Knowledge-Based Systems 48 (2013) 57–69

Contents lists available at SciVerse ScienceDirect

Knowledge-Based Systems

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

Hybrid recommendation approach for learning material based on sequential pattern of the accessed material and the learner's preference tree

Mojtaba Salehi, Isa Nakhai Kamalabadi *

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

ARTICLE INFO

Article history:
Received 17 November 2012
Received in revised form 16 April 2013
Accepted 16 April 2013
Available online 25 April 2013

Keywords:
Personalized recommendation
Collaborative filtering
Sequential pattern mining
Learning material
E-learning
Association rules

ABSTRACT

The explosion of the learning materials in personal learning environments has caused difficulties to locate appropriate learning materials to learners. Personalized recommendations have been used to support the activities of learners in personal learning environments and this technology can deliver suitable learning materials to learners. In order to improve the quality of recommendations, this research considers the multidimensional attributes of material, rating of learners, and the order and sequential patterns of the learner's accessed material in a unified model. The proposed approach has two modules. In the sequential-based recommendation module, latent patterns of accessing materials are discovered and presented in two formats including the weighted association rules and the compact tree structure (called Pattern-tree). In the attribute-based module, after clustering the learners using latent patterns by *K*-means algorithm, the learner preference tree (LPT) is introduced to consider the multidimensional attributes of materials, rating of learners, and also order of the accessed materials. The mixed, weighted, and cascade hybrid methods are employed to generate the final combined recommendations. The experiments show that the proposed approach outperforms the previous algorithms in terms of precision, recall, and intra-list similarity measure. The main contributions are improvement of the recommendations' quality and alleviation of the sparsity problem by combining the contextual information, including order and sequential patterns of the accessed material, rating of learners, and the multidimensional attributes of materials.

1. Introduction

With the explosion of learning materials available on personal learning environments (PLEs), it is difficult for learners to discover the most appropriate materials according to keyword searching method. One way to address this challenge is the use of recommender systems [16]. In addition, up to very recent years, several researches have expressed the need for personalization in e-learning environments. In fact, one of the new forms of personalization in e-learning environments is to provide recommendations to learners to support and help them through the e-learning process [19].

According to the strategies applied, recommender systems can be segmented into three major categories: content-based, collaborative, and hybrid recommendation [1]. Hybrid recommendation mechanisms attempt to deal with some of the limitations and overcome the drawbacks of pure content-based approach and pure collaborative approach by combining the two approaches.

1.1. Motivation

The majority of the traditional recommendation algorithms have been developed for e-commerce applications, which are unable to cover the entire requirements of learning environments. One of these drawbacks is that they do not consider the learning process in their recommendation approach. Since the learning process concerns about repeatability and periodicity, some dependency relations can be found in histories of material access of each learner that might be able to present the material access patterns. This information can be used to reflect the learner's latent preferences. In order to find the latent patterns in accessing the materials, the weighted association rules and Pattern-tree structure have been used in the sequential-based module.

On the other hand, it is necessary to consider the attributes of learning materials such as subject, author, and publisher in order to provide a good recommendation in the learning environment. Therefore, in order to improve the quality and accuracy of recommendations in the learning environment, this research takes the multidimensional attributes of learning material into account in order to model multi-preferences of the learner in the attribute-based module.

* Corresponding author. Tel./fax: +98 2182884323.
E-mail address: nakhai@modares.ac.ir (I. Nakhai Kamalabadi).

0950-7051/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.knosys.2013.04.012

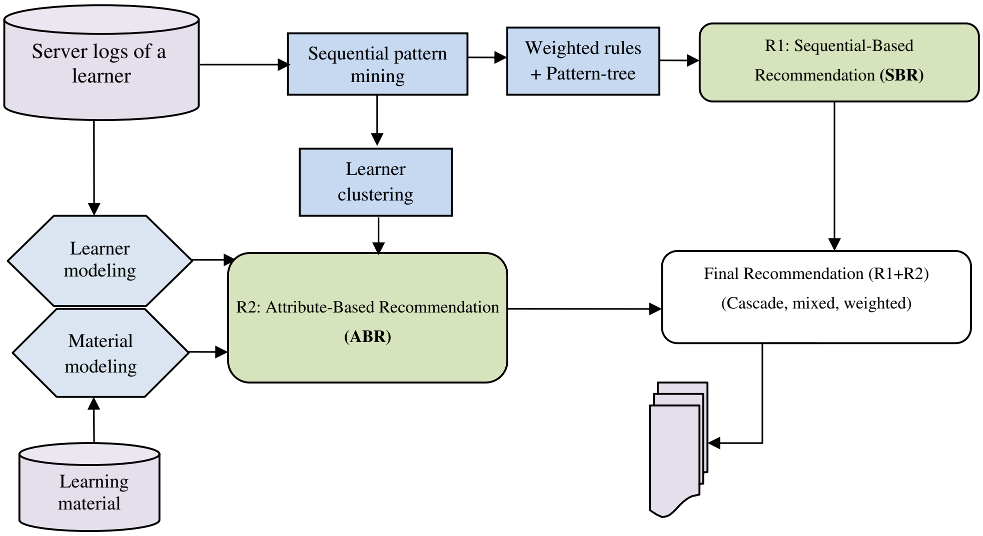

Server logs of a learner
Sequential pattern mining
Weighted rules + Pattern-tree
R1: Sequential-Based Recommendation (SBR)
Learner clustering
Learner modeling
Material modeling
R2: Attribute-Based Recommendation (ABR)
Final Recommendation (R1+R2)
(Cascade, mixed, weighted)
Learning material

Fig. 1. Overall system architecture of the proposed material recommender system for PLE.

Finally, it is an obvious fact that the learners' preferences are changing dynamically. Therefore, in order to make good recommendations in-time while the learners' current interests are changing, a recommendation approach is better to consider the order or time of the accessed material. This research applies the gradual forgetting function (GFF) to model the changes of the learners' interests in the attribute-based module.

1.2. Contribution

As a novel approach, which generates recommendations with higher quality and accuracy, and addresses the sparsity problem in the learning environment, this research combines latent patterns of the accessed materials in the sequential-based module. Moreover, it combines learner rating and multidimensional attributes and the accessed order of materials to model the dynamics and multi-preferences of the learner in the attribute-based module.

In the sequential-based recommendation module, in order to discover the latent patterns of the accessed materials, the modified Apriori and PrefixSpan algorithms are used to explore latent patterns in the shape of weighted association rules and Pattern-tree structure, respectively. Afterward, in the attribute-based recommendation module, through application of these patterns, the learners are clustered using k-means algorithm. Subsequently, in this module, in order to reflect the learner's complete spectrum of interests, the LPT is introduced to consider multidimensional attributes and order of the accessed materials and the learner's rating all at the same time. Truly, the LPT is built based on the accessed records and rating of the target learner and also the multidimensional attributes of materials. Then, a new similarity measure is introduced, which can consider the information of LPTs to calculate the similarity between learners. The results of both modules are combined using three hybrid methods.

2. Methodology

Fig. 1 shows the overall system architecture of the proposed material recommender system for personal learning environments. The sequential-based recommendation (SBR) and attribute-based recommendation (ABR) are applied to make recommendations. In SBR, the weighted rules and Pattern-tree, which indicate the sequential patterns of the learning materials, are generated using PrefixSpan and the modified Apriori, respectively. Then, in the ABR, the first learners are clustered based on the similarity between their accessed material sequences. As a result, the learner model and material model will be constructed. Materials are modeled according to their attributes. In order to model a learner, the server usage logs are collected in a certain period. After cleaning the original logs, we applied the learner modeling approach to build the LPT for each learner. Two recommendation sets of modules were combined by the cascade, mixed, and weighted approaches for final recommendation.

2.1. Sequential-based recommendation

The learning processes and material access sequences in the learning process usually have some time-dependency relationships that are repeatable and periodic. Therefore, the time-dependency relationship between the learning materials in a learning process can reflect a learner's preferences and the material access patterns. For example, the learning material access sequence in a special learning process is: {Lecture notes, Reference, Exercise book}. Therefore, when a learner has recently accessed the lecture notes, it is highly likely that he will access the reference in the near future. Therefore, we can mine the learner's access records to discover the material access sequential patterns. Afterward, using these sequential patterns, we can predict the most probable materials that a learner will probably access in the near future. This will further help to improve the recommendations' accuracy and solve the sparsity problem.

A sequential access pattern is a sequential pattern in a large set of pieces of access logs, which is pursued frequently by users. There are two main groups of algorithms for sequential pattern mining (SPM). The first algorithms are mainly based on the Apriori algorithm such as AprioriAll [2] and Generalized Sequence Pattern (GSP) [40]. These algorithms encounter this problem like almost all the Apriori-based algorithms. They require expensive multiple scans of the databases in order to determine that which candidates are actually frequent. The second group of algorithms that has been proposed to improve the efficiency of SPM is generally based on pattern growth, such as PrefixSpan [33]. The PrefixSpan algorithm has demonstrated a higher efficiency compared to the others [33]. In this research, we use two algorithms for sequential pattern mining including the modified Apriori algorithm to find the weighted association rules and PrefixSpan algorithm to find the Pattern-tree structure.

2.1.1. Weighted association rule mining

The association rule r is an expression of this form: $A \rightarrow B$, where A and B are two sets of items. A is the body and B is the head of the rule. The support for the association rule $A \rightarrow B$ is the percentage of transactions (sequences in this study) that contains both A and B in all transactions. The confidence of the rule $A \rightarrow B$ is the percentage of transactions that contains B among the transactions that contain A. The confidence can be computed as follows:

$$Confidence(A \rightarrow B) = \frac{\sup\ port\ (A \cup B)}{\sup port\ (A)} \tag{1}$$

Rule r must satisfy a minimum confidence and a minimum support threshold, in order to be involved in the discovered rules set.

The traditional model of association rules only considers that whether an item is present in a sequence or not. It is assumed that the entire items have the same importance and do not take the weight of an item within a sequence into account. In order to improve the efficiency of association rules, inspired by Tao et al. [42], to associate a weight parameter with each learning material in an association rule, we implemented a weighted association rules mining approach. The weight of a learning material in the sequence $s = \{M_1, M_2, \ldots, M_n\}$ is computed based on the visiting time-length of the material. The idea of using a weighted association rule mining is beneficial, because by considering the visiting time-length of the material in the rules mining, we can take into account the learner's interests and subsequently, improve the recommendation results. In the following of this section, we will modify the measures of Apriori algorithm [3] to reflect the weighting approach.

Definition 1 (*Item Weight*). The Item Weight represents the significance of an item. The weight scheme for material i in the sequence j is defined as follows:

$$W(M_i) = \frac{\frac{TotalTime(M_i)}{size(M_i)}}{\max\limits_{q \in s}\left(\frac{TotalTime(M_q)}{size(M_q)}\right)} \tag{2}$$

where s is a sequence in the database and M_q is a material in this sequence. Since the learner generally spends more time on a more useful learning material, this research uses the material visit duration as the weighting parameter. However, since a short visiting time might also occur due to the short length of a material, we normalized the time using the total bytes of the material.

Definition 2 (*Weight of an Itemset in a Sequence*). The weight of an itemset X in sequence s, denoted as $W(X, s)$, can be derived from the weights of its enclosing items. A simple way is to use the minimum weights of all the items in the itemset as the total weight of the itemset, as shown in the below equation.

$$W(X,s) = \begin{cases} \min\limits_{\forall i \in X} W(M_i) & X \subseteq s \\ 0 & X \not\subseteq s \end{cases} \tag{3}$$

Definition 3 (*Weighted Support*). The Weighted Support (WSP) is defined as the average of the itemset weight in all the sequences in the database, as follows:

$$WSP(X) = \frac{\sum_{s \in S} W(X,s)}{|S|} \tag{4}$$

where S is the set of all the sequences in the database.

Definition 4 (*Weighted Confidence*). The Weighted Confidence of the weighted association rule can be formulated as follows:

$$WC(X \rightarrow Y) = \frac{WSP(X \cup Y)}{WSP(X)} \tag{5}$$

The generated rules express the behavior characteristics of the user's learning process.

2.1.2. Rule-based recommendation

In order to obtain a recommendation set, based on the weighted association rules for an active learner, a sliding window w is used to control the number of materials to be matched against the association rules in the sequences. It is very important to maintain the historical depth in order to be able to provide reasonable suggestions.

To produce recommendations, we must search the left-hand side of the weighted association rules, which are most similar to the accessed material sequence of an active user, using a similarity degree. We used a similarity measure to find the most similar rules instead of the exact matches between the active learner and rules. The weighted association rules and the accessed material sequence of an active learner are presented as a set of material-weight pairs. This allows us to consider both the sequence of an active learner and the association rules as m-dimensional vectors over the space of material. Thus, the left-hand side of each rule (r_L) can be represented by a vector: $r_L = \{w_1(r_L), w_2(r_L), \ldots, w_m(r_L)\}$, where

$$w_i(r_L) = \begin{cases} \overline{W}(M_i), & if\ M_i \in r_L \\ 0 & otherwise \end{cases} \tag{6}$$

where $\overline{W}(M_i)$ is considered as the mean of the Item Weight defined by Eq. (2) on the sequences, which has the left-hand side of the rule. An active user sequence is also represented by a vector $s_a = \{w_1(s_a), w_2(s_a), \ldots, w_m(s_a)\}$, where $w_i(s_a)$ is the significance weight associated with material M_i and is defined using Eq. (2) for the accessed sequence of the user, If the user has accessed M_i in his/her sliding window (w), and $w_i(s_a) = 0$, otherwise.

Then, inspired by cosine similarity, the matching score is defined as:

$$Match\,score\,(s_a, r_L) = \frac{\sum_{i=1}^{m} w_i(s_a) w_i(r_L)}{\sqrt{\sum_{i=1}^{m} w_i(s_a)^2} \sqrt{\sum_{i=1}^{m} w_i(r_L)^2}} \tag{7}$$

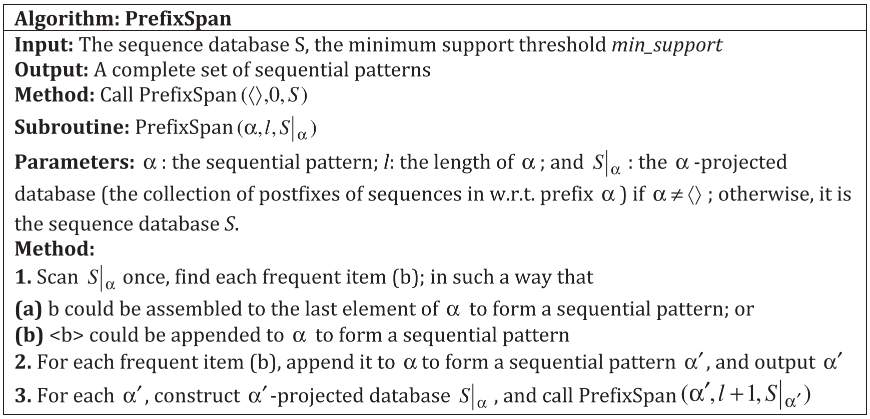
s_a and r_L represent the active learner sequence and the left-hand side of the weighted association rule, respectively. By using this measure, the algorithm finds the rules that are similar to the active user sequence. Finally, a recommendation score can be calculated for each unvisited material by the active learner. In this research, three factors are applied in calculation of the recommendation score: the matching score, the weighted confidence, and the support of the rule:

$$\begin{aligned} \mathrm{Rec}.Score(s_a, r_L \rightarrow M_i) = {} & MatchScore(s_a, r_L) \\ & \times WC(r_L \rightarrow M_i) \times WSP(r_L) \end{aligned} \tag{8}$$

Ultimately, N materials with the highest recommendation scores, which have not already been visited by the active learner, are chosen as the recommendation set. By using the matching measure between the weighted rules and current sequence (instead of the exact match between them) and also by using both weighted confidence and support (instead of just the confidence value), this approach will give more accurate results.

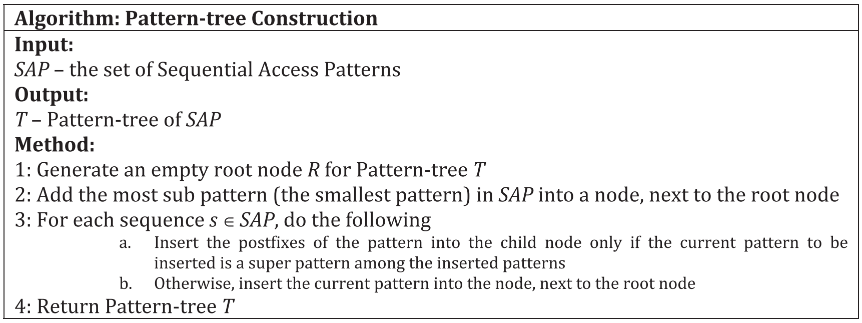
2.1.3. Construction of Pattern-tree

In another method of sequential pattern mining, we use PrefixSpan to find the sequences and make the Pattern-tree. General



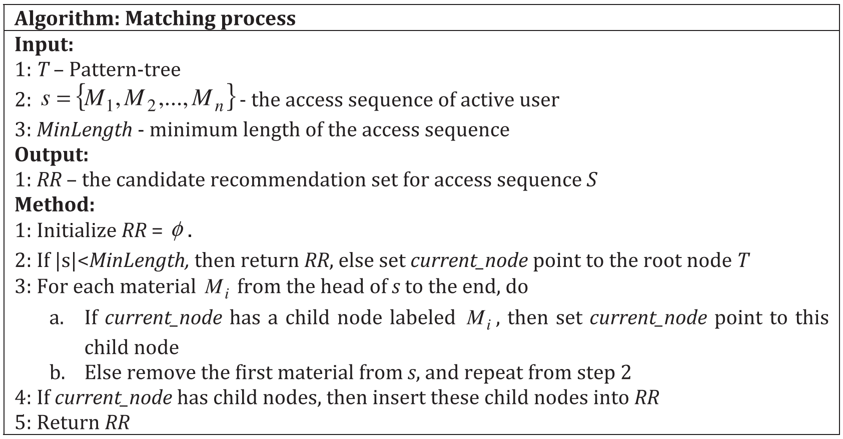
```
Algorithm: PrefixSpan
Input: The sequence database S, the minimum support threshold min_support
Output: A complete set of sequential patterns
Method: Call PrefixSpan ($\langle\rangle, 0, S$)
Subroutine: PrefixSpan ($\alpha, l, S|_{\alpha}$)
Parameters: $\alpha$ : the sequential pattern; l: the length of $\alpha$ ; and $S|_{\alpha}$ : the $\alpha$ -projected
database (the collection of postfixes of sequences in w.r.t. prefix $\alpha$ ) if $\alpha \neq \langle\rangle$ ; otherwise, it is
the sequence database S.
Method:
1. Scan $S|_{\alpha}$ once, find each frequent item (b); in such a way that
(a) b could be assembled to the last element of $\alpha$ to form a sequential pattern; or
(b) <b> could be appended to $\alpha$ to form a sequential pattern
2. For each frequent item (b), append it to $\alpha$ to form a sequential pattern $\alpha'$, and output $\alpha'$
3. For each $\alpha'$, construct $\alpha'$-projected database $S|_{\alpha}$ , and call PrefixSpan ($\alpha', l+1, S|_{\alpha'}$)
```

Fig. 2. PrefixSpan algorithm to find the sequential pattern of materials.



```
Algorithm: Pattern-tree Construction
Input:
SAP – the set of Sequential Access Patterns
Output:
T – Pattern-tree of SAP
Method:
1: Generate an empty root node R for Pattern-tree T
2: Add the most sub pattern (the smallest pattern) in SAP into a node, next to the root node
3: For each sequence $s \in SAP$, do the following
        a. Insert the postfixes of the pattern into the child node only if the current pattern to be
           inserted is a super pattern among the inserted patterns
        b. Otherwise, insert the current pattern into the node, next to the root node
4: Return Pattern-tree T
```

Fig. 3. The algorithm for pattern-tree construction.



```
Algorithm: Matching process
Input:
1: T – Pattern-tree
2: $s = \{M_1, M_2, \ldots, M_n\}$ - the access sequence of active user
3: MinLength - minimum length of the access sequence
Output:
1: RR – the candidate recommendation set for access sequence S
Method:
1: Initialize RR = $\phi$ .
2: If |s|<MinLength, then return RR, else set current_node point to the root node T
3: For each material $M_i$ from the head of s to the end, do
    a. If current_node has a child node labeled $M_i$ , then set current_node point to this
       child node
    b. Else remove the first material from s, and repeat from step 2
4: If current_node has child nodes, then insert these child nodes into RR
5: Return RR
```

Fig. 4. The matching process of the pattern-tree and the users' access sequence.

steps of PrefixSpan are provided: (1) Find length-1 sequential patterns. (2) Divide the search space according to the prefixes. (3) Find the subsets of sequential patterns. (4) Execute step 3 recursively until there would be no more sequential pattern to be discovered. More detailed steps were presented in Fig. 2.

A Pattern-tree model is employed to store sequential access patterns compactly. The pattern tree is constructed using the procedure defined in [30]. They utilized a patricia-based data structure for webpage recommendations, due to its advantages over the trie structure, which had been used previously. To construct a Pattern-tree, we only need to have one scan of all the sequential access patterns. Fig. 3 gives the method for constructing the pattern tree.

2.1.4. Pattern-tree based recommendation

To generate recommendations, we must search for the best matching access path in the Pattern-tree according to the access sequence of the active user. To compute a recommendation set for the current (active) learner sequences, a sliding window (w) is used to control the number of materials to be matched against the discovered sequences. The matching process is implemented on the latest w accessed materials of the target learner as the prefix sequence. It must be noted that w, which is considered as the maximum length of the access sequence for an active learner, should be less than the depth of the Pattern-tree. On the other hand, since the recommendations generated from shorter matching paths usually

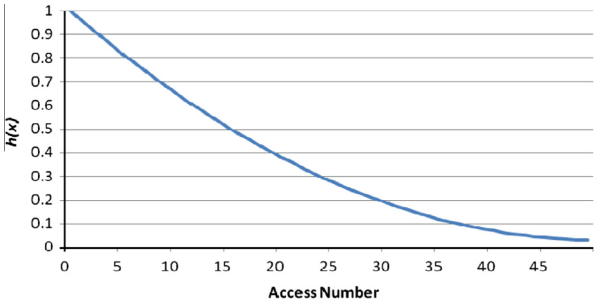


Fig. 5. Gradual forgetting function with $\lambda = 0.95$.

have lower accuracy, only the material access sequence can be processed that its length is not less than a given threshold. In this research, the minimum length of access sequence is set to 2. The matching process between Pattern-tree and the users' access sequence are presented in Fig. 4.

The result of matching process is the candidate recommendation set for access sequence of the active learner. To generate recommendations at this stage, we use the Markov model described in (Faten et al., 2008), which was employed to identify the next page to be accessed by the website user. It is performed based on the sequence of the previously accessed pages of the user.

Every sequence has a support degree that is given in the node of the constructed tree. Using these support degrees, the probability is computed to find the most important sequence for an active learner. The following equation provides the probability of sequence selection for each sequence, in which $M^{\alpha}_{n+1} \in RR$ is a member of the sequence.

$$P^{\alpha}\left(s \in \left\{M_1, M_2, \ldots, M_n, M^{\alpha}_{n+1}\right\}, \ldots M^{\alpha} \middle| \{M_1, M_2, \ldots, M_n\}\right) = \frac{SD(\{M_1, M_2, \ldots, M_n\})}{SD(M_1, M_2, \ldots, M_n, M^{\alpha}_{n+1}, \ldots M^{\alpha})} \tag{9}$$

Using the defined probability, we can predict the best material for the active learner, as follows:

$$M^{*}_{n+1} = \arg\max P^{\alpha}\left(s \in \left\{M_1, M_2, \ldots, M_n, M^{\alpha}_{n+1}, \ldots M^{\alpha}\right\} \middle| \{M_1, M_2, \ldots, M_n\}\right) \tag{10}$$

2.2. Attribute-based recommendation

To improve the scalability of the recommendation process, we firstly cluster the learners according to the sequences of the accessed material by learners. Each cluster contains the learners with similar behaviors and interests, and therefore, similar sequential patterns. The clustering approach can increase the scalability of recommendations that is an important issue in recommender systems. In addition, clustering can personalize the recommendations and improve the accuracy of recommendations. This research uses K-means algorithm, proposed by MacQueen [27], which is one of the simplest unsupervised clustering methods. To implement this algorithm, each learner is represented by her/his sequence weight vector in the material space $s_a = \{w_1(s_a), w_2(s_a), \ldots, w_m(s_a)\}$, as defined in Section 2.1.2. We used the Euclidean distance as a measure to find clusters. Number of clusters is determined based on the within-cluster sums measure. After clustering, we can implement the attribute-based recommendation approach in each cluster.

2.2.1. Material modeling

We can consider some attributes for learning materials such as subject, which is like literature, mathematics, and computer science. In addition, since the ratings of the learner's accessed materials, which have certain attribute values, indicate the importance of these attribute values for the learner, they can be considered as the base for learner modeling. Therefore, in order to consider the learner's preferences accurately, the attributes of learning materials should be taken into account as much as possible. The material attributes' description model can be defined as a vector $M = \langle (A_1, AW_1), (A_2, AW_2), \ldots, (A_m, AW_m) \rangle$, where A_i denotes the ith dimensional attribute's name of material and AW_i denotes the appropriate weight value for the ith attribute, and $AW_1 \geqslant AW_2 \geqslant \cdots \geqslant AW_m$, $\sum_{i=1}^{m} AW_i = 1$. Selection of appropriate attributes may vary in various systems. The system developer must use suitable attributes. In this research, according to simplicity and usefulness, we selected four attributes, including the primary subject, secondary subject, education level, and publisher of material. Based on this description model, the attributes of a certain material M_j can be defined as $M_j = (AK_1, AK_2, \ldots, AW_m)$, where AK_i denotes the ith dimension attribute's keyword of material M_j. The attribute's keywords are determined by the experts, when a material is registered in the system for the first of time. An example has been provided below: $M_i = [(Mathematic, 0.35), (\Pr obability, 0.3), (Bachelor\ \deg\ ree, 0.2), (UT, 0.15)]$.

2.2.2. Learner modeling

In addition to multidimensional attributes of the material, two types of information are used for learner modeling. (1) Rating is

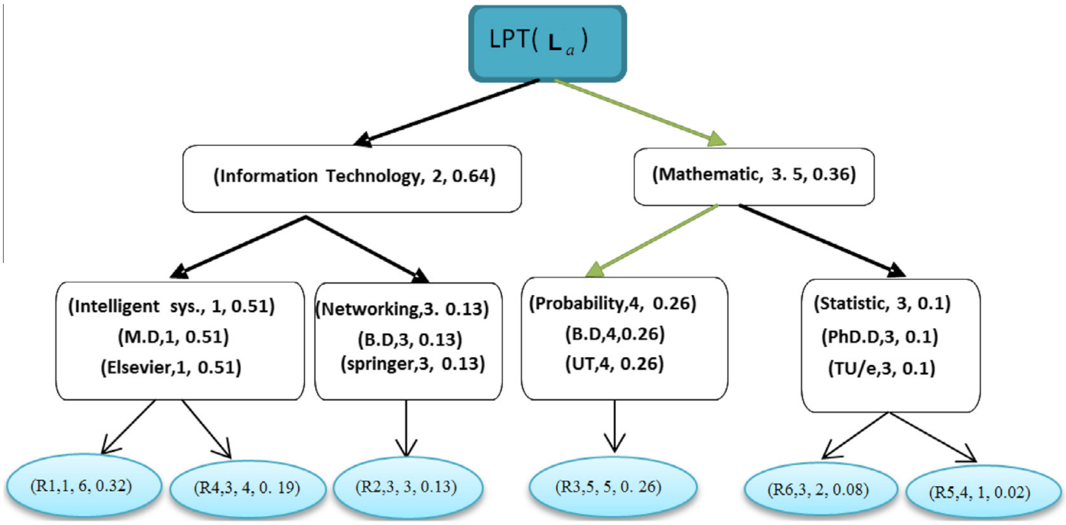


Fig. 6. Learner preference tree based on a four-dimensional attributes description model.

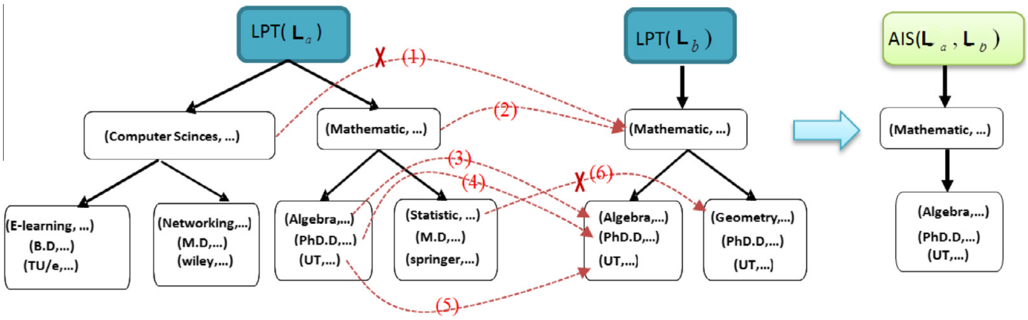


Fig. 7. A sub-tree sample of attributes intersection.

used to model the learner's preferences. (2) Access order of materials is used to model the changes in the learner's interests.

Interest change modeling: In this section, we use the order of the accessed materials as the useful information for interest change modeling. Usually in learning environments, the preferences of a learner may change and the history records are unable to entirely reflect the whole preferences of a learner. Thus, by changing the learner's interests and preferences with the passage of time, the recommender system can no longer produce accurate recommendations. On the other hand, the preferences of the learner's recently accessed materials have an important influence on the future interests. Inspired from Chuan-Chang [10], the gradual forgetting function (GFF) concept is introduced in order to reflect the dynamic interests and preferences of a learner more accurately. We used the following GFF:

$$h(x(M_j)) = 1 - \lambda\left(1 - \left(\frac{x(M_j)-x_{\min}}{x_{\max}-x_{\min}}\right)^2\right) \tag{11}$$

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

where $x(M_j)$ is the order of M_j in the accessed sequence of learner L_i. $x_{\max}$ and $x_{\min}$ are the orders of the latest and first accessed material by L_i, respectively. Therefore, the effect of M_j on L_i's future interest will become smaller during the on-going material access process and $h(M_j)$ will attenuate gradually. In $h(M_j)$, λ is an adjustable parameter used to describe the change rate of the learner's preferences, and the bigger the λ, the faster the forgetting will be.

By using the GFF as a coefficient, we implemented three important rules: (1) Since the materials, which are accessed recently have a larger h, the recommended materials are similar to the recent accessed materials. (2) Since the h is a nonlinear function, the recently accessed material does not have a large value for a long time and this avoids the effect of an occasional material access on the long-term access. (3) In addition, the nonlinear function (h) supports the previous long-term preferences of the learner.

The h attenuation with $\lambda = 0.95$ is shown in Fig. 5. Based on Eq. (11), the h value of the latest accessed material is equal to 1, and with ongoing access, the h value of materials can be updated.

Learner modeling: In order to model learners, inspired from Gu et al. [17], the LPT is introduced, which combines the multidimensional attributes of the learner's accessed materials and the learner's rating information to model multi-preferences of the learner. In addition, the *LPT* uses h value of the learner's accessed materials to model the dynamic interest of the learner.

LPT is defined as a tree with $(m + 1)$ levels, where m indicates the number of attributes for materials. In this tree, the leaf node that represents an accessed material of L_i is defined as a four-tuple: $T_{leaf} = \{MID, MR, x(MID), NH\}$, where MID indicates the accessed material ID by learner L_i, $x(MID)$ indicates the order of material ID in the access order of learner L_i, NH indicates the normalization value of h function for material ID, and MR indicates rating of L_i for the material. The non-leaf node can be defined as a four-tuple: $LT_{nonleaf} = \{KA, MR, NH\}$, where KA is the keyword of the level-th attribute of the material. A four-dimension attributes description model based on LPT is considered in this research. Attributes include the primary subject, secondary subject, publisher, and author. A sample is shown in Fig. 6.

To have a multidimensional learner model, we define:

(1) The NH value of the ith $(0 < i < m + 1)$ level node is defined as the sum of the values of its entire immediate successors, which are placed at the $(i + 1)$th level.
(2) The MR of the non-leaf node k is defined as the mean of MR values of all the leaf nodes that belong to the k's sub-tree.

In this tree, each accessed material 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 M_j's attributes. The green route in Fig. 6 indicates the unique path for the material $M_i = [(Mathematic, 0.35), (\Pr obability, 0.3), (Bachelor\ \deg\ ree, 0.2), (UT, 0.15)]$.

As shown in Fig. 6, at first, the learner prefers to use mathematic materials and then, she/he changes his interests and uses the information technology materials. Our modeling considers this interest change through allocating $NH = 0.64$ to the information technology materials and $NH = 0.36$ to the mathematic materials.

Update strategy of LPT: The LPT must be constructed for each learner and updated according to the following strategy:

Search the keywords of the latest accessed material attributes in *LPT*, from the upper to the lower levels. If the keyword of the ith attribute could not be matched, the $m - i + 1$ new level with latter $m - i + 1$ attribute keywords of the material will be created in a new route and, *NH and MR* will be updated in the entire *LPT*.

2.2.3. Attribute-based recommendation

Most researches have used CF approach to generate recommendations. However, there are several drawbacks in the traditional rating data based on the similarity calculation method: (1) The

Table 1
The characteristics of the dataset used in the experiments.

Number of users	Number of items	Number of transactions	Density (%)	Average person's records
1980	2931	40445	0.70195	20.4267

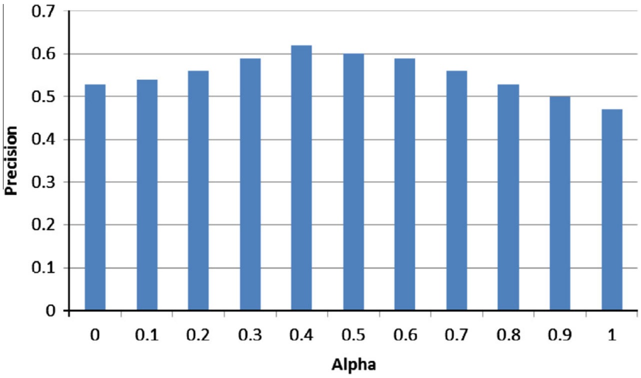


Fig. 8. The precision of ABR recommendation with respect to α.

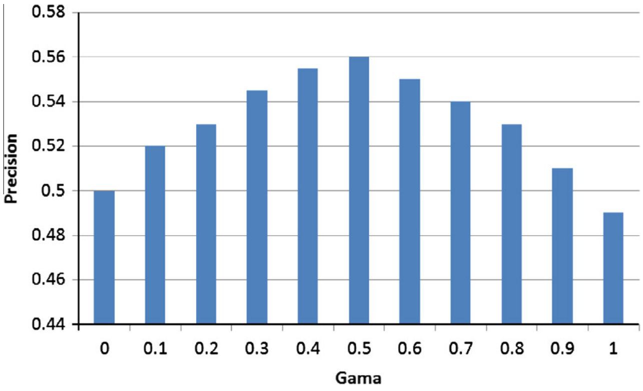


Fig. 9. The precision of SBR with respect to δ.

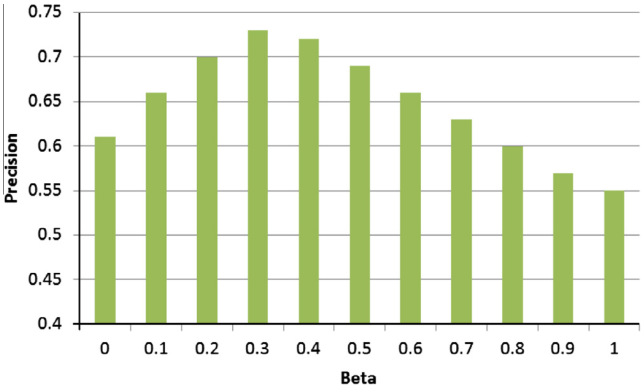


Fig. 10. The precision of the W-SBR–ABR recommendation with respect to β.

Table 2
A comparison of the classification accuracy of various methods.

System	Recall	Precision	F1
SBR	0.381	0.547	0.449
ABR	0.395	0.6183	0.482
M-SBR–ABR	0.493	0.639	0.557
W-SBR–ABR	0.513	0.733	0.604
C-SBR–CB	0.562	0.752	0.643
User-based with DV	0.334	0.601	0.429
Item-based	0.293	0.591	0.392
Pazzani	0.501	0.674	0.575
Melville et al.	0.504	0.689	0.582
Personality diagnosis	0.36	0.668	0.468

CF's dependence on the learner ratings can be regarded as a drawback. Because, it is essential that several learners evaluate each learning material. As a result, the new materials could not be recommended until some learners spend time to evaluate them. This problem refers to the 'data sparsity' and 'cold start problem'. (2) Most of the existing algorithms only consider the learner-material rating matrix, and do not use the context information of the learner and the attributes of material. Thus, it will lead to inaccurate results in the similarity calculation.

To address these drawbacks, a new similarity measure is introduced that can take into account the rating information of a learner, the multidimensional attributes of the accessed materials, and the dynamic preferences information based on the LPT, altogether. As a logical assumption, two learners with similar attribute keywords in their LPTs can be considered as similar neighbors. Therefore, in order to define the similarity degree, three rules are considered:

(1) The more similar the attributes of the accessed materials by learner L_a and learner L_b are, the larger the similarity between them will be.
(2) The more similar the order of the accessed materials of learner L_a and learnerL_b are, the larger the similarity between them will be.
(3) The more similar the rating data of learner L_a and learner L_b are, the larger the similarity between them will be.

The similarity degree between two learners can be calculated based on Attributes Intersection Sub-tree (*AIS*) between their corresponding *LPTs*. The *AIS* between learner L_a and L_b, $AIS(L_a, L_b)$, is defined as the maximum intersection between levels of LPT_a and LPT_b with the same keyword in each level. After the matching process, we have an *AIS*, represented in Fig. 7.

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

$$sim_A(L_a, L_b) = \frac{\sum_{i \in AIS(L_a, L_b)} MW_i \cdot NH_{ai} \cdot NH_{bi}}{\sqrt{\sum_{i \in LPT(L_a)} MW_i \cdot NH_{ai}^2} \cdot \sqrt{\sum_{i \in LPT(L_b)} MW_i \cdot NH_{bi}^2}} \quad (12)$$

where NH_{ai} indicates the value of *NH* in the *i*th level of the matching for learner *a*. MW_i indicates the *i*th level's matching weight of the *LPT*. Since MW_i should rise with increasing the depth (number of levels in *AIS*), in the current paper, it is defined as $MW_i = AW_i^{-1}$. This similarity measure takes into account the number of similar attribute's keywords and the similarity between *NH* values of similar keywords.

In order to reflect the similarity between the rating vectors of two learners, inspired by Pearson similarity, the learner rating based on similarity $sim_R(L_a, L_b)$, can be applied as below:

$$sim_R(L_a, L_b) = \frac{\sum_{i \in AIS} |(MR_{ai.level} - \overline{MR_a}) \cdot (MR_{bi.level} - \overline{MR_b})|}{\sqrt{\sum_{i \in AIS} (MR_{ai.level} - \overline{MR_a})^2} \cdot \sqrt{\sum_{i \in AIS} (MR_{ai.level} - \overline{MR_a})^2}} \quad (13)$$

where $MR_{ai.level}$ indicates the rating of user L_a in row *i* of LPT_a, corresponding with the row *i* of *AIS*. $\overline{MR_a}$ and $\overline{MR_b}$ indicate the mean values of the L_a and L_b's rating data, respectively. In calculation of $sim_R(L_a, L_b)$, the similarity between the *MR* values of the rows on LPT_a and LPT_b that correspond to each row on $AIS(L_a, L_b)$ is computed. It is not required to have the identical accessed materials

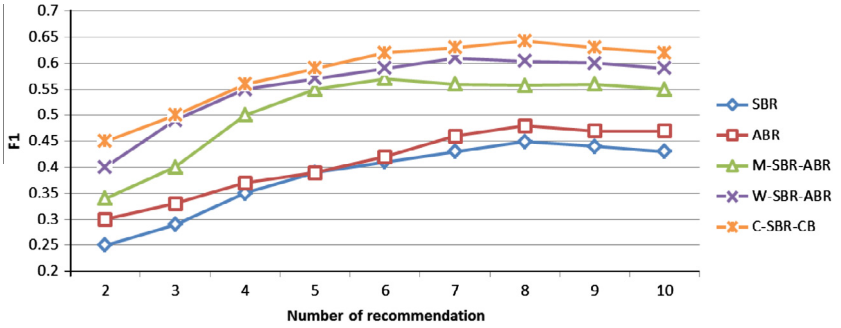


Fig. 11. The precision of the proposed recommendation methods with respect to the number of recommendations.

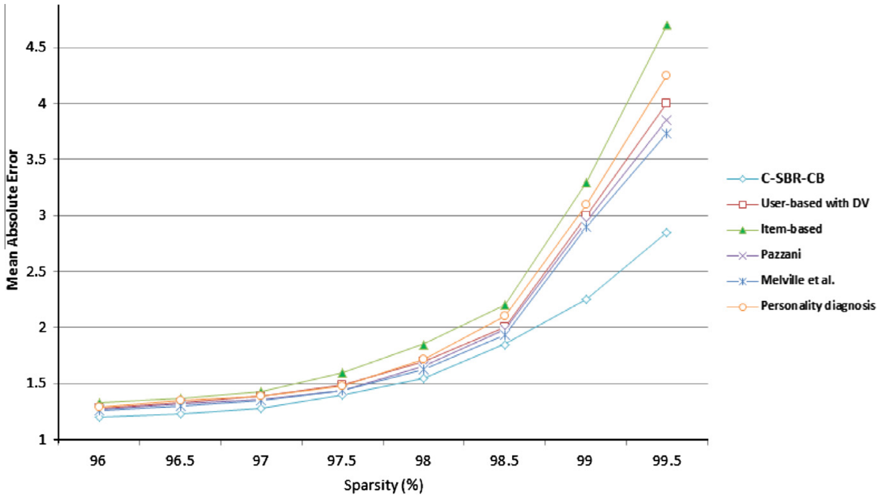


Fig. 12. The performance of algorithms under different sparsity levels.

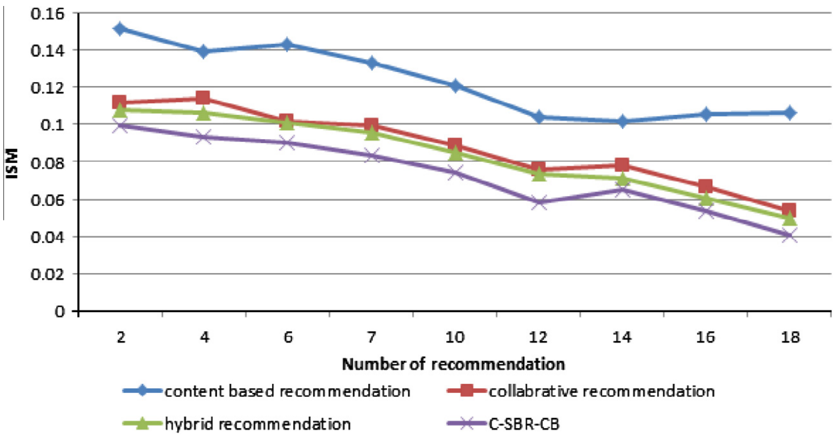


Fig. 13. The ISM of algorithms with respect to *N*.

between the two learners. Using this definition of similarity, we can overcome the sparsity rating problem. In addition, in order to decrease the deviation of the learner's different rating scales, the learner's rating data was modified. Now, we can calculate the similarity between L_a and L_b as follows:

$$Sim(L_a,L_b) = \alpha \cdot sim_R(L_a,L_b) + (1-\alpha)\cdot sim_A(L_a,L_b) \quad 0 \leqslant \alpha \leqslant 1 \quad (14)$$

In which, α indicates the weight between $sim_R(L_a, L_b)$ and $sim_A(L_a, L_b)$. Finally, the recommendation score of material M_j for learner L_i can be calculated via Eq. (15):

$$\mathrm{Rec}.Score(L_i, M_j) = \left(\overline{MR}_i + \frac{\sum_{q\in L_{M_j}} Sim(L_i,L_q)\cdot(MR_{qj} - \overline{MR}_q)}{\sum_{q\in L_{M_j}} Sim(L_i,L_q)}\right) \quad (15)$$

$$L_{M_j} = \{q \in L \mid MR_{qj} \neq \Phi\}$$

where MR_{qj} is the rate of learner L_q for material M_j. L_{M_j} indicates the learners that have rated M_j. $\overline{MR_i}$ indicates the mean value of the L_i's rating data. The final step in this phase is to eventually

Table 3
An overview of the recommendation strategies (input, output, process, and some of researches).

Strategies	Input	Output	Process	Some of researches
Data mining	Access the history of learners	Materials, related links, learning activities	Discover suitable rules for recommendation using the sequential pattern mining	Romero et al. [35] Khribi et al. [19] García et al. [14] Wang and Liao [43] Lucas et al. [24]
Collaborative filtering	Access the history and rating of learners	Materials related links, learning activities	Identify the learners with similar rating or behavior and extrapolate from their ratings	Lemire et al. [21] Bobadilla et al. [5] Milicevic et al. [29] Bobadilla et al. [6] Bobadilla et al. [7] Salehi and Kmalabadi [36]
Content-based filtering	The attributes of material and learner	Materials, related links, learning activities	Generate a classifier that fits the learner's rating behavior	Liang et al. [22] Khribi et al. [19]
Hybrid approach	Access the history and rating of learners, The attributes of material and learner	Materials, related links, learning activities	Combine the learner's ratings and the attributes of material and learner, in order to predict the learner's rating	Tai et al. [41] Zheng and Li [44] Klašnja-Milićević et al. [20] Chen et al. [9]

derive the top-N recommendations. For an active learner, we produce a recommendation set of N materials according to the higher recommendation scores. It must be noted that the previously selected materials are excluded from the recommendation set.

2.3. Final recommendation

In order to improve the accuracy of recommendations in this research, at first, the results of the rule-based recommendations and the pattern-tree based recommendations are combined, as shown in the below:

$$\begin{aligned} Score_{Sequential-based}(L_a, M_i) = \delta \cdot Nor(Score_{Rule-based}(L_a, M_i)) \\ + (1-\delta) \cdot Nor(Score_{Pattern-tree-based}(L_a, M_i)) \end{aligned} \tag{16}$$

where $Nor(x)$ is a normalization function. Then, two recommendation sets including SBR (sequential-based recommendation) and ABR (attribute-based recommendation) are combined through the following hybrid recommendation approaches:

1. Mixed of SBR and ABR (M-SBR–ABR): This hybrid method is based on the merging and presentation of two ranked lists into one. ABR and FBR are components of this method. Each component of this hybrid produces recommendation lists with ranks and the core algorithm of mixed hybrid merges them into a single ranked list. The new rank scores are produced simply by adding each rank score.
2. Weighted of SBR and ABR (W-SBR–ABR): A linear combination of SBR and ABR is used for recommendations. By normalizing the recommendation score for each method, we have:

$$\begin{aligned} FinalScore(L_a, M_i) = \beta.Nor(Score_{Sequential-based}(L_a, M_i)) \\ + (1-\beta).Nor(Score_{Attribute-based}(L_a, M_i)) \end{aligned} \tag{17}$$

3. Cascade of SBR and ABR (C-SBR–ABR): The recommendation results of SBR are ranked using ABR method. In other words, we firstly produce recommendations using SBR method, and then, we rank them using the recommendation score of ABR method.

Each of methods, SBR and ABR uses the special information for recommendation and if one of SBR and ABR fails in the combination process, we lose some information in the recommendation process. In this situation, we can use the other methods for combination process to compensate the lost information. For example if ABR fails, we can combine SBR with the other methods that use rating information such as item based CF or Matrix factorization based methods and if SBR fails, we can combine ABR with the other methods that use sequential patterns information such as association rules based recommendation approaches.

3. Implication

In this section, a set of experiments has been conducted on a set of parameters and the effectiveness of our proposed recommender system has been examined in terms of the recommendations' accuracy and quality.

3.1. The experiment environment and data set

A real-world dataset is applied in our experiments. The dataset of learning records comes from the usage data of the course management system, Moodle. MOODLE (Modular Object-Oriented Dynamic Learning Environment) is defined as a course management system (CMS), a free and open source software package designed using pedagogical principles, to help educators by creating effective online learning communities. Moodle stores the detailed records of students' activities. The educator can access the summarized reports about these activities according to the categories specified by the Moodle system.

Moodle has a number of interactive learning activity modules, such as forums, chats, quizzes, and assignments. In addition, through Moodle, one can register and track a user's accesses (user identification, IP, and time) and the activities and materials that have been accessed by the user. In each record, the system stores the time of the access, the IP address of the user's computer, the user identification, and types of activities or the materials that have been visited. By this log, Moodle is able to generate the activity reports with charts and details, for each student about the times that a material has been visited. Therefore, each record includes (1) IP address, (2) Date and times of the access, (3) Complete name of the user who has accessed, (4) Type of the access (material view, course view, etc.), and (5) The specific element of the course, which has been visited. This information is used to make and update the leaner tree in every cluster of the learners.

The used dataset contains 40445 lending records by 1980 users on 2931 materials, where each record contains timestamp information. In addition, it contains the material information including the primary subject, secondary subject, publisher, author, and also the users' basic information such as log files data. Using this information, we can build vector of materials and also the learner's preference tree. This information is applied for material and learner modeling. In experiments, the dataset is ordered by the users' access timestamp, and then, is divided into the training set and test set. The characteristics of this dataset are summarized in Table 1. In this table, the density is obtained from dividing the number of transactions by the number of cells in the rating matrix. The average record by a person is the mean value of the transactions by each user.

3.2. Evaluation metrics

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

The decision support accuracy metrics assume the prediction process as a binary operation, in which items are predicted as either good or not bad. The Precision and Recall are the popular and well-established metrics in this category. The Precision and Recall are defined as the followings:

$$\text{Precision} = \frac{\sum_{i=1}^{p} |TS(U_i) \cap \text{Rec}(U_i)|}{\sum_{i=1}^{p} |\text{Rec}(U_i)|} \tag{18}$$

$$\text{Recall} = \frac{\sum_{i=1}^{p} |TS(U_i) \cap \text{Rec}(U_i)|}{\sum_{i=1}^{p} |TS(U_i)|} \tag{19}$$

where p is the size of user set. $\text{Rec}(U_i)$ denotes the recommendation set of user U_i. TS denotes the test set or the relevant items set (that must be recommended) and $|TS(U_i) \cap \text{Rec}(U_i)|$ indicates the hits number of the U_i's recommendation set. The test data set or relevance set is the real tracking of the users in the sequential-based approach. But in the attribute-based approach, we consider a threshold for the predicted rating to determine the test set. This threshold is set to 3.5. It means that if an item is rated 3.5 or higher, it will be accepted by the user.

Since increasing the size of the recommendation set leads to an increase in recall and at the same time a decrease in precision, we can use measure F_1 [39], which is a well-known combination metric, with the following formula:

$$F_1 = 2.\frac{\text{Precision.Recall}}{\text{Precision} + \text{Recall}} \tag{20}$$

The other metrics that we use are predictive accuracy metrics. In this category, Mean Absolute Error (MAE) is used with the following formula:

$$MAE = \frac{\sum_{i=1}^{i=T} |R^i - P^i|}{T} \tag{21}$$

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

Users interact with the recommendation list, and the accuracy metrics are unable to see this problem, because they have been designed to judge the accuracy of individual item predictions; in fact, they do not judge the contents of the entire recommendation lists. Since the recommendation list should be judged for its usefulness as a complete entity, not just as a collection of individual items, we also define in this study an Intra-List Similarity Metric inspired from Herlocker et al. [18], which is provided in the below:

$$ISM(List) = \frac{\sum_{M_i \in List} \sum_{M_j \in List, i \neq j} f(M_i, M_j)}{\binom{|List|}{2}} \tag{22}$$

where

$$f(M_i, M_j) = \frac{mat(M_i, M_j)}{m} \tag{23}$$

where *mat* indicates the number of matching attributes' keywords between materials M_i and M_j. As expressed before, m is number of the considered attributes for the material. Higher similarity denotes lower diversity. This measure is applied to evaluate the quality of recommendations.

3.3. Performance evaluation

In this section, the impact of input parameters is firstly analyzed on the recommendation performance. Then, to evaluate the proposed algorithm, it will be compared with five important algorithms from the literature.

3.3.1. Impact of parameters

For rule-based recommendations, important parameters are the minimum support and minimum confidence threshold, and also the sliding window size. In material recommendation, it is important to show the learners only the good materials (a small number of rules with a higher value of support and confidence). Therefore, we must select an appropriate minimum support and minimum confidence threshold. According to the experimental results, we can see that in order not to obtain numerous rules, a good minimum support and minimum confidence threshold is 0.25 and 0.5, respectively. To consider the impact of the window size on the proposed recommendation algorithm, we changed the window sizes from 1 to 10. A large sliding window provides more information to the system, while on the other hand, makes a larger state space with the sequences that occur less frequently in the dataset. A window of the size 1 cannot hold enough information for the recommendation. Therefore, the accuracy improves with increasing the window size. But the difference in accuracy between the window size 5 and 6 is not very much. It is because the states contain the material visits' sequences (study), occur less frequently in the dataset. Therefore, we consider $w = 5$.

For the Pattern tree-based recommendations, important parameters are the minimum support and also sliding window size. In this approach, we also consider $w = 5$ and $min\_support = 0.25$.

For the attribute-based recommendation approach, important parameters are the number of clusters (C), λ and α. According to the within-cluster sums measure, we have $C = 8$ for our dataset. To determine the best values for λ and α, we run an experiment that the results for α are shown in Fig. 8. According to the experimental results, we set $\lambda = 0.65$ and $\alpha = 0.8$. Fig. 8 also shows the impacts of α on the precision of ABR recommendation while $C = 8$. It indicates that using the attributes information of materials and learners will lead to the better recommendation performance, and the best precision can be obtained with $\alpha = 0.8$. The optimal value depends on the dataset and it must be adjusted for different datasets.

Final parameters are δ and β that are used to combine the recommendation results. Fig. 9 exhibits the impacts of δ on the precision of the SBR recommendation. According to the results, we have $\delta = 0.5$. Fig. 10 illustrates the impacts of β on the precision of the W-SBR–ABR recommendation while $C = 8$. It indicates that a weighted combination of SBR and ABR can improve the recommendation performance, and the best precision can be obtained with $\beta = 0.6$.

In order to compare the relative performance of the SBR, ABR, M-SBR–ABR, W-SBR–ABR, and C-SBR–ABR methods, an experiment is performed. The entire parameters were set and then, these methods were applied on the data. This comparison is based on the number of recommendations (NR) for measure F_1, which is presented in Fig. 11. As Fig. 11 shows, the three proposed combination methods have presented better performances. The relative performance of these methods for different numbers of recommendations is different; however, in general, we can say the best performance is from 7 to 10.

3.3.2. Comparative study

In order to evaluate the effectiveness of our proposed approach, a comparative study was implemented. Table 2 demonstrates a comparative study for classification accuracy between the SBR, ABR, M-SBR–ABR, W-SBR–ABR, and C-SBR–ABR methods and the optimal states of five different algorithms, including the user-based CF using Pearson correlation with default voting (DV) [8], the item-based CF using the adjusted cosine similarity [38], two hybrid recommendation algorithms used by Pazzani [32] and Melville et al. [28], the personality diagnosis algorithm [34] for making probabilistic recommendations. As can be seen, the W-SBR–ABR, and C-SBR–ABR methods generate better recommendations compared with the other algorithms. Comparisons were produced for $C = 8$, and $NR = 8$.

By increasing p, which is the number of participated users in the recommendation process, our approach can produce higher performance than the other algorithms. When p is small, the user's information cannot be utilized efficiently. By increasing p and by utilizing more user's information efficiently, the performance of our algorithms will be enhanced gradually. However, the C-SBR–ABR recommendation algorithm that considers the dynamics and multi-preferences of the learner, multidimensional attributes of the materials, and sequential patterns of materials, is able to provide better results than other algorithms, whether the number of p is small or large.

Performance evaluation for different sparsity levels: To illustrate that the proposed approach can alleviate the sparsity problem, we increased the sparsity level of the training set by dropping some randomly selected entries. However, we kept the test set the same for each sparse training set. The performance of the C-SBR–ABR algorithm was compared with other algorithms. Fig. 12 clarifies that the performance is degraded rapidly in the proposed algorithm. It is because the attributes of an item can still be used for finding the similar items. Furthermore, this algorithm enriches the item and user profiles by combining the attributes and sequential pattern information.

Performance evaluation for recommendation quality: In the final experiment, in order to evaluate the quality of recommendations by the C-SBR–CB, it was compared with the content-based recommendation algorithm [1], collaborative-based recommendation algorithm [26], and hybrid recommendation algorithm [4] based on the defined Intra-List Similarity Metric. As shown in Fig. 13, the C-SBR–CB has lower ISM than any other algorithm, which suggests the higher diversity. Therefore, our method can alleviate the overspecialization problem. As Fig. 13 indicates, the content-based filtering has the lowest diversity, and the diversity in collaborative and hybrid recommendations is almost equal. By increasing the number of recommendations, the diversity decreases for all the algorithms.

4. Related works

The first recommender system was developed in the mid-1990s [13]. Many recommendation systems in various fields such as movies, music, news, commerce, and medicine have been developed but few in the education field. Park et al. [31] reviewed many papers in this area. With the appearance of e-learning, the learning material (learning content or learning resource) and learning activity recommendation have been investigates by researchers [25,37]. According to the applied strategies for material recommendations, we categorize the previous works into four groups: data mining, content-based filtering, collaborative filtering, and hybrid approach. It must be noted that most works used a combination of these strategies. The overview of the recommendation strategies and some of the related works were presented in Table 3. We briefly surveyed some of the important works and explained the drawbacks of them that can be addressed by our proposed approach.

4.1. Data mining

Data mining techniques use the gathered information about the learner behavior (e.g. navigation history) to produce recommendations. These techniques are suitable to recommend the sequence of learning materials (i.e. learning path) rather than the learning materials itself. Romero et al. [35] developed a specific web mining tool for discovering suitable rules in the recommender engine. Their objective was to be enabled to recommend to a student the most appropriate links/web pages for his/her next visit. Lobo et al. [23] used a classification algorithm for the data selected from Moodle database to classify the data. Then, they used the Apriori Association Rule algorithm for the recommender.

4.2. Content-based filtering

This strategy uses the items' features for recommendations. In the existing content-based recommendation algorithms, due to considering the learners' preference information alone and not considering the similarity between the learners, only certain materials, which are similar to the learners' historical preferences, could be recommended. This has led to produce overspecialized recommendations that only include the items that are very similar to those the user already knows. In order to avoid the overspecialization of content-based methods, researchers have proposed new personalization strategies, such as collaborative filtering and hybrid approaches in order to combine the both techniques.

4.3. Collaborative filtering

The majority of researchers have used the collaborative filtering (CF) based recommendation system [6,29]. Bobadilla et al. [5] used a new equation to incorporate the learners' scores obtained from a test into the calculations in collaborative filtering for material prediction. Their experiment showed that the method obtained high accuracy in item prediction. A typical neighborhood-based set of CF algorithms has been used in order to support the learning object recommendation by Manouselis et al. (2010).

Since in the e-learning environment, the learning resources are in a variety of multimedia formats including text, hypertext, image, video, audio, and slides, it is difficult to calculate the content similarity of two items [9]. In this sense, the CF applies the users' preference information that is a good indication for recommendation in e-learning systems. Regardless of its success in many application domains, applicability and quality of CF is limited due to the so-called sparsity problem, which occurs when the available data are insufficient for identification of similar users. In addition, this method neglects the content-based relativity between the materials and other contextual information.

4.4. Hybrid approach

To overcome the drawbacks of other strategies, researchers used a hybrid approach for material recommendation. Khribi et al. [19] used the learners' recent navigation histories to find the similarities and dissimilarities among user preferences and among the contents of the learning materials for online automatic recommendations. They implemented web usage mining techniques with the content-based and collaborative filtering to compute the relevant links to recommend to active learners. García et al. [15] applied the association rule mining to discover interesting information through the student's usage data in the form of IF-THEN recommendation rules and then, used a collaborative recommender system to share and score the recommendation rules obtained by teachers with similar profiles along with other experts in education. García et al. [14] described a collaborative educational data mining tool based on association rule mining for the ongoing improvement of e-learning courses, which also allows the teachers with similar course profiles to share and score the discovered information. This two works also do not consider the attributes' information and the dynamic preferences of the learners. Klašnja-Milićević et al. [20] described a recommendation module of a programming tutoring system, which could automatically adapt to the interests and knowledge levels of the learners. This system recognized different patterns of the learning style and the learner's habits through testing the learning styles of the learners and mining their server logs.

In addition, in the hybrid model based on recommendation strategy, some of the other techniques such as clustering techniques, metadata, item repository theory, and neural networks are used. Usually these recommendations require intensive computation. For example, De Meo et al., [11] presented an XML-based multi-agent system to support e-learning activities, which take into account the profile, past behaviors, preferences, and needs of the users, as well as the characteristics of the devices they use for these activities. Wang and Liao [43] proposed an adaptive learning using the e-learning system. It considers various characteristics of the student to teach them English, as the second language.

An appropriate recommendation technique must be selected according to pedagogical reasons. These pedagogical reasons are derived from specific demands of lifelong learning. Therefore, some recommendation techniques are more suitable for specific demands of lifelong learning than others. A method to implement pedagogical decisions into a recommender system is to apply a variety of recommendation techniques in a recommendation strategy [12]. According to this idea, this paper used two recommendation modules based on the attributes of material and the sequential patterns of the learner. Using this new hybrid approach, we addressed the sparsity and overspecialization problems that are the drawbacks of collaborative filtering and content-based filtering, respectively. In addition, the proposed approach outperformed the previous algorithms on precision, and recall measure.

5. Conclusion

Due to the success of the recommender systems, the researches on technology-enhanced learning have started to deal with the recommender strategies for learning. Considering the contextual learner and the material information, in order to improve the quality of recommendations, we proposed a hybrid recommendation approach based on the sequential patterns and the attributes of the accessed materials. The experiment results show that our algorithm can outperform the traditional recommendation algorithms significantly in precision, recall, and intra-list similarity and could be more suitable for personalization of learning environments.

Additionally, there are some limitations that could determine some possible directions for further researches. In learning environments, as a logical assumption the learners with greater knowledge have to be considered more important in the calculation of the recommendations, compared to the learners with less knowledge. Thus, in order to improve the recommendation results, one way could be to implement a new approach to take the knowledge of learner into account and develop the existing equations for recommendations accordingly.

Acknowledgment

We would like to thank the Iran Research Institute for ICT (ITRC) that supported this research work.

References

[1] G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions, IEEE Trans. Knowl. Data Eng. 17 (6) (2005) 734–749.

[2] R. Agrawal, R. Srikant, Mining sequential patterns, in Proceedings of the11th International Conference on Data, Engineering, 1995, pp. 3–14.

[3] R. Agrawal, R. Srikant, Fast algorithms for mining association rules in large databases, in Proceedings of the 20th International Conference on Very Large Databases VLDB'94, Santiago, Chile, 1994, pp. 487–499.

[4] P. Baudisch, Joining collaborative and content-based filtering, in: Proceedings of the Conference on Human Factors in Computing Systems, 1999.

[5] J. Bobadilla, F. Serradilla, A. Hernando, MovieLens, Collaborative filtering adapted to recommender systems of e-learning, Knowl.-Based Syst. 22 (4) (2009) 261–265.

[6] J. Bobadilla, F. Serradilla, J. Bernal, A new collaborative filtering metric that improves the behavior of recommender systems, Knowl.-Based Syst. 23 (6) (2010) 520–528.

[7] J. Bobadilla, A. Hernando, F. Ortega, J. Bernal, A framework for collaborative filtering recommender systems, Expert Syst. Appl. 38 (12) (2011) 14609–14623 (Original Research Article).

[8] J.S. Breese, D. Heckerman, C. Kadie, Empirical analysis of predictive algorithms for collaborative filtering, in: UAI'98 Proceedings of the Fourteenth Conference on Uncertainty in, Artificial Intelligence, 1998, pp. 43–52.

[9] W. Chen, Z. Niu, X. Zhao, Y. Li, A hybrid recommendation algorithm adapted in e-learning environments. World Wide Web, 2012, pp. 1–14.

[10] L. Chuan-chang, Telecommunications search recommendation model based on user search behavior and gradual forgetting collaborative filtering strategy, J. China Univer. Posts Telecommun. 17 (3) (2010) 110–117.

[11] P. De Meo, A. Garro, G. Terracina, D. Ursino, Personalizing learning programs with X-Learn, an XML-based, "user-device" adaptive multi-agent system, Inf. Sci. 177 (8) (2007) 1729–1770.

[12] H. Drachsler, H.G. Hummel, R. Koper, Personal recommender systems for learners in lifelong learning networks: the requirements, techniques and model, Int. J. Learn. Technol. 3 (4) (2008) 404–423.

[13] A. Felfernig, G. Friedrich, L. Schmidt-Thieme, Introduction to the IEEE intelligent systems special issue: recommender systems, IEEE Comput. Soc. (2007).

[14] E. García, C. Romero, S. Ventura, C.d Castroa, A collaborative educational association rule mining tool, Int. High. Educ. 14 (2) (2011) 77–88.

[15] E. García, C. Romero, S. Ventura, C. Castro, An architecture for making recommendations to courseware authors using association rule mining and collaborative filtering, User Model. User-Adapted Int. 19 (1–2) (2009) 99–132.

[16] K.I. Ghauth, N.A. Abdullah, Learning materials recommendation using good learner's ratings and content-based filtering, Educ. Technol. Res. Develop. 58 (6) (2010) 711–727.

[17] T. Gu, S.-J. Hong, J.-B. Yoo, K.-M. Lee, Personal preference for reliable transaction identification on web service, in: ICIW '08 Proceedings of the 2008 Third International Conference on Internet and Web Applications and Services, 2008, pp. 150–155.

[18] J.L. Herlocker, J.A. Konstan, L.G. Terveen, J.T. Riedl, Evaluating collaborative filtering recommender systems, ACM Trans. Inf. Syst. (TOIS) 22 (1) (2004) 5–53.

[19] M.K. Khribi, M. Jemni, O. Nasraoui, Automatic recommendations for e-learning personalization based on web usage mining techniques and information retrieval, Educ. Technol. Soc. 12 (4) (2009) 30–42.

[20] A. Klašnja-Milićević, B. Vesin, M. Ivanović, Z. Budimac, E-Learning personalization based on hybrid recommendation strategy and learning style identification, Comput. Educ. 56 (3) (2011) 885–899.

[21] D. Lemire, H. Boley, S. McGrath, M. Ball, Collaborative filtering and inference rules for context-aware learning object recommendation, Int. J. Interact. Technol. Smart Educ. 2 (3) (2005) 1–11.

[22] G. Liang, K. Weining, L. Junzhou, Courseware recommendation in e-learning system, Adv. Web Based Learn. (2006) 10–24.

[23] L.M. Lobo, R.J. Sunita, B. Aher, Mining association rule in classified data for course recommender system in e-learning, Int. J. Comput. Appl. 39 (7) (2012) 1–7.

[24] J.P. Lucas, S. Segrera, M.N. Moreno, Making use of associative classifiers in order to alleviate typical drawbacks in recommender systems, Expert Syst. Appl. 39 (1) (2012) 1273–1283.

[25] J. Lu, Framework and approach for developing intelligent personalized learning recommender systems, in: 2nd International Conference on Information Technology and Applications, 8–11 January 2004, Harbin, China, 2004.

[26] H. Ma, I. King, M.R. Lyu, Effective missing data prediction for collaborative filtering, in: Proceedings of the 30th International ACM SIGIR Conference on Information Retrieval, 2007, pp. 39–46.

[27] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, 1967, pp. 281–297.

[28] P. Melville, R.J. Mooney, R. Nagarajan, Content boosted collaborative filtering for improved recommendations, in: Eighteenth National Conference on Artificial Intelligence, 2002, pp. 187–192.

[29] A.K. Milicevic, A. Nanopoulos, M. Ivanovic, Social tagging in recommender systems: a survey of the state-of-the-art and possible extensions, Artif. Intell. Rev. 33 (3) (2010) 187–209.

[30] U. Niranjan, R.B.V. Subramanyam, V. Khanaa, Developing a web recommendation system based on closed sequential patterns, Commun. Comput. Inf. Sci. 101 (1) (2010) 171–179.

[31] D.H. Park, H.K. Kim, Y. Choi, J.K. Kim, A literature review and classification of recommender systems research, Expert Syst. Appl. 39 (11) (2012) 10059–10072.

[32] M.J. Pazzani, A framework for collaborative, content-based and demographic filtering, Artif. Intell. Rev. 13 (5–6) (1999) 393–408.

[33] J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal, M.-C. Hsu, Mining sequential patterns by pattern-growth: the prefixspan approach, IEEE Trans. Knowl. Data Eng. 16 (2004) 1424–1440.

[34] D. Pennock, E. Horvitz, S. Lawrence, C. Giles, Collaborative filtering by personality diagnosis: a hybrid memory and model-based approach, in: Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence, 2000, pp. 473–480.

[35] C. Romero, S. Ventura, A. Zafra, P. Bra, Applying Web usage mining for personalizing hyperlinks in Web-based adaptive educational systems, Comput. Educ. 53 (2009) 828–840.

[36] M. Salehi, I.N. Kmalabadi, Attribute-based recommender system for learning resource by learner preference tree, in: 2012 2nd International eConference on Computer and Knowledge Engineering (ICCKE), IEEE, 2012, pp. 133–138.

[37] M. Salehi, M. Pourzaferani, S.A. Razavi, Hybrid attribute-based recommender system for learning material using genetic algorithm and a multidimensional information model, Egyptian Inf. J. 14 (1) (2013) 67–78.

[38] B. Sarwar, G. Karypis, J. Konstan, J. Reidl, Item based collaborative filtering recommendation algorithms, in: Proceedings of the 10th International Conference on World Wide Web, 2001, pp. 285–295.

[39] Y.Y. Shih, D.R. Liu, Product recommendation approaches: collaborative filtering via customer lifetime value and customer demands, Expert Syst. Appl. 35 (1–2) (2008) 350–360.

[40] R. Srikant, R. Agrawal, Mining sequential patterns: generalizations and performance improvements, in: Proceedings of the 5th International Conference on Extending Database Technology (EDBT), Avignon, France, 1996, pp. 3–17.

[41] D.W. Tai, H. Wu, P. Li, Effective e-learning recommendation system based on self-organizing maps and association mining, Electron. Lib. 26 (2008) 329–344.

[42] F. Tao, F. Murtagh, M. Farid, Weighted association rule mining using in weighted support and significance framework, in: Proceedings of the Ninth International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 2003, pp. 661–666.

[43] Y.-H. Wang, H.-C. Liao, Data mining for adaptive learning in a TESL-based e-learning system, Expert Syst. Appl. 38 (6) (2011) 6480–6485.

[44] Y. Zheng, L. Li, A three-dimensional context-awareness model for peer recommendation in the e-learning context, Int. J. e-Learn. 7 (1) (2008) 153–168.