An Automated Recommender System for Course Selection

Amer Al-Badarenah

Computer Information Systems Department
Jordan University of Science and Technology
Irbid 22110, Jordan

Jamal Alsakran

Computer Science Department
University of Jordan
Amman 11942, Jordan

Abstract—**Most of electronic commerce and knowledge management` systems use recommender systems as the underling tools for identifying a set of items that will be of interest to a certain user. Collaborative recommender systems recommend items based on similarities and dissimilarities among users' preferences. This paper presents a collaborative recommender system that recommends university elective courses to students by exploiting courses that other similar students had taken. The proposed system employs an association rules mining algorithm as an underlying technique to discover patterns between courses. Experiments were conducted with real datasets to assess the overall performance of the proposed approach.**

Keywords—collaborative recommendation; association rule mining; data mining; recommender system; course selection

I. INTRODUCTION

In our daily life, we make our choices at most cases relying on recommendations from newspapers, people, or the Internet (e.g., book reviews, movie, restaurant rating, etc.). However, as the amount of information available on the Internet grows, searching for and making decisions about information becomes difficult. New technologies are required to assist Internet users to cope with information overload. Recommender systems [1, 2, 3, 4, 5, 6, 7, 8] have been an important application area and the focus of considerable recent academic and commercial interests. They are widely used by many commercial and non-profit web sites to help users to select items based on users' preferences. The goal of a recommender system is to provide recommendations that users will evaluate favorably and accept [9].

The main steps a recommender system utilizes to propose an item to a user include analyzing user data, extracting useful information, and finally predicting items to users [10]. By and large, recommender systems use combinations of different kinds of user data, including item ratings [11, 12, 35], item features [13, 14], purchase history [15, 16], user demographic data [17, 18], text comments [19, 20], and contextual information [21, 22]. Ideally, a recommender system should allow users to find the preferable items quickly and to alleviate the information overload problem [24, 25]. Currently, a wide range of electronic commerce web sites that sell products, such as books, movies, music, vehicles, and rental apartments, employ online recommendation to fulfill customers' needs [16, 26, 27, 28].

A. Classifications of Recommender Methods

In general, recommender methods are usually grouped into broad four categories, based on how recommendations are made [4, 15]: collaborative [5, 10, 12, 13, 29, 40, 62, 63], content-based [11, 17, 21, 32, 33, 34], knowledge-based [27, 65, 66, 67, 68, 69], and hybrid recommender approaches [4, 17, 37, 46, 70]. An excellent survey of different recommender systems for various applications can be found in [30]. Pure content-based recommender methods typically propose items to a target user based on affinity between items' contents and the user profile, ignoring data from other users. On the other hand, in pure collaborative recommender methods, items are recommended to a target user based on similarities with other users' preferences (e.g., purchase histories and user ratings), ignoring items' features. Knowledge-based recommender methods exploit inferences, often adopting techniques from artificial intelligence, to deduce a match between user and item [13]. Knowledge-based methods use deep knowledge about features of items rather than users' ratings. To overcome deficiencies associated with pure recommender systems, hybrid approaches based on combining of collaborative recommender method with content-based or knowledge-based approach were proposed [46, 74].

A. Content-Based Recommender Methods

Content-based recommenders propose items (e.g., articles, music, images) to a target user based on similarities between the content of the yet unseen items and the user's preferences (i.e., items that the user has liked in the past). For example, the system may try to correlate the occurrences of keywords in a web page with the user's interest [36]. A profile of the user's interests is created by analyzing the content of the items that have been rated by the user. Later on, when a user interacts with the recommender system, the items proposed are similar to those items in the user's profile. A variety of techniques from approximation theory, machine learning, and various heuristics have been used for analyzing the items' content and finding regularities in this content that can serve as the basis for making recommendations. Commonly used techniques are nearest neighbor formation [33], Bayesian classifier [38, 39], Neural Networks [40], and Association Rule Mining [41]. Several existing content-based recommender systems have been employed in industry. Among these systems are ACR News [42], InfoFinder [43], and WebSail [44].

A pure content-based recommender system has some drawbacks. Specifically, the content feature extraction methods

are suitable to analyze only textual data [45]. On the other hand, applying content-based systems to non-textual data, say multimedia (e.g., video and audio streams), has remained a difficult task. Such data needs to be manually annotated with content information. A second problem, known as over-specialization [46], is that content-based systems can only propose items that are closely related to the user's profile; therefore, the user will only get to see items that are similar to previous items that the user has indicated interest in. However, in certain cases, the user may be interested in other potential items that are outside his usual preferences. One of the ways to overcome this problem is to filter out high-relevance items if they are too similar to the user profile [47]. Another way to tackle this problem is to use the diversity, in addition to the similarity, for ranking items for recommendation [48]. Finally, a new user with an empty profile has to judge an adequate number of items in order to get a reliable recommendation. This is often referred to as a new user problem, since a new user, having very few ratings, often is not able to get appropriate recommendations [21].

B. Collaborative Recommender Methods

Collaborative recommender systems recommend items to a target user based on similarity between past preferences of the target user and other similar users. Unlike content-based systems, collaborative systems predict items based on users' ratings and not machine analysis of content. Typically, users are grouped together by employing methods, like clustering [49], based on their preferences. Once the clusters are created, the clusters that have the strongest correlations with the target user are used in making recommendations for a user. Different analytical methods have been used in collaborative recommender systems to represent affinities among users' preferences. Commonly used methods are correlation-based [19], Bayesian network [56], and association rules techniques [57, 58]. Most recently, collaborative recommender systems have gained much attention from academia and industry. Among these systems are GroupLens [19], Ringo [45], the Bellcore Video Recommender [35], the Jester system, which recommends jokes [59], and the PHOAKS system that assists users finding relevant data on the Web [60].

Some of the problems of the content-based systems described before can be fixed using collaborative systems, since collaborative systems do not depend on error-prone machine analysis of the content. The system proposes items to users based on quality of items (i.e., preferences generated by users' ratings), rather than more objective properties of the items themselves (i.e., items' content). This makes the system capable of recommending items to the user, which are very different (content-wise) from what the user has indicated liking before [45]. Unlike content-based systems, a collaborative recommender system can deal with contents of variety of types (e.g. text, artwork, music, video clip); which can enhance the quality of recommendations.

Although collaborative recommender systems have been successfully utilized in both practice and research, they still suffer from some problems including the early rater problem, the sparsity problem, and the gray sheep problem. The early rater problem (also called the cold start problem [53]) refers to the case when recommendations are required for a new item that has not been yet rated or for a user that just starts to use the system. In this case, the collaborative recommender does not work well, since there is no much rating information on either the item or the user. The sparsity problem, which can also be related to the rater problem [54], occurs when available items ratings are insufficient for identifying similarities between users, leading to poor recommendation. This problem is also known as reduced coverage [26]. The gray sheep problem, on the other hand, occurs when a poor recommendation may be proposed to users with odd preferences compares to the rest of the users, since there will not be any other user who has similar preferences [46].

C. Knowledge-Based Recommender Methods

Knowledge-based recommendation approaches are especially appropriate for domains where profound product knowledge is needed to recognize and justify solutions; like financial services for instance. In comparing users purchasing simple items like videos or books, with users purchasing financial services, the latter are much more in need of information and of intelligent interaction mechanisms that support appropriate solutions. Therefore, explicit representation of product, marketing, and sales knowledge is needed [12].

Knowledge-based recommender systems use inferences, often adopting techniques from AI, to infer a match between a buyer and a product based on the features of a product rather than the buyers' ratings. Knowledge-based recommender systems are receiving increasing research attention [4, 5, 12, 22, 24, 39]. Many knowledge-based recommender systems that are in existence are an Multimedia enhanced product recommendation [24], an integrated Natural Language Interfaces with personalized recommender system to reduce system-user interactions applied to a restaurant recommender system [39].

D. Hybrid Recommender Methods

Because of the deficiencies of pure recommender systems, several hybrid recommender systems, that combine collaborative methods with content-based or knowledge based approaches, were proposed. The main goal of a hybrid system is to improve recommendation accuracy as well as to avoid certain drawbacks (e.g., new item and new user problems) of traditional recommender approaches. In his study of hybrid recommender systems [4], Burke describes a taxonomy that consists of seven types of hybrid recommender systems: weighted, switching, mixed, feature combination, cascade, feature augmentation, and meta-level. The study pointed out that most hybrid systems involve the combination of collaborative recommendation with either a content-based or a data mining technique.

This paper presents a novel recommendation methodology that recommends elective courses to a target university student based on affinity between courses taken by the target student and other students. The proposed method is based on collaborative recommender approach that employs association rule mining to discover courses' patterns in order to recommend courses. In this study, clustering is first employed to a group of students based on their courses' grades, then nearest neighborhood is applied to select the students' group

that is most similar to the target student, and finally, association rule mining is used to provide course recommendations to the target student.

The rest of the paper is organized as follows. Section 2 describes background information and related work. Section 3 describes the proposed courses recommendation system. Section 4 presents experimental results with discussion and finally in Section 5 we conclude the paper and suggests possible future work.

II. RELATED WORK

This section first presents a general formalization of collaborative filtering (CF) and its applicability to course recommendation problem. All the symbols used throughout the paper are summarized in Table 1.

A. Collaborative Recommendation

Collaborative filtering (CF) techniques have been proven to provide satisfying recommendations to users [35, 45]. CF-based techniques rely on the experiences of similar users, i.e., users who share the same preferences. Specifically, the CF methods recommend the target user new items that have been chosen by those users whose preferences are likely to coincide (at least to some extent) with the target user's preferences. Thus, the goal of a CF algorithm is to predict the rating of an item for a particular user, given the same item's ratings of other users. The nearest-neighbor users are those that exhibit the strongest relevance to the target user. Most of CF-based algorithms use *k*-nearest-neighborhood algorithm to recommend items [56]. A typical nearest-neighborhood method uses the following steps in making a recommendation to a user [77].

1) Construct a profile vector PV for a target user ut by collecting his ratings of some items. PV(ut) = {R(ut, i) for some i ∈ I}

2) Compute the pair-wise similarity S(ut, ua) between this profile PV(ut) and the profile of each other user PV(ua). The actual definition of similarity function S depends on the CF algorithm used.

3) Construct a list of target user's nearest neighbors NN(ut) sorted descending by similarity value and take the k users most similar to user ut. NN(ut) = Top(S, ut, n, k, minSim), where Top function returns the top k similarity values S(ut, ua) between user ut and any user ua such that S(ut, ua) ≥ minSim, a = 1, …, n, and a ≠ t.

4) Use the nearest neighbor list to calculate a predication rating P(ut, i) for a new item i for user ut. If P(ut, i) ≥ minPred, then recommend item i to user ut.

B. Association Rules Mining

The association rule mining (ARM) has received considerable attention over the last decade. The task of ARM is to find the correlations between items in a dataset by discovering items frequently appeared together in a transactional dataset. Formally, ARM is defined as follows [78].

Let $I = \{i_1, i_2, …, i_m\}$ be a set of items. Let $D = \{T_1, T_2, …, T_n\}$ be a set of transactions, where each transaction T is a set of items such that $T \subseteq I$. An association rule is an implication of the form $X \Rightarrow Y$, where $X \subset I$, $Y \subset I$, $X \cap Y = \varnothing$, and $X \neq Y \neq \varnothing$. The rule $X \Rightarrow Y$ holds in the transaction set D with confidence c if $c\%$ of transactions in D that contains X also contain Y. The rule $X \Rightarrow Y$ has support s in the transaction set D if $s\%$ of transactions in D contains $X \cup Y$.

Confidence and support are the two fundamental quality heuristics that are used to measure the interestingness of the association rule. The confidence measures the validity of $X \Rightarrow Y$ as a rule; the lesser the exceptions to the rule, the greater its validity. The support measures the efficiency of the rule. The rules that have confidence and support greater than the user specified minimum confidence and minimum support are called *interesting rules*.

Since the introduction of association rules problem in [78], considerable work has been devoted to design efficient algorithms for mining such rules [79, 80]. These algorithms achieved significant improvements over the previous algorithm and were applicable to large databases. Recent survey of association rules mining can be found in [81].

C. Recommendations Using Association Rules Mining

This section describes briefly some collaborative recommendation systems that employ association rule mining. Mobasher et al. in [71] proposed a technique for web personalization based on association rule discovery from previous user's transaction data. In this technique, recommendations are produced based on matching the current user session against patterns discovered through association rules on user transaction data. Changchien et al. in [57] presented a method to support on-line recommendation by customers and products fragmentation. It consists of two essential modules; one is clustering based on neural networks, which groups tasks on a large amount of database records, while the other extracts association rules by employing rough set theory.

Additionally, a new personalization recommendation system that integrates both user clustering and association rules techniques was proposed in [72]. The clustering method is used to cluster users' time-framed navigation sessions that are analyzed using association rules technique to establish recommendations for other similar users in the future. In addition, Wang et al. in [73] proposed a personalized recommendation system that incorporates content-based, collaborative, and association rules.

TABLE I.   SYMBOLS USED IN THE PAPER

Symbol	Meaning
U	The set of users
n	The number of users, |U|
$u, u_1, …, u_n$	Users
I	The set of items
m	The number of items, |I|
$i, i_1, …, i_m$	Items
$S(u_t, u_a)$	The similarity function used
PV(u)	The profile vector for a user u

R(u, i)	Rating of the user u for the item i
P(u, i)	Predication of user u's rating for item i
minSim	minimum similarity threshold value
minPred	minimum prediction threshold value

Liu et al. in [58] have proposed a product recommendation methodology that combines group decision-making and association rules. The system addressed the lifetime value of a customer to a firm. This system employed analytical hierarchy process to evaluate customer lifetime value (CLV) then clustering were used to group customers according to the CLV. Finally, an association rule method was implemented to provide product recommendation to each customer group.

D. Clustering Techniques

Traditional collaborative filtering techniques are often based on matching the current user profile against clusters of similar profiles obtained by the system over time from other users. Clustering is the process of grouping the data into classes or clusters so that the objects within a cluster have high similarity in comparison to one another, but are very dissimilar to objects in other clusters. Dissimilarities are assessed based on the attributes values describing the objects. Often distance measures are used. There exist a large number of clustering algorithms in the literature. The choice of clustering algorithms depends both on the type of the data available and on the particular purpose and application. k-mean clustering technique is used in this work because of its simplicity and being suitable to be used with numerical unsupervised data like student courses' grades.

K-means clustering technique is one of the simplest unsupervised learning algorithms that do not rely on predefined classes to solve the clustering problem. The main idea is to define k centers, one for each cluster. The next step is to take each point belonging to a given data set and associate it to the nearest centers. At this point, we need to re-compute k new centers of the clusters resulting from the previous step. After we have these k new centers, a new binding has to be done between the same data set points and the nearest new centers. A loop has been generated. As a result of this loop, we may notice that the k centers change their location step by step until no more changes are done. In other words, centers do not move any more. Finally, this algorithm aims at minimizing a criterion function, in this case a squared error function. The criterion function J is defined by

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2$$

where $\left\| x_i^{(j)} - c_j \right\|^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster center c_j, is an indicator of the distance of the n data points from their respective cluster centers.

E. Similarities Techniques

Similarity is quantity that reflects the strength of relationship between two objects or two features. This quantity is usually having range of either -1 to +1 or normalized into 0 to 1. If the similarity between object i and object j is denoted by S_{ij}, we can measure this quantity in several ways depending on the scale of measurement (or data type) that we have.

Similarity distance measures are commonly used for computing the similarity of objects described by interval scaled variables. Interval scale variables are continuous measurements of roughly linear scale. Typical examples include weight, height, grade, and weather temperature. The similarity between the objects described by interval-scaled variables is typically computed based on the distance between each pair of the objects. Different measures, including Manhattan, Euclidean, and Minkowski, can be used. In this paper, we used Manhattan distance.

III. THE PROPOSED RECOMMENDATION SYSTEM

Recently, many aspects of receiving a college education have been changed. The volume of course-related information available to students is rapidly increasing. This abundance of information has created the need to help students find, organize, and use resources that match their individual goals, interests, and current knowledge. One of the concerns students have is to make decisions about which courses to take. The concern is more serious for graduate students who have more freedom to choose courses while they care more about taking courses that contribute to their progress towards career goals. To make these decisions, they use information from course catalogs and schedules, consult with their advisors, and seek guidance from their classmates, especially those with similar interests. To give better decision making support to students who wish to make relevant course choices, we have developed a course recommendation system that recommends courses to students based on other similar students.

Our collaborative recommendation system tries to recommend elective courses to students based on what other similar students have taken. It recommends courses and specifies expected grades for these courses. Accordingly, the student may take a course that is recommended by the system with an acceptable grade. Typically, students have the choice to take courses from a set of elective courses and in most cases, the students take the advices from other students that took such courses. In our recommendation system, we automatically find similar students and then apply association rule mining algorithm on their courses to create courses association rules. Discovered courses association rules are used to get recommendation.

To obtain courses association rules, courses dataset is built by mapping each course either compulsory or elective to an item and each student to a transaction. For each student, a transaction is created that contains the grades of all courses taken by the student. Table 2 shows an example of courses dataset, where -1 indicates that the student does not take the corresponding course. Examples of courses association rules that can be generated are "70% of students who got **A** in **Algo** course and **E** in **Parallel** course may also get **B** in **OS** course" and "90% of students who got **D** in **Net** course may also get **D** in **Mining** course".

The system takes as an input the specified minimum support, the specified minimum confidence, and the courses

dataset. As an output, the system generates courses association rules that satisfy the support and confidence constraints. Then the system uses these rules to generate courses recommendations. Fig. 1 illustrates our recommendation methodology. The main steps of our system are described in the following subsections.

TABLE II.   AN EXAMPLE OF COURSES DATASET

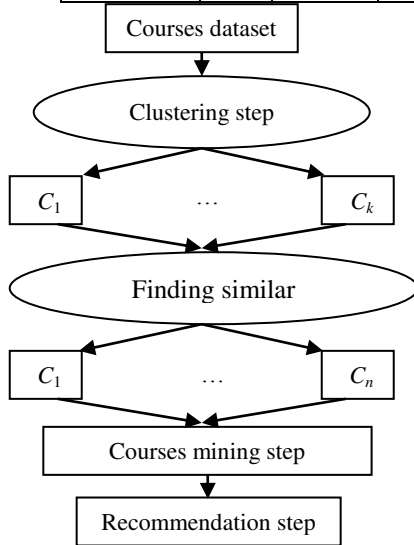| Student ID	Alg	Mining	OS	Parallel	Net
1000	A	A	-1	F	B
1001	F	F	C	D	E
1002	B	A	F	E	E
1003	E	D	B	-1	A

Fig. 1.   Proposed recommendation methodology

A. Clustering Step

This step applies clustering technique on courses dataset to group similar students to the same cluster. The clustering algorithm used in this step is *k-means* clustering algorithm [21], where each cluster is represented by the mean value of students in the cluster. A cluster of similar students is treated as one group in this step. The similarity between the courses is typically computed based on the distance between each pair of courses. A well-known distance measure is *Manhattan* distance [21], which measures the distance between two *p*-dimensional data objects $i = (x_{i1}, x_{i2}, …, x_{ip})$ and $j = (x_{j1}, x_{j2}, …, x_{jp})$ as follow $d(i, j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + … + |x_{ip} - x_{jp}|$. In our case, *i* and *j* represent two students to calculate similarity between them, $x_{i1}, x_{i2}, …, x_{ip}$ represent the courses' grades for student *i*, while $x_{j1}, x_{j2}, …, x_{jp}$ represent the courses' grades for student *j*. For example, *Manhattan* distance between the first two students, 1000 and 1001 of Table 2 is:

$$d(1000, 1001) = |A - F| + |A - F| + |F - D| + |B - E| = 15$$

where the grades (**A**, **B**, **C**, **D**, **E**, and **F**) are mapped to the integers (1, 2, 3, 4, 5, and 6), respectively. Here, we did not include the distance between students' grades in **OS** course because the first student did not take this course.

B. Finding Similar Students

In this step, the group most similar to the target student is selected by comparing his/her previous courses' grades with the courses' grades of the mean student of the *n*-cluster. We used *n-nearest neighborhood* technique [22] to select the most *n* similar groups generated in Step I. In this way, all of the clusters centers are represented in a *p*-dimensional pattern space. When given an unknown student, an *n*-nearest neighborhood searches the pattern space for the *n* clusters that are closest to the unknown student. These *n* clusters are the *n* "nearest neighbors" of the unknown sample. Closeness is defined in terms of Manhattan distance.

C. Courses Mining

Association rules mining is used to discover courses association rules. The algorithm is applied on the *n* similar groups created from Step II. Table 3 shows an example of the transactional courses dataset that is used in the mining step, where each transaction contains transaction Id (*TID*) that corresponds to student *ID* and set of items. Items are in the form of [*course*:*grade*] pairs. The rationale of using [*course*:*grade*] pairs is that our system tries, in addition to recommend courses, also to specify expected grades for those courses. This will help students to select elective courses with high grade expectation. An example of courses association rule is [**Alg**:**A**] ∧ [**OS**:**D**] ⇒ [**Parallel**:**C**], this indicates that if the target student got **A** in **Algo** and **D** in **OS** courses, then the system recommends the student to take **Parallel** course with expected grade **C**.

D. Recommending courses

The courses association rules, generated in Step III, are used to recommend elective courses. This step is the core step in our recommendation system. The recommendation is done after interesting courses association rules are created. The association rules are in the form of $[c_1:g_1] ∧ … ∧ [c_n:g_n] ⇒ [c_{n+1}:g_{n+1}] ∧ … ∧ [c_m:g_m]$, where [*c*:*g*] represents a course *c* with its grade *g*. The whole recommendation strategy is described as follows:

1) If the target student has taken the courses [c₁, …, cₙ], the ones in the antecedent part of the rule, with grades [g₁, …, gₙ] respectively, then the system recommends the student to take the elective courses [cₙ₊₁, …, cₘ], the ones in the consequent part of the rule, with expected grades [gₙ₊₁, …, gₘ], respectively. The system may also recommend courses from the consequent part of the rule if they met specified minimum grades. In this step, the student can specify a minimum grade in which the system will not recommend any course with grade that does not exceed this minimum threshold. In the following example, the system recommends courses for a student who has taken the following courses [DB:A], [Parallel:C], [Net:D], and specified the minimum grade to C.

Rule#1: [**DB**:**A**] ∧ [**Parallel**:**C**] ⇒ [**Mining**:**B**]
Rule#2: [**DB**:**A**] ∧ [**Net**:**D**] ⇒ [**AI**:**B**] ∧ [**Archit**:**D**]

　　1. Rule#1 recommends **[Mining:B]**
　　2. Rule#2 recommends [**AI**:**B**]

TABLE III. AN EXAMPLE OF TRANSACTIONAL COURSES DATASET USED IN MINING STEP

TID	Items
1000	[**Alg**:A], [**Mining**:A], [**Parallel**:F], [**Net**:B]
1001	[**Alg**:F], [**Mining**:F], [**OS**:C], [**Parallel**:D], [**Net**:E]
1002	[**Alg**:B], [**Mining**:A], [**OS**:F], [**Parallel**:E], [**Net**:E]
1003	[**Alg**:E], [**Mining**:D], [**OS**:B], [**Net**:A]

Because the student has taken the courses in the antecedent part of rule#1, it recommends the student the **Mining** course with expected grade is **B**. in the same way rules#2 recommends the student the **AI** course with expected grade is **B**. The rule#2 does not recommend the student to take the **Archit** course because the expected grade **D** is less than the minimum threshold already specified by that student.

2) The target student might not take all the courses in the antecedent part of the rule. The rule still can be used in recommendation and a new constraint called match is used to assess the quality of rule in the recommendation. The match is defined as the number of matched courses between the target student's courses and the courses in the antecedent part of the rule to the total number of courses in the antecedent part of the rule. matchrule =

$$\frac{matched\ courses\ between\ student\ and\ antecedent\ part\ of\ rule}{total\ courses\ in\ antecedent\ part\ of\ rule}$$

If *matchrule* value is greater than a threshold value, then the rule is used for recommendation.

3) If different rules recommend the target student different courses, then the student can select the top-N courses recommended by best quality rules, i.e., rules that have the highest supports, the highest confidences, and the highest match. If there a tie between rules, the system recommends courses in this order of preferences: highest support rule, highest confidence, and highest match.

4) The recommendation system does not recommend new students since they do not have courses taken yet.

For example, consider the following four course association rules.

Rule#1: [**DB**:A] ∧ [**Parallel**:C] ⇒ [**Mining**:B]
Rule#2: [**DB**:A] ∧ [**Net**:D] ⇒ [**AI**:B] ∧ [**Archit**:D]
Rule#3: [**Mining**:E] ∧ [**OS**:C] ⇒ [**IR**:D]
Rule#4: [**Net**:D] ∧ [**Parallel**:C] ⇒ [**Image**:B]

Table 4 shows the proposed recommendations, that are derived from the above listed four rules, for a student who took the following courses: [**DB**:A], [**Parallel**:C], [**Net**:D], with **C** as a minimum accepted grade and 50% as a match threshold value. As shown in Table 4, Rule#1 has *match*$_{rule}$ value equal to 66%, which is above the match threshold 50%. Then this rule recommends the elective course in the consequent part **Mining** with **B** as expected grade. Rule#2 has also *match*$_{rule}$ value equals to 66%. The elective courses in the consequent part of the rule are [**AI:B**] and [**Archit**:D]. The rule recommends only [**AI:B**], but not **Archit** course, because the expected grade **D** is less than the specified minimum accepted grade **C**. Rule#3 is not used in recommendation because its *match*$_{rule}$ value (zero value) is less than 50%, the match

threshold value. The last rule Rule#4 recommends the elective course [**Image**:B].

TABLE IV. THE PROPOSED RECOMMENDATIONS USING THE GIVEN FOUR COURSES ASSOCIATION RULES

Rule used	Recommendation	match$_{rule}$
#1	[**Mining:B**]	66%
#2	[**AI:B**]	66%
#2	[**Archit:D**] is not recommended since D < C	66%
#3	No recommendation since matchrule < 50%	0%
#4	[**Image:B**]	66%

Another example consists of eight course association rules.

Student courses: [**DB**:A], [**Parallel**:C], and [**Net**:D]
Rule#1: [**DB**:A] ∧ [**Parallel**:C] ⇒ [**Mining**:B]
Rule#2: [**DB**:A] ∧ [**Net**:D] ⇒ [**AI**:B] ∧ [**Archit**:D]
Rule#3: [**Mining**:E] ∧ [**OS**:C] ⇒ [**IR**:D]
Rule#4: [**Net**:D] ∧ [**Parallel**:C] ∧ [**Net**:D] ⇒ [**Image**:B]
Rule#5: [**DB**:A] ∧ [**Parallel**:D] ∧ [**IR**:D] ⇒ [**Alg**:B]
Rule#6: [**DB**:B] ∧ [**Net**:D] ∧ [**OS**:C] ⇒ [**AI**:B] ∧ [**Archit**:D]
Rule#7: [**Alg**:E] ∧ [**OS**:C] ∧ [**DB**:A] ∧ [**Image**:B] ⇒ [**IR**:D]
Rule#8: [**Net**:D] ⇒ [**Artchit**:B] ∧ [**Image**:B]

Table 5 shows the proposed recommendations, that are derived from the above listed eight rules, for a student who took the following courses: [**DB**:A], [**Parallel**:C], [**Net**:D], with **C** as a minimum accepted grade and 50% as a match threshold value. As shown in Table 5, Rule#1 has *match*$_{rule}$ value equal to 66%, which is above the match threshold 50%. Accordingly, this rule recommends the elective course in the consequent part **Mining** with **B** as expected grade. Rule#2 has also *match*$_{rule}$ value equals to 66%. The elective courses in the consequent part of the rule are [**AI:B**] and [**Archit**:D]. The rule recommends only [**AI:B**], it does not recommend **Archit** course, because the expected grade **D** is less than the specified minimum accepted grade **C**. Rules #3, #5, #6, and #7 are not used in recommendation because their *match*$_{rule}$ values 0%, 33%, 0%, and 25% respectively, are less than 50%, the match threshold value. [**Image:B**] is recommended by rule#4 that has *match*$_{rule}$ value 100%. Two courses are recommended using rule#8, [**Artchit:B**] and [**Image:B**].

TABLE V. THE PROPOSED RECOMMENDATIONS USING THE GIVEN EIGHT COURSE ASSOCIATION RULES

Rule used	Recommendation	match$_{rule}$
#1	[**Mining:B**]	66%
#2	[**AI:B**]	66%
#2	[**Archit:D**] is not recommended since D < C	66%
#3	No recommendation since matchrule < 50%	0%
#4	[**Image:B**]	100%
#5	No recommendation since matchrule < 50%	33%
#6	No recommendation since matchrule < 50%	0%
#7	No recommendation since matchrule < 50%	25%
#8	[**Artchit:**B] and [**Image:B**]	100%

IV. EXPERIMENTS

We performed experiments using courses dataset taken by 2000 graduate students of Electrical Engineering. The total

number of courses is 54, where five of them are compulsory, while the remaining are elective courses. Each student has a choice of selecting four elective courses. As early mentioned, *k*-means clustering algorithm is used, with *k* = 5. In the mining step, we used *n* = 3 as the most similar *n* clusters. We divided our dataset into two disjoint sets: the *training set* and the *test set*. *K*-fold cross validation technique is applied on the dataset, in which we run *k* experiments, each time setting aside different 1/*k* of the data to test on, and average the result. Popular values for *k* are 5 and 10, we use *k* = 5. Mining association rules algorithm is applied on training set to generate courses association rules. Then we measured the percentage of students in the test set that were correctly recommended by courses association rules.

To evaluate the performance of our recommendation system, we use two standard information retrieval measures, *precision* and *recall* [24]. Precision is the percentage of the number of recommended courses taken to the total number of recommended courses, while recall is the percentage of the number of recommended courses taken to the total number of courses taken by the students. More precisely

$$Precision = \frac{\#\ of\ recommended\ courses\ taken}{total\ \#\ of\ recommended\ courses}$$

$$Recall = \frac{\#\ of\ recommended\ courses\ taken}{total\ \#\ of\ couses\ taken\ by\ students}$$

For recommendation systems, precision is more significant than recall, because we concern more on getting high quality recommendation than just recommending a large number of courses. Therefore, our goal is to achieve a high precision with reasonably high recall. The main parameters used in our experiments are minimum confidence, minimum match, minimum specified grade, and minimum support. The following sections show experiments that were performed in order to choose the appropriate values for the parameters.

A. Minimum Confidence

Both minimum support and minimum confidence are important factors that influence the performance of the whole process of recommendation system. Since minimum confidence is used during recommendation step, it would be interested to study the performance of our recommendation system using different minimum confidence values. The results are shown in Fig. 2. As shown in Fig. 2, the minimum confidence value has a significant impact on the performance, i.e., the higher the minimum confidence, the higher the precision but the lower the recall. We achieved the highest precision of 0.95 with a recall of 0.5 for the minimum confidence of 90%. Even though we think that the precision is the most important factor in recommendation systems, the best combination of precision and recall values, which occurs with minimum confidence of 80%, is also important in the sense that we can achieve higher values of both precision and recall.
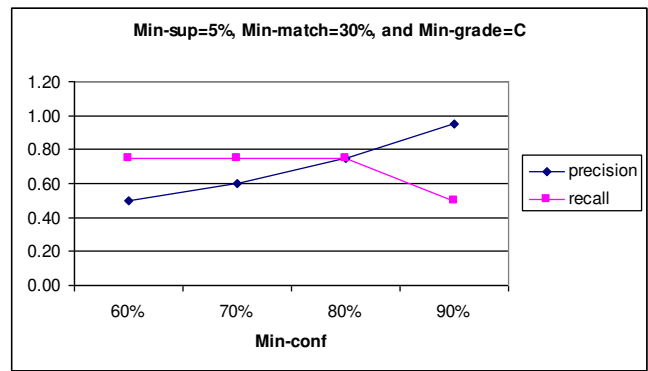

Fig. 2. Performance using different minimum confidence values

B. Minimum Match

In order to select the appropriate minimum match, we performed some experiments using different minimum match values. As shown in Fig. 3, the higher the minimum match the higher precision but the lower the recall. We achieved the highest precision with minimum match 90%. When the minimum match varies, a tradeoff between precision and recall values is noticed.

C. Minimum Grade

Minimum specified grade is also an important parameter of our approach. Fig. 4 gives the performance for different minimum grades. We could see a general fact that the minimum grade has a similar influence on the performance as the minimum confidence, i.e., the higher the minimum grade, the higher the precision but the lower the recall. When the minimum grade is varied, there shows a tradeoff between the precision and the recall.
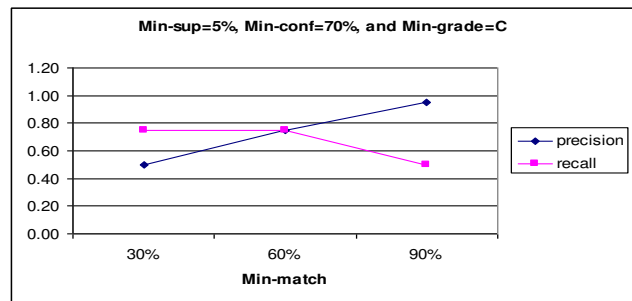

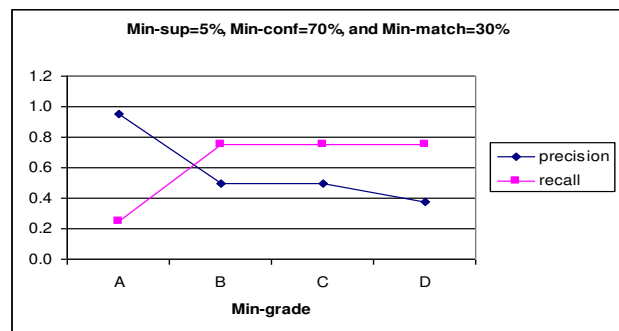Fig. 3. Performance using different minimum match values

Fig. 4. Performance using different minimum grade values

D. Minimum Support

In our experiments for courses associations, we tested the performance for different minimum supports, i.e., we only use rules above a specified minimum support for recommendation during a test. The results are shown in Fig. 5.

Form our observations, we found that when a target student's minimum support determined by the mining process is very low, it takes a very long time to mine rules for this student and at the same time the performance is bad. While, if a student's minimum support is greater than a threshold, then we get a better performance.

V. CONCLUSION AND FUTURE WORK

The main contribution of this paper is a new collaborative recommendation system that employed association rules algorithm to recommend university elective courses to a target student based on what other similar students have taken. The experiments shown that association rule is a desirable tool for making recommendation to a target student. Through our experiments, we noticed the patterns of influence of different parameters on the performance of the system. The confidence and match of a rule have a great impact on the performance, but the highest confidence or match may not be the best choice. By choosing a relatively high confidence or match, we can achieve a better performance.

Much work can be performed in the future such as doing comparison between our method and other typical methods. In addition, further experimental evaluation, joining collaborative and content-based recommendations, and applying the new recommendation system in other domains of interest is expected as future work.
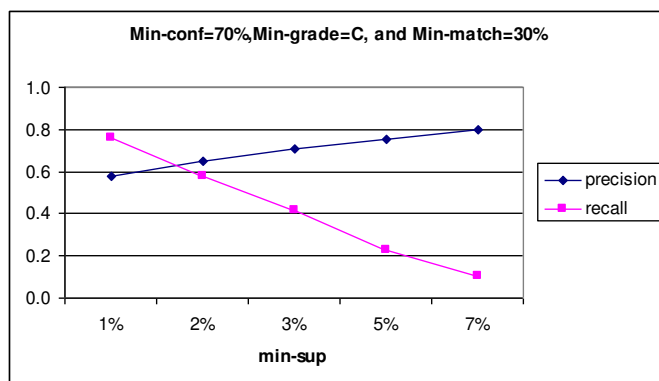

Fig. 5. Performance using different minimum support values

REFERENCES

[1] Resnick P. and Varian H., Recommender Systems, *Communication of the ACM*, 40(3), 56-58, March 1997.

[2] Montaner M., Lopez B., and Josep R., A Taxonomy of Recommender Agents on the Internet, *Artificial Intelligence Review*, 19, 285-330, 2003.

[3] Bridge D., Product recommendation systems: A new direction. *In R.Weber, Wangenheim, C., eds.: Procs. of the Workshop Programme at the Fourth International Conference on Case-Based Reasoning.* Vancouver, Canada, 79-86, 2001.

[4] Burke R., Hybrid Recommender Systems: Survey and Experiments, *User Modelling and User-Adapted. Interaction* 12, 4, 331-370, 2002.

[5] Herlocker J., Konstan J., Tervin L., and Riedl J., Evaluating Collaborative Filtering Recommender Systems, *ACM Transactions on Information Systems*, 22(1), 5-53, 2004.

[6] Mirza J., Keller B., and Ramakrishnan N., Studying Recommendation Algorithms by Graph Analysis, *Journal of Intelligent Information Systems*, 20(2), 131-160, 2003.

[7] Ariely D., Lynch, J., and Aparicio M., Learning by Collaborative and Individual-Based Recommendation Agents. *Journal of Consumer Psychology*, 14(1 & 2), 81–95, 2004.

[8] Ziegler C., McNee S., Konstan J., and Lausen G., Improving Recommendation Lists Through Topic Diversification. *In Proceedings of 14th Int. WWW Conference*, Chiba, Japan, 22–32, 2005.

[9] Gretzel U. and Fesenmaier D., Persuasion in Recommender Systems, *International Journal of Electronic Commerce*, 11(2), 81-100, 2006.

[10] Chen A. and McLeod D. (2006). Collaborative Filtering for Information Recommendation Systems, *Encyclopedia of E-Commerce, E-Government, and Mobile Commerce*. IGI Global, pp. 118-123.

[11] Melville P., Mooney R., and Nagarajan R., Content-Boosted Collaborative Filtering for Improved Recommendations, *In Proceedings of the 18th National Conference on Artificial Intelligence (AAAI-2002)*, (July 2002), Edmonton, Canada, 187-192.

[12] McLauglin M. and Herlocher J., A Collaborative Filtering Algorithm and Evaluation Metric that Accurately Model the User Experience, *In Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. (2004), Sheffield, United Kingdom, 329-336.

[13] Ahn H. J., Utilizing Popularity Characteristics for Product Recommendation. *International Journal of Electronic Commerce*, (2006), 11(2), 59-80.

[14] Nikolaeva R. and Sriram S., The Moderating Role of Consumer and Product Characteristics on the Value of Customized Online Recommendations, *International Journal of Electronic Commerce*. (2006), 11(2), 101-124.

[15] Huang Z., Chung W., and Chen H., A Graph Model for E-Commerce Recommender Systems, *Journal of the American Society for Information Science and Technology (JASIST)*, (2004), 55(3), 259-274.

[16] Schafer J., Konstan J., Riedl J., E-Commerce Recommendation Applications, *Data Mining and Knowledge Discovery*, (2001), 5(1-2), 115-153.

[17] Pazzani M., A Framework for Collaborative, Content-Based and Demographic Filtering, *Artificial Intelligence Review*, (1999), 13(5-6), 393-408.

[18] Chen M., Chiu A., and Chang H., Mining Changes in Customer Behavior in Retail Marketing. *Expert Systems with Applications*, (2005), 28(4), 773-781.

[19] Resnick P., Iacovou N., Suchak M., Bergstrom P., and Riedl J., GroupLens: An Open Architecture for Collaborative Filtering of Netnews, *In Proceedings of ACM Conference on Computer Supported Cooperative Work*, (1994), Chapel Hill, NC, USA, 175-186.

[20] Goldberg D., Nichols D., Oki B., and Terry D., Using Collaborative Filtering to Weave an Information Tapestry, *Communications of the ACM*, (1992), 35(12), 61-70.

[21] Adomavicius G., Sankaranarayanan R., Sen S., and Tuzhilin A., Incorporating Contextual Information in Recommender Systems Using a Multidimensional Approach, *ACM Transactions on Information Systems*, (2005), 23(1), 103-145

[22] Yap G., Tan A., and Pang H., Dynamically Optimized Context in Recommender Systems, *In the Proceedings of the 6th International Conference on Mobile Data Management*, (May 2005), Ayia Napa, Cyprus, 265-272.

[23] Good N., Scafer J., Konstan J., Borchers Al, Sarwar B., Herlocaker J., and Riedl J., Combining Collaborative Filtering with Personal Agents

for Better Recommendations, *In Proceedings of the Eleventh Innovative Applications of Artificial Intelligence Conference*, July 18-22, 1999, American Association for Artificial Intelligence, Menlo Park, CA, USA, 439-446.

[24] Maes P., Agent that Reduce Work and Information Overload, *Communication of the ACM*, (1994), 37(7), 31-40.

[25] Sarwar B., Karypis G., Konstan J., and Riedl J., Analysis of Recommendation Algorithms for E-Commerce. *In Proceedings of the 2nd ACM Conference on Electronic Commerce (EC-00)*, October 17-20, 2000, Minneapolis, MN, USA, 158-167.

[26] Burke R., Knowledge-Based Recommender Systems, In A. Kent (ed.) *Encyclopedia of Library and Information Systems,* (2000), vol 69(32).

[27] Zhang J. and Pu P., A Comparative Study of Compound Critique Generation in Conversational Recommender Systems. *In Proceedings of the 4th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH2006)*, June 2006, Dublin, Ireland, Volume 4018 of Lecture Notes in Computer Science, 234-243, Springer.

[28] Wiranto, Winarko E., Hartati S., and Wardoyo R., Improving the Prediction Accuracy of Multicriteria Collaborative Filtering by Combination Algorithms, International Journal of Advanced Computer Science and Applications, 5(4), pp. 52-58, 2014.

[29] Adomavicius G. and Tuzhilin A., Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Transactions on Knowledge and Data Engineering*, (June 2005), 17(6), 734-749.

[30] Lee T., Chun J., Shim J., and Lee S., An Ontology-Based Product Recommender System for B2B Marketplaces, *International Journal of Electronic Commerce*, (Jan 2007), 11(2), 125-154.

[31] Jian C., Jian Y., and Jin H., Automatic Content-Based Recommendation in e-Commerce, *In the proceedings of IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'05),* (March 29-April 01 2005), Hong Kong, China, 748-753.

[32] Philip S., Shola P.B., John A.O., Application of Content-Based Approach in Research Paper Recommendation System for a Digital Library, International Journal of Advanced Computer Science and Applications, 5(10), pp. 37-40, 2014.

[33] Hill W., Stead L., Rosenstein M., and Furnas G., Recommending and Evaluating Choices in a Virtual Community of Use*, In the Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, (May 07-11 1995), Denver, Colorado, United States, 194-201.

[34] Pazzani M., Muramatsu J., and Billsus D., Syskill & Webert: Identifying Interesting Web Sites, *In Proceedings of the National Conference on Artificial Intelligence (AAA1996),* (1996), Portland, 54-61.

[35] Shih Y. and Liu D., Hybrid Recommendation Approaches: Collaborative Filtering via Valuable Content Information, *In the Proceedings of the 38th Annual Hawaii International Conference HICSS '05,* 2005, 217-223.

[36] Mooney R. and Roy L., Content-Based Book Recommending Using Learning for Text Categorization, *In Proceedings of the 5th ACM Conference on Digital Libraries*, (June 2-7 2000), San Antonio, Texas, USA, 195-204.

[37] Condliff M., Lewis D., Madigan D., and Posse C., Bayesian Mixed-Effects Models for Recommender Systems, *In ACM SIGIR'99 Workshop on Recommender Systems: Algorithms and Evaluation*, 1999, 23-30.

[38] Kim E., Kim M., and Ryu J., Collaborative Filtering Based on Neural Networks Using Similarity, *In the Proceedings of Computational Science and Its Applications conference (ICCSA 2),* (May 2005), Singapore, 355-360.

[39] Smyth B, McCarthy K., Reilly J., O'Sullivan D., and McGinty L., David C. Wilson, Case-Studies in Association Rule Mining for Recommender Systems, *In Proceedings of the 2005 International Conference on Artificial Intelligence, ICAI 2005,* (June 27-30 2005), Las Vegas, Nevada, USA, 809-815.

[40] Mobasher B., Cooley R., and Srivastava J., Automatic Personalization Based on Web Usage Mining, *Communications of the ACM*, (2000), 43(8), 142–151.

[41] Krulwich B. and Burkley C., The InfoFinder Agent: Learning User Interests through Heuristic Phrase Extraction, *IEEE Expert: Intelligent Systems and Their Applications*, (1997), 12(5), 22-27.

[42] Chen Z, Meng X, Zhu B., and Fowler R., WebSail: From On-line Learning to Web Search, *In Proceedings of the 2000 International Conference on Web Information Systems Engineering*, (2000), Hong Kong, China, 206-213.

[43] U. Shardanaand P. and Maes P., Social Information Filtering: Algorithms for Automating "Word of. Mouth" *Proceedings of the SIGCHI conference on Human factors in computing systems*, (1995), Denver, Colorado, United States, 210–217.

[44] Balabanovic M. znd Shoham Y., Fab: Content-Based, Collaborative Recommendation, *Communications of the ACM*, (1997), 40(3), 66-72.

[45] Billsus D. and Pazzani M., User Modeling for Adaptive News Access, *Journal of User Modeling and User-Adapted Interaction*, (2000), 10(2-3), 147-180.

[46] Smyth B. and McClave P., Similarity vs. Diversity, *In Proceedings of the 4th International Conference on Case-Based Reasoning: Case-Based Reasoning Research and Development,* (August 02, 2001), Vancouver, British Columbia, Canada p, 347-361.

[47] Kanungo T., Mount D., Netanyahu N., Piatko C., Silverman R., and Wu A., An Efficient k-Means Clustering Algorithm: Analysis and Implementation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7), 881-892, 2002.

[48] Schein A., Popescul A., Ungar L., and Pennock D., Methods and Metrics for Cold-Start Recommendations. *In Proceedings of the 25th th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, (2002), Tampere, Finland , 253-260

[49] Papagelis M., Plexousakis D., and Kutsuras T., Alleviating the Sparsity Problem of Collaborative Filtering Using Trust Inferences, *In Proceedings of iTrust 2005*, (2005), Paris, France, 224-239.

[50] Breese J., Heckerman D., and Kadie C., Empirical Analysis of Predictive Algorithms for Collaborative Filtering, *In Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, (1998), Madison, WI, 43-52.

[51] Changchien S. and Lu T., Mining Association Rule Procedure to Support On-Line Recommendation by Customers and Products Fragmentation, *Expert Systems with Application* , (2001), 20, 325-335.

[52] Liu D. and Shih Y., Integrating AHP and Data Mining for Product Recommendation Based on Customer Lifetime Value, *Information and Management*, (March 2005), 42(3), 387-400.

[53] Goldberg K, Roeder T, Gupta D., and Perkins C., Eigentaste: A Constant Time Collaborative Filtering Algorithm*, Information Retrieval*, (July 2001), 4(2), 133-151.

[54] Terveen L., Hill W., Amento B, McDonald D., and Creter J., PHOAKS: A System for Sharing Recommendations, *Communications of the ACM*, (March 1997), 40(3), 59-62.

[55] Greco G., Greco S., and Zumpano E., Collaborative Filtering Supporting Web Site Navigation, *AI Communications,* (2004), 17(3), 155–166.

[56] Konstan J., Miller B., Maltz D., Herlocker J., Gordon L., and Riedl J., GroupLens: Applying Collaborative Filtering to Usenet News, *Communications of the ACM*, (March 1997), 40(3), 77-87.

[57] Lee D., Kim G., and Choi H., A Web-Based Collaborative Filtering System, *Pattern Recognition,* (2003), 36, 519-526.

[58] Burke R., Hammond K., and Young B., The FindMe Approach to Assisted Browsing, *IEEE Expert: Intelligent Systems and Their Applications*, (July 1997), 12(4), 32-40.

[59] Felferng A. and Gula B., An Empirical Study on Consumer Behavior in the Interaction with Knowledge-Based Recommender Applications, In the Proceedings of the 8th IEEE International Conference on E-Commerce Technology and the 3rd IEEE International Conference on Enterprise Computing, E-Commerce, and E-Services (CEC/EEE'06), 2006, 37-45.

[60] Felfernig A., Friedrich G., Jannach D., and Zanker M., An Integrated Environment for the Development of Knowledge-Based Recommender Applications, *International Journal of Electronic Commerce*, (2006), 11(2), 11-34

[61] Felfernig, A., A. Knowledge-Based Interactive Selling of Financial Services with FSAdvisor. *In N. Jacobstein and B. Porter (eds.), Seventeenth Innovative Applications of Artificial Intelligence Conference*. (2005), Menlo Park, CA, 1475-1482.

[62] Towle B. and Quinn C., Knowledge Based Recommender Systems Using Explicit User Models. *In Knowledge-Based Electronic Markets, Papers from the AAAI Workshop, AAAI Technical Report WS-00-04*. (2000), Menlo Park, CA, 74-77.

[63] Wakil K., Bakhtyar R., Ali K., and Alaadin K., Improving Web Movie Recommender System Based on Emotions, *International Journal of Advanced Computer Science and Applications*, 6(2), 2015, 218-226.

[64] Mobasher B., Dai H., Luo T., and Nakagawa M., Effective Personalization Based on Association Rule Discovery from Web Usage Data, *In the Proceedings of the 3rd International Workshop on Web Information and Data Management*, (November 09-01, 2001), Atlanta, Georgia, USA, 9-15.

[65] Wang F. and Shao H., Effective Personalized Recommendation Based on Time-Framed Navigation Clustering and Association Mining. *Expert Systems with Application*, (2004), 27(3), 365-377.

[66] Wang Y., Chuang Y., Hsu M., Keh H., A Personalized Recommender System for the Cosmetic Business. *Expert Systems with Application,* (2004), 26, 427-434.

[67] Burke R., Integrating Knowledge-Based and Collaborative-Filtering Recommender Systems, In *AAAI Workshop on AI in Electronic Commerce*, 1999, 69-72.

[68] Chen Y., Cheng L., and Chuang C., A Group Recommendation System with Consideration of Interactions Among Group Members, Expert Systems with Applications (2008), 34(3), 2082-2090.

[69] Agrawal R., Imilienski T., Swami A., Mining Association Rules Between Sets of Items in Large Databases, In Proceedings of the ACM SIGMOD International Conference on Management of Data, 1993, Washington, DC 207–216.

[70] Agrawal R. and Srikant R., Fast Algorithms for Mining Association Rules, In Proceedings of the 20th International Conference on Very Large Databases, Santiago, Chile: Sept. 1994, 487-499.

[71] Tsay YJ and Chiang JY, CBAR: An Efficient Method for Mining Association Rules. Knowledge Based Systems, 2004, 1-7.

[72] Ceglar A. and Roddick J., Association Mining, ACM Surveys, 2006, V 38(2), article No. 5.