



Cairo University
Egyptian Informatics Journal

www.elsevier.com/locate/eij
www.sciencedirect.com



ORIGINAL ARTICLE

Hybrid attribute-based recommender system for learning material using genetic algorithm and a multidimensional information model

Mojtaba Salehi ^{a,*}, Mohammad Pourzaferani ^b, Seyed Amir Razavi ^c

^a Department of Industrial Engineering, Tarbiat Modares University, Tehran, Iran

^b Department of Computer Engineering, University of Isfahan, Isfahan, Iran

^c Department of Computer Engineering and Information Technology, Amirkabir University of Technology, Tehran, Iran

Received 6 May 2012; revised 6 December 2012; accepted 11 December 2012

Available online 3 January 2013

KEYWORDS

Collaborative filtering;
Sparsity;
Personalized recommender;
Learning material;
Genetic algorithm

Abstract In recent years, the explosion of learning materials in the web-based educational systems has caused difficulty of locating appropriate learning materials to learners. A personalized recommendation is an enabling mechanism to overcome information overload occurred in the new learning environments and deliver suitable materials to learners. Since users express their opinions based on some specific attributes of items, this paper proposes a hybrid recommender system for learning materials based on their attributes to improve the accuracy and quality of recommendation. The presented system has two main modules: explicit attribute-based recommender and implicit attribute-based recommender. In the first module, weights of implicit or latent attributes of materials for learner are considered as chromosomes in genetic algorithm then this algorithm optimizes the weights according to historical rating. Then, recommendation is generated by Nearest Neighborhood Algorithm (NNA) using the optimized weight vectors implicit attributes that represent the opinions of learners. In the second, preference matrix (PM) is introduced that can model the interests of learner based on explicit attributes of learning materials in a multidimensional information model. Then, a new similarity measure between PMs is introduced and recommendations are generated by NNA. The experimental results show that our proposed method outperforms current algorithms on accuracy measures and can alleviate some problems such as cold-start and sparsity.

© 2012 Faculty of Computers and Information, Cairo University.
Production and hosting by Elsevier B.V. All rights reserved.

* Corresponding author. Tel./fax: +98 2182884323.

E-mail addresses: m_salehi61@yahoo.com, mojtaba.salehi@modares.ac.ir (M. Salehi).

Peer review under responsibility of Faculty of Computers and Information, Cairo University.



Production and hosting by Elsevier

1. Introduction

With the growth of technology in educational organizations at recent years, Web-based learning environments are becoming very popular. Typical electronic learning (e-learning) environments that can be accessed by mobile, such as Moodle and Blackboard include course content delivery tools, synchronous

and asynchronous conferencing systems, Forums, quiz modules, sharing materials, white boards and etc. [1,2].

One of the important parts in the new learning environments is recommender system (RS). A recommender system in an e-learning context is a software agent that tries to "intelligently" recommend actions to a learner based on the actions of previous learners. This recommendation could be an on-line activity such as doing an exercise, reading posted messages on a conferencing system, or running an on-line simulation, or could be simply a web material [3]. One of the most important applications of recommender systems in learning environments is material recommendation. RSs use opinions of a community of users to help individuals identify material and content of interest from a potentially overwhelming set of choices more effectively [4]. By using material recommender systems in learning environments, we can address two problems, personalization and information overload. In this situation, recommender system offers which learning objects should learners study next [5], or offers learning objects in order to contribute to the learners' progress towards particular goals [6].

While the recommender system algorithms try to address information overload and personalization problems, with growing numbers of existing users and items tremendously, these algorithms will suffer serious scalability and sparsity problems. In addition, most of traditional recommendation algorithms have been developed for e-commerce applications that cannot cover some necessary requirements of learning environments. One of these drawbacks is that they usually consider user's rating information alone and cannot take into account contextual information of user and item such as their attributes. But considering attributes of learner and learning materials such as subject and publisher is a necessary requirement to have a good recommendation. Therefore, it is necessary to consider attributes of materials and learners to improve the quality and accuracy of recommendations in learning environment.

On the other hand, we can consider two groups of attributes for learning material including explicit attributes and implicit (latent) attributes. Explicit attributes are known such as subject and publisher for learning materials and can be extracted by experts, but implicit attributes that are latent can be inferred by historical ratings of learners. Some researches tried to combine attributes (features) of users or items with historical rating for recommendation. Robin [7] reviewed several hybrid recommender methods developed to combine the external (we called explicit) features and historical rating data for higher prediction accuracy. According to the experiment results reported, it is believed that both features and the historical ratings have great values to estimate the prediction function for recommendation.

In order to generate recommendations with higher quality and accuracy and alleviating some problems in existing recommender algorithms such as sparsity in learning environment, this research combines explicit and implicit attributes of learners and materials in the unified model. This model has two modules. In the implicit attribute-based module, genetic algorithm is used for extracting implicit attributes of learners from historical rating in the shape of weight vectors. In the explicit attribute-based module, preference matrix (PM) is introduced that can model the interest of learners based on explicit attributes of learning materials in a multidimensional space. The main contribution of this paper is improving the quality and

accuracy of recommendation and addressing sparsity problem using combining implicit and explicit attributes of learners in a unified model by genetic algorithm and a multidimensional information model. Using this recommender system, tutors can improve the performance of the teaching process and learners can find their suitable online materials.

Rest of this paper is organized as follows: In Section 2, the previous related works on e-learning material recommender systems are discussed. Section 3 introduces the overall system framework and describes the proposed mechanism step by step. Experiment section applies the proposed algorithm for a dataset to evaluate and analyze the performance of method. Finally, Conclusion section provides the concluding remarks.

2. Literature review

Recommender systems have already been implemented in real e-commerce applications such as Amazon [8] and CDNow [7] where they are used to recommend to online shoppers, products and services that they might otherwise never discover on their own. There have also been several pioneering research system prototypes, such as Syskill and Webert [9], Fab [10], and GroupLens [11]. Many recommendation systems in various fields such as movies, music, news, commerce and medicine have been developed but few in education field [12]. With the appearance of e-learning, learning material (learning content or learning resource) recommendation is a new topic in recommendation systems.

Most of recommendation systems are designed either based on content-based filtering or collaborative filtering. Both types of systems have inherent strengths and weaknesses, where content-based approaches directly exploit the product information, and the collaboration filtering approaches utilize specific user rating information. In addition, to produce the accurate and effective recommendations, researchers proposed several different algorithms, some of which derive from the achievements of data mining. Some of recommending algorithms are user-based collaborative filtering [13], Item-based collaborative filtering [10], Cluster-based collaborative filtering [14], Dimension reduction based collaborative filtering [15], Horting Graph-theoretic collaborative filtering [16] and Bayesian network based recommendation [17]. In the following of this section, we explain some researches about recommender systems in the area of e-learning in four categories.

Collaborative filtering: Majority of researchers used collaborative filtering (CF) based recommendation system [18–21]. Based on the assumption that users with similar past behaviors have similar interests, a CF system recommends items that are liked by other users with similar interests. Collaborative filtering methods are completely independent of the intrinsic properties of the items being rated or recommended. CF was used by Soonthornphisaj et al. [22] for prediction the most suitable materials for the learner. At first, the weight between all users and the active learner is calculated by Pearson correlation. Then, n users that have the highest similarity to the active learner are selected as the neighborhoods. Finally, using the weight combination obtained from the neighborhoods, the rating prediction is calculated. This strategy considers learner's rating information alone and neglects content-based relativity between

materials and other contextual information. Bobadilla et al. [23] incorporated the learners score (obtained from a test) into the calculations by a new equation for CF for material prediction. Their experiment showed that the method obtained high item-prediction accuracy. However CF approach does not consider attribute of items and users.

Content based filtering: The recommendations are done based only on the profile made taking into consideration the object content analysis the user has evaluated in the past. The content based RSs are mainly used to recommend documents, Web pages, publications, jokes or news. Khribi et al. [24] used learners' recent navigation histories and similarities and dissimilarities among the contents of the learning materials for online automatic recommendations.

Data mining: The data mining techniques use the gathered information about the learner behavior, such as navigation history, to produce recommendations. These techniques are suitable to recommend the sequence of learning materials (i.e., learning path) rather than the learning materials itself. For example, Romero et al. [25] developed a specific Web mining tool for discovering suitable rules in recommender engine. Their objective was to be able to recommend to a student the most appropriate links/WebPages to visit next. This strategy does not consider contents of learning materials for improving the accuracy of recommendation. Clustering was proposed by Hammouda and Kamel [26] to group learning documents based on their topics and similarities. Data mining techniques such as Association Rule mining, and inter-session and intra-session frequent pattern mining, were applied by Zaiane [3]. Sunita and Lobo used a classification algorithm for the data selected from Moodle database to classify the data, then they used Apriori Association Rule algorithm for Recommendation [27].

Hybrids: Each recommendation strategy has its own strengths and weaknesses. Hence, combining several recommendation strategies can be expected to provide better results than either strategy alone [28,29]. Most hybrids work by combining several input data sources or several recommendation strategies. Liang et al. [30] implemented the combination of content-based filtering and collaborative filtering to make personalized recommendations for a courseware selection module. Liu and Shih [31] designed a material recommendation system based on association rule mining and collaborative filtering. Since the user's preferences were predetermined (from the result of the web usage mining), the system was able to reduce the work load that was required to develop the system as search engine. Khribi et al. [32] proposed two modules: an off-line module which preprocesses data to build learner and content models, and an online module which uses these models on-the-fly to recognize the students' needs and goals, and predict a recommendation list. Li et al. [33] discovered content-related item sets CF then applied the item sets to sequential pattern mining and generated sequential pattern recommendations to learners. Some researchers also try to use semantic information for recommendation [34].

As said before some researches combine attributes (features) of items or users with historical ratings to get better recommendations. These researches really combine only explicit attributes. Claypool and Gokhale [35] introduced a simple lin-

ear combination of recommendation scores from different recommenders. Robin [7] reviewed some of main approaches that use this approach. In summary, in order to improve the learning material recommendation efficiency and alleviate some problems such as sparsity, this research develops a unified model for combining multi-dimensional attributes of materials and learner's rating information. In addition, this research introduces implicit attributes and uses genetic algorithm for optimized extraction of these attributes.

3. Proposed recommendation approach

In this section, at first the system framework is presented and then the proposed recommendation mechanism is described step by step.

3.1. Proposed recommender framework

Vector space model is implemented for user's preference modeling in the most of recommendation algorithms. These vectors are ratings of user for items. In these approaches, according to the similarity between vectors or similarity between ratings of users, most relevant items are recommended to users. But these approaches do not have adequate accuracy for learning environment, because materials usually have several kinds of attributes with different values and different learners may place different emphases on these attributes. For example materials have subject, sub-subject and publisher as attributes and each attribute has values for example for subject we have "mathematic, computer science".

Truly, rating of a user for an item represents the overall rating value on different attribute value of product. Therefore, two users that have similar overall rating values for a specific item may place different emphases on its attributes. As a result, to have a good personalization in e-learning recommender systems, it is necessary to consider different attributes of materials [36].

Learning materials usually have several kinds of attributes. Therefore, in order to consider learner's preference accurately, attributes of learning materials should be taken into account as much as possible. Since the ratings depend on needs and attributes of learners and also attributes of materials, the rating function could be denoted as $\varphi(M, \vec{U}, \vec{T})$, M is a prediction model learned from the historical rating data. \vec{U} and \vec{T} are attributes of the learners and learning materials, respectively. Based on this view, the objective of recommender system problem is to find a fit relationship between spaces attributes of user and items to generate appropriate recommendation. Unfortunately, in most cases we cannot use the mentioned model. Because the selection of all suitable attributes for the learner and material in a CF problem is an almost impossible mission. Even if the attribute set is chosen, it is approximately impossible to collect the corresponding data because some data are involved the privacy of people or some attributes could not be described and coded formally. This leads to low accuracy of prediction as it is only based on the limited observed attributes [37,38].

However, we can use the historical rating data in a user-item matrix for discovering some valuable attributes of learner and learning material that are called implicit attributes reflecting characteristics of learning material and learner.

Thus, we can use the prediction models built based on the observed attributes or explicit attributes plus latent attributes or implicit attributes to improve the recommendation process for acquiring higher prediction accuracy [36]. In this research, the explicit attributes were modeled using PM and also genetic algorithm is used to find the relationship between the overall rating and the underlying implicit attributes weight vector for each learner. More specifically, given the ratings data of a learner, GA computes his/her preference model in terms of implicit attributes weight.

Fig. 1 shows the framework of the proposed recommender system. The proposed framework has two recommenders. In the multidimensional information model based recommender for learners' modeling, server usage logs of learners are collected in the certain period. Then using this information and rating information, PM is built for each learner. Then, according to the new similarity between learners, ratings are predicted. In genetic based recommender module, the weights of implicit attributes for each learner are calculated using genetic algorithm. The proposed genetic algorithm can calculate the interest of learners for each attribute of learning materials. In the online mode, the material access history of the active learner is extracted from the server log file, starting from the time that the learner connected to the e-learning system until s/he asks for recommendations. Finally, results of two recommender systems are combined with each other. In the following of this section, the detailed steps are presented.

3.2. Genetic based recommender

With growing number of users and items tremendously for learning environment, recommendation algorithms will suffer serious scalability problems, with computational materials going beyond practical or acceptable levels. Therefore, this research uses genetic algorithm as a metaheuristic algorithm for optimization of attributes weight.

3.2.1. Optimization of implicit attributes

The GA mimics the process of natural evolution by combining the survival of the fittest among solution structures with a structured, yet randomized, information exchange and creates

offspring. Each candidate solution is represented by a sequence of numbers known as chromosome. In this research, each element (gene) in a string represents an implicit attribute weight. A judiciously selected set of chromosomes is called a population and the population at a given time is a generation. The population size, which remains fixed from generation to generation, has a significant impact on the performance of the GA. This size is to be specified by the user depending upon the number of elements in the string and the problem complexity. In this research, this parameter is selected by trial and error. A randomly generated set of strings makes the initial population. Optimization of the initial population is done by GA, using an appropriately defined fitness function. In the following of this section we, describe GA process step by step.

Coding strategy: Let $w_i = (w_{i1}, w_{i2}, \dots, w_{iK})$ and $e_i = (e_{i1}, e_{i2}, \dots, e_{iK})$ indicate attributes weight vector for user i item i where K is number of defined attributes and $\sum_{j=1}^K w_{ij} = 1$, $\sum_{j=1}^K e_{ij} = 1$. In this research, each weight vector will be represented by the following string of 0 s and 1 s:

$$b_1^0 \dots b_1^1 b_1^0 \dots b_2^0 \dots b_2^1 b_2^0 \dots b_{K-1}^0 \dots b_{K-1}^1 b_{K-1}^0 \dots b_K^0 \dots b_K^1 b_K^0$$

Since the value of each weight is continuous and also between 0 and 1, we make 1/1000th precision for each attribute weight by 10 bits. These 10 bit binary numbers are transformed into decimal floating numbers, ranging from 0 to 1 by applying the following equation:

$$x' = \frac{x}{2^{10} - 1} \quad (1)$$

where x is the decimal number of the binary code for each attribute weight. Two matrixes of attributes weight $W_U = (w_1, w_1, \dots, w_N)^T$ and $W_I = (e_1, e_1, \dots, e_M)^T$ that indicate attributes weight vectors for N users and M items respectively become the optimizing targets. Its initial solution could be some random values gained by an off-line process. At the basis of initial population, new individuals produced in each iteration are evaluated by fitness function.

Fitness function: Fitness is an evaluated function to analyze the attributes weight of individual and judge its prediction

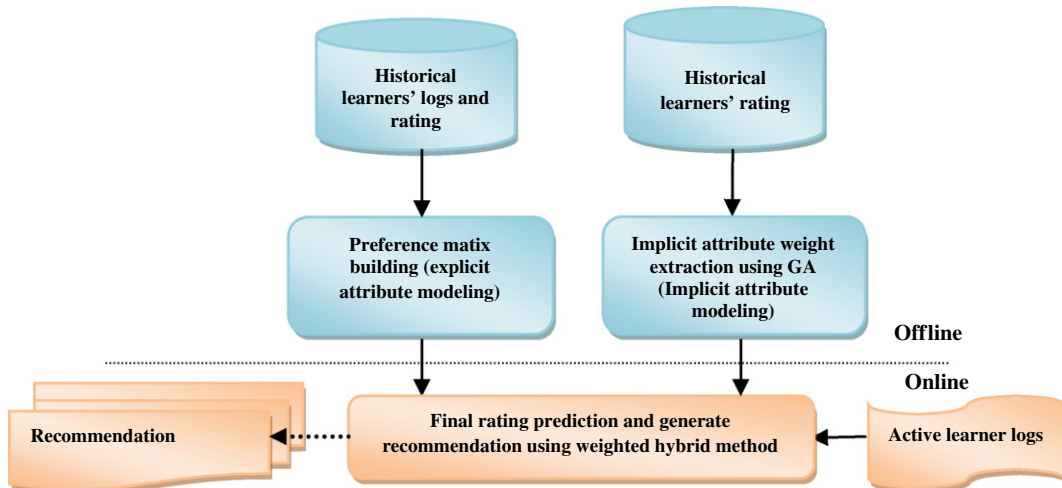


Figure 1 System framework of the proposed material recommender system.

accuracy. When individual w_i is applied to generate recommendation's results for user i , the similarity between prediction rating with the actual rating values can express its prediction accuracy. It is the basis of fitness. So, the accuracy function is defined as follows:

$$f(W_U, W_I) = \sum_{i=1}^N f(w_i) = \sum_{i=1}^N \sum_{j=1}^{M_i} \left| \sum_{k=1}^K w_{ik} \cdot e_{jk} - r_{ij} \right| \quad (2)$$

where r_{ij} is actual rating of item j by user i , w_{ik} and e_{jk} are weight of attribute k for user i and item j respectively and M_i is number of rated items by user i . When $f(W_U, W_I)$ is lower, the accuracy prediction would be higher.

Selection operation: The selecting of selection operators is an important part in genetic algorithms. This part is independent of other parts in genetic algorithms and has no direct relation with the problem and with the fitness function, crossover operator and mutation operator used in genetic algorithms [39]. In this research, a probabilistic selection is performed based upon the individual's fitness such that the better individuals have an increased chance of being selected. Here, the universal sampling method is adopted for selecting the good strings and the probability of selecting each string is calculated by:

$$p_c(W_U, W_I) = 1 - \frac{f_c(W_U, W_I)}{\sum_{c=1}^{PS} f_c(W_U, W_I)} \quad (3)$$

where $f_c(W_U, W_I)$ denotes the value of fitness function for chromosome c , PS is number of individuals in the population or population size and $p_c(W_U, W_I)$ denotes the probability of selecting chromosome c . Because the sum of fitness in a population is constant, an individual with lower fitness (higher prediction accuracy) has larger probability to be chosen. We find that the universal sampling method scheme yields a good individual to be selected for the reproduction of the next population. We hope it would be helpful in improving the efficiency of our algorithm.

Crossover and mutation operation: Crossover is a process of taking more than one parent solutions and producing a child solution from them. The crossover operator takes two chromosomes selected and tries to mate them generating the individuals for the next generation. In this work, one-point crossover is used to produce offspring. Single crossover point on both parents' strings is selected randomly. All gens beyond that point in either string is swapped between the two parent chromosomes.

Mutation operator is used to investigate some of the unvisited points in the search space, and also to avoid premature convergence of the entire feasible space caused by some super chromosomes. This operator makes random changes in one or more elements of the string. Mutation is done with a small probability, called mutation probability or rate. According to mutation rate, randomly selecting some elements of individual, and changing its value, new individual can be gained. It is a local random searching method to keep diversity of population.

3.2.2. Recommendation

After implicit attributes weight optimization, similarity degree between learners by implicit attribute based (IAB) method can be calculated by following formula that is a cosine similarity:

$$sim_{IAB}(L_a, L_b) = \frac{\sum_{i=1}^K w_{ai} \cdot w_{bi}}{\sqrt{\sum_{i=1}^K w_{ai}^2} \cdot \sqrt{\sum_{i=1}^K w_{bi}^2}} \quad (4)$$

The prediction rating of learning material i by L_a using implicit attribute based method is $P_{IAB}(L_a, i)$ that is gained by the rating of L_a neighborhood, $N_{IAB}(L_a)$, that have rated i before. The computation formula is as follows:

$$P_{IAB}(L_a, i) = \bar{R}_{L_a} + \frac{\sum_{j \in N_{IAB}(L_a)} sim_{IAB}(L_a, L_j) \times (R_{L_j}(i) - \bar{R}_{L_j})}{\sum_{j \in N_{IAB}(L_a)} sim_{IAB}(L_a, L_j)} \quad (5)$$

where \bar{R}_{L_a} and \bar{R}_{L_j} denote rating average of learning materials rated by active learner L_a and L_j respectively and $sim_{IAB}(L_a, L_j)$ is the similarity between active learner L_a and L_j that is a member of $N_{IAB}(L_a)$. However, if a learner does not have enough similar learners, traditional algorithms will generate a lot of dissimilar learners which will definitely decrease the prediction accuracy of active learner. Thus, in order to enhance efficiency of calculation, learners set should be preliminarily filtered via setting a similarity matching threshold τ . The two learners are effective similar neighbor only if the similarity between them is at least τ .

3.3. Multidimensional information model based recommender

In this section, learner interests are modeled as a multidimensional data structure according to explicit attributes of learning materials. Then, to generate recommendation, similarity between learners is computed based on similarity between their multidimensional data structures.

3.3.1. Multidimensional information model

Rating of a material that has certain explicit attribute values indicates the importance of these explicit attribute values for the learner; it can be considered as base for weighting of explicit attributes for the learner. Therefore, in order to consider learner's preference accurately, attributes of learning materials should be taken into account. Therefore, the material attributes' description model can be defined as a vector $C = \langle A_1, A_2, \dots, A_m \rangle$ where A_t denotes the t th dimensional attribute's name of material.

This research introduces a multidimensional attribute-based framework for recommendation that involves attributes of materials in the recommendation process, but selection of appropriate attributes may vary in the different systems. System developer can use Learning Object Metadata (LOM) to select suitable attributes. In this research, according to the simplicity and usefulness, we select four attributes including: subject, secondary subject, education type (Bachelor Degree (B.D.), Master Degree (M.D.), PhD Degree (PhD.D.)) and publisher of material. Based on this description model, a certain material is defined as $MA_j = [(AK_1, AW_1), (AK_2, AW_2), \dots, (AK_m, AW_m)]$, where AK_i denotes i th dimension attribute's keyword of material M_j and AW_i denotes the appropriate weight value for i th attribute and $AW_1 \geq AW_2 \geq \dots \geq AW_m$ and $\sum_{i=1}^m AW_i = 1$. For example:

$$M_j = [(Mathematic, 0.35), (Probability, 0.3), (Master \times degree, 0.2), (Author5, 0.15)].$$

we use the order of accessed material as useful information for learners' dynamic interest modeling. The preference of a learner's recently accessed materials has an important role to the future interests. However, in the existing vector-space based preference modeling methods, the dynamic changes of learner's preference are neglected and always all accessed materials treat equally. Thus, by changing the learner's interests and preferences with the passage of time, the recommender system cannot produce the accurate recommendations. Herein, Gradual Forgetting Function (GFF) concept is introduced in order to reflect dynamic interests and preference of a learner more accurately. In this research, we introduce a quadratic function, as follows:

$$h(x(M_i)) = 1 - \lambda \left(1 - \left(\frac{x(M_i) - x_{\min}}{x_{\max} - x_{\min}} \right)^2 \right) \quad (6)$$

$$x_{\min} < x < x_{\max}, 0 \leq \lambda \leq 1, 0 < h(x(M_i)) < 1$$

where $x(M_i)$ is the order of M_i in the material access order by learner L_i , x_{\max} is the order of the latest accessed material and x_{\min} is the order of the first accessed material. Therefore, the effect of M_j to L_i 's future interest will becoming smaller with material access process going on and $h(x)$ should be attenuated gradually. In $h(x)$, λ is an adjustable parameter used to describe the change rate of learner's preference, and the bigger of λ , the quicker of the forgetting.

The $h(x)$ attenuation with $\lambda = 0.95$ is shown in Fig. 2. Based on Eq. (6), $h(x)$ value of the latest accessed material is equal to 1, and with access going on, $h(x)$ value of materials could be updated.

The central element of all recommender systems is the user model that contains knowledge about the individual preferences which determine his or her behavior in a complex environment of web-based system. The attention-degree of learners is inferred by learners rating. In this paper, a multidimensional information model is introduced to combine attributes of accessed materials and learner's rating information for making a preference model for learner. For each learner, we consider a set of accessed materials by the learner and a preference matrix that model learner's preference. Fig. 3 shows a preference matrix (PM) and a set of accessed materials (AM) by learner L_i .

Definition 1. Each accessed material by learner L_i is defined as a four-tuple $(MID, X(MID), NH, RR)$, where MID denotes accessed material ID of learner L_i , $x(MID)$ denotes the order of MID in the material access order by learner L_i , NH denotes the normalization value of $h(x)$ function for accessed material MID by learner L_i and RR denotes the rating of L_i to material

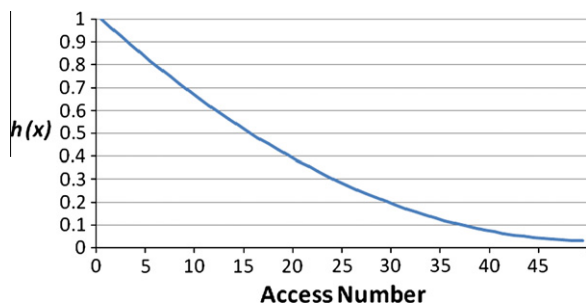


Figure 2 Gradual forgetting function.

MID . For example in Fig. 3 material $M1$ is the first material that have been studied by L_i , therefore $x(M1) = 1$ and $h(1) = 0.02$. Learner L_i has rated this material four, therefore $RR = 4$.

Definition 2. The preference model of learner L_i is defined as a matrix with m rows in which m denotes the number of attributes of materials. Each cell in this matrix is defined as a four-tuple $(KA, NH, RR, level)$, where KA is the keyword of the $level$ th attribute accessed materials by learner L_i and $level$ denotes the row number for this tuple.

Definition 3. NH of each tuple in PM is defined as the sum of the NH value of all materials in AM that their attributes' keyword is same with the attributes' keyword of tuple.

Definition 4. RR of each tuple in PM is defined as the average of the RR value of all materials in AM that their attributes' keyword is same with the attributes' keyword of tuple.

We make and update the preference matrix by the following strategy:

Search the keywords of the latest accessed material attributes $(MA_j = [AK_{j1}, AK_{j2}, \dots, AK_{jm}])$ in PM from the upper row to the bottom row. If the keyword of i th attribute cannot be matched, a new column is created that the first $i - 1$ its rows give the corresponding matched attributes and the next $m - i + 1$ its rows give the latter $m - i + 1$ attributes of material. Then the NH s and RR s of matrix are updated according to Definitions 3 and 4.

PM can model preferences of a learner. System can transfer preference of a learner from accessed materials to high-level attributes and indicate importance each attribute for the learner. In this matrix, each accessed material corresponds to a unique path from the first row to the last row, and the keywords of all tuples located in this path correspond to the relevant keywords of the material's attributes.

3.3.2. Recommendation

As a logical assumption, two learners with similar attribute keywords in their PM can be considered as similar neighbors. Based on this assumption, we can solve sparsity problem. For defining similarity degree, three rules are implemented:

- (1) The more similar attributes of learner L_a and learner L_b 's accessed materials, the larger similarity between them.
- (2) The more similar the order of accessed materials of learner L_a and learner L_b , the larger similarity between them.
- (3) The more similar the rating data of learner L_b and learner L_b , the larger similarity between them.

Therefore, the similarity degree between two learners can be calculated based on Attributes Intersection Vector (AIV) between two their corresponding PMs. AIV between learner L_a and L_b , $AIV(L_a, L_b)$, is defined as the maximum intersection between columns of PM_a and PM_b with same keyword in each row. After matching process, we have an AIV such as Fig. 4 shows.

The calculation of similarity between two learners can be divided into two aspects as: preference based similarity and

$$PM(L_i) = \begin{bmatrix} (\text{Info. Tech., 0.1, 4}) & (\text{Info. Tech., 0.1, 4}) & (\text{Math., 0.9, 3}) & (\text{Math., 0.9, 3}) & (\text{Math., 0.9, 3}) & (\text{Math., 0.9, 3}) \\ (\text{Intell. sys., 0.08, 5}) & (\text{Net., 0.02, 3}) & (\text{Probab., 0.32, 2.5}) & (\text{Probab., 0.32, 2.5}) & (\text{Stat., 0.58, 3.5}) & (\text{Stat., 0.58, 3.5}) \\ (\text{Ph.D.D., 0.08, 5}) & (\text{B.D., 0.02, 3}) & (\text{M.D., 0.32, 2.5}) & (\text{M.D., 0.32, 2.5}) & (\text{M.D., 0.58, 3.5}) & (\text{M.D., 0.58, 3.5}) \\ (\text{A1, 0.08, 5}) & (\text{A2, 0.02, 3}) & (\text{A3, 0.19, 3}) & (\text{A4, 0.13, 2}) & (\text{A5, 0.26, 3}) & (\text{A6, 0.32, 4}) \end{bmatrix}$$

(a) Preference Matrix sample

$$AM(L_i) = \{(M2, 2, 0.08, 4), (M1, 1, 0.02, 4), (M4, 4, 0.19, 4), (M3, 3, 0.13, 4), (M5, 5, 0.26, 4), (M6, 6, 0.32, 4)\} \quad (\text{b})$$

(b) Accessed Material set sample**Figure 3** Components of multidimensional information model.

learner rating based similarity. The preference based similarity sim_{PB} can reflect the similarity between learners based on attributes. Inspired by Cosine similarity, the calculation of $sim_{PB}(L_a, L_b)$ can be defined as follows:

$$sim_{PB}(L_a, L_b) = \frac{\sum_{i \in AIV(L_a, L_b)} AW_i \cdot NH_{ai} \cdot NH_{bi}}{\sqrt{\sum_{i \in PM_a} AW_i \cdot NH_{ai}^2} \cdot \sqrt{\sum_{i \in PM_b} AW_i \cdot NH_{bi}^2}} \quad (7)$$

where NH_{ai} indicates the value of NH in the i -th row's matching for learner a . AW_i indicates attribute weight that was defined before. For reflecting the similarity between the rating vectors of two learners, the learner rating based similarity can be applied to overcome sparsity rating problem. Inspired from Pearson, this similarity can be defined as follows:

$$sim_{RB}(L_a, L_b) = \frac{\sum_{i \in AIV(L_a, L_b)} [RR_{i,level}(L_a) - \overline{RR}_{AIV}(L_a)] \cdot [RR_{i,level}(L_b) - \overline{RR}_{AIV}(L_b)]}{\sqrt{\sum_{i \in AIV(L_a, L_b)} (RR_{i,level}(L_a) - \overline{RR}_{AIV}(L_a))^2} \cdot \sqrt{\sum_{i \in AIV(L_a, L_b)} (RR_{i,level}(L_b) - \overline{RR}_{AIV}(L_b))^2}} \quad (8)$$

where $RR_{i,level}(L_a)$ indicates rating of user L_a in row i of PM_a corresponding with row i of AIV . $\overline{RR}_{AIV}(L_a)$ indicates the mean value of L_a 's rating in rows of PM_a corresponding with rows of AIV .

It must be noted, in the calculation of $sim_{RB}(L_a, L_b)$ that computes the similarity between RR value of rows on PM_a and PM_b which correspond to each row on $AIV(L_a, L_b)$, does not need to have the identical accessed materials between two learners. By this definition of similarity, we can overcome sparsity rating problem. Finally, Explicit Attribute Based (EAB) similarity between L_a and L_b can be calculated as follows:

$$Sim_{EAB}(L_a, L_b) = \beta \cdot Sim_{PB}(L_a, L_b) + (1 - \beta) \cdot Sim_{RB}(L_a, L_b) \quad (9)$$

The prediction rating of learning material i by L_a using implicit attribute based method is $P_{EAB}(L_a, i)$ that is gained by the rating of L_a neighborhood, $N_{EAB}(L_a)$, that have rated i before. The computation formula is as the follows:

$$P_{EAB}(L_a, i) = \overline{R}_{L_a} + \frac{\sum_{j \in N_{EAB}(L_a)} sim_{EAB}(L_a, L_j) \times (R_{L_j}(i) - \overline{R}_{L_j})}{\sum_{j \in N_{EAB}(L_a)} sim_{EAB}(L_a, L_j)} \quad (10)$$

$$AIV(L_a, L_b) = \begin{bmatrix} (\text{Info. Tech., ...}) \\ (\text{Intell. sys., ...}) \\ (\text{Ph.D.D., ...}) \end{bmatrix}$$

Figure 4 Attributes intersection vector sample.

where \overline{R}_{L_a} and \overline{R}_{L_j} denote rating average of learning materials rated by active learner L_a and L_j respectively and $sim_{EAB}(L_a, L_b)$ denote the similarity between active learner L_a and L_j that is a member of $N_{EAB}(L_a)$.

3.4. Final recommendation

The development of final recommendation is done in this stage. Truly, we proposed two methods for learning material recommendation: Explicit Attribute Based Collaborative Filtering (EAB-CF) and Implicit Attribute Based Collaborative Filtering (IAB-CF). These two methods can be combined for final recommendation. A linear combination of EAB-CF and IAB-CF is used for recommendation (EB-IB-CF). Therefore for rating prediction the following formula is used:

$$P_{L_a}(i) = \alpha \cdot P_{IAB}(L_a, i) + (1 - \alpha) \cdot P_{EAB}(L_a, i) \quad (11)$$

where $P_{L_a}(i)$ denotes final prediction rate for learning material i by L_a . Finally top N -learning materials with higher predicted rate are considered as recommendation results. Each of the approaches uses some of useful information in their recommendation process. Thus, hybrid approach can resolve their weak points and improve the accuracy and quality of recommendation results.

4. Experiments

We have conducted a set of experiments to set parameters and examine the effectiveness of our proposed recommender system in terms of recommendation accuracy and quality.

4.1. Evaluation metrics and data set

In order to check the performance of the proposed algorithm, a real-world dataset is applied in our simulations. MACE¹ dataset that is pan-European initiative to interconnect and disseminate digital information about architecture is used for experiment. This dataset is issued from MACE project that is done from September 2006 to September 2010. This dataset contains 1148 learners and 12,000 materials.

The precision and recall are most popular metrics that evaluate decision support accuracy. For the evaluation of recommender system, they have been used by various researchers [12,14]. The precision is a measure of exactness and recall is a measure of completeness. Several ways to evaluate precision and recall exists [40]. When referring to Recommender Systems the recall can be defined as follows:

¹ Metadata for Architectural Contents in Europe.

$$\text{Recall} = \frac{|test \cap top - N|}{|test|} \quad (12)$$

where $top - N$ denotes the recommendation set and $test$ denotes the test set. The precision when referring to recommender systems can be defined as follows:

$$\text{Precision} = \frac{|test \cap top - N|}{NR} \quad (13)$$

where NR denotes number of recommendations. Since increasing the size of the recommendation set leads to an increase in recall but at the same time a decrease in precision, we can use F_1 measure [41] that is a well-known combination metric with the following formula:

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (14)$$

To evaluate prediction accuracy, we have used the Mean Absolute Error (MAE), a statistical accuracy metric, [40,42] is computed as

$$MAE = \frac{\sum_{U=1}^{U=N} |R_U(i) - P_U(i)|}{|N|} \quad (15)$$

where $P_U(i)$ is the predicted rating for material i by learner U , $R_U(i)$ is the learner given rating for material i by learner U , and N is the total number learners. We have used the Rooted Mean Square (RMS) error, and 0/1 loss error also.

4.2. Parameters setting

In this section, first the impact of input parameters is analyzed on the recommendation performance.

4.2.1. Parameters of IAB-CF

The probabilities of Crossover and mutation operators have an important role in GA, so it is necessary to define proper operator's probability to achieve a better performance. However, the optimal values of crossover and mutation probabilities are problem specific that often are obtained by trial and error. Therefore, we examine the impacts of various combinations of PC (probability of crossover) and PM (probability of mutation) on the recommendations quality of the proposed approach. According to the experiments, $PC = 0.83$ and $PM = 0.17$ give good results for our problem. In order to choose the population size, we have considered the criterion of using a number of individuals in the population which is the double of the number of bits used to represent each individual [43]. Consequently, since we used $K \times 10 = 15 \times 10 = 150$ bits for each attributes weight vector, we select 300 as population size. The number of individuals keeps constant through every generation. We only keep the 5% of the best individuals from each generation to obtain the next one (elitist selection). The genetic algorithm stops when there is an individual in the population with a fitness value lower than a constant γ . We

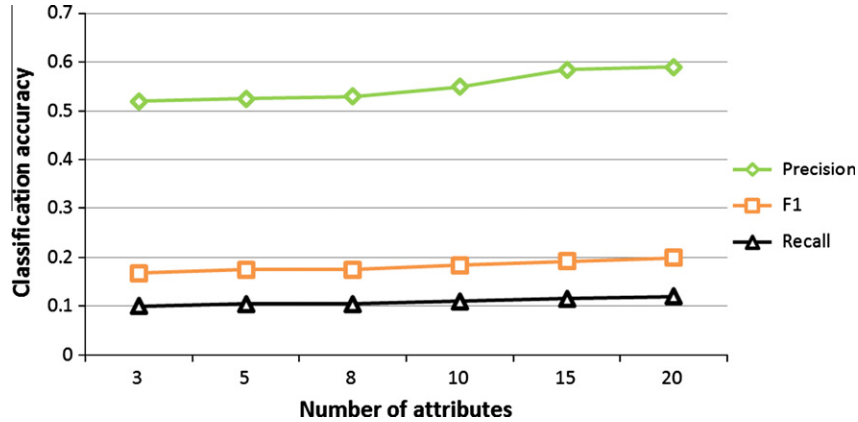


Figure 5 Performance of IAB-CF with respect of K .

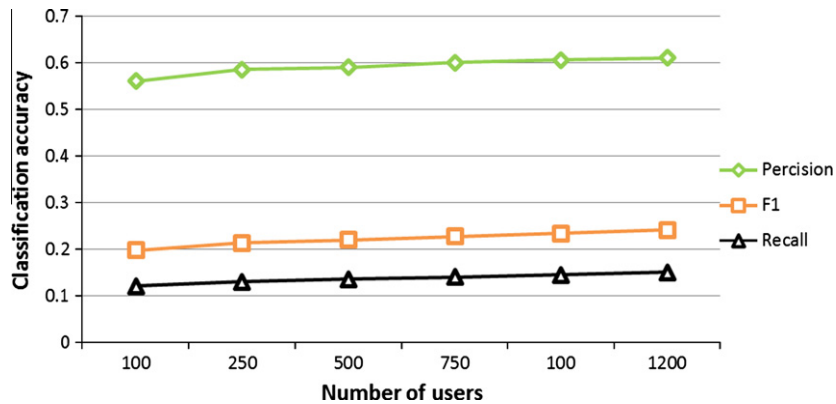


Figure 6 Performance of IAB-CF with respect of the number of users.

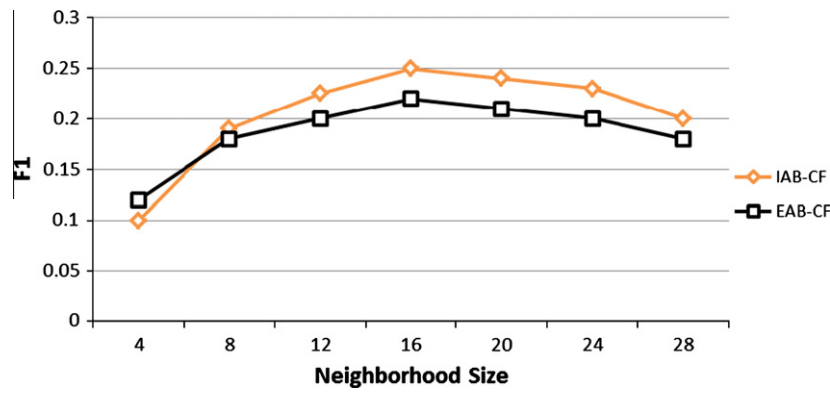


Figure 7 Performance results of proposed approaches with respect to neighborhood size.

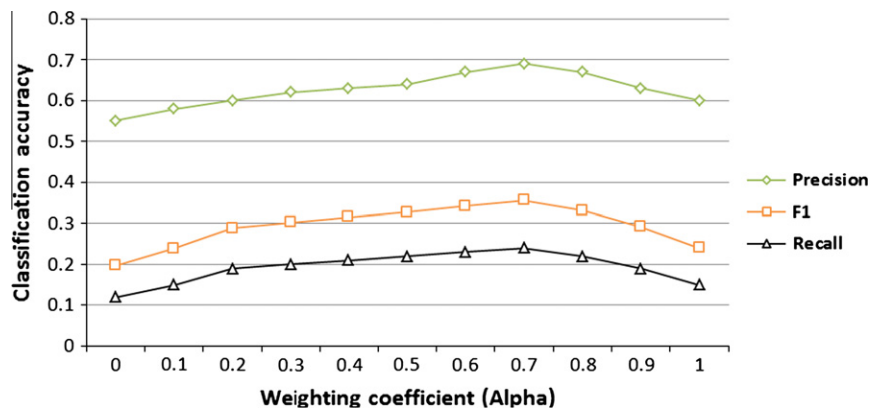


Figure 8 Effect of α on precision, recall and F1 of *EB-IB-CF*.

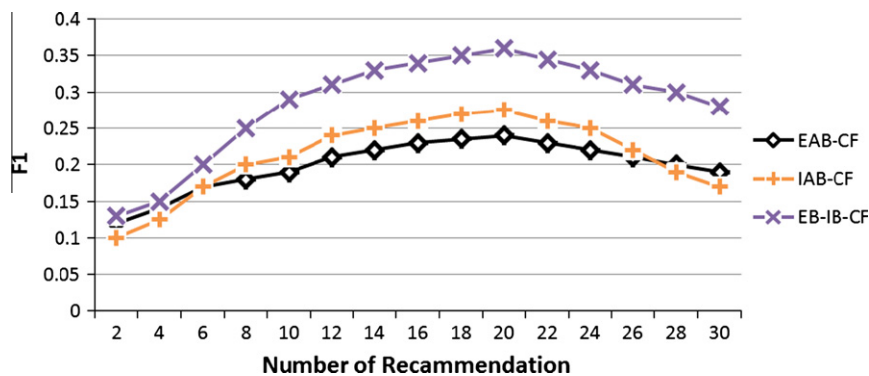


Figure 9 Comparison of the proposed approaches with respect to number of recommendation.

select γ according to the optimal values found in results of the other recommendation approaches.

One of the parameters for the IAB-CF is number of attributes; K . The performance of method may vary with varying number of considered attributes for users. Fig. 5 shows the results obtained for the proposed model with different number of attributes, where the minimum number of rating required for test users, M , was 30, the number of user, N , was 500. It can be seen that the performance improves steadily with the number of attributes increasing, but not very much.

Since number of attributes of user could be seen as the user communities, this number should be a relative small number according to experience. Fig. 6 shows the results obtained

Table 1 A comparison of prediction accuracy of various methods.

Method	Error		
	MAE	RMS	0/1 loss
EB-IB-CF	0.787	1.051	62.7
EAB-CF	0.891	1.181	65.2
IAB-CF	0.832	1.163	63.2
User based	0.873	1.172	64.7
Mixture pLSA	0.848	1.170	63.4
CR [45]	0.994	–	–
BC [45]	1.103	–	–
BN [45]	1.066	–	–

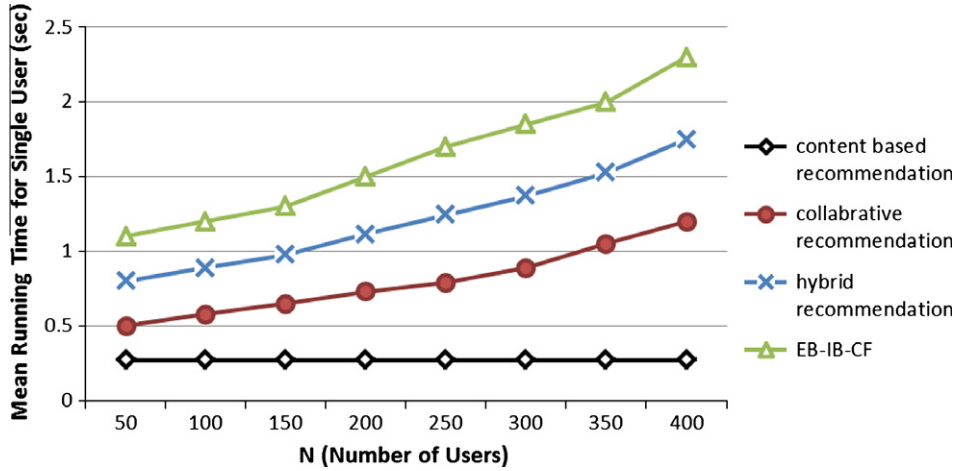


Figure 10 The mean running times for single learner of algorithms with respect of N .

for the IAB-CF method, while parameters set as $M = 30$, $K = 15$, and the number of user was changed from 100 to 1200. According to Fig. 6, the performance of the method is good for different values of users. It means that the choosing the number of attributes, K is not strict.

In addition, there are three other parameters including: number of neighborhoods, weighting coefficient in the combination of two type of similarity in EBA-CF (beta) and EAB-CF and IAB-CF (alpha) that must be adjust for better recommendation. According to previous experiments, EBA-CF for $\beta = 0.35$ gives the best prediction accuracy.

4.2.2. Neighborhood size

Fig. 7 shows the influence of neighborhood size on the performance of EAB-CF and IAB-CF with F1 metric while $N = 500$, $M = 50$, $K = 15$. It was observed that the size of the neighborhood affects the quality of top-N recommendations. Considering this diagram, we select 16 as the optimal choice of neighborhood size.

4.2.3. Weighting coefficient (α)

Fig. 8 shows the impacts of α on the precision, recall and F1 of EB-IB-CF while $N = 500$, $M = 50$ and $K = 15$. It indicates that taking into consideration a combination of EAB-CF and IAB-CF to predict rating will play a positive role in the recommendation process, but α does not acknowledge ‘the larger the better’ rule: the best precision can be obtained with $\alpha = 0.7$.

4.3. Performance comparison

In experiments, the data is ordered by learners’ access timestamp, and then is divided into a training set and a test set. In order to increase the number of records in test set as much as possible so as to eliminate the effect of accidental factor, the top 60% access records of each learner in ordered dataset are used as training set and the remnant 40% access records are used as test set. The algorithm is then trained on the training set and top N-learning materials are predicted from that learners’ test set.

To evaluate the sensitivity of different algorithms on number of recommendation (NR), we compare proposed

approaches that is presented in Fig. 9 while $N = 500$, $K = 15$, and $M = 50$. As Fig. 9 shows combination of explicit based and implicit based collaborative filtering has the best performance. The relative performance of these methods for different number of recommendations is different but as in general we can say the best performance is from 18 to 22.

Although, this paper presents a recommender system for learning material, but the proposed approach can be used for some of other area of recommender system. Therefore, to compare our approach with other approach of recommender system, EachMovie dataset has been used. Table 1 presents the experimental results obtained by EB-IB-CF, EAB-CF, IAB-CF, a memory-based method using nearest-neighbor users to predict ratings, results of normalized Gaussian PLSA² mixture method published in Hofmann [44] and results published in Breese et al. [45] including Bayesian clustering (BC), Bayesian networks (BN), Correlation(CR) for EachMovie dataset. Since the data set will influence the results of CF algorithm, comparing of different algorithms is difficult. For the mixture PLSA, results are chosen the best results in Hofmann [44]. The results of user-based and proposed method obtained from the same data set. Comparisons were produced for $N = 500$ users with the average number of ratings about 100, and $M = 50$. As can be seen, the proposed multi-attribute based method has better prediction accuracy of the memory-based, mixture PLSA method and other methods in terms of MAE.

In the final comparison experiment, the mean running times for single learner of EB-IB-CF algorithm, vector space model-based content-based recommendation algorithm [46], user and item combined collaborative-based recommendation algorithm and hybrid recommendation algorithm [47] is compared with respect to N or number of participated learner which are selected from MACE dataset while $K = 15$, $M = 50$ and $NR = 20$. As shown in Fig. 10 at all times, content-based algorithm is faster than any other algorithms. The running time of the proposed and improved hybrid recommendation algorithms are slightly larger than collaborative based algorithm. According these experiments, although the proposed recommendation will get higher precision in most case, it will cost

² Probabilistic latent semantic analysis.

the largest running time. Therefore, there is a trade-off between algorithm running time and recommendation precision when choosing the proposed recommendation algorithm.

5. Conclusions

One of the most important applications of recommendation systems in e-learning environment is personalization and recommendation of learning materials. However, since the repository of learning materials is very massive and these materials have several attributes, there are several drawbacks such as sparsity when applying the existing recommendation algorithms. To address these problems and have a good recommendation for learner, this paper presents a novel personalized recommender system that utilizes explicit and implicit attributes of materials in the unified model. The experiment results show that the proposed approach performs better than the traditional approaches. The main contribution of this paper is improving the quality of recommendations and addressing sparsity problem using genetic algorithm and a multidimensional information model.

The learning processes (resource access processes) usually have some time-dependency relationship and are repeatable and periodic. Therefore, the time-dependency relationship between learning resources in a learning process can reflect learner's resource access latent pattern and preference. For further research, to improve the recommendation process we can make a hybrid approach and mine learner's historical access records for discovering the resource access sequential patterns. Then, using these sequential patterns, we can predict the most probable resource that a learner will access in near future.

References

- [1] Greenhow C, Robelia B, Hughes JE. Learning, teaching, and scholarship in a digital age: Web 2.0 and classroom research: what path should we take now? *Educ Res* 2009;38(4):246–59.
- [2] Kardan AA, Abbaspour S, Hendijanifard F. A hybrid recommender system for e-learning environments based on concept maps and collaborative tagging. In: *The 4th international conference on virtual learning ICVL*; 2009. p. 300–7.
- [3] Zaiane OR. Building a recommender agent for e-learning systems. In: *Proceedings international conference on computers in education*; 2002. p. 55–9.
- [4] Resnick P, Iacovou N, Suchak M, Bergstrom P, Riedl J. Grouplens: an open architecture for collaborative filtering of netnews. In: *Proceedings of the 1994 ACM conference on computer supported cooperative work*; 1994. p. 175–86.
- [5] Klasnja-Milicevic A, Vesin B, Ivanovic M, Budimac Z. E-learning personalization based on hybrid recommendation strategy and learning style identification. *Comput Educat* 2011;56(3):885–99.
- [6] Farzan R, Brusilovsky P. Social navigation support in a course recommendation system. In: *Adaptive hypermedia and adaptive web-based systems*; 2006. p. 91–100.
- [7] Burke R. Hybrid: recommender systems: survey and experiments. *J User Model User-Adapted Interact* 2002;12(4):331–70.
- [8] Mobasher B, Cooley R, Srivastava J. Automatic personalization based on Web usage mining. *Commun ACM* 2000;43(8):142–51.
- [9] Fayyad M, Piatetsky-Shapiro G, Smyth P, Uthurusamy R. *Advances in knowledge discovery and data mining*; 1996.
- [10] Schafer JB, Konstan JA, Riedl J. E-commerce recommendation applications. *Data Mining Knowl Discov* 2001;5(1):115–53.
- [11] Anderson CR, Domingos P, Weld DS. Relational Markov models and their application to adaptive web navigation. In: *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*; 2002. p. 143–52.
- [12] Drachsler H, Hummel HGK, Koper R. Recommendations for learners are different: applying memory-based recommender system technique. In: *Proceedings of the 1st workshop on social information retrieval for technology-enhanced learning and exchange*; 2007. p. 18–26.
- [13] Sarwar B. Application of dimensionality reduction in recommender system – a case study. In: *ACM WEBKDD workshop, DTIC document*; 2000.
- [14] Lee WS. Collaborative learning for recommender systems. In: *Proceedings of the 18th international conference on machine learning*; 2001. p. 314–21.
- [15] Sarwar B, Karypis G, Konstan J, Riedl J. Analysis of recommendation algorithms for e-commerce. In: *Proceedings of the 2nd ACM conference on electronic commerce*; 2000. p. 158–67.
- [16] Aggarwal CC, Wolf JL, Wu K-I, Yu PS. Horting hatches an egg: a new graph-theoretic approach to collaborative filtering. In: *Proceedings of the fifth ACM SIGKDD international conference on knowledge discovery and data mining*; 1999 p. 201–12.
- [17] Herlocker JL. Understanding and improving automated collaborative filtering systems. *Doctoral dissertation*. Citeseer; 2000.
- [18] Tan H, Guo J, Li Y. e-Learning recommendation system. In: *International conference on computer science and software engineering*; 2008. p. 430–3.
- [19] Santos OC, Boticario JG. Recommendation strategies for promoting eLearning performance factors for all. In: *6th Workshop on intelligent techniques for web personalization & recommender systems in conjunction with the 23nd AAAI conference on artificial intelligence*; 2008.
- [20] García E, Romero C, Ventura S, Castro C. An architecture for making recommendations to courseware authors using association rule mining and collaborative filtering. *User Model User-Adapted Interact* 2009;19(1):99–132.
- [21] García E, Romero C, Ventura S, Castroa Cd. A collaborative educational association rule mining tool. *Internet Higher Educat* 2011;14(2):77–88.
- [22] Soonthornphisaj N, Rojsattarat E, Yim-ngam S. Smart e-learning using recommender system. In: *ICIC'06 proceedings of the 2006 international conference on intelligent computing*; 2006. p. 518–23.
- [23] Bobadilla J, Serradilla F, Hernando A. MovieLens, Collaborative filtering adapted to recommender systems of e-learning. *Knowl-Based Syst* 2009;22(4):261–5.
- [24] Khribi MK, Jemni M, Nasraoui O. Automatic recommendations for e-learning personalization based on web usage mining techniques and information retrieval. In: *ICALT'08. Eighth IEEE international conference on advanced learning technologies*; 2008. p. 241–5.
- [25] Romero C, Ventura S, Zafra A, de Bra P. Applying Web usage mining for personalizing hyperlinks in Web-based adaptive educational systems. *Comput Educat* 2009;53:828–40.
- [26] Hammouda K, Kamel M. Collaborative document clustering. In: *Proceedings of the sixth SIAM international conference on data mining (SDM06)*; 2006. p. 453–63.
- [27] Lobo LM, Sunita RJ, Aher B. Mining association rule in classified data for course recommender system in e-learning. *Int J Comput Appl* 2012;39(7):1–7.
- [28] Chen MS, Han J, Yu PS. Data mining: an overview from a database perspective. *IEEE Trans Knowl Data Eng* 1996;8(6):866–83.
- [29] Mahdavi M, Shepherd J. Enabling dynamic content caching in Web portals. In: *RIDE '04 Proceedings of the 14th international workshop on research issues on data engineering: web services for e-commerce and e-government applications (RIDE'04)*; 2004. p. 129–36.
- [30] Liang G, Weining K, Junzhou L. Courseware recommendation in e-learning system. In: *Advances in web based, learning*; 2006. p. 10–24.

- [31] Liu F, Shih B. Learning activity-based e-learning material recommendation system. In: *Proceedings of the ninth IEEE international symposium on multimedia workshops*; 2007. p. 343–8.
- [32] Khribi MK, Jemni M, Nasraoui O. Automatic recommendations for e-learning personalization based on web usage mining techniques and information retrieval. *Educ Technol Soc* 2009;12(4):30–42.
- [33] Li Y, Niu Z, Chen W, Zhang W. Combining collaborative filtering and sequential pattern mining for recommendation in e-learning environment. *Lecture Notes Comput Sci* 2011;7048:305–13.
- [34] Benharzallah S, Kazar O, Caplat G. Intelligent query processing for semantic mediation of information systems. *Egyptian Inform J* 2011;12(3):151–63.
- [35] Claypool M, Gokhale A. Combining content-based and collaborative filters in an online newspaper. In: *SIGIR '99 workshop on recommender systems: algorithms and evaluation*; 1999. p. 195–204.
- [36] salehi M, Nakhai Kamalabadi I. A hybrid attribute based recommender system for e-learning material recommendation. *IERI Procedia* 2012;2:565–70.
- [37] Zhong J, Li X. Unified collaborative filtering model based on combination of latent features. *Expert Syst Appl* 2010;37:5666–72.
- [38] salehi M, Nakhai Kamalabadi I, Ghaznavi ghoushci MB. A new recommendation approach based on implicit attributes of learning material. *IERI Procedia* 2012;2:571–6.
- [39] Salehi M, Tavakkoli-Moghaddam R. A genetic algorithm-based grouping method for a cell formation problem with the efficacy measure. *Int J Ind Syst Eng* 2010;6(3):340–59.
- [40] Herlocker JL, Konstan JA, Terveen LG, Riedl JT. Evaluating collaborative filtering recommender systems. *ACM Trans Inform Syst (TOIS)* 2004;22(1):5–53.
- [41] Shih YY, Liu DR. Product recommendation approaches: collaborative filtering via customer lifetime value and customer demands. *Expert Syst Appl* 2008;35(1–2):350–60.
- [42] Zhao Z, Bing S. An adaptive algorithm for personal recommendation. *J Changchun Univ* 2005;1(1):22–9.
- [43] Alander JT. On optimal population size of genetic algorithms. *Comput Syst Software Eng* 1992:65–70.
- [44] Hafmann T. Latent semantic models for collaborative filtering. *ACM Trans Inform Syst* 2004;22:89–115.
- [45] Breese JS, Heckerman D, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering. In: *Proceedings of the 14th conference on uncertainty in artificial intelligence*; 1998. p. 43–52.
- [46] Adomavicius G, Tuzhilin A. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Trans Knowl Data Eng* 2005;17(6):734–49.
- [47] Baudisch P. Joining collaborative and content-based filtering. In: *Proceedings of the conference on human factors in computing systems*; 1999.