex preferences and constraints. This recommender system can be generalized, and added to any interactive recommender application where the user and customer are involved in the procedure of choosing their products.

References

- Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules. *Proceedings of the 20th Int'l Conference on Very Large Databases* (pp. 487-499). New York: IEEE press.
- Alanazi, E., Mouhoub, M., & Mohammed, B. (2012). A preference aware interactive system for online shopping. *Computer and Information Science*, 5(6), 33-42. <http://dx.doi.org/10.5539/cis.v5n6p33>
- Boutilier, C., Brafman, R. I., Hoos, H. H., & Poole, D. (1999). Reasoning with conditional ceteris paribus preference statements. In *UAI* (pp. 71-80).
- Brafman, R. I., & Domshlak, C. (2009). Preference handling - an introductory tutorial. *AI Magazine*, 30(1), 58-86.
- Cheung, D. W., Ng, V. T., Fu, A. W., & Fu, Y. (1996). Efficient Mining of Association Rules in Distributed Databases. *IEEE Transactions on knowledge and Data Engineering*, 8(6), 911-922. <http://dx.doi.org/10.1109/69.553158>
- Chevalayre, Y., Endriss, U., Lang, J., & Maudet, N. (2008). Preference handling in combinatorial domains: From ai to social choice. *AI Magazine*, 29(4), 37-46.
- Deshpande, M., & Karypis, G. (2004). *Item-based top-n recommendation algorithms* (pp. 143-177). ACM Transactions on Information Systems.
- Eaton, E., & Wagstaff, K. L. (2006). Learning user preferences for sets of objects. In *Proceedings of the Twenty-Third International Conference on Machine Learning* (pp. 273-280). ACM.
- Furnkranz, J., & Hallermeier, E. (2010). *Preference Learning* (1st ed.). New York, NY, USA: Springer-Verlag New York, Inc.
- Gemmis, M. D., Iaquinta, L., Lops, P., Musto, C., Narducci, F., & Semeraro, G. (2009). Preference learning in recommender system. In *In Preference Learning (PL-09) ECML/PKDD-09 Workshop*.
- Han, J., & Kamber, M. (2006). *Data Mining: Concepts and Techniques* (2nd ed.). Morgan Kaufmann.
- Han, J., Kamber, M., & Pei, J. (2011). *Data Mining: Concepts and Techniques* (3rd ed.). Morgan Kaufmann.
- Han, J., Pei, J., & Yin, Y. (2000). Mining Frequent Patterns without Candidate Generation. In *ACM SIGMOD International Conference on Management of Data*. Dallas. <http://dx.doi.org/10.1145/335191.335372>
- Hipp, J., Guntzer, U., & Nakhaeizadeh, G. (2000). Algorithms for Association Rule Mining – A General Survey and Comparison. *SIGKDD Explorations*, 2(1), 58-64. <http://dx.doi.org/10.1145/360402.360421>
- Kaur, G., & Aggarwal, N. (2010). Association Rule Mining in XML databases: Performance Evaluation and Analysis. *International Journal on Computer Science and Technology*, 1(2), 33-37.
- Li, Z. C., He, P. L., & Lei, M. (2005). A High Efficient AprioriTid Algorithm for Mining Association Rule. *Proceedings of the Fourth International Conference on Machine Learning and Cybernetics* (pp. 18-21). Guangzhou.
- Mouhoub, M., Mohammed, B., & Alanazi, E. (2012). Approximating user preferences in online store systems. In *CCECE* (pp. 1-4). Montreal, Quebec, Canada.
- Peng, Y., & Xiong, Y. (2005). Improvement of AprioriTid algorithm for mining association rules. *Journal of Computer Applications*, 25(5), 979-981.
- Rossi, F. (2005). Preference reasoning. In *CP* (pp. 9-12).
- Suguna, R., & Sharmila, D. (2013). An Efficient Web Recommendation System using Collaborative Filtering and Pattern Discovery Algorithms. *International Journal of Computer Applications*, 37-44.
- Tan, P. N., Steinbach, M., & Kumar, V. (2005). *Introduction to Data Mining* (1st ed.). Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.
- Tran, T., Phung, D. Q., & Venkatesh, S. (2012). *A Sequential Decision Approach to Ordinal Preferences in Recommender Systems*. Toronto, Ontario, Canada: Association for the Advancement of Artificial

Intelligence.

Walsh, T. (2007). Representing and reasoning with preferences. *AI Magazine*, 28(4), 59-70.

Wu, H., Lu, Z., Pan, L., Xu, R., & Jiang, W. (2009). An improved Apriori-based algorithm for association rules mining. In *Proceedings of the 6th international conference on Fuzzy systems and knowledge discovery - Volume 2, FSKD'09* (pp. 51-55). Piscataway, NJ, USA: IEEE Press.

Copyrights

Copyright for this article is retained by the author(s), with first publication rights granted to the journal.

This is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).