Data & Knowledge Engineering 87 (2013) 130–145

Contents lists available at ScienceDirect

Data & Knowledge Engineering

journal homepage: www.elsevier.com/locate/datak

Editorial

Application of implicit and explicit attribute based collaborative filtering and BIDE for learning resource recommendation

Mojtaba Salehi *

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

ARTICLE INFO

Article history:
Received 15 December 2011
Received in revised form 29 June 2013
Accepted 2 July 2013
Available online 13 July 2013

Keywords:
Personalized recommendation
Information filtering
Sequential pattern mining
Learning resource
BIDE
Data mining

ABSTRACT

Explosive growth of e-learning in the recent years has caused difficulty of locating appropriate learning resources to learners in these environments. Recommender system is a promising technology in e-learning environments to present personalized offers and deliver suitable learning resources for supporting activity of users. Compared with resource recommendation in e-commerce systems, users in e-learning systems have topic preferences in e-learning systems. However, e-learning systems have their own characteristics and current e-commerce algorithms cannot effectively use these characteristics to address needs of recommendations in these environments. To address requirement of e-learning resource recommendation, this research uses attribute of resources and learners and the sequential patterns of the learner's accessed resource in recommendation process. Learner Tree (LT) is introduced to take into account explicit multi-attribute of resources, time-variant multi-preference of learner and learners' rating matrix simultaneously. Implicit attributes are introduced and discovered using matrix factorization. BIDE algorithm also is used to discover sequential patterns of resource accessing for improving the recommendation quality. Finally, the results recommendation of implicit and explicit attribute based collaborative filtering and BIDE are combined. The experiments show that our proposed method outperforms the previous algorithms on precision and recall measures and the learner's real learning preference can be satisfied accurately according to the real-time up dated contextual information.

1. Introduction

In the recent years, with advances in wireless networking and mobile broadband Internet access technologies, and also the maturing of portable mobile devices, online e-learning has become a relatively widespread learning method. The popularity of e-learning has created huge amounts of educational resources (learning materials). Hence, locating the suitable learning resources has become a big challenge. One way to address this challenge is the use of recommender systems [1]. In addition, one of the new forms of personalization in e-learning environment that has been expressed as a need by several researches is to give recommendations to learners in order to support and help them through the e-learning process [2]. The task of delivering personalized learning resource is often framed in terms of a recommendation task in which a system recommends items to an active user [3].

Recommender systems use three main strategies to generate recommendations including content-based, collaborative, and hybrid recommendation [4]. Content-based recommendation uses the features of items and user and then builds a matching model for them. Recommendations are made based on comparison of user's preference and item's features. On the other hand, collaborative filtering assumes that users who had similar choices before will make the same selection in the future. CF approaches used in

* Tel.: +98 9151866393.
E-mail address: m_salehi61@yahoo.com.

0169-023X/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.datak.2013.07.001

e-learning environments focus on the correlations among users having similar interests. Combining several recommendation strategies can be expected to provide better results than either strategy alone [5]. Hybrid recommendation mechanisms attempt to deal with some of limitation and overcome drawbacks of pure content-based approach and pure collaborative approach by combining the two approaches.

An appropriate recommendation technique must be chosen according to pedagogical reasons. These pedagogical reasons are derived from specific demands of lifelong learning [6]. 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. The decision to change from one recommendation technique to another can be done according to pedagogical reasons, derived from specific demands of lifelong learning [7]. This paper uses two recommendation techniques based on attributes of learners and resources and sequential patterns of resource accessing.

The first technique introduced two types of attributes including implicit attributes and explicit attributes. Implicit attributes were discovered by matrix factorization. Learner Tree has been developed for integrating multi-dimensional explicit attributes of resources; learner's rating information and time-variant multi-preferences of learner. Our proposed framework can use this information simultaneously to model adaptive multi-preference of learners. The second technique integrates sequential patterns of resource accessing to improve the accuracy of recommendations based on learning process.

Using this hybrid approach, first, we take in account contextual information of learner and learning resources by implicit and explicit attribute CF. Second, implement a time-variant approach for producing recommendations. Third, improve accuracy of recommendation using implicit attribute when we do not have adequate information about explicit attributes of resources or coding and describing these attributes are impossible. Fourth, obtain more accurate recommendation based on the order of accessed resources in learning process.

The rest of this paper is organized as follows. In the Literature analysis section, the previous related works on e-learning resource recommender systems is discussed. The Methodology section introduces the overall system framework and describes the proposed algorithms step by step. The Implication section applies the proposed framework for a dataset to evaluate and analyze the performance and limitations of it. Finally, the Conclusion section provides the concluding remarks along with suggestions for future work.

2. Literature analysis

Recommender systems have been researched extensively by the Technology Enhanced Learning (TEL) community during the last decade [8]. TEL aims to design, develop and test sociotechnical innovations that will support and enhance learning practices of both formal and informal learning [9]. Recommender system is one of the technologies that have been used for e-learning environments to recommend useful resources to users. These systems address information overload and make a personal learning environment (PLE) for users. PLEs refer to a set of learning tools, services, and artifacts gathered from various contexts to be used by the learners. We briefly survey some of the important works and explain the drawbacks that can be addressed by our proposed approach.

2.1. Content based filtering

This strategy uses the features of items for recommendation [2]. These features may be used by case based reasoning (CBR) or data mining techniques for recommendation. CBR approach recommends new but similar items. However data mining techniques recommend items based on the matching of their attributes to the user profile. CBR 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. 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. As examples for e-learning application, Khribi et al. [2] used learners' recent navigation histories and similarities and dissimilarities among the contents of the learning resources for online automatic recommendations. Sharif et al. [10] provided a framework for discovering most relevant resources from CiteULike for learners. In their approach, the keywords of learning resource are matched with tags of CiteULike using Direct Match, Partial Match, and Synonym Match. Then, the resources are further ranked based on number of weights.

In the existing content based recommendation algorithms, due to considering learner's preference information alone and not considering similarity between learners, only certain resources which are similar to learner's historical preference could be recommended. 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

Majority of researchers used collaborative filtering based recommendation system [11–14]. CF approaches can be divided in to three categories. Neighbor-based CF finds similar items or users based on rating data and predicts rating using weighted average of similar users or items. Model-based techniques predict rating of a user by learning of 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. As an example in e-learning area, Bobadilla et al. [15] used a new equation for incorporating the learners score obtained from a test into the calculations in collaborative filtering for resources prediction. Their experiment showed that the method obtained high item-prediction accuracy.

Since in the e-learning environment learning resources are in a variety of multimedia formats including text, hypertext, image, video, audio and slides, it is difficult to calculate content similarity of the two items [16]. In this sense, we can use users' preference information that is a good indication for recommendation in e-learning systems. 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 researches were run to alleviate the sparsity problem using data mining techniques. Second, it requires knowing many user profiles in order to elaborate accurate recommendations for a given user. Therefore, in some e-learning environments that number of learners is low; recommendation results do not have adequate accuracy. Lobo and Sunita [17] used a classification algorithm for the data selected from Moodle database to classify the data, then they used Apriori Association Rule algorithm for recommender. Dwivedi and Bharadwaj [18] presented an approach that calculates the similarity between a pair of friends in a social network using their mutual ratings' history and their preferences. A learner asks a resource rating query to his instant and distant friends through the social network and based on their query responses, association retrieval technique is implemented to infer recommendation score. Salehi and Kamalabadi [19] propose a tree to model the interest of learner based on attributes of learning resources in multidimensional space. Recommendations are generated using a new similarity measure between the trees of learners. The experimental results show that their proposed method outperforms current algorithms and alleviates problems such as cold-start and sparsity.

2.3. Hybrid approach

To overcome drawbacks of these strategies, most of the researchers used a hybrid approach for resource recommendation [20–23]. García et al. [23] 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. Liang et al. [24] implemented the combination of content-based filtering and collaborative filtering to make personalized recommendations for a courseware selection module. At first, a user *u* enters keywords on the portal of courseware management system. Then, the courseware recommendation module finds within the same user interest group of user *u* the *k* courseware with the same or similar keywords that others choose. An appropriateness degree will be calculated for each *k* courseware by multiplying the degree of trust between user *u* and other users and the evaluation of courseware by user *u*. Finally the top five recommended

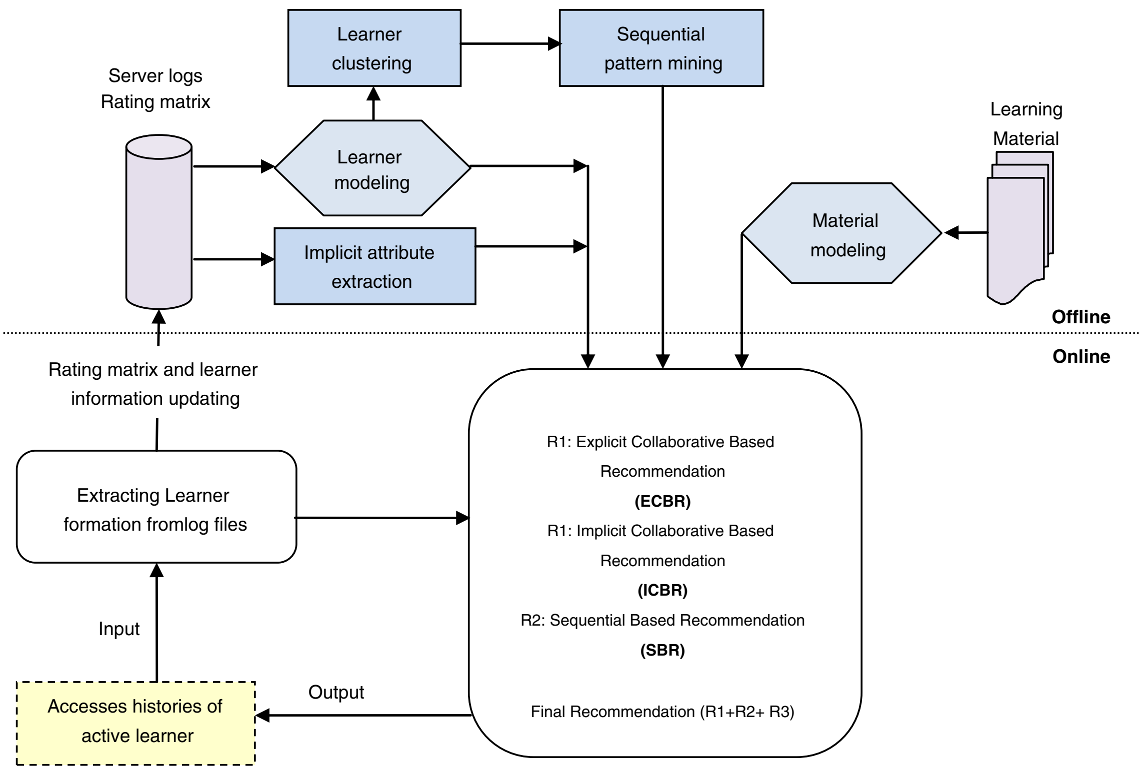

Fig. 1. Overall system architecture of the proposed e-learning recommender system.

courseware is outputted according to the recommendation degree. Khribi et al. [2] used learners' recent navigation histories and similarities and dissimilarities among user preferences and among the contents of the learning resources for online automatic recommendations. They implemented web usage mining techniques with content-based and collaborative filtering for computing relevant links to recommend to active learners. This approach doesn't use some of contextual information, such as rating information of learner and multi-attribute of learning resources. Klašnja-Milićević et al. [25] described a recommendation module of a programming tutoring system, which could automatically adapt to the interests and knowledge levels of learners. This system recognized different patterns of learning style and learner's habits through testing the learning styles of learners and mining their server logs.

In summary, to adapt a recommendation approach for learning environment, we develop a framework for integrating contextual information learners and resources and also latent patterns of resource accessing. Most of the researches only use some of this information in the resource recommendation process.

3. Methodology

We assume that the learning resources have been created using the converting and authoring tools by instructors and all the resources have been stored in a resource repository. Therefore, learners are able to view and rate the viewing resources.

The overall system architecture of the proposed e-learning recommender system is shown in Fig. 1. In the offline mode, we use rating information to extract the implicit attributes of resources and learners by matrix factorization algorithms. In addition, the learner model and resource model are built. For learner's modeling, server usage logs of learners are collected in the certain period. Since the collected logs data contain many uninteresting elements (graphics, icons and etc.), they must first be pre-processed. After cleaning the original logs, we apply the learner modeling approach to build a Learner Tree (LT) for each learner. Then, learners are clustered and access patterns of learning resources are mined. In the online phase, the resource 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 he/she asks for recommendations. Two approaches, implicit and explicit collaborative filtering recommendation and sequential based recommendation, are used for producing two recommendation sets. Finally, two recommendation sets are combined for the final recommendation.

3.1. Collaborative based recommendation

The developments of implicit and explicit attribute based CF recommendation are described in this section.

3.1.1. Implicit collaborative based recommendation

According to content-based assumption, user ratings on items are determined by the attributes of user and item corporately. In other words, if two items have similar attributes, the users would like 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 recommendation process.

Since ratings depend on needs and attributes of users and also attributes of items, the rating prediction function could be denoted as $\varphi = (M,U_i,I_j)$, M is a prediction model learned from the historical rating data H; $U_i = (w_{i1}, w_{i2},..., w_{ip})$ and $I_j = (e_{j1}, e_{j2},..., e_{jq})$ are attributes weight of the user i and item j, respectively. Based on this view, the objective of the recommender system problem is to find a fit relationship between spaces attributes of users and items to generate an appropriate recommendation. Unfortunately, in most cases we cannot use the mentioned model. Because the selection of all suitable attributes for a user and item is an almost impossible mission. Even if the attribute set is chosen, it is approximately impossible to collect the corresponding data because some data involve the privacy of people or some attributes could not be described and coded formally. This leads to the low accuracy of prediction only based on the limited observed attributes [26].

However, we can use the historical rating data in a user-item matrix for discovering some of the valuable attributes of the user and learning item that are called implicit attributes reflecting the characteristics of the learning item and user. The predication models built based on the observed attributes plus latent (implicit) attributes should have relatively higher prediction accuracy.

Let U^I and I^I denote implicit attribute space for users and items respectively. Let the vector $U_i^I = \left(w_{i1}^I, w_{i2}^I, \dots, w_{iK}^I\right)$ and $I_j^I = \left(e_{j1}^I, e_{j2}^I, \dots, e_{jK}^I\right)$ represent user i and item j implicit attribute weight respectively. The prediction function could be denoted as $\varphi' = (M,U_i,I_j,U_i^I,I_j^I)$ the historical rating data could be converted to $H' = \left\{\left(U_i, I_j, U_i^I, I_j^I, R_i^j\right)\right\}$. This research uses nearest neighborhood as a prediction model and also uses matrix factorization to discover implicit attributes.

In this research, we decompose rating matrix into two matrixes using the matrix factorization technique. In the attribute space, different people may place different emphases on the interrelated attributes. The goal of matrix factorization is to find the relationship between the overall rating and the underlying attributes rating for each user. However to improve result of matrix factorization we use genetic algorithm for the initialization of factors or implicit attributes of user and items. More specifically, given the ratings data of a user, GA computes his/her preference model in terms of implicit attribute weight. Truly, we use genetic algorithm as a supervised learning task whose fitness function is the Mean Absolute Error (MAE) of the RS.

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

$$f\left(U^I, I^I\right) = \sum_{i=N}^{N} \sum_{j=1}^{M_i} \left| \sum_{k=1}^{K} w_{ik}.e_{jk} - r_{ij} \right| \tag{1}$$

where r_{ij} is the actual rating of item j by user i, w_{ik} and e_{jk} are weight of feature k for user and item respectively and M_i is the number of rated items by user i. When $f(U^I,I^I)$ is lower, the forecasting accuracy would be higher. Finally, recommendation score item j for user i can be calculated using implicit attributes as follows:

$$Rrec.sco.\left(U_i, I_j\right) = \sum_{k=1}^{K} w_{ik}.e_{jk} \tag{2}$$

3.1.2. Explicit collaborative based recommendation

In addition to the implicit attributes that can be inferred from rating data, to improve the quality of recommendations, in this research we use explicit attributes that are known to us. In the remainder of this section, attribute represents explicit attribute.

3.1.2.1. Resource modeling. Learning resources can be categorized according to different area which each resource belongs to, such as mathematics, physics and computer science. Since the weight value of a specific area for a particular learner represents the importance of this area to this learner, it could be computed according to the number of learner's accessed educational resources which belong to the respective area. On the other hand, educational resources usually have several types of attributes (not only as subject but also as secondary subject, education type, author and so on). Therefore, for reflecting learner's preference accurately, the multidimensional attributes of educational resource must be considered.

By considering the attention degree of learners to each attribute of learning resource, the resource's attribute-based model is defined as a multidimensional vector $M = [(AK_1, AW_1),(AK_2, AW_2), \ldots,(AK_m, AW_m)]$ where AK_t indicates the t-th dimensional attribute's name of resource, AW_t denotes the appropriate weight value. In addition, we assume $AW1 \geq AW2 \geq \ldots \geq AWm$ and $\sum_{t=1}^{m} AW_t = 1$.

Based on this description model, the attributes of a certain resource M_j can be defined as $MA_j = [AK_{j1}, AK_{j2},\ldots AK_{jm}]$ where AK_{jt} denotes the t-th dimension attribute's keyword of resource M_j. For example: $M_i = [(Mathematics,0.35),(Probability, 0.3),(Master's\ degree, 0.2),(Author5, 0.15)]$".

3.1.2.2. Learner modeling. To model the dynamic multi-preference of a learner, we use rating information and also access order of resources. Basically, two main approaches can be outlined for rating of learning resources by learners: explicit rating and implicit rating. Explicit rating approach requires learners to provide explicit information about their preferences and needs, but implicit rating approach gathers information based on the online behavior and activities (i.e. implicit feedback) of learners. Since based on Nielson's 90–9–1 principle [27], more people will lurk in a virtual community than will participate, this research uses implicit rating. Based on this theory, user participation often more or less follows a 90–9–1 rule [27]:

- 90% of users are lurkers (i.e., read or observe, but don't contribute).
- 9% of users contribute from time to time, but other priorities dominate their time.
- 1% of users participate a lot and account for most contributions.

Rating of a learner can be determined in a number of ways. However the rating schema must precisely model the user's interest. In e-learning environments, the relative resource access operations of a certain learner can be regarded as an access sequence defined as $S_{L_i} = \{M_1, M_2, \ldots, M_n\}$. So, according to each access sequence of all learners, the access sequence set S can be obtained.

In our rating schema, both access time-length of a resource and visiting frequency of a resource are used to estimate its importance for a learner. The idea of using resource visit duration as one of the weighting parameters is reasonable because, it reflects the relative importance of each resource, because a learner generally spends more time on a more useful learning resource and if a learner is not interested in a resource, he/she do not spend much time on viewing the resource and usually jumps to another resource quickly. However, a quick jump might also occur due to the short length of a resource so the size of a resource may affect the actual visiting time. Hence, it is more appropriate to accordingly normalize time by the total bytes of the resource. The formula of time is given in Eq. (3). Frequency is the number of times that a resource is accessed by a learner. It seems reasonable to assume that resources with a higher frequency are of stronger interest to learners. The formula of frequency is given in Eq. (4). We use time spent by a learner for viewing a resource and frequency of visiting as two very important pieces of information in measuring the learner's interest on the resource, so we assign a significant rate to each resource for a learner according to these definitions as Eq. (5).

$$Time\left(L_i, M_j\right) = \frac{\dfrac{TotalTime\left(L_i, M_j\right)}{size\left(M_j\right)}}{\max_{q \in S_{L_i}}\left(\dfrac{TotalTime\left(L_i, M_q\right)}{size\left(M_q\right)}\right)} \tag{3}$$

$$Frequency\left(L_i, M_j\right) = \frac{Number\ \ of\ \ visits\left(L_i, M_j\right)}{\sum_{q \in S_{L_i}} Number\ \ of\ \ visits\left(L_i, M_q\right)} \tag{4}$$

$$MR\left(L_i, M_j\right) = \begin{cases} 5 \times Nor\left(Frequency\left(L_i, M_j\right) \times Time\left(L_i, M_j\right)\right) & \text{if learner } L_i \text{ observes material } M_j \\ \phi & \text{othewise} \end{cases} \tag{5}$$

where *Nor* is the normalization function. The scope of rating is set 1–5 in this paper. Ratings of a learner updates as the leaner proceeds in the system. $MR(L_i,M_j)$ denotes rate of learner L_i for resource M_j. It must be noted that some resources are more difficult or/and their navigation is complicated. We address this drawback by normalization of users' implicit rating on resources. Since understanding a complicated resource is difficult for all users, this normalization on rating of users can remove the impact of this factor somewhat.

In addition, we use the order of accessed resource as useful information for dynamic interest modeling. Usually in e-learning environments, the preference of a learner may change and the history records couldn't entirely reflect the whole preference of a learner. On the other hand, the preference of a learner's recently accessed resources 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 resources 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 an exponential function, as follows:

$$\begin{aligned} &h\left(x\left(M_j\right)\right) = \exp\left(-\lambda\left(x\left(M_j\right) - 1\right)\right) \\ &, 0 \le \lambda \le 1, 0 < h(x) < 1 \end{aligned} \tag{6}$$

where $x(M_j)$ is the M_j access order in the session of L_i. Therefore, the effect of M_j to L_i's interest will become smaller with resource 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 a learner's preference, and the bigger the λ, the quicker the forgetting.

By using GFF, we implement three important rules: (1) Since the resources which are accessed recently have a larger $h(x)$, recommended resources is similar to the recent accessed resources. (2) Since the $h(x)$ is a nonlinear function, the recently accessed resource does not have a large value for a long time and this prevents the effect of an occasional resource access to long term access. (3) In addition, the nonlinear function, $h(x)$ supports the previous long term preference of learner.

The $h(x)$ attenuation with $\lambda = 0.95$ is shown in Fig. 2. Based on Eq. (6), the $h(x)$ value of the latest accessed resource is equal to 1, and with access going on, the $h(x)$ value of resources could be updated. Now by using the two pieces of information, rating information and dynamic interest information, we can implement learner modeling.

A drawback of existing algorithms is that there is no modeling of dynamic interest and multi-preference of the learner. To use this information in a recommendation process for increasing the accuracy of a recommendation, we introduce the Learner Tree.

LT combines multidimensional attributes of learner's accessed resources and learner's rating information to model multi-preference of the learner. In addition, LT uses $h(x)$ value of learner's accessed resources to model dynamic interest of the learner.

In this research, learner L_i is defined as a tree with $(m + 1)$ levels in which m indicates the number of attributes of resources. In this tree, the leaf node which represents an accessed resource of L_i is defined as a four-tuple: $LT_{leaf} = \{MID, OR, NH, MR\}$, where *MID* indicates accessed resource ID, *OR* indicates current resource access order of learner L_i, *NH* indicates the normalization value of $h(x)$ for accessed resources of L_i, and *MR* indicates the rating of L_i to a certain resource. The non-leaf node can be defined as a three-tuple: $LT_{nonleaf} = \{KA, NH, MR\}$, where *KA* is the keyword of the *level*-th attribute of resource. A four-dimension attribute description model based LT is considered in this research including subject, secondary subject, education type and author, a sample of which is shown in Fig. 3.

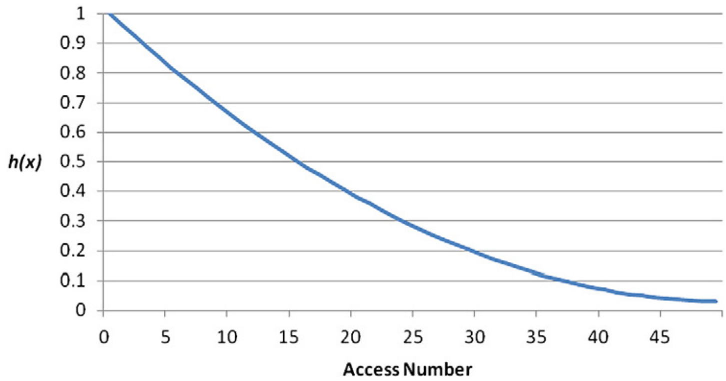

Fig. 2. Gradual forgetting function.

The *NH* value of *i*-th $(0 < i < m + 1)$ level node can be calculated as the sum value of its entire immediate successor which is placed at $(i + 1)$-th level. The *MR* of non-leaf node *k* can be calculated as the mean of *MR* value of all leaf nodes which belong to *k*'s sub-tree. In this tree, each accessed resource corresponds to a unique path from root to relevant leaf node, and the keywords of all nodes located in this path correspond to the relevant keywords of M_j's attributes.

As Fig. 3 shows, first the learner prefers to use information technology resource and then he/she changes his/her interest and uses mathematics resource. Our modeling considers this dynamic interest by allocating $NH = 0.24$ to the information technology resources and $NH = 0.76$ to the mathematics resources.

LT must be constructed for each learner and updated according to the following strategy: Search the keywords of the latest accessed resource attributes ($MA_j = [AK_{j1}, AK_{j2},..., AK_{jm}]$) in *LTM* from the upper level to the bottom level. If the keyword of the *i*-th attribute cannot be matched, the $m - i + 1$ new node(s) with latter $m - i + 1$ attribute(s) of resource will be created and updated *NH*, *MR* in the whole of tree.

3.1.2.3. Explicit CF recommendation. Most of the existing collaborative based recommendation algorithms usually represent the similarity between learners by calculating similarity degree between two rating vectors in learner–resource rating matrix. However, there are several drawbacks of the traditional rating data based similarity calculation method: (1) CF's dependence on learner ratings can be a drawback. Because several learners must evaluate each learning resource and new resources cannot be recommended until some learners have taken the time to evaluate them. This problem is referred to as 'data sparsity' that makes 'cold start problem' also. (2) Most existing algorithms only consider the learner–resource rating matrix, and don't use the context information of learner and attributes of resource. So, it will result to inaccurate results of similarity calculation.

To address these drawbacks, a new similarity measure is introduced based on learner's LT. As a logical assumption, two learners with similar attribute keywords in their Learner Tree can be considered as similar neighbors. Based on this assumption,

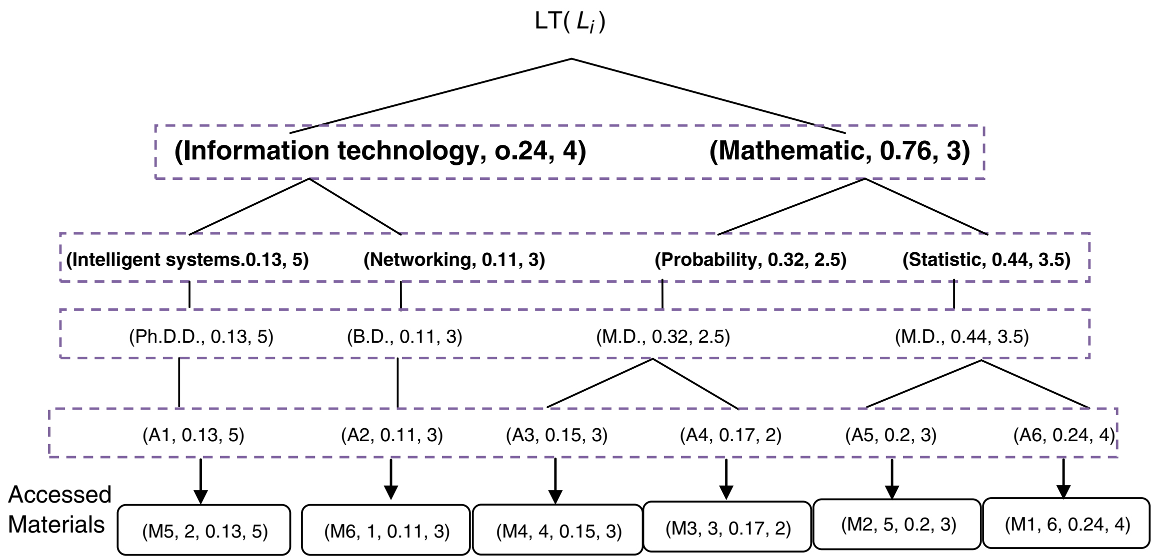

Fig. 3. Learner Tree sample.

we can solve the sparsity problem and use the context information of learner and attributes of resource to improve the quality and accuracy of the recommendation. For defining similarity degree, three rules must be considered:

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

Therefore, the similarity degree between two learners can be calculated based on the Attributes Subscription Vector (*ASV*) between their corresponding *LTM*. *ASV* between learner L_a and L_b, $ASV(L_a,L_b)$, is defined as the maximum intersection between levels of LT_a and LT_b with the same keyword in each level. After the matching process, we have an *ASV* such as that shown in Fig. 4.

The calculation of similarity between two learners can be divided into two aspects as: dynamic preference based similarity and learner rating based similarity. The attribute based similarity $sim_I(L_a,L_b)$ can reflect the similarity between learner L_a and learner L_b based on dynamic preferences simultaneously. Inspired by Cosine similarity, the calculation of $sim_I(L_a,L_b)$ can be defined as follows:

$$sim_I(L_a,L_b)=\frac{\sum_{i\in ASV(L_a,L_b)}MW_i.NH_{ai}.NH_{bi}}{\sqrt{\sum_{i\in LT(L_a)}MW_i.NH_{ai}^2}.\sqrt{\sum_{i\in LT(L_b)}MW_i.NH_{bi}^2}} \tag{7}$$

where NH_{ai} indicates the value of *NH* in the *i*-th level's matching for learner *a*. MW_i indicates the *i*-th level's matching weight of *LT*. Since MW_i should increase with increasing depth, in this paper, it can be defined as $MW_i=AW_i^{-1}$.

For reflecting the similarity between the rating vectors of two learners, inspired by Pearson similarity, the learner rating based similarity $sim_R(L_a,L_b)$ can be applied as follows:

$$sim_R(L_a,L_b)=\frac{\sum_{i\in L}\left|(MR_{ai}-\overline{MR_a}).(MR_{bi}-\overline{MR_b})\right|}{\sqrt{\sum_{i\in L}(MR_{ai}-\overline{MR_a})^2}.\sqrt{\sum_{i\in L}(MR_{ai}-\overline{MR_a})^2}} \tag{8}$$

where *L* indicates the leaf node set of $ASV(L_a,L_b)$. $\overline{MR_a}$ and $\overline{MR_b}$ indicate the mean value of L_a and L_b's rating data respectively. In the calculation of $sim_R(L_a,L_b)$ that computes similarity between *MR* values of nodes on LT_a and LT_b which correspond to each leaf node on $ASV(L_a,L_b)$, it does not need to have identical accessed resources between two learners. By this definition of similarity, we can overcome the sparsity rating problem. In addition, to decrease the deviation of learner's different rating scales, learner's rating data was modified.

Now we can calculate Collaborative Based Similarity (CBS) between L_a and L_b as follows:

$$CBS(L_a,L_b)=\alpha.sim_R(L_a,L_b)+(1-\alpha).sim_A(L_a,L_b) \tag{9}$$

where α indicates the weight between $sim_R(L_a,L_b)$ and $sim_A(L_a,L_b)$. After obtaining similarity between learners, we can calculate Appropriateness Degree (AD) of resource M_j for learner L_i by Eq. (10):

$$AD\left(L_i,M_j\right)=\overline{MR}_i+\frac{\sum_{q\in L_{M_j}}CBS\left(L_i,L_q\right)\cdot\left(MR_{qj}-\overline{MR}_q\right)}{\sum_{q\in L_{M_j}}CBS\left(L_i,L_q\right)}$$
$$L_{M_j}=\left\{q\in L\middle|MR_{qj}\neq\Phi\right\} \tag{10}$$

where MR_{qj} is rate of learner L_q for resource M_j, L_{M_j} indicates learners that have rated M_j, and $\overline{MR_i}$ indicates the mean value of L_i's rating data. The final step in this phase is to ultimately derive the top-n recommendation. For an active learner, we produce a recommendation set of *n* resources according to a greater appropriateness degree. It must be noted that previously selected resources are excluded from the recommendation set.

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

Fig. 4. Attributes Subscription Vector sample.

3.1.2.4. Time complexity analysis. In $AD(L_i,M_j)$ calculation, according to formulas (7)–(9) and relevant definitions, as it does not need to consider item-based similarity calculation the similarity between L_i and any other learner can be computed within $O(\text{p} \cdot |\text{AIS}|)$, where p is the size of learner set. Therefore the explicit CF recommendation can be calculated within $O(\text{p} \cdot |\text{AIS}| + k \cdot |\text{RS}|)$ where k is number of neighbors and |RS| is the size of candidate resource. However, collaborative filtering can be computed within $O(\text{p} \cdot |\text{Union Set}| + \text{k}. |\text{RS}|)$. As |Union Set| and |AIS| usually have the same order, thus the ECBR algorithm has the same time complexity with collaborative filtering recommendation algorithm.

3.2. Sequential based recommendation

E-learning systems have their own characteristics and simply transferring a recommender system from an existing (e.g. commercial) content to e-learning systems may not accurately meet the needs of the targeted users. Compared with resource recommendation in e-commerce systems, users in e-learning systems have topic preferences in e-learning systems. However, users' behaviors in learning systems are in a more consistent and coherent way [16]. Learning items have some intrinsic orders in users' learning processes. For example, a user will probably learn from easy resources to difficult ones; for a single knowledge point, a user will probably learn from theoretical to practical. In another example, the learning resource access sequence in a special learning process is: {Lecture notes, Reference, Exercise book}. Therefore, the time-dependency relationship between learning resources in a learning process can reflect learner's latent resource access pattern and preference. However, conventional CF approaches cannot reflect these characteristics. Therefore, it is necessary to find a method to represent the sequences of the e-learning resources. We can 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 the near feature to further improve quality of recommendations and solve new user problem.

3.2.1. Mining sequential pattern using BIDE

Sequential pattern mining has received considerable attention among the researchers with broad applications. The sequential pattern algorithms generally face problems when mining long sequential patterns or while using very low support threshold. One possible solution of such problems is considering only the closed sequential patterns, which is a condensed representation of sequential patterns. BI-Directional Extension based frequent closed sequence mining (BIDE) which is an efficient algorithm for mining frequent closed sequences without candidate maintenance is used in this research for mining closed sequential patterns of resource accessing [28]. It adopts a novel sequence closure checking scheme called BI-Directional Extension, and prunes the search space more deeply compared to the previous algorithms by using the BackScan pruning method and the ScanSkip optimization technique.

In order to improve the efficiency of recommendation based on sequential patterns of resource accessing, we can apply a clustering algorithm to group learners. Each cluster contains learners with similar behavior and similar interests and so similar sequential pattern. Clustering approach can increase the scalability of recommendation that is an important problem in recommender systems. In addition, clustering can personalize recommendations and improve accuracy. By using collaborative based similarity and the *k*-means algorithm [29], it is easy to discover clusters of learners. Then, we can discover sequential patterns of each cluster. We apply *k*-means based on collaborative based similarity in off-line in the initial stage of system development using gathered data in the certain period. But after determining center of each cluster using *k*-means, in online mode the active learner first is assigned to an appropriate cluster based on her/his Euclidean distance with the center of clusters

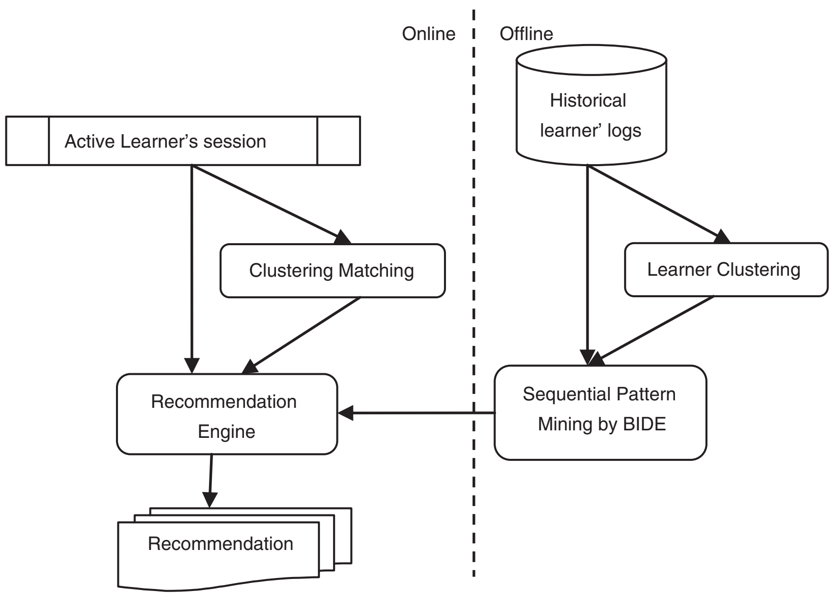

Fig. 5. Sequential pattern based recommendation.

and then gets recommendation using collaborative filtering. Since Learner Tree is updated in interaction with the learning environment, we must update the center of clusters by k-means algorithm in offline after a certain period. Using this approach we can decrease computation cost in online and improve the scalability of system (Fig. 5).

3.2.2. A recommendation mechanism using BIDE

In this section, a recommendation mechanism based on BIDE is developed.

3.2.2.1. Sliding window. To compute a recommendation set based on the discovered sequences for the current (active) learner sequence, a sliding window is used to control the number of resources to be matched against the patterns in the sequences [30]. It is very important to maintain a historical depth for providing reasonable suggestions.

In the most of works [31] a fixed size sliding window (w) is used in the active learner sequences to obtain the historical depth of current learner and generate recommendations. In this situation, recommendation results are affected by only the last visited w resources. Usually all the w last visited resources are treated equally. However, in e-learning environments, the preference of a learner may change and w last visited resources couldn't entirely reflect dynamic interest of a learner. Also, the preference of a learner's recently accessed resources has an important role to the future interests. Therefore, recently accessed resources by a learner are more suitable for recommendation. Thus, in order to signify resources in an accessed resource sequence by an active learner, a coefficient is defined as follows:

$$h(x) = \frac{x}{|w|} \quad x = 1, 2, \ldots, |w| \tag{11}$$

where $|w|$ the size of sliding window and x is the location of resource in the sliding window. 1 is assigned to the first visited resource. Obviously, the last visited resources have the largest coefficient. In order to reflect the impact of recently accessed resource, the interest of each resource is defined as follows:

$$Interest(M_i) = h(x).\frac{W(M_i)}{\sum_{j=1}^{n} W\left(M_j\right)} \tag{12}$$

w accessed resources with more interest are selected as sliding window. Eq. (12) guarantees that interest of a resource is high only when $W(M_i)$ and $h(x)$ are both high.

3.2.2.2. Recommendation score. After extracting sequential patterns for each cluster, the recommendation engine will calculate the recommendation score (SR) of candidate resources according to learner's sliding window that was defined in the previous section to predict the most possible resources which will be accessed in the future. Matching process is implemented on the sliding window of target learner (SW_w) as prefix sequence. Assume that MS_{pos} indicates the resources which appear in the successor of a sequential pattern including SW_w and also assume that the sequential pattern set with SW_w as prefix is indicated as PS_{pos}. Fig. 6 shows an illustrative example for learner's sliding window that is the prefix of a sequential pattern.

In this research, we combine tree rules in the calculation of recommendation score of candidate resources for each learner:

(1) The more times M_i appears in PS_{pos}, the larger the recommendation score of this resource will be.
(2) The more anterior M_i appears in a sequential pattern, the larger the recommendation score of this resource will be.
(3) The larger the ratio between support of sequential pattern s_i in which resource M_i is located and the support of its predecessor sequential pattern, the larger the recommendation score of this resource will be.

Based on defined rules, the recommendation score M_i for learner L_a can be defined as formula (13).

$$RS_{sequence}(L_a, M_i) = \frac{\sum_{s \in PS_{pos}} \varphi(find(s - SW_w, M_i)) \frac{SP(s)}{SP(SW_w)}}{\sum_{s \in PS_{pos}} \varphi(find(s - SW_w, M_i))} \tag{13}$$

where the *SP* function denotes the support of certain sequential pattern; the *find* function can return the position of resource M_i in the sequential pattern without a corresponding prefix, and the $\phi(x)$ function denotes the order weight of resource in the sequential

$$s_i = < \overbrace{M_1, M_2, \ldots, M_w}^{SW_w}, M_{w+1}, M_{w+2}, \ldots, M_i, \ldots, M_{w+n} >$$

Fig. 6. An illustrative example for learner's sliding window that is the prefix of a sequential pattern.

pattern, where x denotes the resource's order in certain sequential pattern. Since based on rule (2) $\phi(1) > \phi(2) > \ldots > \phi(n)$, in this research we suppose $\phi(x) = \exp(-x)$.

Finally n resources with the highest recommendation score that are not already visited by the active learner are chosen as the recommendation set.

3.3. Final recommendation

In order to improve the accuracy of recommendations in this research, at first, the results of the implicit and explicit CF recommendations are combined to make collaborative based recommendation (CBR), as shown below:

$$Score_{CBR-based}\left(L_i, M_j\right) = \beta.Nor\left(AD\left(L_i, M_j\right)\right) + (1-\beta) \cdot Nor\left(Recsco\left(L_i, M_j\right)\right) \tag{14}$$

where $Nor(x)$ is a normalization function. Then, two recommendation sets including sequential-based recommendation (SBR) and CBR are combined through cascade approach. In other words, first we produce a primary recommendation list using CBR method. We select only items that their recommendation score calculated by Eq. (14) is more than the user defined value γ. We rank the primary recommendation list using SBR recommendation method and finally N-items with the highest recommendation score are chosen as the recommendation set.

4. Implication

In this section, we conduct several experiments to evaluate recommendation performance of the proposed recommendation model. Overall our experiments have verified the effectiveness of the proposed model in resource recommendation.

4.1. Experiment environment and data set

All experiments in this paper were performed on DELL 1400 Vostro with Intel Pentium 1.6 GHz CPU and 1 GB RAM, and Windows 7 operating system. The algorithm is implemented by Matlab 2009.

As a real-world dataset, D-Lib records are applied in our experiments in order to evaluate the proposed algorithm. The D-Lib records dataset comes from the usage data of digital library that is an informal learning environment integrated with the course management system Moodle. In this environment, learners are responsible for their own learning pace and path [32]. The learning process depends to a large extent on individual preferences or choices and is often self-directed [33]. The resources come from sources such as expert communities, work context, training or even friends that might offer an opportunity for an informal competence development.

The D-Lib records dataset is from August, 2009 to February, 2011. This dataset contains 52,345 lending records from 2364 users on 3763 books where each record contains a timestamp; in addition it contains book type information and user's basic information.

The characteristics of this dataset are summarized in Table 1. In this table, density is obtained from dividing the number of transactions by the number of cells in the rating matrix. Average person record is the mean number of transactions for each learner. In the experiments, the dataset is ordered by user's access timestamp, and then is divided into training set and test set. In order to increase the number of records in the test set as much as possible so as to eliminate the effect of accidental factor, the top 60% access records of each user in ordered dataset are used as the training set and the remaining 40% access records are used as test set.

4.2. Evaluation metrics

The experimental evaluation is based on two metrics: Precision and Recall defined as following:

$$Precision = \frac{\sum_{i=1}^{p} |TS(U_i) \cap Rec(U_i)|}{\sum_{i=1}^{p} |Rec(U_i)|} \tag{15}$$

$$Recall = \frac{\sum_{i=1}^{p} |TS(U_i) \cap Rec(U_i)|}{\sum_{i=1}^{p} |TS(U_i)|} \tag{16}$$

Table 1
The characteristics of the dataset used in experiments.

Dataset	Number of users	Number of items	Number of transactions	Density	Average person records
D-Lib records	2764	3763	52,345	0.503%	18.94

where p is the size of the user set, $Rec(U_i)$ denotes the recommendation set of user U_i, TS denotes the test set or the relevant items set (that must be recommended) and $|TS(U_i) \cap Rec(U_i)|$ indicates the hits number of U_i's recommendation set. Precision measure represents the proportion of relevant recommendations to the total number of obtained recommendations, whereas Recall measure or coverage represents the proportion of relevant recommendations to all learning resources that should be recommended [3].

Precision and recall metrics measure the decision-support accuracy that indicates how effectively predictions help a learner to select high-quality resource from the resource set [30]. However, since usually there is a trade-off between algorithm running time and recommendation precision, we measure mean running time for single user that can help us for an appropriate decision.

4.3. Performance evaluation

In this section, first the impact of input parameters is analyzed on the recommendation performance. Then, for evaluation of the proposed approach, it is compared with vector space content-based recommendation algorithm [34], collaborative based recommendation algorithm [35] and improved hybrid recommendation algorithm [36]. In these comparisons, N denotes the number of recommendations; p denotes the number of participating users which are selected from the dataset to build an experiment dataset and G is number of neighborhoods.

4.3.1. Impact of parameters

At first, we will analyze how α, β, w and C or number of clusters affect the recommendation performance in order to determine the optimal values of these parameters. Fig. 7 shows the impacts of α and β on the precision of recommendation while $N = 16$, $p = 300$, $G = 10$. Fig. 7A indicates that using contextual information of resources and learners including attributes of resource, access order of resource, and rating of learners will lead to the better recommendation performance, and the best precision can be obtained with $\alpha = 0.7$. Fig. 7B also indicates that combination of explicit and implicit CF can improve the accuracy of recommendation and the best results can be obtained with $\beta = 0.6$.

In the next experiment, we have executed k-means algorithm in order to find the best values for number of clusters and size of the sliding widow that is more appropriate to use with D-Lib records data. In this experiment, number of clusters C varies from 2 to 8 while size of sliding widow, w varies from 2 to 6, $N = 16$, $p = 300$ and $G = 10$.

A large sliding window provides more information to the system while on the other hand it increases the computation time. As Fig. 8 shows a window of size 2 cannot hold enough information for the recommendation. Therefore, the accuracy improves with increasing window size. However, the difference of accuracy between window size 5 and 6 is not very much. In this research, we consider $w = 5$. On the other hand, Fig. 8 shows that with increasing number of clusters, the precision of SBR decreases. The reason is that with increasing number of clusters, neighbors of each learner decrease. In this research, to balance the scalability of recommender and precision of results, we consider $C = 4$.

4.3.2. Comparative study

In the first experiment, the precision of proposed algorithms is compared with respect to G while $N = 16$ and $p = 300$. As shown in Fig. 9, when G is limited in a certain value range, with the increasing of G, the precision of each algorithm is increasing except the SBR algorithm. When G reaches a certain extent, with an increase of G, the precision of each algorithm is decreasing, especially for implicit attribute CF. Moreover during the change of G, the proposed method (hybrid combination of SBR and CBR) always produces better performance than any other algorithm. The reason is that we set a threshold in the similar learner's calculation process to guarantee their quality and therefore in this situation, dissimilar learners cannot be denoted as similar learners. Meanwhile the dynamic learners' preference and the resource's attributes are taken into account based on traditional collaborative-based mechanism. Since neighbors aren't used by the SBR method to generate recommendation, this method is not sensitive to G.

In the second comparison experiment, the precision of recommendation algorithms is compared with respect to N while $p = 300$ and $G = 10$. As shown in Fig. 10, with the increase of N, the precision of algorithms is decreasing. However, the proposed algorithm

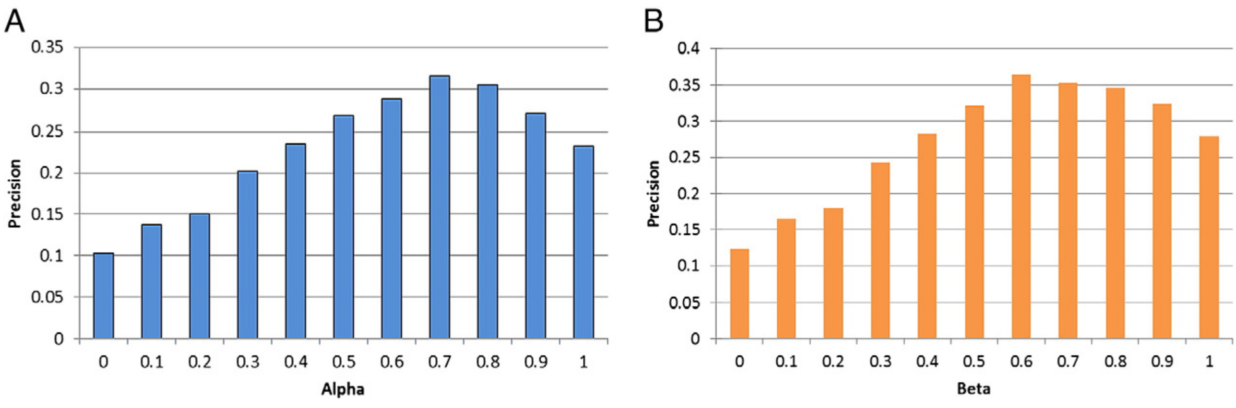

Fig. 7. The precision of explicit based CF and CBR with respect to α and βrespectively.

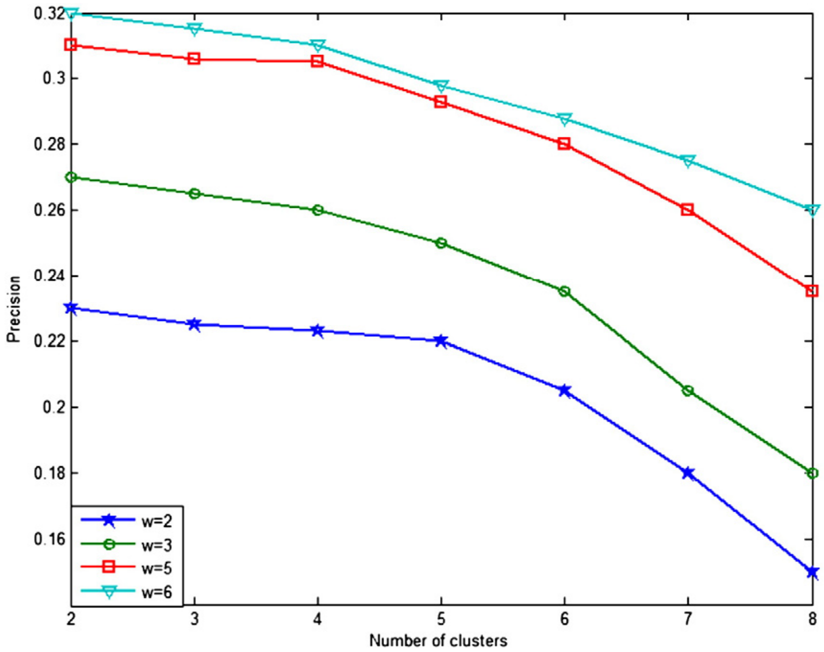

Fig. 8. The precision of sequential based recommendation (SBR) with respect to C and w.

always produces better performance than other algorithms, especially when N is small because during the changing process, according to precision formula, the numerator and denominator of precision will increase synchronously, but denominator gets the higher increasing rate. In addition, the proposed algorithm by integrating multi-dimensional attributes of resource and user's rating can reflect the actual preference and interests of users accurately. Based on them, the proposed algorithm can filter and rank the candidate resources much more efficiently.

In the third comparison experiment, the mean running times for single learner of all algorithms are compared with respect to p while $N = 16$ and $G = 10$. As shown in Fig. 11 at all times, content-based algorithm is faster than any other algorithms. The running time of the proposed and improved hybrid recommendation algorithms are a little larger than collaborative based

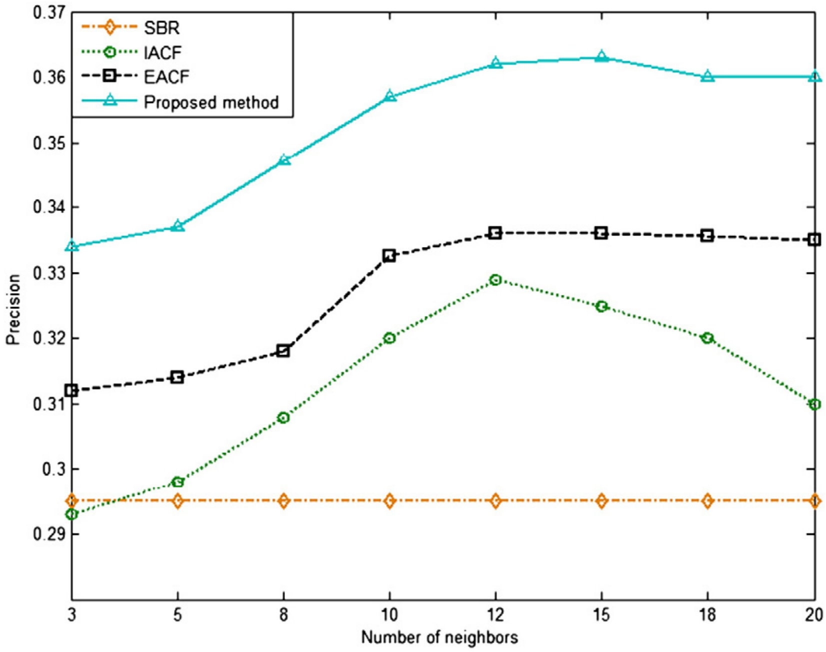

Fig. 9. The precision of proposed algorithms with respect to p.

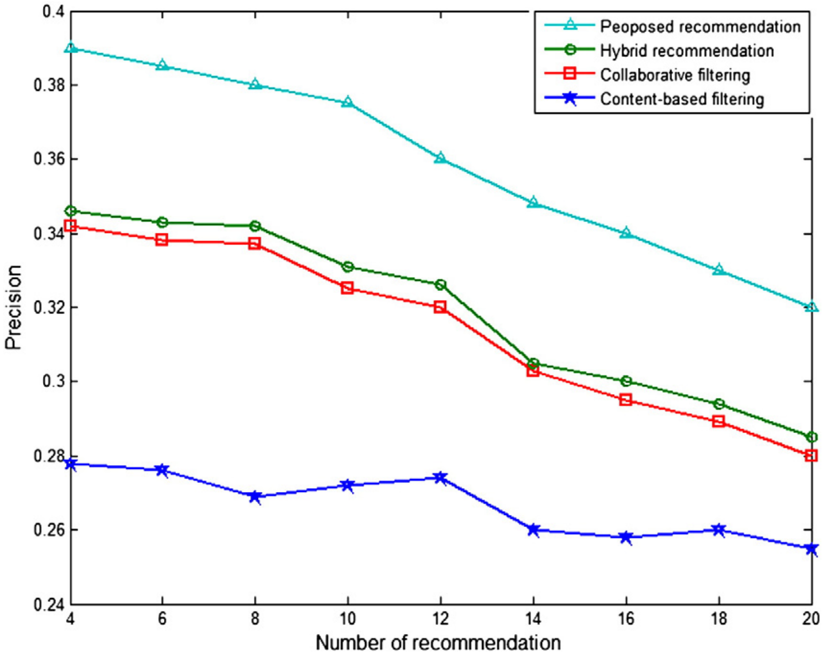

Fig. 10. The precision of algorithms with respect to N.

algorithm. According these experiments, although the proposed recommendation will get higher precision in most case, it will cost the largest running time. Therefore, there is a trade-off between algorithm running time and recommendation precision when choosing the proposed recommendation algorithm.

In the fourth comparison experiment, the recall of recommendation algorithms is compared with respect to p while $N = 16$ and $G = 10$. As shown in Fig. 12, at all times, the proposed algorithm has higher recall than any other algorithms. When p is low, the recall of all algorithms is approximately equal, but by increasing of p, the proposed recommendation has better results. According these experiments, the proposed recommendation has higher precision and recall in most cases.

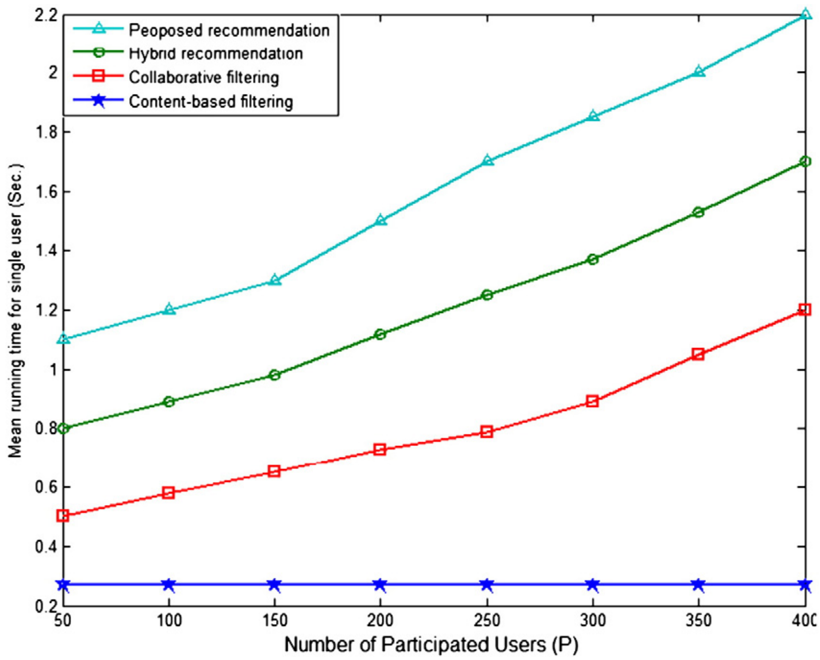

Fig. 11. The mean running times for single learner of algorithms with respect to p.

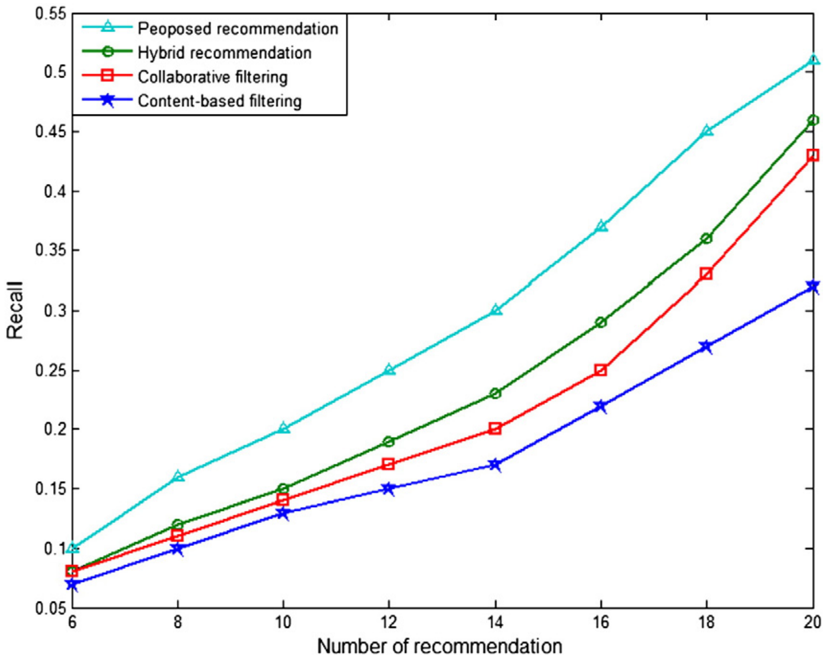

Fig. 12. The recall of algorithms with respect to *p*.

5. Conclusion

Personalized recommendations are used to support learning of users in the learning environments. However, there are several drawbacks when applying the existing recommendation algorithms. To address these drawbacks in this paper, we proposed a new resource recommender system framework for e-learning based on implicit and explicit collaborative filtering and sequential based recommendation that are able to take into account the multidimensional information of resource and learner and the latent patterns of resource access information for the recommendation process. Two approaches were used for producing of recommendation. In the first approach, Collaborative Based Recommendation, Learner Tree was introduced to consider multi-dimensional attributes of resources, time-variant multi-preference of learner, and relevant rating information simultaneously in the explicit CF. In addition, in this approach, implicit attributes are introduced to reinforce recommendation results. In the second approach, sequential based recommendation, we cluster learners by the new similarity measure and then BIDE algorithm was implemented in each cluster to find sequential patterns in accessing of materials for recommendation. The experiment results show that our algorithm can outperform traditional recommendation algorithms significantly in precision, recall and could be more suitable for e-learning environments. Based on the proposed algorithm, the learner's real learning preference can be satisfied accurately according to the real-time up dated contextual information.

Since in e-learning environments as logical assumption learners with greater knowledge have to be more important in the calculation of the recommendations than the learners with less knowledge, in the future, we plan to continue our work on a new approach that can take into account the knowledge of a learner and extend the existent equations of recommendation.

References

[1] K.I. Ghauth, N.A. Abdullah, Learning materials recommendation using good learner's ratings and content-based filtering, Educational Technology Research and Development 58 (6) (2010) 711–727.
[2] M.K. Khribi, M. Jemni, O. Nasraoui, Automatic recommendations for e-learning personalization based on web usage mining techniques and information retrieval, Educational Technology and Society 12 (4) (2009) 30–42.
[3] B. Mobasher, Data mining for personalization, in: P. Brusilovsky, A. Kobsa, W. Nejdl (Eds.), The Adaptive Web: Methods and Strategies of Web Personalization, 2007, pp. 1–46.
[4] G. Adomavicius, A. Tuzhilin, Toward the Next generation of recommender systems: a survey of the state-of-the-art and possible extensions, IEEE Transactions on Knowledge and Data Engineering 17 (6) (2005) 734–749.
[5] M.S. Chen, J. Han, P.S. Yu, Data mining: an overview from a database perspective, IEEE Transactions on Knowledge and Data Engineering 8 (6) (1996) 866–883.
[6] R. Gluga, J. Kay, T. Lever, Modeling long term learning of generic skills, Intelligent Tutoring Systems 1 (2010) 85–94.
[7] H. Drachsler, H. Hummel, R. Koper, Personal recommender systems for learners in lifelong learning: requirements, techniques and model, International Journal of Learning Technology 3 (4) (2008) 404–423.
[8] K. Verbert, N. Manouselis, X. Ochoa, M. Wolpers, H. Drachsler, I. Bosnic, E. Duval, Context-aware recommender systems for learning: a survey and future challenges, IEEE Transactions on Learning Technologies 5 (4) (2012) 318–335.
[9] N. Manouselis, H. Drachsler, R. Vuorikari, H. Hummel, R. Koper, Recommender systems in technology enhanced learning, in: P.B. Kantor, F. Ricci, L. Rokach, B. Shapira (Eds.), Recommender Systems Handbook, 2011, pp. 387–415.
[10] N. Sharif, M.T. Afzal, D. Helic, A framework for resource recommendations for learners using social bookmarking, Computing and Networking Technology (ICCNT), 2012 8th International Conference on, IEEE, 2012, pp. 71–76.

[11] D. Lemire, H. Boley, S. McGrath, M. Ball, Collaborative filtering and inference rules for context-aware learning object recommendation, International Journal of Interactive Technology and Smart Education 2 (3) (2005) 1–11.
[12] C. Romero, S. Ventura, A. Zafra, P. Bra, Applying web usage mining for personalizing hyperlinks in web-based adaptive educational systems, Computers in Education 53 (2009) 828–840.
[13] M. Salehi, I. Nakhai Kamalabadi, M.B. Ghaznavi Ghoushchi, Personalized recommendation of learning material using sequential pattern mining and attribute based collaborative filtering, Education and Information Technologies (2012) 1–23, http://dx.doi.org/10.1007/s10639-012-9245-5.
[14] M. Salehi, 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 48 (2013) 57–69.
[15] J. Bobadilla, F. Serradilla, A. Hernando, MovieLens, collaborative filtering adapted to recommender systems of e-learning, Knowledge-Based Systems 22 (4) (2009) 261–265.
[16] W. Chen, Z. Niu, X. Zhao, Y. Li, A hybrid recommendation algorithm adapted in e-learning environments, World Wide Web (2012) 1–14.
[17] S.B. Aher, L. Lobo, Applicability of data mining algorithms for recommendation system in e-learning, Proceedings of the International Conference on Advances in Computing, Communications and InformaticsACM, 2012, pp. 1034–1040.
[18] P. Dwivedi, K.K. Bharadwaj, e-learning recommender system for learners in online social networks through association retrieval, Proceedings of the CUBE International Information Technology Conference, ACM, 2012, pp. 676–681.
[19] M. Salehi, I. Nakhai Kamalabadi, Attribute-based recommender system for learning resource by learner preference tree, Computer and Knowledge Engineering (ICCKE), 2012 2nd International e-Conference on, IEEE, 2012, pp. 133–138.
[20] M. Salehi, M. Pourzaferani, S.A. Razavi, Hybrid attribute-based recommender system for learning material using genetic algorithm and a multidimensional information model, Egyptian Informatics Journal 14 (1) (2013) 67–78.
[21] L. Wan, C. Zhao, A hybrid learning object recommendation algorithm in E-learning context, International Journal of Digital Content Technology and its Applications 6 (18) (2012) 442–448.
[22] A. Ruiz-Iniesta, G. Jiménez-Díaz, M. Gómez-Albarran, A hybrid user-centred recommendation strategy applied to repositories of learning objects, International Journal of Web Based Communities 8 (3) (2012) 302–321.
[23] E. García, C. Romero, S. Ventura, C.D. Castroa, A collaborative educational association rule mining tool, The Internet and Higher Education 14 (2) (2011) 77–88.
[24] G. Liang, K. Weining, L. Junzhou, Courseware recommendation in e-learning system, In Advances in web based learning (2006) 10–24.
[25] A. Klašnja-Milićević, B. Vesin, M. Ivanović, Z. Budimac, E-learning personalization based on hybrid recommendation strategy and learning style identification, Computers in Education 56 (3) (2011) 885–899.
[26] J. Zhong, X. Li, Unified collaborative filtering model based on combination of latent features, Expert Systems with Applications 37 (8) (2010) 5666–5672.
[27] J. Nielsen, Participation Inequality: Lurkers vs. Contributors in Internet Communities, http://www.useit.com/alertbox/participation_inequality.html 2006.
[28] J. Wang, J. Han, BIDE: efficient mining of frequent closed sequences, Data Engineering, 2004. Proceedings. 20th International Conference on, IEEE, 2004, pp. 79–90.
[29] J. Li, O. Zaïane, Combining usage content and structure data to improve web site recommendation, Proceedings of the international conference one-commerce and web technologies, China, 2004, pp. 305–315.
[30] B. Mobasher, H. Dai, T. Luo, M. Nakagawa, Effective personalization based on association rule discovery from web usage data, Proceedings of the third ACM Workshop on Web Information and Data Management, Atlanta, Georgia, 2001.
[31] M. Nakagawa, B. Mobasher, A hybrid web personalization model based on site connectivity, The fifth international WEBKDD workshop: Web mining as a premise to effective and intelligent web applications, 2003, pp. 59–70.
[32] N. Longworth, Lifelong Learning in Action — Transforming Education in the 21st Century, Kogan Page, London, 2003.
[33] R.G. Brockett, R. Hiemstra, Self-direction in Adult Learning: Perspectives on Theory, Research and Practice, Routledge, London, 1991.
[34] G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions, IEEE Transactions on Knowledge and Data Engineering 17 (6) (2005) 734–749.
[35] H. Ma, I. King, M.R. Lyu, Effective missing data prediction for collaborative filtering, Proceedings of the 30th International ACM SIGIR Conference on, Information Retrieval, 2007, pp. 39–46.
[36] P. Baudisch, Joining collaborative and content-based filtering, Proceedings of the Conference on Human Factors in Computing Systems, 1999.

Mojtaba Salehi received his B.Sc. degree from Shahid Bahonar University in 2004, his M.Sc. degree from Tehran University in 2006, and his Ph.D. degree from Tarbiat Modares University in 2013. During 2012, he was a researcher in Eindhoven University of Technology in Netherlands, Mathematics and Computer Science Department. He is currently working as an Assistant Professor of K. N. Toosi University of Technology (KNUT). His research areas of interest include operation research, applied multivariate analysis, soft computing, data mining and recommender systems.