An Effective Recommendation Framework for Personal Learning Environments Using a Learner Preference Tree and a GA

Mojtaba Salehi, Isa Nakhai Kamalabadi, and Mohammad B. Ghaznavi Ghoushchi, *Member, IEEE*

Abstract—Personalized recommendations are used to support the activities of learners in personal learning environments and this technology can deliver suitable learning resources to learners. This paper models the dynamic multipreferences of learners using the multidimensional attributes of resource and learner ratings by using data mining technology to alleviate sparsity and cold-start problems and increase the diversity of the recommendation list. The presented approach has two main modules: an explicit attribute-based recommender and an implicit attribute-based recommender. In the first module, a learner preference tree (LPT) is introduced to model the interests of learners based on the explicit multidimensional attributes of resources and historical ratings of accessed resources. Then, recommendations are generated by nearest neighborhood collaborative filtering (NNCF). In the second module, the weights of implicit or latent attributes of resources for learners are considered as chromosomes in a genetic algorithm (GA), and then this algorithm optimizes the weights according to historical ratings. Then, recommendations are generated by NNCF using the optimized weight vectors of implicit attributes. The experimental results show that the proposed method outperforms current algorithms on accuracy measures and can alleviate cold-start and sparsity problems and also generate a more diverse recommendation list.

Index Terms—Collaborative filtering, learning environment, sparsity, personalized recommender, genetic algorithm, explicit attribute, implicit attribute

1 Introduction

Web-based education has undergone rapid development in recent years. With the growth of many online learning systems, a huge number of e-learning resources have been generated which are highly heterogeneous and in various media formats [1]. The task of delivering personalized learning resources is often framed in terms of a recommendation task in which a system recommends items to an active user [2]. To address information overload and personalization problems in e-learning environments, recommender systems (RS) have been proposed by many researchers. This research also proposes an effective recommendation framework for personal learning environments (PLE) based on the attributes of learners and resources. Using this approach, tutors can improve the performance of the teaching process and learners can find suitable online resources.

1.1 Motivation

With the explosion of e-learning resources and the digitalization of a lot of conventional learning resources, it is difficult for learners to discover the most appropriate resources using a keyword search method. On the other hand, several research works have addressed the need for personalization in web-based learning environments. Researchers utilize recommendation techniques to resolve information overload in the new learning environment. However, there are still several challenging problems.

The first important problem relates to sparsity and cold-start problems in the e-learning environment. The sparsity problem occurs when rating data are insufficient for identifying similar users (neighbors). In practice, recommender systems are used to evaluate very large data sets, and since each user only rates a small number of items, the number of ratings given by the users is very small in comparison with the total number of (user; item) pairs in the system. Cold-start refers to the situation in which an item cannot be recommended unless it has been rated by a substantial number of users. This problem is particularly detrimental to users. Likewise, a new user has to rate a sufficient number of items before the recommendation algorithm is able to provide reliable and accurate recommendations [3]. In the e-learning environment, since various learners have different knowledge and different preferences, the commonly used items (resources) between them are few, and therefore, the similarity value between users will be unreliable. This leads to sparsity and also the cold-start problem.

The second important problem refers to the overspecialized recommendation results that occur when recommended items are very similar to each other and the recommendation list is not diverse. The goal of recommendation diversification is to identify a list of items that are dissimilar, but nonetheless relevant to the user's interests. To the best of our knowledge, the diversity of the recommended

- *M. Salehi is with the Department of Industrial Engineering, K.N. Toosi University of Technology, Tehran, Iran. E-mail: m_salehi61@yahoo.com.*
- *I.N. Kamalabadi is with the Department of Industrial Engineering, Tarbiat Modares University, Tehran, Iran. E-mail: nakhai@modares.ac.ir.*
- *M.B.G. Ghoushchi is with the Electrical Department, Shahed University, Tehran, Iran. E-mail: ghaznavi@shahed.ac.ir.*

Manuscript received 16 July 2012; revised 31 Oct. 2012; accepted 18 July 2013; published online 20 Aug. 2013.
For information on obtaining reprints of this article, please send e-mail to: lt@computer.org, and reference IEEECS Log Number TLT-2012-07-0092.
Digital Object Identifier no. 10.1109/TLT.2013.28.

1939-1382/13/$31.00 © 2013 IEEE Published by the IEEE CS & ES

list is defined by how much each item in the list differs from the others in terms of their attribute values [4]. On the other hand, to find similarity between items and users in the usual attribute-based recommendation algorithms, an exact match between their attributes is necessary. This approach first leads to low accuracy when there is not adequate attribute information about users and items, and second, leads to the other well-known problem in recommender systems, overspecialization.

The third vital challenge relates to how to gather attributes' information and use it for modeling the multipreferences of a learner in the resource recommendation. This is challenging because the selection of all suitable attributes for a learner and resource is an almost impossible mission. In addition, it is nearly impossible to collect the corresponding data because some attributes cannot be described and coded formally.

Such challenges are potent motivators to find suitable recommendation techniques capable of discovering learning resources for the users.

1.2 Contribution

In response to the aforementioned problems, a new hybrid RS applicable for e-learning environments is proposed in this study. We consider two groups of attributes for learning resources including explicit attributes and implicit (latent) attributes. Explicit attributes, such as the subject and publisher for learning resources, are known and can be extracted by experts, but implicit attributes are latent and can be inferred by historical ratings of learners. Some research works combine the attributes (features) of users or items with historical ratings for recommendation. Burke [5] reviewed several hybrid recommender methods developed to combine the external (which we call explicit) features and historical rating data for higher predication accuracy. According to the experiment results reported, it is believed that both the features and the historical ratings have great value to estimate the predication function for recommendation. Therefore, the contribution of this study can be summarized as follows:

1. Establishment of a new recommendation approach based on the explicit and implicit attributes of learning resources.
2. Implication of a learner preference tree to model the multipreferences of learners based on the multidimensional explicit attributes of resources and the ratings of learners to simultaneously alleviate the sparsity and cold-start problems and also improve the diversity of the recommendation list.
3. Introduction of implicit attributes and optimization of the weight of these attributes by a genetic algorithm (GA) for each user and item to improve the accuracy of recommendation when the information about explicit attributes is low.

2 Literature Review

In recent years, most innovations in the area of educational systems have introduced new web-based technologies to train learners any time and any place. The creation of the technology for personalized lifelong learning has been recognized as a grand challenge by peak research bodies [6]. The goal of technology-enhanced learning (TEL) is designing, developing, and testing sociotechnical innovations that will support and enhance the learning practices of both individuals and organizations.

Similar to other fields where there is a massive increase in product variety, in TEL, there is also a need for better findability of (mainly digital) learning resources. For instance, in recent years, numerous repositories with digital learning resources have been set up [7], such as MERLOT (http://www.merlot.org), which has more than 40,000 learning resources and about 110,000 registered users. Considering this proliferation of online learning resources and the various opportunities for interacting with such resources in both formal and nonformal settings, it is necessary to create a technology to help user groups identify suitable learning resources from a potentially overwhelming variety of choices. As a consequence, the concept of recommender systems has already appeared in TEL.

Recommender systems address information overload and make a PLE for users. PLE's solutions should provide facilities for empowering learners in using this kind of technology. Using this approach, we can improve a personal learning path according to pedagogical issues and available resources. In the TEL domain, a number of recommender systems have been introduced to propose learning resources to users. Such systems could potentially play an important educational role, considering the variety of learning resources that are published online and the benefits of collaboration between tutors and learners [8].

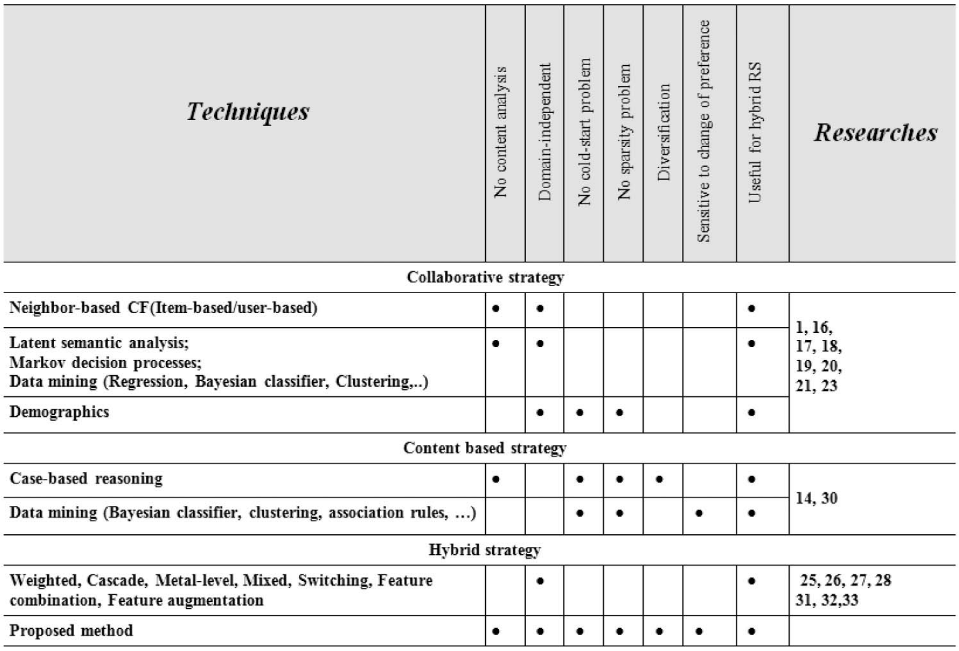
The recommender systems support a number of relevant user tasks within some particular application content. Generally speaking, most of recommendation goals and user tasks in other areas, such as e-commerce, are valid in the case of TEL recommender systems as well. However, recommendation in a TEL context has many particularities that are based on the richness of the pedagogical theories and models [9].

Most recommendation systems are designed either based on content-based filtering or collaborative filtering (CF). Content-based filtering techniques suggest items similar to the ones that each user liked in the past, taking into account the object content analysis that the user has evaluated in the past [10]. On the other hand, based on the assumption that users with similar past behaviors have similar interests, a collaborative filtering system recommends items that are liked by other users with similar interests. Both types of systems have inherent strengths and weaknesses. Table 1 presents an overview of the recommendation strategies in a TEL context. We briefly explain some of the important research works here.

2.1 Content-Based Filtering

This strategy uses the features of items for recommendation. As Table 1 indicates, these features may be used by case-based reasoning (CBR) or data mining techniques for recommendation. CBR assumes that if a user likes a certain item, she/he will probably also like similar items. This approach recommends new but similar items. However, data mining techniques recommend items based on the matching of their attributes to the user profile. CBR

TABLE 1
Overview of the Recommendation Strategies



Techniques	No content analysis	Domain-independent	No cold-start problem	No sparsity problem	Diversification	Sensitive to change of preference	Useful for hybrid RS	*Researches*
Collaborative strategy								
Neighbor-based CF(Item-based/user-based)	•	•					•	**1, 16, 17, 18, 19, 20, 21, 23**

**Latent semantic analysis;
Markov decision processes;
Data mining (Regression, Bayesian classifier, Clustering,..)**
	•	•					•	

Demographics		•	•	•			•	
Content based strategy								
Case-based reasoning	•		•	•	•		•	**14, 30**
Data mining (Bayesian classifier, clustering, association rules, …)			•	•		•	•	
Hybrid strategy								
Weighted, Cascade, Metal-level, Mixed, Switching, Feature combination, Feature augmentation		•					•	**25, 26, 27, 28 31, 32,33**
Proposed method	•	•	•	•	•	•	•	

mechanisms have to evaluate all the cases in the case base to retrieve those most similar case(s), which makes their efficiency strongly and negatively related to the size of the applicable case base [11]. The performances of CBR mechanisms are closely related to the case representation and indexing approach, so their superior performances are unstable and cannot be guaranteed. Semantic and multicriteria recommender systems also consider attributes of items. Semantic recommender systems, instead of using syntactic matching techniques, use inference techniques borrowed from the Semantic Web. This approach uses reasoning about the semantics of items and user preferences to discover complex associations between them [12]. Rating systems can model a user's utility for a given item with the user's ratings for each individual criterion [13]. Since more people will lurk in a virtual community than will participate, they usually do not spend time to rate based on each individual criterion in multicriteria recommenders. Khribi et al. [14] used learners' recent navigation histories, similarities, and dissimilarities among the contents of the learning resources for online automatic recommendations.

In fact, the existing metrics in content-based filtering only detect the similarity between items that share the same attributes. Indeed, the basic process performed by a content-based recommender consists of matching up the attributes of a user profile in which preferences and interests are stored with the attributes of a content object (item) to recommend to the user new interesting items [10]. This causes overspecialized recommendations that only include items very similar to those the user already knows. To avoid the overspecialization of content-based methods, researchers proposed new personalization strategies, such as collaborative filtering and hybrid approaches mixing both techniques.

2.2 Collaborative Filtering

Collaborative filtering is regarded as one of the important and useful strategies in recommender systems [15]. CF approaches used in e-learning environments focus on the correlations among users having similar interests and can be divided into three categories that are shown in Table 1. Neighbor-based CF finds similar items or users based on rating data and predicts ratings using the weighted average of similar users or items. Model-based techniques predict the ratings of a user by learning from complex patterns based on the training data (rating matrix). In the demographics approach, users with similar attributes are matched; then, this method recommends items that are preferred by similar users.

The collaborative e-learning field is strongly growing [16], [17], converting this area into an important receiver of applications and generating numerous research papers.

One of the first attempts to develop a collaborative filtering system for learning resources was the Altered Vista system [18]. The proposed system collects user-provided evaluations from learning resources and then propagates them into the form of word-of-mouth recommendations about the qualities of the resources. Lemire et al. [19] proposed a rule-applying collaborative filtering (RACOFI) composer system. RACOFI combines two recommendation approaches by integrating a collaborative filtering engine, which works with ratings that users provide for learning resources, with an inference rule engine that is mining association rules between the learning resources and using them for recommendation. The questions sharing and interactive assignments (QSIA) for learning resources sharing, assessing, and recommendation were developed by Rafaeli et al. [20].

Manouselis et al. [21] tried to use a typical neighborhood-based set of CF algorithms to support learning object recommendation. Their research considers multidimensional ratings that users provide for learning resources. According the results of this study, it seems that the performance of the same algorithms changes depending on the context where testing takes place.

Since, in an e-learning environment, learning resources are provided in a variety of multimedia formats, including text, hypertext, image, video, audio, and slides, it is difficult to calculate the content similarities of two items [1]. In this sense, since CF is completely independent of the intrinsic properties of the items being rated or recommended, we can use users' preference information as a good indication for recommendation in e-learning systems [22]. Regardless of its success in many application domains, collaborative filtering has two serious drawbacks.

First, its applicability and quality is limited by the so-called sparsity problem, which occurs when the available data are insufficient for identifying similar users. Therefore, many research works have been run to alleviate the sparsity problem using data mining techniques. For example, Romero et al. [23] developed a specific web mining tool for discovering suitable rules in a recommender engine. Their objective was to recommend to a student the most appropriate links/webpages to visit next.

Second, it requires knowing many user profiles to elaborate accurate recommendations for a given user. Therefore, in some e-learning environments, that number of learners is low; recommendation results have no adequate accuracy.

2.3 Hybrids

Combining several recommendation strategies can be expected to provide better results than either strategy alone [24]. Most hybrids work by combining several input data sources or several recommendation strategies. Table 1 lists some techniques that are used for hybrid recommendation.

Li et al. [25] discovered content-related item sets by CF, then applied the item sets to sequential pattern mining and generated sequential pattern recommendations for learners. Nadolski et al. [26] created a simulation environment for different combinations of recommendation algorithms in a hybrid recommender system to compare them against each other with regard to their impact on learners in informal learning networks. Tang and McCalla [27] proposed an evolving e-learning system, open to new learning resources that may be found online, which includes a hybrid recommendation service. A learning object recommendation model (LORM) was proposed by Tsai et al. [28] which also follows a hybrid recommendation algorithmic approach and describes resources using multiple attributes, but has not yet been reported to be implemented in an actual system.

An appropriate recommendation technique must be chosen according to pedagogical reasons. These pedagogical reasons are derived from specific demands of lifelong learning [29]. Therefore, some recommendation techniques are more suitable for specific demands of lifelong learning than others. One way to implement pedagogical decisions into a recommender system is to use a variety of recommendation techniques in a recommendation strategy. This paper uses two recommendation techniques based on explicit and implicit attributes of learners and resources.

3 Explicit and Implicit Attributes

The most important task of a recommendation system is modeling the user's preferences and computing the appropriateness degree of items for a target user. There are various preference elicitation methods employed in current decision support systems. Since the traditional methods are too time consuming and tedious, computer-aided decision support systems have appeared to simplify the task by making the assumption of additive preferential independence.

To improve the accuracy of preferences elicited as well as to save the decision maker's effort, another research branch on preference elicitation has aimed at releasing all assumptions on preference structure by matching a new user's preferences to other users' preference models. Collaborative filtering is also based on a similar idea, but the preferences matched are item ratings provided by different users [34]. Currently, the CF method uses a rating matrix that is the result of interaction between users and items. Each cell of this matrix can be illustrated as a triple data set $H = \{U_i, I_j, R_i^j\}$, which represents a user's historical preferences, where R_i^j is a user preference. These algorithms assume that user ratings of items are determined by the attributes of user and item corporately. In other words, if two items have similar attributes, users would be likely to rate them similarly. On the other hand, if two users have similar attributes, they are more likely to choose the same items. In this research, we use this reasonable assumption for the recommendation process.

Since ratings depend on the needs and attributes of learners and also the attributes of resources, the rating prediction function could be denoted as $\phi = \{M, U_i, I_j\}$, where M is a prediction model learned from the historical rating data H, and $U_i = (w_{i1}, w_{i2}, \dots, w_{ip})$ and $I_j = (e_{i1}, e_{i2}, \dots, e_{iq})$ are attribute weights of learner i and resource j, respectively. Based on this view, the objective of the recommender system problem is to find a fitting relationship between the space attributes of learners and resources to generate an appropriate recommendation. Unfortunately, in most cases, we cannot use the mentioned model because the selection of all suitable attributes for a learner and a resource is an almost impossible mission. Even if the attribute set is chosen, it is nearly impossible to collect the corresponding data because some data involves the privacy of people and some attributes cannot be described and coded formally. This leads to low prediction accuracy because it is only based on limited observed attributes [35].

However, we can use historical rating data in a user-item matrix for discovering some valuable implicit attributes, which are reflected characteristics of the learning resource and learner. The prediction models built based on observed attributes plus latent (implicit) attributes should have relatively higher prediction accuracy. In this research, matrix factorization (MF) is used for implicit attribute extraction. However, a GA is used to generate the initial solution of matrix factorization.

Let U_i^I and I_j^I denote the implicit attributes space for users and items, respectively. Let vector $U_i^I = (w_{i1}^I, w_{i2}^I, \dots, w_{iK}^I)$ and $I_j^I = (e_{i1}^I, e_{i2}^I, \dots, e_{iK}^I)$ represent the user i and item j implicit attributes' weights, respectively. The prediction function could be denoted as $\varphi' = (M, U_i, I_j, U_i^I, I_j^I)$ and

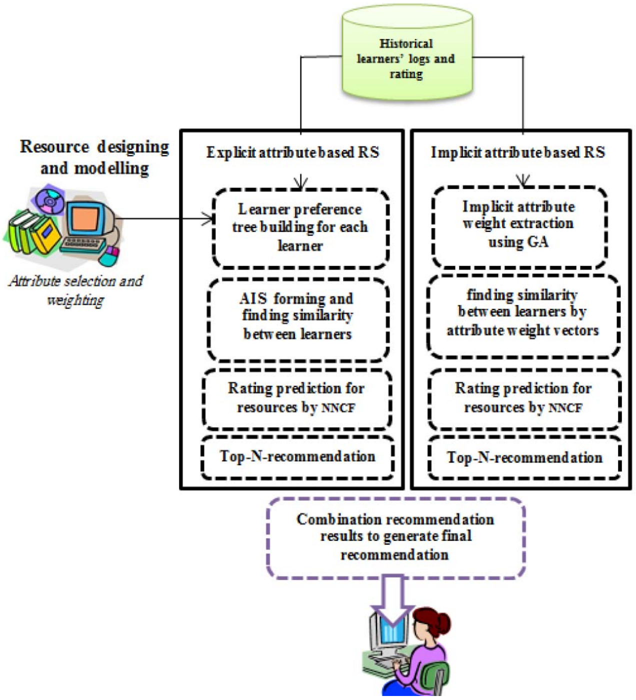

Historical learners' logs and rating
Resource designing and modelling
Attribute selection and weighting
Explicit attribute based RS
Learner preference tree building for each learner
AIS forming and finding similarity between learners
Rating prediction for resources by NNCF
Top-N-recommendation
Implicit attribute based RS
Implicit attribute weight extraction using GA
finding similarity between learners by attribute weight vectors
Rating prediction for resources by NNCF
Top-N-recommendation
Combination recommendation results to generate final recommendation

Fig. 1. System framework of the proposed resource RS.

the historical rating data could be converted to $H' = \{(U_i, I_j, U_i^I, I_j^I, R_i^j)\}$. This research uses nearest neighborhood as a prediction model and also uses a GA to discover implicit attributes.

4 Methodology

The proposed framework shown in Fig. 1 is composed of two main modules. In the explicit attribute-based (EAB) module, using server usage logs of learners and ratings, LPT is built for each learner. Then, according to the defined similarity between LPTs, ratings are predicted. In the implicit attribute-based (IAB) module, the weights of implicit attributes for each learner are calculated using a GA and MF. The proposed GA can calculate the interest of a learner for each attribute of the learning resources.

4.1 Explicit Attribute-Based Module

The idea of using LPT to model the explicit interest of a learner is useful because this approach can reflect learners' complete spectrum of interests and can take into account resources' multidimensional attributes and learners' rating information simultaneously. This approach traces learner preferences and can adapt recommendation results based on changes in learner behavior. This tracing is performed based on changes in the ratings of learners for multidimensional attributes of resources.

4.1.1 Interest Learner Modeling

We can consider some attributes for each learning resource, for example, a subject that has some value such as literature or mathematics. In addition, since a learner's ratings for accessed resources that have certain explicit attributes indicates the importance of these attributes to the learner, they can be considered as base for weighting explicit attributes for the learner. Therefore, the resource attributes' description model can be defined as a vector $C = <(A_1, AW_1), (A_1, AW_2), \ldots, (A_1, AW_m)>$, where A_t denotes the tth dimensional attribute's name of the resource, AW_t denotes the relevant weight value, $AW_1 \geq AW_2 \geq \cdots \geq AW_m$, and $\sum_{t=1}^{m} AW_t = 1$. Based on this description model, the attributes of a certain resource R_j can be defined as $R_j = (KA_1, KA_2, \ldots, AK_m)$, where KA_t denotes the tth dimension attribute's keyword of resource R_j. To further clarify the explicit-based module, we use a running example in this section. In this research, we consider

$$C = [(Primary\ subject, 0.4), (Secondary\ subject, 0.25), (Education\ level, 0.2), (Publisher, 0.15)].$$

After registering each resource in the system, the system developer must determine its attribute's keywords, such as

$$R_6 = (Mathematic,\ statstic,\ PhD,\ TU/e).$$

This research introduces a multidimensional attribute-based framework for recommendation that involves the explicit attributes of resources in the recommendation process; however, the selection of appropriate attributes may vary in the different systems. Relevant attributes of learning resources, including type of object, author, owner, terms of distribution, format, and pedagogical attributes, such as teaching or interaction style, could be selected. In this research, according to simplicity and usefulness, we selected four attributes including: primary subject, secondary subject, education level (Bachelor's Degree (B.D.), Master's Degree (M.D.), PhD Degree (PhD.D.)), and publisher of the resource. The most important attributes of a learning resource for the interest modeling of learners are the primary subject and secondary subject; therefore, in this research, we use a weighting approach that allocates a higher weight for the primary subject and secondary subject and a lower weight for other attributes.

LPT is introduced to combine the multidimensional attributes of accessed resources and the learner's rating information for making an information model of the learner's preferences. An LPT has a $(m+1)$-level (starting from 0), in which m denotes the number of attributes of the resource. In this tree, the leaf node that represents an accessed resource of the learner is defined as $LPT_{leaf} = \{\mathrm{RID}, \mathrm{RR}\}$, where RID denotes the accessed resource ID of the learner and RR denotes the rating the learner gave to the resource (ratings are between 1 and 5). The nonleaf node can be defined as $LPT_{nonleaf} = \{\mathrm{KA}, \mathrm{RR}, level\}$, where KA is the keyword of the *level*th attribute of the resource and *level* denotes the layer number of this node. A typical LPT that has four levels (corresponding with the four attributes of the resource) is shown in Fig. 2. In this tree, each accessed resource corresponds to a unique path from the root to the relevant leaf node, and the keywords of all nodes located in this path correspond to the relevant keywords of the resource's attributes. The corresponding route of R_6 has been highlighted in red.

As we can see in Fig. 2, the RR of nonleaf node k is defined as the mean of RR values of all leaf nodes that belong to ks subtree. For example, for the information technology node in the first level, we have $2 = (3+1)/2$. Using this definition, the tree can transfer preferences of the

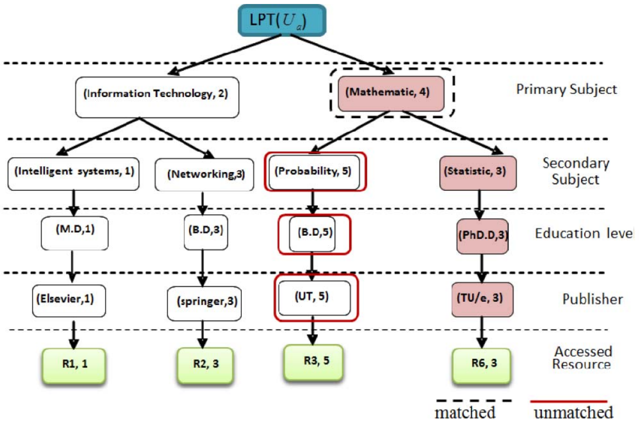


Fig. 2. Learner preference tree sample.

learner from accessed resources to the highest level of attributes and indicate the importance of each attribute for the learner.

The system updates LPTs using the following strategy:

1. Search the keywords of the latest accessed resource attributes in the LPT from top to bottom. If the keywords of the ith attribute cannot be matched, the $m-i+1$ nodes with latter $m-i+1$ attribute keywords of the resource will be created.
2. Add a leaf node for the latest accessed resource in level $m+1$ at the proper position.
3. Calculate RR of the new node's all predecessors.

If learner U_a visits $R_3 = (Mathematic, probability, B.D, UT)$, to update the tree in Fig. 2, search the "Mathematic" keyword at the first level. Since there is this keyword at the first level of the tree, it is matched. Then, search the "probability" keyword at the second level. As you can see, we do not have a match event at this level; therefore, we add the remaining nodes with their attributes' keywords to the next levels. R_3 is also added to level $m+1$, and then, the RR of the mathematic keyword is updated (changed from 3 to 4).

4.1.2 Rating Prediction

As a logical assumption, two learners with similar attribute keywords in their LPTs can be considered similar neighbors. Based on this assumption, we can solve the sparsity problem. To define the similarity degree, two rules are implemented:

1. The more similar the attributes of learner U_a and learner U_b's accessed resources, the larger the similarity between them.
2. The more similar the rating data of learner U_a and learner U_b, the larger the similarity between them.

Therefore, the similarity degree between two learners can be calculated based on the attributes' intersection subtree (AIS) between two relevant LPTs. The AIS between learner U_a and U_b, $AIS(U_a, U_b)$, is defined as the maximum connected intersection between LPT_a and LPT_b with the same node's keyword. We generated the AIS between U_a and U_b, shown in Fig. 3. The AIS is generated as follows: The traversing process is started on the most left nodes of LPT_a and LPT_b. If the attributes' keywords of two trees are the same, then the two trees are matched at this level. At this time, a new node will be added into the proper position in the AIS. Then, the traversing process is continued from the most left node of the immediate successors. If the attributes' keywords of two trees are different, the process is continued from the right node, and if we reach the bottom of the tree, then we backtrack to the last matching node between the two LPTs and continue the process from the right node. This process is continued until all nodes are traversed.

Fig. 3 shows how we can obtain $AIS(U_a, U_b)$. To predict ratings for learners, we calculate the EAB similarity between two learners that reflects the similarity between them based on explicit attributes. Therefore, inspired from cosines similarity [38], this similarity can be defined by (1).

$$sim_{EAB}(U_a, U_b) = \left(\sum_{i \in AIS(U_a,U_b)} AW_i.RR_{i.level}(U_a).RR_{i.level}(U_b) \right) \Big/ \left(\sqrt{\sum_{i \in LPT(U_a)} AW_i.RR_{i.level}(U_a)^2} \cdot \sqrt{\sum_{i \in LPT(U_b)} AW_i.RR_{i.level}(U_b)^2} \right), \tag{1}$$

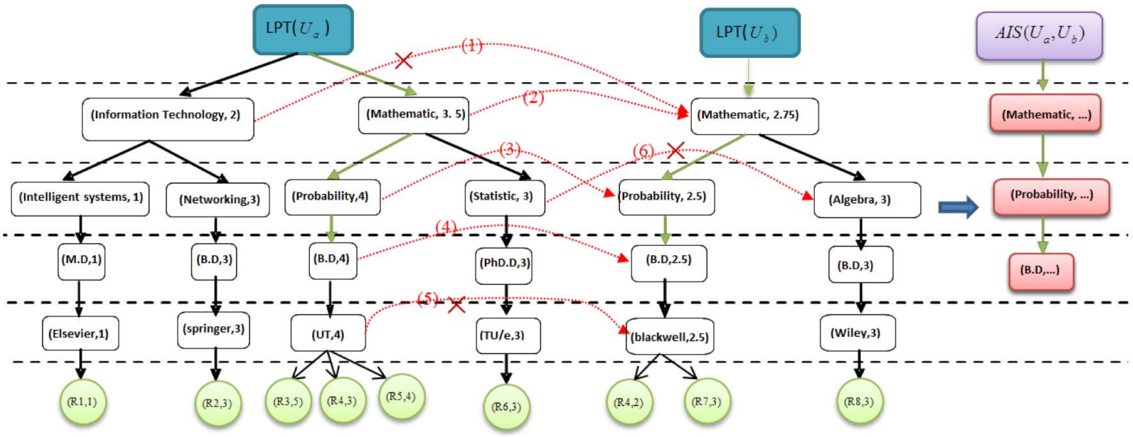


Fig. 3. Attributes intersection subtree sample.

$$
\begin{aligned}
&sim_{EAB}(U_a, U_b)\\
&\quad = h(0.4 \times 3.5 \times 2.75 + 0.25 \times 4 \times 2.5 + 0.2 \times 4 \times 2.5)/\\
&\qquad \sqrt{0.4 \times 3.5^2 + 0.25 \times 4^2 + 0.2 \times 4^2}\\
&\qquad \sqrt{0.4 \times 2.75^2 + 0.25 \times 2.5^2 + 0.2 \times 2.5^2} \approx 0.411,
\end{aligned}
$$

where $RR_{i.level}(U_a)$ indicates the rating of user U_a at level i of LPT_a, corresponding with level i in the AIS. We used this similarity for our example between U_a and U_b. The result has been shown above.

The predicated rating of learning resource i by U_a using the explicit attribute-based method, $P^i_{EBA.U_a}$, is obtained by the rating of U_a neighborhood, $N_{EBA}(U_a)$, which rated i before. The computation formula is as follows:

$$
P^i_{EBA.U_a} = \overline{R}_{U_a} + \frac{\sum_{j \in N_{EBA}(U_a)} sim_{EBA}(U_a, U_j) \times \left(R^i_{U_j} - \overline{R}_{U_j}\right)}{\sum_{j \in N_{EBA}(U_a)} sim_{EBA}(U_a, U_j)}, \tag{2}
$$

where $\overline{R}_{U_a}$ and $\overline{R}_{U_j}$ denote the rating average of learning resources rated by active learners U_a and U_j, respectively, and $sim_{EAB}(U_a, U_b)$ is the similarity between active learners U_a and U_j , which is a member of $N_{EAB}(U_a)$. 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 the active learner. Thus, to enhance the efficiency of the calculation, the neighborhood set should be preliminarily filtered via setting a similarity matching threshold τ. The two learners are effectively similar neighbors only if the similarity between them is at least τ. Therefore, the top k learners, which have the largest k similarity between them and U_a and meanwhile satisfy the effective requirement, are defined as U_a's neighbors. After rating prediction, we rank resources according to their ratings and recommend the top N resources to the active learner. In our example, if we consider U_b and U_c (we assume $sim_{EAB}(U_a, U_c) = 0.380$, $\overline{R}_{U_c} = 3.80, R^7_{U_c} = 3$) as neighbors of U_a, the predicted rating for R_7 will be

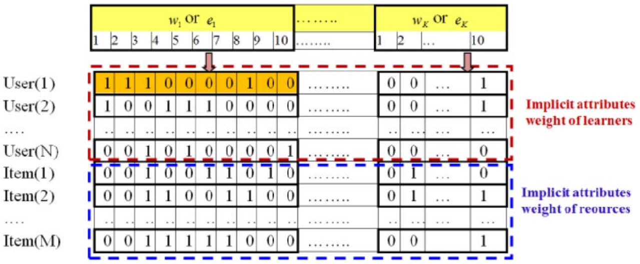
$$
\begin{aligned}
P^7_{EBA.U_a} &= 3.167 + \frac{0.411 \times (3 - 2.667) + 0.38 \times (3 - 3.8)}{0.411 + 0.38}\\
&= 2.92.
\end{aligned}
$$

4.2 Implicit Attribute-Based Module

The time complexity of executing the collaborative filtering algorithm grows linearly with the number of items and the number of users. Therefore, if the number of users and resources grows tremendously for learning environments, recommendation algorithms will suffer serious scalability problems, with computational resources going beyond practical or acceptable levels [38]. In general, GAs are believed to be effective on NP-complete global optimization problems, and they can provide good near-optimal solutions in reasonable time. Therefore, this research uses a GA for optimization of implicit attributes' weight.

4.2.1 Implicit Attribute Optimization

In the attribute space, different people may place different emphases on interrelated attributes. The goal of a GA is to find the relationship between the overall rating and the underlying attributes' rating for each learner. More specifically, given the rating data of a learner, a GA computes his/her preference model in terms of the implicit attributes' weight. Truly, we use a GA as a supervised learning task whose fitness function is the mean absolute error (MAE) of the RS.



	w_1 or e_1										....	w_K or e_K			
	1	2	3	4	5	6	7	8	9	10	....	1	2	...	10
User(1)	1	1	1	0	0	0	0	1	0	0	....	0	0	...	1
User(2)	1	0	0	1	1	1	0	0	0	0	....	0	0	...	1
....	..	..	..	..	..	..	..	..	..	..	....	..	...	...	...
User(N)	0	0	1	0	1	0	0	0	0	1	....	0	0	...	0
Item(1)	0	0	1	0	0	1	1	0	1	0	....	0	1	...	0
Item(2)	0	0	1	1	0	0	1	1	0	0	....	0	1	...	1
....	..	..	..	..	..	..	..	..	..	..	....	..	...	...	...
Item(M)	0	0	1	1	1	1	1	0	0	0	....	0	0		1

Implicit attributes weight of learners
Implicit attributes weight of reources

Fig. 4. Representation of implicit attributes' weight vectors in a chromosome.

Coding strategy. The chromosome scheme shown in Fig. 4 represents the implicit attributes' weights for users and items, where $U^I_i = (w^I_{i1}, w^I_{i2}, \ldots, w^I_{iK})$ and $I^I_j = (e^I_{j1}, e^I_{j2}, \ldots, e^I_{jK})$ indicate the implicit attributes' weight vector for user i and item j, respectively, and where K is the number of attributes and $\sum_{j=1}^{K} e^I_{ij} = 1$, $\sum_{j=1}^{K} w^I_{ij} = 1$. Each chromosome has $N + M$ rows corresponding with N users and M items, and also has $K \times 10$ columns corresponding with K implicit attributes. Each attribute weight is coded in the form of a binary string. Since the value of the weight is continuous and also between 0 to 1, to express them with 1/1,000th precision, 10 binary bits are used for each attribute weight because $512 = 2^9 < 1{,}000 < 2^{10} = 1{,}024$. Therefore, the length of the chromosome will be $\times 10$. These 10-bit binary numbers are transformed into decimal floating numbers, ranging from 0 to 1 by applying the following equation:

$$
x' = \frac{x}{2^{10} - 1}, \tag{3}
$$

where x is the decimal number of the binary code for each attribute's weight. For example, the binary code for the weight of the first feature of user 1 in Fig. 4 is $(1110000100)_2$. The decimal value of it is $(900)_{10}$ and it is interpreted as

$$
x' = \frac{900}{2^{10} - 1} = 0.8797653 \approx 0.880.
$$

Fitness function. Two matrixes, $W_U = (w^I_1, w^I_2, \ldots, w^I_N)$ and $W_I = (I^I_1, I^I_2, \ldots, I^I_M)$ that indicate the implicit attributes' weight vectors for N users and M items, respectively, become the optimizing targets. Their initial solution could be some random values. The fitness function will be the MAE of the RS (indeed we only use the training rating since we are trying to find two optimal latent attributes' weight matrixes) for particular matrixes, W_U and W_I. The MAE is obtained by comparing the real ratings with the predicted ratings made based on two matrixes. To calculate the MAE of the RS for particular matrixes, W_U and W_I, which have been generated in one of the generations of GA, we use the following equation:

$$
fitness = \sum_{i=1}^{N} \sum_{j=1}^{M_i} \left| \sum_{k=1}^{K} w_{ik}.e_{jk} - R^j_i \right|, \tag{4}
$$

TABLE 2
The Optimized Implicit Attributes' Weight for U_a and U_b

k	1	2	3	4	5	6
U_a	0.380	0	0	0.230	0.130	0.26
U_b	0.420	0.173	0.090	0.2	0.117	0

where R_i^j is real rating of item j by user i, w_{ik} and e_{jk} are the weight of attribute k for user i and item j, respectively, and M_i is the number of rated items by user i. It must be noted that, since we only use the training rating in the learning stage of the model, some rated items by each user are selected as the training set and the remaining rated items are considered the test set. When the fitness is lower, the rating prediction accuracy will be higher.

Selection. A probabilistic selection is performed based on the individual's fitness such that the better individuals have an increased chance of being selected. Therefore, the selection probability for each string is calculated by

$$p_c = 1 - \frac{fitness_c}{\sum_{C=1}^{PS} fitness_C}, \tag{5}$$

where $fitness_c$ denotes the value of the fitness function for chromosome c, PS is number of individuals in the population or population size, and p_c denotes the selection probability for chromosome c. Since the sum of the fitness in a population is constant, an individual with lower fitness (higher prediction accuracy) has a larger probability to be chosen. We find that the universal sampling method scheme yields a good individual to be selected for reproduction of the next population.

Crossover and mutation. This study performs crossover using a uniform crossover routine. The uniform crossover routine uses a fixed mixing ratio between two parents. Single-point and two-point crossover methods may bias the search with the irrelevant position of the attributes. However, the uniform crossover method is considered better at preserving the schema and can generate any schema from the two parents. A single-point mutation technique is used to introduce diversity.

4.2.2 Rating Prediction

After the implicit attributes' weight optimization, the similarity between learners using the IAB method can be calculated by the following formula, which is a cosine similarity [36]:

$$sim_{IAB}(U_a, U_b) = \frac{\sum_{i=1}^{K} w_{ai}.w_{ai}}{\sqrt{\sum_{i=1}^{K} w_{ai}^2}.\sqrt{\sum_{i=1}^{K} w_{bi}^2}}. \tag{6}$$

To clarify our approach and run our example, we assume that Table 2 indicates the optimized implicit attributes' weight by GA for U_a and U_b. Therefore, the implicit attribute-based similarity between U_a and U_b is as follows:

$$sim_{IAB}(U_a, U_b) = \frac{0.38 \times 0.42 + 0.23 \times 0.2 + 0.13 \times 0.117}{\sqrt{0.282} \times \sqrt{0.268}},$$

$$sim_{IAB}(U_a, U_b) = 0.803.$$

The predicted rating of learning resource i by U_a using the implicit attribute-based method is $P_{IAB.U_a}^i$, which is obtained by the rating of U_a neighborhood $N_{IBA}(U_a)$, which has rated i before. The computation formula is as follows:

$$P_{IBA.U_a}^i = \overline{R}_{U_a} + \frac{\sum_{j \in N_{IBA}(U_a)} sim_{IBA}(U_a, U_j) \times (R_{U_j}^i - \overline{R}_{U_j})}{\sum_{j \in N_{IBA}(U_a)} sim_{EBA}(U_a, U_j)}, \tag{7}$$

where $\bar{R}_{U_a}$ and $\bar{R}_{U_j}$ denote the average rating of learning resources rated by active learners U_a and U_j, respectively, and $sim_{IBA}(U_a, U_j)$ is the similarity between active learners U_a and U_j, which are members of $N_{IBA}(U_a)$. In this module, neighbors are determined similarly to the explicit-based attribute module. After rating prediction, we rank resources according to their ratings and recommend the top N resources to active learners.

In our example, similar to the explicit attribute-based method, if we consider U_b and U_c (we assume $sim_{IAB}(U_a, U_c) = 0.530$) as neighbors of U_a, the predicted rating for R_7 will be

$$P_{IBA.U_a}^7 = 3.167 + \frac{0.803 \times (3 - 2.667) + 0.40 \times (3 - 3.8)}{0.803 + 0.40} = 3.19.$$

4.3 Recommendation

We proposed two methods for learning resource recommendation: explicit attribute-based collaborative filtering (EAB-CF) and implicit attribute-based collaborative filtering (IAB-CF). To improve the quality of recommendations, we create a hybrid of two methods by the weighted combination method. 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:

$$P_{U_a}^i = \alpha.P_{IBA.U_a}^i + (1 - \alpha).P_{EBA.U_a}^i, \tag{8}$$

where $P_{U_a}^i$ denotes final predicted rate for learning resource i by U_a. For our example, we have (we assume $\alpha = 0.7$):

$$P_{U_a}^7 = 0.7.P_{IBA.U_a}^7 + 0.3.P_{EBA.U_a}^7 \approx 3.11.$$

Finally, the top N learning resources with the higher predicted rate are considered as the recommendation results.

5 Experiments

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

5.1 Simulation Environment and Data Set

Finding appropriate data sets for experimentation can be a challenging task in TEL, as there are various sources of data that have not been identified and documented exhaustively. This research uses a real-world data set in its experiment. The MACE[1] data set is a Pan-European initiative to interconnect and disseminate digital information about architecture used for experiments. This data set was issued the from MACE project that was done from September 2006 to September 2010. MACE enables architecture students to search through

[1] Metadata for Architectural Contents in Europe.

and find learning resources that are appropriate for their context. This data set contain 1,148 learners and 12,000 resources. These objects hold together about 47,000 tags, 12,000 classifications terms, and 19,000 competency values. Tags were assigned by logged-in users and the classification and competency terms were assigned by domain experts.

Most user actions within the MACE portal were logged, including search activities using faceted search, social tags, geographical locations, classifications and/or competencies, accessing learning resources, downloading resources, social tagging (including the options to add a tag, add a comment, and add a rating), and accessing user pages. The time of each user activity was recorded. In addition to explicit rating feedback, access time, downloads, tags, and comments provide useful implicit indications that can be used to gain knowledge about user interests.

Data sets that can be used by the new proposed algorithms must include some features about learners, their action types, learning resources, and also some features about the context of learning and results [37]. Learners must log-in, search appropriate learning resources, and also rate them. Our approach does not use special information about learners, but it uses available attributes from learning materials. The presented framework in this research can use different attributes for learning resources based on simplicity and usefulness. Therefore, according to our data set, the relevant attributes of learning resources (including the type of object, author, owner, terms of distribution, format, and pedagogical attributes, such as teaching or interaction style), can be used. It must be noted that some data sets do not have explicit relevance indicators in the form of ratings that are relevant for research on recommendation algorithms for learning.

5.2 Evaluation Metrics

In this paper, the evaluation metrics of recommendation algorithms are divided into three categories.

Decision support accuracy metrics assume the prediction process as a binary operation; either items are predicted (good) or not (bad). Precision and recall are the most popular metrics in this category. When referring to recommender systems, recall and precision can be defined as follows:

$$\text{Recall} = \frac{tp}{tp + fp}, \tag{9}$$

$$\text{Pr}ecision = \frac{tp}{tp + fn}, \tag{10}$$

where tp stands for true positive, fp stands for false positive, and fn stands for false negative. The threshold for determining a true positive is set to 3.5, meaning that if an item is rated 3.5 or higher, it is considered to be accepted by the user.

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 the F_1 measure [38], which is a well-known combination metric, with the following formula:

$$F_1 = 2.\frac{\text{Pr}ecision \cdot \text{Recall}}{\text{Pr}ecision + \text{Recall}}. \tag{11}$$

The other metrics that we use are predictive accuracy metrics. These statistical metrics measure how close the recommender system's predicted ratings are to the true user ratings. In this category, MAE is used with following formula:

$$MAE = \frac{\sum_{i=1}^{i=S} |R^i - P^i|}{|S|}, \tag{12}$$

where P^i is the predicted rating for resource i, R^i is the learner given rating for resource i, and S is the total number of the pair ratings P^i and R^i.

Since most recommendation algorithms have been developed based on accuracy measures, the recommendation lists produced by them contain similar items. Going to Amazon.com for a book by Isaac Asimov, for example, will give you a recommendation list full of his other books. In this case, the Item-Item collaborative filtering algorithm can trap users in a "similarity hole," only giving exceptionally similar recommendations. Since accuracy metrics are designed to judge the accuracy of individual item predictions, they cannot see this problem.

The recommendation list should be judged for its usefulness as a complete entity, not just as a collection of individual items. Therefore, in this research, we introduce a new measure to compute the similarity between recommended items in the recommendation list. The lower the similarity between the items in the recommendation list, the more diversity there is between them. An intra-list similarity metric (ISM) [39] is defined as follows:

$$ISM(List) = \frac{\sum_{R_i \in list} \sum_{R_j \in list, i \neq j} f(R_i, R_j)}{\binom{\#List}{2}}, \tag{13}$$

where

$$f(R_i, R_j) = \frac{mat(R_i, R_j)}{m}, \tag{14}$$

where the mat function indicates the number of matched attributes (similar attributes) between resource R_i and R_j, and, as it was said before, m is number of explicit attributes for resources. Higher similarity denotes lower diversity. This measure is used to evaluate the quality of recommendations.

5.3 Performance Evaluation

In this section, the proposed recommendation approaches are compared with a content-based recommendation algorithm [36], collaborative-based recommendation algorithm [40], and hybrid recommendation algorithm [41]. A content-based recommendation algorithm extends the state-of-the-art in recommender systems by using multiple TF-IDF vectors to keep track of user interests in different domains. This approach uses the feature of resources in the recommendation process. A collaborative-based recommendation algorithm does not use multidimensional attributes of resources. This algorithm is based on the memory-based CF to extend the state-of-the-art in two ways. First, it uses the enhanced Pearson correlation coefficient (PCC) algorithm to add one parameter which overcomes the potential

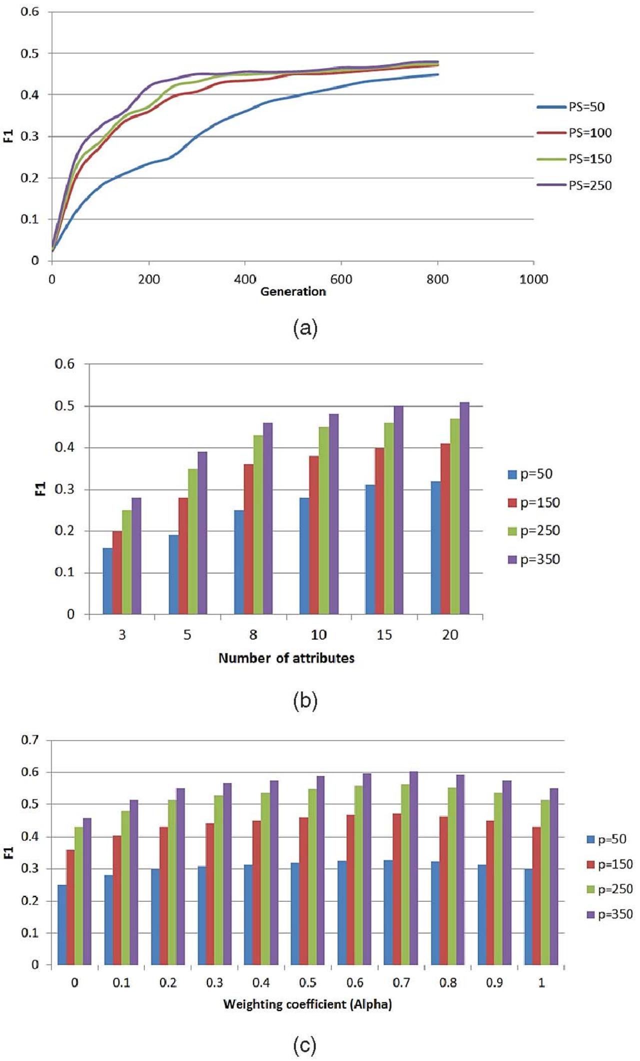
TABLE 3
Attributes' Intersection Subtree Sample

parameters	Value range
l	10, 15, 20, 25, 30, 35
p	50, 100, 150, 200, 250, 300, 350
$\#N_{EAB}$	5, 10, 15, 20, 25, 30
$\#N_{IAB}$	5, 10, 15, 20, 25, 30

decrease of accuracy when computing the similarity of users or items. Second, it uses an effective missing data prediction algorithm, in which information about both users and items is taken into account. While both content-based and collaborative filtering methods have their own advantages, individually, they fail to provide good recommendations in many situations. The hybrid recommendation algorithm used in this research applies an effective framework for combining content and collaboration. It uses a content-based predictor to enhance existing user data and then provides personalized suggestions through collaborative filtering.

To increase the number of records in a test set as much as possible, so as to eliminate the effect of an accidental factor, the top 60 percent of the access records for each learner in an ordered data set are used as the training set and the remaining 40 percent of the access records are used as the test set.

In relevant input parameters, l denotes the number of recommendation resources and p denotes the number of participating users that are selected from the data set to build the simulation data set. We select only users that have rated at least 30 items in the data set. $\#N_{EAB}$ and $\#N_{LAB}$ denote the number of neighborhoods in the explicit and implicit attribute-based recommendation module, respectively. The value ranges of $\#N_{EAB}$, $\#N_{LAB}$, l, and p are set in Table 3.



(a)

(b)

(c)

Fig. 5. (a) Performance of IAB-CF with respect to population size. (b) Performance of IAB-CF with respect to K. (c) Effect of α on F1 of EB-IB-CF.

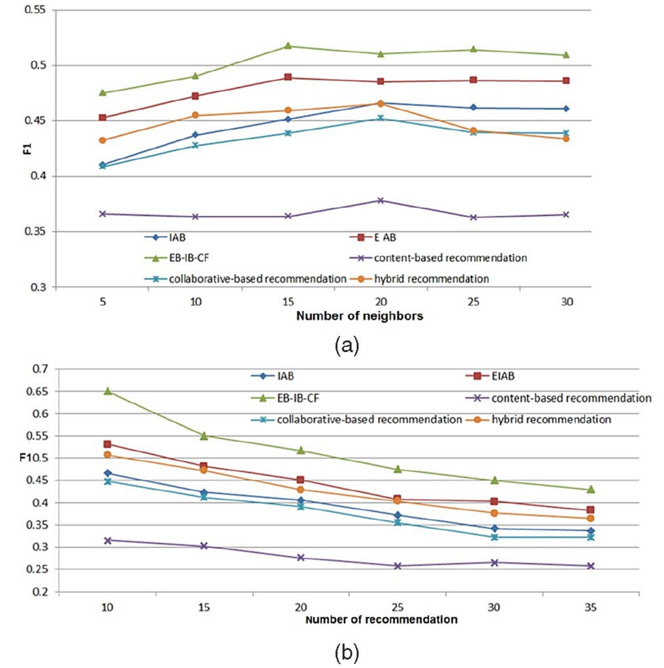
5.3.1 Parameter Setting

First, we analyze how some parameters affect the recommendation performance of the proposed algorithms. The goal of the following experiments is to determine the values of these parameters (as different data sets may correspond to different optimal values of these parameters). For the GA search, we examine the impacts of various combinations of crossover rate and mutation rate on the precision of the implicit attribute-based approach. According to the experiments, a crossover rate = 0.8 and a crossover rate = 0.15 give favorable results for our problem.

To compare the effect of changing the initial size of the population on the GA efficiency and results while $l = 20$, $p = 200$, $N_{IAB} = 15$, and $K = 8$, an experiment was set up. Generated numbers were chosen from 0 (initial generated data without running the algorithm) to 800 generations with a step size of 50 generations. The algorithm has been run 50 times for each population size and each generation value. Fig. 5a shows the results. The figure only compares the average of the best found solutions (in population). The results show that a higher population size provides a higher diversity and, as a result, converges to better solutions sooner than smaller population sizes. On the other hand, a higher population size needs more time for the algorithm to run. In this experiment, a population size of 20 lost diversity before reaching an acceptable solution. However, a population size of 200 does not provide much benefit over the population size of 100. We will therefore use a population size of 100 in four experiments. The GA stops when there is an individual in the population with a fitness value lower than a constant γ. We use $\gamma = 1.5$. We have used this value according to the optimal values found in our experiments.

Fig. 5b shows the results obtained for the implicit attribute-based approach with a different number of implicit attributes while $l = 20$, $p = 200$, and $N_{IAB} = 15$. It can be seen that the performance improves steadily with the number of attributes increasing. To have good efficiency in the computation, we set $K = 8$.

Fig. 5c shows the impact of α on the F_1 of EB-IB-CF, while $l = 20$, $p = 200$, $\#N_{IAB} = \#N_{IAB} = 15$, and $K = 8$. It indicates that taking into consideration a combination of EAB-CF and IAB-CF to predict ratings will play a positive role in the recommendation process, but α does not



(a)

(b)

Fig. 6. (a) Comparison of different algorithms with respect to the number of neighbors. (b) Comparison of different algorithms with respect to the number of recommendations.

acknowledge the "larger the better" rule. The best precision can be obtained with $\alpha = 0.7$.

5.3.2 Performance Comparison

Fig. 6a compares the proposed recommendation algorithms with three traditional algorithms according to the number of similar neighbors while $l = 20$, $K = 8$, and $p = 250$. As the number of similar neighbors increases, the F_1 of each algorithm increases, except for the content-based algorithm. When this number is increased to a certain point, the precision of each algorithm begins to decrease. The reason is that, with an increasing number of neighbors to a certain point, several dissimilar users may be denoted as similar users by the collaborative-based algorithm; therefore, the corresponding recommendation accuracy will decrease. For the hybrid algorithm, which also considers the content-based mechanism, there is less performance degradation. However, our proposed methods will set a threshold for the similar users calculation process to guarantee their quality. Meanwhile, the resources' attributes are taken into account based on the traditional collaborative-based mechanism. Therefore, they can effectively find similar users more accurately with only a little performance degradation.

Fig. 6b compares algorithms with respect to the number of recommendation resources while $\#N_{IAB} = \#N_{IAB} = 15$, $K = 8$, and $p = 250$. As l increases, the F_1 of each algorithm decreases. Moreover, EB-IB-CF always produces better performance than any other algorithm, especially when l is small. This is because the proposed method makes good use of the advantages of the content-based and collaborative-based recommendation mechanism while integrating three kinds of information—multidimensional attributes of a resource, users' ratings, and implicit attributes—hence, the actual preferences and interests of users are reflected accurately.

TABLE 4
Result of the Comparison Statistical Tests between the EAB-CF and Hybrid Algorithms

l	EAB	Hybrid Rec.	Significant	P-value
10	1.729	1.754	Yes	< 0.05
15	1.359	1.375	No	> 0.05
20	0.986	1.096	No	> 0.05
25	0.904	0.912	No	> 0.05
30	0.888	0.893	No	> 0.05
35	0.871	0.874	No	> 0.05

Statistical performance analysis. To evaluate our proposed method, we implemented some statistical tests and compared the EAB-CF and EB-IB-CF algorithms with the best results from the traditional algorithms (the result of the hybrid algorithm) using the original educational data on MSE.

The results are shown in Tables 4 and 5, along with indications of statistical significance. The P-value indicates the strength of significance as measured by a large-sample-paired hypothesis test. With the exception of one data point (for $l = 10$) in Table 4, there is no statistical difference between EAB-CF and the hybrid algorithm. But as Table 5 indicates, EB-IB-CF performs significantly better than the hybrid algorithm.

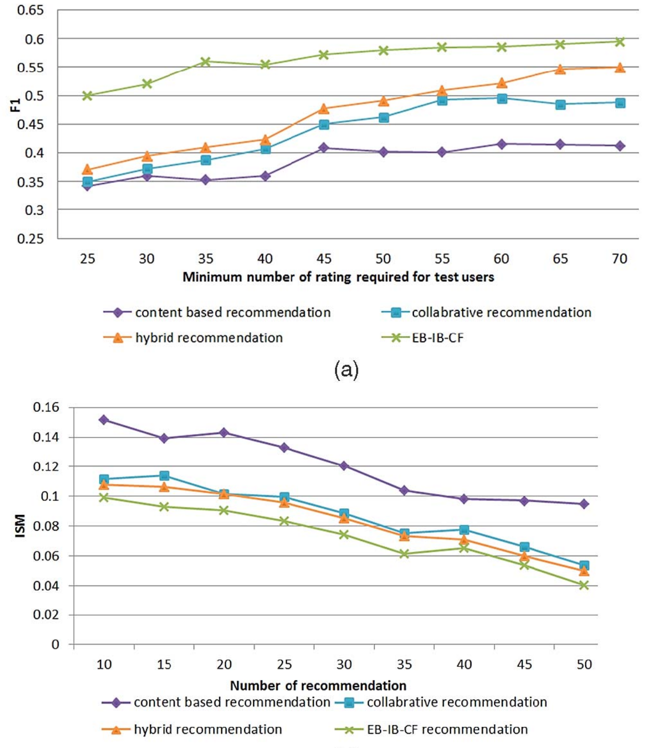
Performance evaluation under the sparsity problem. To evaluate our proposed approach for sparsity data, we change the minimum number of ratings required for test users from 25 to 70 and compare the results of EB-IB-CF with the traditional algorithms. As Fig. 7a shows, with increasing sparsity in the data or decreasing the value of the minimum numbers of ratings required for test users, the performance superiority of EB-IB-CF increases.

Performance evaluation under the diversity problem. As shown in Fig. 7b, the proposed algorithm has lower ISM than any other algorithms, which means higher diversity. By increasing the number of recommendations, diversity decreases for all algorithms. This result proves improvement in the quality of recommendations by the proposed method. Content-based filtering has the lowest diversity, and diversity in collaborative and hybrid recommendations is approximately equal.

This approach can calculate similarity based on partial matching between the attributes of visited resources by users, and it is not necessary to have an exact matching between their attributes. This property can diversify the

TABLE 5
Result of the Comparison Statistical Tests between the EA-IB-CF and Hybrid Algorithms

l	EB-IB-CF	Hybrid Rec.	Significant	P-value
10	1.589	1.754	Yes	< 0.01
15	1.253	1.375	Yes	0.001
20	1.000	1.096	Yes	< 0.001
25	0.836	0.912	Yes	< 0.001
30	0.837	0.893	Yes	< 0.001
35	0.829	0.874	Yes	< 0.001



(a)

(b)

Fig. 7. (a) Comparison of EB-IB-CF with traditional methods with respect to M. (b) The ISM of different algorithms with respect to N.

recommendation list because it allows establishing correspondences between the user preferences and other items appealing to him/her that do not necessarily share the same attributes.

Performance evaluation under the cold-start problem. To test the proposed method for the new item cold-start problem, we selected 500 random samples of item/user pairs from the test set. To make a prediction for a target item, we kept the number of users in the training set who have rated target items 1, 2, 5, and 10 and considered the corresponding MAE from MAE1, MAE2, MAE5, and MAE10. According to Table 6, the proposed method works well in the new item cold-start problem, as it does not solely depend on the number of users who have rated the target item for finding similarity.

Like the new user cold-start problem, we selected 500 random samples of item/user pairs from the test set. To make a prediction for a target user, we considered the number of rated items by the target user in training sets 5, 10, and 20. Then, we calculated the corresponding MAE using different algorithms. According to Table 7, the proposed method also works well in the new user cold-start problem.

TABLE 6
Performance Evaluation under the
New Item Cold-Start Problem

Algorithm	*MAE1*	*MAE2*	*MAE5*	*MAE10*
EB-IB-CF	2.36	2.17	2.11	2.05
Collaborative	2.83	2.56	2.34	2.16
Content based	2.59	2.47	2.32	2.18
Hybrid	2.53	2.35	2.19	2.09

TABLE 7
Performance Evaluation under the
New User Cold-Start Problem

Algorithm	*MAE5*	*MAE10*	*MAE20*
EB-IB-CF	2.33	2.26	2.17
Collaborative	2.52	2.42	2.30
Content based	2.49	2.39	2.34
Hybrid	2.47	2.33	2.19

The proposed algorithms were introduced to address the drawbacks of recommender systems in TEL. However, since some drawbacks in RecSys are common, the proposed new algorithms can be adapted and used in another area. However, there is a tradeoff between the complexity of algorithms and the required accuracy in the application area. Since getting more accuracy in the e-learning area is more important than the e-commerce area, these new algorithms that are based on the attributes of learning resources and learners are more suitable for the learning environment.

6 Conclusion

One of the most important applications of recommendation systems in an e-learning environment is personalization and recommendation of learning resources. To address the sparsity and cold-start problems and have a more diverse recommendation list for each learner, this paper presents a novel personalized recommendation framework that utilizes explicit and implicit attributes of resources in the unified model. The explicit attribute-based recommender which uses LPTs for modeling the multipreferences of learners can alleviate the sparsity and cold-start problems and also generate a more diverse recommendation list than traditional recommender systems. In addition, the implicit attribute-based recommender which uses GAs for the weight optimization of implicit attributes can increase the accuracy of recommendations. In addition, our approach considers the knowledge concept in the recommendation process implicitly. We can infer the knowledge level from the preferences because the learner preference tree is formed based on ratings and the behavior of the learner. Since a learner gives higher ratings to resources that are in her/his knowledge scope (primary subject and secondary subject) and also in her/his knowledge level (education type), the system traces this rating and automatically recommends appropriate resources based on user knowledge. If a learner gives a higher rating to a special resource in a certain scope and level of knowledge, it means this resource can improve her/his knowledge better than other resources. For example, if a leaner has knowledge in the applied mathematical area and wants to improve her/his knowledge and read more resources in the route of improving knowledge, she/he will give a higher rating to applied mathematical resources rather than pure mathematical resources; therefore, the system traces this behavior and automatically recommends appropriate resources.

However, there are some limitations that can determine some possible directions for further research work:

First, many systems need to react immediately to online requirements and make recommendations for all users regardless of ratings history on visited resources, which demands a high scalability of a CF system.

Second, there are some access rules for learners that this research does not use. For example, the learning processes (resource access processes) usually have some time-dependence relationship, including repeatability and periodicity. Therefore, the time-dependence relationship between learning resources in a learning process can reflect a learner's resource access latent pattern and preference.

Therefore, for further research, we can implement some techniques to increase the scalability of systems. For example, clustering algorithms are good choices that can cluster users based on their behaviors and address the scalability problem by seeking users for recommendation within smaller and highly similar clusters instead of the entire database.

In addition, we can mine learner's historical access records for discovering resource access sequential patterns. Then, using these sequential patterns, we can predict the most probable resource that a learner will access in the near future to further improve the quality of recommendations and solve the new user problem. In addition, the use of real-coded GAs can be investigated for further research work.

References

[1] W. Chen, Z. Niu, X. Zhao, and Y. Li, "A Hybrid Recommendation Algorithm Adapted in E-Learning Environments," *World Wide Web*, Sept. 2012, doi:10.1007/s11280-012-0187-z.

[2] B. Mobasher, "Data Mining for Personalization," *The Adaptive Web: Methods and Strategies of Web Personalization*, P. Brusilovsky, A. Kobsa, and W. Nejdl, eds., pp. 1-46, Springer, 2007.

[3] A.I. Schein, A. Popescul, and L.H. Ungar, "Methods and Metrics for Cold-Start Recommendations," *Proc. 25th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval*, pp. 253-260, 2002.

[4] S. McNee, J. Riedl, and J.A. Konstan, "Being Accurate Is Not Enough: How Accuracy Metrics Have Hurt Recommender Systems," *Proc. ACM SIGCHI Extended Abstracts on Human Factors in Computing Systems (CHI EA '06)*, pp. 1097-1101, 2006.

[5] R. Burke, "Hybrid: Recommender Systems: Survey and Experiments," *J. User Modeling and User-Adapted Interaction*, vol. 12, no. 4, pp. 331-370, 2002.

[6] J. kay, "Lifelong Learner Modeling for Lifelong Personalized Pervasive Learning," *IEEE Trans. Learning Technology*, vol. 1, no. 4, pp. 215-228, Oct. 2008.

[7] A. Tzikopoulos, N. Manouselis, and R. Vuorikari, "An Overview of Learning Object Repositories," *Learning Objects for Instruction: Design and Evaluation*, P. Northrup, ed., pp. 29-55, Idea Group, 2007.

[8] V. Kumar, J. Nesbit, and K. Han, "Rating Learning Object Quality with Distributed Bayesian Belief Networks: The Why and the How," *Proc. Fifth IEEE Int'l Conf. Advanced Learning Technologies (ICALT '05)*, pp. 685-687, 2005.

[9] N. Manouselis, H. Drachsler, R. Vuorikari, H. Hummel, and R. Koper, "Recommender Systems in Technology Enhanced Learning," *Recommender Systems Handbook*, P.B. Kantor, F. Ricci, L. Rokach, and B. Shapira, eds., pp. 387-415, Springer, 2011.

[10] P. Lops, M. de Gemmis, and G. Semeraro, "Content-Based Recommender Systems: State of the Art and Trends," *Recommender Systems Handbook*, pp. 73-105, Springer, 2011.

[11] P.-C. Chang and C.-Y. Lai, "A Hybrid System Knowledge-Based Systems, Combining Self-Organizing Maps with Case-Based Reasoning in Wholesaler's New-Release Book Forecasting," *Expert Systems with Applications*, vol. 29, no. 1, pp. 183-192, 2005.

[12] Y. Blanco-Fernandez et al., "A Flexible Semantic Inference Methodology to Reason About User Preferences in Knowledge-Based Recommender Systems," *Knowledge-Based Systems*, vol. 21, no. 4, pp. 305-320, 2008.

[13] G. Adomavicius, N. Manouselis, and Y. Kwon, "Multi-Criteria Recommender Systems," *Recommender Systems Handbook*, pp. 769-803, Springer, 2011.

[14] M.K. Khribi, M. Jemni, and O. Nasraoui, "Automatic Recommendations for E-Learning Personalization Based on Web Usage Mining Techniques and Information Retrieval," *Educational Technology and Soc.*, vol. 12, no. 4, pp. 30-42, 2009.

[15] J. Bobadilla, F. Serradilla, and J. Bernal, "A New Collaborative Filtering Metric That Improves the Behavior of Recommender Systems," *Knowledge Based System*, vol. 23, no. 6, pp. 520-528, 2010.

[16] E. García, C. Romero, S. Ventura, and C. Castro, "An Architecture for Making Recommendations to Courseware Authors Using Association Rule Mining and Collaborative Filtering," *User Modeling and User-Adapted Interaction*, vol. 19, no. 1, pp. 99-132, 2009.

[17] E. García, C. Romero, S. Ventura, and C.D. Castroa, "A Collaborative Educational Association Rule Mining Tool," *Internet and Higher Education*, vol. 14, no. 2, pp. 77-88, 2011.

[18] A. Walker, M. Recker, K. Lawless, and D. Wiley, "Collaborative Information Filtering: A Review and an Educational Application," *Int'l J. Artificial Intelligence and Education*, vol. 14, pp. 1-26, 2004.

[19] D. Lemire, H. Boley, S. McGrath, and M. Ball, "Collaborative Filtering and Inference Rules for Context-Aware Learning Object Recommendation," *Int'l J. Interactive Technology and Smart Education*, vol. 2, no. 3, pp. 179-188, 2005.

[20] S. Rafaeli, Y. Dan-Gur, and M. Barak, "Social Recommender Systems: Recommendations in Support of E-Learning," *Int'l J. Distance Education Technologies*, vol. 3, no. 2, pp. 29-45, 2005.

[21] N. Manouselis, R. Vuorikari, and F. Van Assche, "Simulated Analysis of MAUT Collaborative Filtering for Learning Object Recommendation," *Proc. First Workshop Social Information Retrieval for Technology Enhanced Learning*, pp. 27-35, 2007.

[22] L. Yu, Q. Li, H. Xie, and Y. Cai, "Exploring Folksonomy and Cooking Procedures to Boost Cooking Recipe Recommendation," *Proc. 13th Asia-Pacific Web Conf. Web Technologies and Applications (APWeb '11)*, pp. 119-130, 2011.

[23] C. Romero, S. Ventura, A. Zafra, and P. de Bra, "Applying Web Usage Mining for Personalizing Hyperlinks in Web-Based Adaptive Educational Systems," *Computers and Education*, vol. 53, no. 3, pp. 828-840, 2009.

[24] M.S. Chen, J. Han, and P.S. Yu, "Data Mining: An Overview from a Database Perspective," *IEEE Trans. Knowledge and Data Eng.*, vol. 8, no. 6, pp. 866-883, Dec. 1996.

[25] Y. Li, Z. Niu, W. Chen, and W. Zhang, "Combining Collaborative Filtering and Sequential Pattern Mining for Recommendation in E-Learning Environment," *Proc. 10th Int'l Conf. Advances in Web-Based Learning*, pp. 305-313, 2011.

[26] R.J. Nadolski, B. Van den Berg, A.J. Berlanga, H. Drachsler, H.G. Hummel, R. Koper, and P.B. Sloep, "Simulating Light-Weight Personalised Recommender Systems in Learning Networks: A Case for Pedagogy-Oriented and Rating-Based Hybrid Recommendation Strategies," *J. Artificial Societies and Social Simulation*, vol. 12, no. 1, 2009.

[27] T.Y. Tang and G.I. McCalla, "Smart Recommendation for an Evolving E-Learning System: Architecture and Experiment," *Int'l J. E-Learning*, vol. 4, no. 1, pp. 105-129, 2005.

[28] K.H. Tsai, T.K. Chiu, M.C. Lee, and T.I. Wang, "A Learning Objects Recommendation Model Based on the Preference and Ontological Approaches," *Proc. IEEE Sixth Int'l Conf. Advanced Learning Technologies*, pp. 36-40, 2006.

[29] R. Gluga, J. Kay, and T. Lever, "Modeling Long Term Learning of Generic Skills," *Intelligent Tutoring Systems*, vol. 1, pp. 85-94, 2010.

[30] K.I. Ghauth and N.A. Abdullah, "Learning Materials Recommendation Using Good Learners' Ratings and Content-Based Filtering," *Educational Technology Research and Development*, vol. 58, no. 6, pp. 711-727, 2010.

[31] M. Salehi, I. Nakhai Kamalabadi, and M.B. Ghaznavi Ghoushchi, "Personalized Recommendation of Learning Material Using Sequential Pattern Mining and Attribute Based Collaborative Filtering," *Education and Information Technologies*, pp. 1-23, Dec. 2012, doi:10.1007/s10639-012-9245-5.

[32] M. Salehi and I. Nakhai Kamalabadi, "Hybrid Recommendation Approach for Learning Material Based on Sequential Pattern of the Accessed Material and the Learner's Preference Tree," *Knowledge-Based Systems*, vol. 48, pp. 57-69, 2013.

[33] M. Salehi, M. Pourzaferani, and S.A. Razavi, "Hybrid Attribute-Based Recommender System for Learning Material Using Genetic Algorithm and a Multidimensional Information Model," *Egyptian Informatics J.*, vol. 14, no. 1, pp. 67-78, 2013.

[34] L. Chen and P. Pu, "Survey of Preference Elicitation Methods," *Proc. First Workshop Social Information Retrieval for Technology-Enhanced Learning and Exchange,* http://ceur-ws.org/Vol-307/paper04.pdf, 2004.

[35] J. Zhong and X. Li, "Unified Collaborative Filtering Model Based on Combination of Latent Features," *Expert Systems with Applications,* vol. 37, no. 8, pp. 5666-5672, 2010.

[36] G. Adomavicius and A. Tuzhilin, "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," *IEEE Trans. Knowledge and Data Eng.*, vol. 17, no. 6, pp. 734-749, June 2005.

[37] K. Verbert, N. Manouselis, H. Drachsler, and E. Duval, "Dataset-Driven Research to Support Learning and Knowledge Analytics," *Educational Technology and Soc.*, vol. 15, no. 3, pp. 133-148, 2012.

[38] Y.Y. Shih and D.R. Liu, "Product Recommendation Approaches: Collaborative Filtering via Customer Lifetime Value and Customer Demands," *Expert Systems with Applications,* vol. 35, nos. 1/2, pp. 350-360, 2008.

[39] J.L. Herlocker, J.A. Konstan, L.G. Terveen, and J.T. Riedl, "Evaluating Collaborative Filtering Recommender Systems," *ACM Trans. Information Systems,* vol. 22, no. 1, pp. 5-53, 2004.

[40] H. Ma, I. King, and M.R. Lyu, "Effective Missing Data Prediction for Collaborative Filtering," *Proc. 30th Int'l ACM SIGIR Conf. Information Retrieval,* pp. 39-46, 2007.

[41] P. Baudisch, "Joining Collaborative and Content-Based Filtering," *Proc. Conf. Human Factors in Computing Systems,* 1999.

Mojtaba Salehi received the BSc degree from Shahid Bahonar University in 2004, the MSc degree from Tehran University in 2006, and the PhD degree from Tarbiat Modares University, Tehran, Iran, in 2013. During 2012, he was a researcher in the Department of Mathematics and Computer Science, TU/e, Netherlands. He is currently working as an assistant professor at the K.N. Toosi University of Technology. His research interests include soft computing, data mining, recommender systems, and applied multivariate analysis.

Isa Nakhai Kamalabadi received the BSc degree from Shahid Beshti University in 1979, the MSc degree from Tarbiat Modares University, Tehran, Iran, and the PhD degree in industrial engineering from the University of Toronto in 1995. He is currently an associate professor at Tarbiat Modares University. His research interests include optimization, operational research, information technology, scheduling and sequencing, and SCM.

Mohammad B. Ghaznavi Ghoushchi received the BSc degree from Shiraz University, Iran, in 1993 and the MSc and PhD degrees both from the Tarbiat Modares University (TMU), Tehran, Iran, in 1997 and 2003, respectively. During 2003-2004, he was a researcher at the Institute of Information Technology, TMU. He is currently an assistant professor with Shahed University, Tehran. His interests include VLSI design, computer-aided design automation for mixed-signal and UML-based designs for SOC, and mixed signal.