

“© 2014 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.”

# A Fuzzy Tree Matching-Based Personalised E-Learning Recommender System

Dianshuang Wu, Guangquan Zhang and Jie Lu

**Abstract**—The rapid development of e-learning systems provides learners great opportunities to access the learning activities online, which greatly supports and enhances learning practices. However, too many learning activities are emerging in the e-learning system, which makes it difficult for learners to select proper ones for their particular situations since there is no personalised service function. Recommender systems, which aim to provide personalised recommendations, can be used to solve this issue. However, e-learning systems have two features to handle: (1) data of learners and leaning activities often present tree structures; (2) data are often vague and uncertain in practice. In this study, a fuzzy tree-structured data model is proposed to comprehensively describe the complex learning activities and learner profiles. A tree matching method is then developed to match the similar learning activities or learners. To deal with the uncertain category issues, a fuzzy category tree and relevant similarity measure are developed. A hybrid recommendation approach, which considers precedence relations between learning activities and combines the semantic and collaborative filtering similarities between learners, is developed. The proposed approach can handle the special requirements in e-learning environment and make proper recommendations in e-learning systems.

## I. INTRODUCTION

**E**-LEARNING environments are becoming increasingly popular in educational establishments due to the development of information and communication technologies. The rapid growth of e-learning has changed traditional learning behaviour and presented a new situation to learners (students), which greatly supports and enhances learning practices online. In the meantime, because of the emerging of numerous kinds of learning activities, such as subjects, learning materials, and so on, in the e-learning environment, learners find it difficult to select the learning activities that best meet their situations. The information overload problem is increasingly severe in the big data era [1], [2]. It would be very useful if an e-learning system could automatically generate recommendations to guide a learner's activities [3]. An e-learning recommender system is necessary to make personalised recommendations [4]. The motivation of this study is to develop a recommendation approach to support learners in the selection of the most appropriate learning activities in e-learning environment.

A recommender system [5], one of the most popular applications of personalization techniques, is first proposed

Dianshuang Wu, Guangquan Zhang and Jie Lu are with Decision Systems & e-Service Intelligence Lab, Centre for Quantum Computation & Intelligent Systems, Faculty of Engineering and Information Technology, University of Technology, Sydney, Australia (email: Dianshuang.Wu@student.uts.edu.au, {Guangquan.Zhang, Jie.Lu}@uts.edu.au).

The work presented in this paper was supported by the Australian Research Council (ARC) under discovery grant DP110103733.

and applied in e-commerce area for product purchase. Recommender systems can be defined as programs which attempt to recommend items (products and services) to users (individuals and businesses) by predicting a user's interest in a given item based on various types of information, including particulars about items, users and the interactions between users and items. They have been widely used in various web-based applications in e-commerce, e-business [6][7], e-tourism [8], e-government [9], and so on. A variety of recommendation approaches, such as data mining [10], [11], agents [12] and reasoning, have been developed and applied into recommender systems [13], [14]. In general, the most common used three recommendation approaches are collaborative filtering (CF), content-based (CB) and knowledge-based (KB) approaches [15]. CF approach helps people make choices based on the opinions of other people who share similar interests [16]. It can be further divided into user-based and item-based CF approaches. CB approaches recommend items that are similar to those previously preferred by a specific user [17]. KB approaches offer items to users based on knowledge about the users and items [15]. Each approach has its limitations, such as the item content dependency problem, overspecialization problem for CB [5], [17], the cold start problem and the sparsity problem for CF [5]. To gain higher performance and avoid the drawbacks of the typical recommendation approaches, a hybrid recommendation approach can be proposed by combining the best features of two or more recommendation approaches into one hybrid approach [18]. In recommender systems, the similarity measure, which is used to find similar users or items, or to match items with users' requirements, is the core technique in various recommendation approaches.

E-learning activities have some special features and demands different from commercial products [19], which brings special requirements on recommendation approaches and similarity measures. First, both learning activities and learner profiles have complex descriptions and features. A learning activity contains several aspects of information, such as the content description, lecture information, prerequisite information and so on, while a learner profile contains the learner's background, learning goals, prior knowledge, learner characteristics, and so on. Each aspect of information can be described in detail with several sub-aspects. Thus, the data in the e-learning environment present a hierarchical structure, and the structure is not fixed. For example, the prerequisite of a learning activity may have one or several subjects. Second, there are some precedence relationships among learning activities. Learners always want to learn

something new or with higher difficulty levels, which makes the basic CB recommendation approach not work effectively. Third, it is not feasible to differentiate two learning activities just from their IDs or names, because learning activities provided from different schools may have different names but the same or similar content. Fourth, there are some complex and uncertain relationships in e-learning context, such as the categories of learning activities. One subject may be under several categories with different degrees.

To deal with the above special requirements in e-learning recommender systems, this study develops a fuzzy tree-structured data model to describe learner profiles and learning activities, proposes a tree matching method to match the similar learning activities and to compare learners, and presents a hybrid recommendation approach to recommend learning activities to learners. To deal with the uncertain category issues, a fuzzy category tree and relevant similarity measure are developed.

The paper makes contributions to both theoretical and practical aspects. At the theoretical level, a fuzzy tree-structured data model and tree matching method is developed. At the practical level, a hybrid recommendation approach for e-learning systems is proposed.

The remainder of the paper is organised as follows. In Section II, the fuzzy tree-structured data model and tree matching method is presented. Section III describes the fuzzy tree-structured learner profile and learning activities. The hybrid recommendation approach for learning activities is developed in Section IV. Section V gives an illustrative example to calculate the predicted rating by use of the proposed recommendation approach. Finally, the proposed approach is discussed in Section VI, and the future study is also given.

## II. A FUZZY TREE-STRUCTURED DATA MODEL AND TREE MATCHING METHOD

In this section, a fuzzy tree-structured data model, which is used to represent tree-structured learning activities or learner profiles, is defined first. A tree matching method, which is used to construct a map to identify the parts of two trees that most correspond and compare two trees, is then presented.

### A. A Fuzzy Tree-Structured Data Model

The fuzzy tree-structured data model is based on the basic tree definition, which is given as follows.

*Definition 1:* [20] A tree is defined as a directed graph  $T = (V, E)$ , in which  $V$  is a finite set of nodes,  $E$  is a binary relation on  $V$  where each pair  $(u, v) \in E$  represents the parent-child relationship between two nodes  $u, v \in V$ , and the underlying undirected graph of  $T$  has no cycles and there is a distinguished root node in  $V$ , denoted by  $root(T)$ , so that for any node  $v$ , there is a path in  $T$  from  $root(T)$  to node  $v$ .

The definition only defines the hierarchical relations among the nodes. In real applications, the definition is usually extended to represent practical objects. In this research, a tree-structured data model is defined.

*Definition 2:* A tree-structured data model is a tree, in which the following features are added to the tree nodes:

1) A set of attributes  $A = \{a_1, a_2, \dots, a_n\}$  are introduced, in which each attribute  $a_i \in A$  represents one aspect of the semantic meanings of a node. A value domain set  $D = \{d_1, d_2, \dots, d_n\}$  is defined accordingly. For each attribute  $a_i$ , a value assignment function  $a_i : V \rightarrow d_i$  is defined so that each node can be assigned values for its attributes.

2) A set of similarity measures  $S = \{s_1, s_2, \dots, s_m\}$  are defined on the node attributes to evaluate the similarity between nodes from different points of views. Each similarity measure  $s_i$  is defined as a function  $s_i : \Delta \times \Delta \rightarrow [0, 1]$ , where  $\Delta \in 2^D$ , and  $\Delta$  can be specified according to specific applications. Two commonly defined similarity measures are concept similarity and value similarity to compare the concepts and values of two tree nodes respectively.

3) A weight function  $w : V \rightarrow [0, 1]$  is defined to assign a weight to each node to represent its importance degree to its siblings.

In the above definition, a tree node can be assigned several attributes. For example, each node can be assigned a label from a term set to express some semantic meanings. A node can also be assigned numerical values to express the degrees of the relevant attribute. Various similarity measures between tree nodes can be introduced. For example, concept similarity and value similarity between nodes can be defined properly, which can be used to match two tree-structured data or evaluate the similarity between two trees.

In real applications, the data or relations are usually vague and uncertain. For example, a subject in e-learning context may belong to several categories with different degrees; and the concept similarity between two node labels may be given by domain experts subjectively by use of some linguistic terms, such as “very similar”, “absolutely different”. To deal with these issues, fuzzy set theory and techniques are applied. A fuzzy tree-structured data model is defined.

*Definition 3:* A fuzzy tree-structured data model is a tree-structured data whose node features, i.e. the node attribute values, similarity measures between nodes, or node weights, are represented as fuzzy sets.

In the following sections, trees and nodes are represented with the following symbols. Suppose that we have a numbering for each tree. Let  $t[i]$  be the  $i$ th node of tree  $T$  in the given numbering,  $T[i]$  be the sub-tree rooted at  $t[i]$ ,  $F[i]$  be the unordered forest obtained by deleting  $t[i]$  from  $T[i]$ , and  $t[i_1], t[i_2], \dots, t[i_{n_i}]$  be the children of  $t[i]$ .

### B. A Tree-Structured Data Matching Method

This sub-section proposes a tree matching method to construct a map to identify the parts of two trees that most correspond.

Let two trees to be matched be denoted as  $T_1[i]$  and  $T_2[j]$ . A maximum conceptual similarity tree mapping [21], which is a kind of edit distance mapping, is constructed to identify the parts of the two trees that most conceptually correspond. When constructing the mapping, tree structures, node concepts and node weights are all taken into consideration. As

discussed in the tree-structured data definition, a concept similarity measure between tree nodes  $sc(\cdot)$  is pre-defined based on the node attributes.

It should be noted that in contrasting application scenarios, the requirements to match two trees are different. For example, when comparing two trees, the weights of both trees should be considered. In another situation, when matching a sub-tree to a target tree to find out whether the target tree includes the sub-tree, the weights of the sub-tree should mainly be weighted. Therefore, the matching method should consider the two types of matching situations respectively. In the former situation, the matching is called symmetric matching, while in the latter situation the matching is called asymmetric matching.

The maximum conceptual similarity tree mapping maps the most conceptually similar parts of two trees. This mapping can be constructed during the computation of the conceptual similarity between two trees. The conceptual similarity also has two types, symmetric and asymmetric, depending on the matching types. They are denoted as  $sc_{Tsym}$  and  $sc_{Tasym}$  when the matching type needs to be specified. When computing the conceptual similarity between two trees, not only the concepts of two roots, but also the concepts of their sub-trees should be considered.

Given two trees  $T_1[i]$  and  $T_2[j]$  to be compared, their conceptual similarity is calculated as follows.

According to the condition of whether  $t_1[i]$  and  $t_2[j]$  are leaves, four situations are listed below.

*Situation 1:*  $F_1[i] = \phi, F_2[i] = \phi$

In this situation,  $t_1[i]$  and  $t_2[j]$  are both leaves, and their conceptual similarity is equivalent to the conceptual similarity of the two nodes:

$$sc_T(T_1[i], T_2[j]) = sc(t_1[i], t_2[j]). \quad (1)$$

*Situation 2:*  $F_1[i] = \phi, F_2[i] \neq \phi$

In this situation,  $t_1[i]$  is a leaf node and  $t_2[j]$  is an inner node. As the concept of a tree is dependent not only on its root's concept, but also on its children's, the children of the inner node  $t_2[j]$  are also considered in the formula:

$$sc_T(T_1[i], T_2[j]) = \alpha \cdot sc(t_1[i], t_2[j]) + (1 - \alpha) \cdot \sum_{t=1}^{n_j} w_{j_t} \cdot sc_T(T_1[i], T_2[j_t]), \quad (2)$$

where  $w_{j_t}$  is the weight of  $t_2[j_t]$ ,  $\alpha$  is the influence factor of the parent node, and  $n_j$  is the number of children of  $t_2[j]$ .

*Situation 3:*  $F_1[i] \neq \phi, F_2[i] = \phi$

The situation is similar to the *Situation 2*. The children of the inner node  $t_1[i]$  are considered in the formula:

$$sc_T(T_1[i], T_2[j]) = \alpha \cdot sc(t_1[i], t_2[j]) + (1 - \alpha) \cdot \sum_{t=1}^{n_i} w_{i_t} \cdot sc_T(T_1[i_t], T_2[j]), \quad (3)$$

where  $w_{i_t}$  is the weight of  $t_1[i_t]$ ,  $\alpha$  is the influence factor of the parent node, and  $n_i$  is the number of children of  $t_1[i]$ .

*Situation 4:*  $F_1[i] \neq \phi, F_2[i] \neq \phi$

In this situation,  $t_1[i]$  and  $t_2[j]$  are both inner nodes.

$$sc_T(T_1[i], T_2[j]) =$$

$$\alpha \cdot sc(t_1[i], t_2[j]) + (1 - \alpha) \cdot sc_F(F_1[i], F_2[j]), \quad (4)$$

where  $\alpha$  is the influence factor of the parent node, and  $sc_F(\cdot)$  is the conceptual similarity between two forests.

In the last situation, both  $t_1[i]$  and  $t_2[j]$  have children. Their children construct two forests  $F_1[i]$  and  $F_2[j]$ , which are compared with the forest similarity measure  $sc_F(F_1[i], F_2[j])$ . To find the node pairs that most correspond between the roots of the two forests, a maximum weighted bipartite matching (MWBM) problem [22] is resolved. A MWBM between the roots of the two forests,  $M_{ij}$  is constructed. The conceptual similarity between  $F_1[i]$  and  $F_2[j]$  is calculated as:

$$sc_F(F_1[i], F_2[j]) = \sum_{(t_1[i_p], t_2[j_q]) \in M_{ij}} w_{i_p, j_q} \cdot sc_T(T_1[i_p], T_2[j_q]), \quad (5)$$

where  $w_{i_p, j_q}$  is the weight of the matching node pair. If the matching is a symmetric matching, both of the corresponding nodes' weights should be considered,  $w_{i_p, j_q} = (w(t_1[i_p]) + w(t_2[j_q]))/2$ . If the measure is an asymmetric matching, only the first node's weight is considered,  $w_{i_p, j_q} = w(t_1[i_p])$ .

During the computation process of the conceptual similarity between two trees, the maximum weighted bipartite matching results are recorded. Based on the records, the most corresponding nodes among two trees can be identified. The roots of two trees are corresponding node pairs. Then the corresponding nodes in the children of two roots are identified based on two roots' children's maximum weighted bipartite matching. Other corresponding nodes are identified in the same way.

### III. FUZZY TREE-STRUCTURED LEARNER PROFILES AND LEARNING ACTIVITIES

In this section, the data structures of learner profiles and learning activities in the e-learning recommender system are presented.

When a learner selects an learning activity, various kinds of information, such as the learner's background, learning goals, and learned learning activities, will influence the learner to make a decision. In our recommender system, all the information is taken into consideration when making recommendations. Based on the tree-structured data model proposed in last section, a learner's profile is represented as a tree-structured data. The learning activities are described from several aspects, such as the prerequisite courses, the content, the lecture, and so on. A learning activity is also described as a tree-structured data in our system. The structures of a learner profile and a learning activity are illustrated in Fig. 1.

To express the semantic meanings of the tree-structured data, attributes are defined on the tree nodes. In particular, a *label* attribute and a *category* attribute are defined to express the node concept. The *label* assigns each node a name. The *category* specifies which category a learning activity, a major,

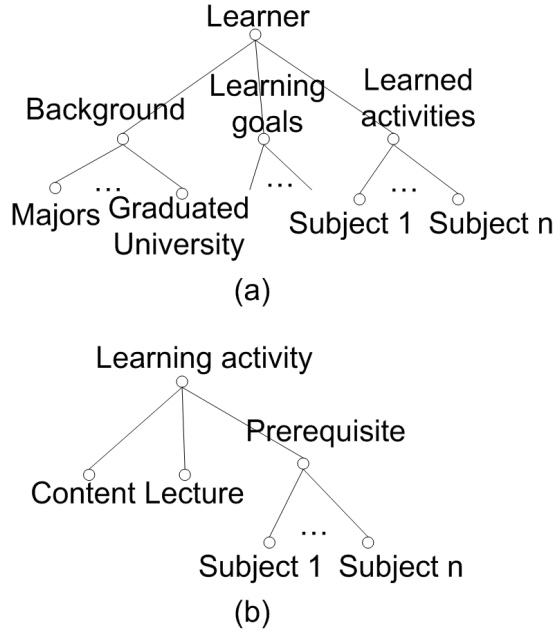


Fig. 1. The structures of a learner profile and a learning activity.

or a career belongs to, which is used to infer the semantic relations between tree nodes. In the proposed system, the concept similarity between two nodes is calculated based on the two attributes. If two nodes are assigned *category*, the category similarity will be taken as the concept similarity. Otherwise, their *labels* are compared. In our system, the value of a *category* is a fuzzy category tree. The fuzzy category trees and their similarity measure are described in detail as follows.

#### A. Fuzzy Category Tree and The Category Similarity

To describe the semantic relations between the learning activities, a category is introduced in our recommender system. The category has two levels, which construct a tree structure. There are six general categories, which are “IT/Computer Science”, “Nature Science”, “Humanities/Social Sciences”, “Business”, “Engineering/Technology”, and “Medicine/Health”. Each general category is divided into several sub-categories. For example, “IT/Computer Science” category can be divided into four sub-categories, which are “Internet”, “Software”, “Hardware”, and “Business Intelligence”. The structure of the category is illustrated in Fig. 2.

In real applications, each learning activity may belong to several categories with different degrees. For example, the subject *Business Intelligence* is under the categories “Business Intelligence”, “Software”, “Marketing”, and “Management” with different membership degrees, as shown in Fig. 3 (a), in which the number under each sub-category represents the membership degree of the subject belonging to the sub-category. Therefore, the category of a learning activity is represented as a fuzzy category tree. The structure of the fuzzy category tree is shown in Fig. 2. Only leaves of the

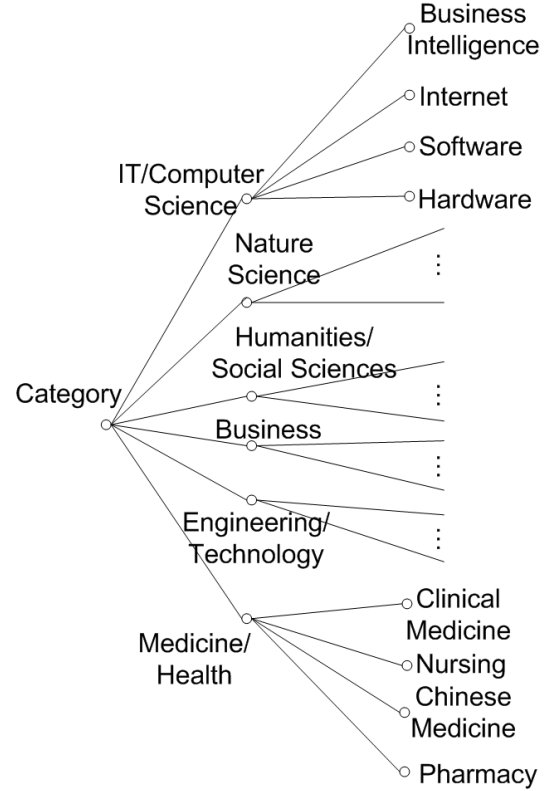


Fig. 2. The category tree of the learning activities.

fuzzy category tree are assigned values, which represent the membership degrees of the learning activity belonging to the relevant sub-categories.

The category similarity between two learning activities is calculated as follows. Let  $T_1[i]$  and  $T_2[i]$  represent two fuzzy category trees of two learning activities  $l_1$  and  $l_2$  respectively. According to the conditions whether  $t_1[i]$  and  $t_2[i]$  have children or not, four situations are considered in the equation. The category similarity between  $l_1$  and  $l_2$  is calculated as:

$$cs(T_1[i], T_2[i]) = \begin{cases} v(t_1[i]) \wedge v(t_2[i]), & F_1[i] = \phi, F_2[i] = \phi \\ v(t_1[i]) \wedge v(T_2[i]), & F_1[i] = \phi, F_2[i] \neq \phi \\ v(T_1[i]) \wedge v(t_2[i]), & F_1[i] \neq \phi, F_2[i] = \phi \\ \alpha^{h-d_i} \cdot (v(T_1[i]) \wedge v(T_2[i])) + (1 - \alpha^{h-d_i}) \cdot (\bigvee_{j=1}^{n_i} cs(T_1[i_j], T_2[i_j])), & F_1[i] \neq \phi, F_2[i] \neq \phi \end{cases} \quad (6)$$

where  $v(t_1[i])$  and  $v(t_2[i])$  represent the values of nodes  $t_1[i]$  and  $t_2[i]$  respectively;  $v(T_1[i])$  and  $v(T_2[i])$  represent the values of sub-trees under  $t_1[i]$  and  $t_2[i]$  respectively, which is calculated by Equation 7;  $\alpha$  is the influence factor of the parent node,  $h$  is the height of the category tree, and  $d_i$  is the depth of node  $i$  in the category tree.

$$v(T[i]) = \begin{cases} v(t[i]), & F[i] = \phi \\ \bigvee_{j=1}^{n_i} v(T[i_j]), & F[i] \neq \phi \end{cases} \quad (7)$$

When calculating the category similarity, the categories in all levels are taken into consideration, and the lower level gains more weight as the categories in lower level are more

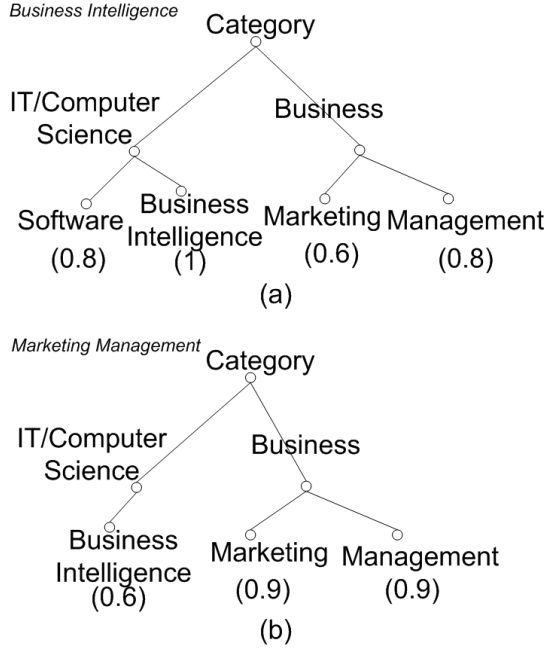


Fig. 3. Two fuzzy category trees of two learning activities. (a) is the fuzzy category tree of the subject *Business Intelligence*. (b) is the fuzzy category tree of the subject *Marketing Management*.

specific. The coefficient  $\alpha^{h-d_i}$  in Equation 6 reflects the point.

Taking two subjects *Business Intelligence* and *Marketing Management*, which are illustrated in Fig. 3, as examples. Fig. 3 shows the fuzzy category trees of the two subjects. Let  $\alpha$  be 0.5. The category similarity between these two subjects is calculated as 0.675 by use of the Equation 6.

The category similarity between the *category* attributes of two nodes is used to evaluate the concept similarity between the two nodes.

#### IV. A HYBRID FUZZY TREE MATCHING-BASED RECOMMENDATION APPROACH

In this section, a hybrid recommendation approach for learning activities is developed based on the fuzzy tree-structured data model. The approach first selects the recommendation alternatives which satisfy the prerequisite requirement. For each recommendation alternative, the predicted rating is computed by weighted aggregating the ratings of similar learners to the target learner. The similarity between learners hybridizes both the semantic and CF similarities. The proposed approach takes the user-item rating matrix which contains the existing ratings, the learner profile trees and learning activity trees as input. For a target learner  $u_t$ , the recommendation process is described in six steps as follows.

1) *Step 1: Determine the recommendation alternatives:* To make recommendations to  $u_t$ , the recommendation alternatives must be determined first. Since the learning activities often have some prerequisites, only part of the unlearned activities of  $u_t$  can be recommended. Let the profile tree of the target learner  $u_t$  be denoted as  $T_t$ . The sub-tree of  $T_t$ ,

which represents the learned learning activities, is denoted as  $T_{t,l}$ . As mentioned before, it is usually impossible to match two learning activities just from their IDs or names. The proposed tree matching method is used to check if a learning activity is suitable for the learner.

For a learning activity  $a$ , which is new to  $u_t$ , its prerequisite sub-tree is denoted as  $T_{a,p}$ . A sub-tree matching is calculated as

$$s_a = s_{CT_{asym}}(T_{a,p}, T_{t,l}) \quad (8)$$

A matching similarity threshold  $s_{thres}$  is defined. If  $s_a > s_{thres}$ , then learning activity  $a$  can be selected as a recommendation alternative. Through this step, a set of recommendation alternatives are chosen.

2) *Step 2: Calculate the semantic similarity between users:* For an alternative learning activity  $a$ , the users who have rated it are selected, denoted as  $U_a = \{u_1, u_2, \dots, u_m\}$ . For each user  $u_i \in U_a$ , let the profile tree be  $T_i$ . The semantic similarity between  $u_t$  and  $u_i$  is calculated as:

$$s_{sem}(u_t, u_i) = s_{CT_{sym}}(T_t, T_i) \quad (9)$$

During the calculation process of  $s_{sem}(u_t, u_i)$ , a maximum conceptual similarity tree mapping between the profile trees of  $u_t$  and  $u_i$  is constructed. Their most similar learned activities can be matched. Let the matched learning activities be recorded in  $M_{t,i}$ . For any  $(p, q) \in M_{t,i}$ ,  $p$  and  $q$  are the activities rated by  $u_t$  and  $u_i$  respectively.

3) *Step 3: Calculate the CF similarity between users:* A category similarity threshold  $cs_t$  is predefined. For any learning activity pair  $(p, q)$ ,  $p$  and  $q$  will be taken as irrelevant if their category similarity  $cs(p, q) \leq cs_t$ . Given the matched learning activity set  $M_{t,i}$  of  $u_t$  and  $u_i$ , a sub set of  $M_{t,i}$  is selected as  $M'_{t,i} = \{(p, q) : (p, q) \in M_{t,i}, cs(p, q) > cs_t\}$ . Based on  $M'_{t,i}$ , the CF similarity between  $u_t$  and  $u_i$  is calculated by the Pearson correlation formula as:

$$s_{CF}(u_t, u_i) = \frac{\sum_{(p,q) \in M'_{t,i}} (r_{t,p} - \bar{r}_t) \cdot (r_{i,q} - \bar{r}_i)}{\sqrt{\sum_{(p,q) \in M'_{t,i}} (r_{t,p} - \bar{r}_t)^2} \cdot \sqrt{\sum_{(p,q) \in M'_{t,i}} (r_{i,q} - \bar{r}_i)^2}} \quad (10)$$

where  $r_{t,p}$  is the rating of item  $p$  from user  $u_t$ , and  $\bar{r}_t$  and  $\bar{r}_i$  are the average ratings from  $u_t$  and  $u_i$  respectively.

4) *Step 4: Select top-N similar users:* The total similarity between users  $u_t$  and  $u_i$  is computed by integrating the two similarity measures computed in the last two steps.

$$s_u(u_t, u_i) = \beta \times s_{sem}(u_t, u_i) + (1 - \beta) \times s_{CF}(u_t, u_i) \quad (11)$$

where  $\beta \in [0, 1]$  is a semantic combination parameter specifying the weight of similarity in the integrating measure. The users in  $U_a$  are ordered according to the total similarity. The top- $N$  most similar users are selected as neighbors to predict ratings.

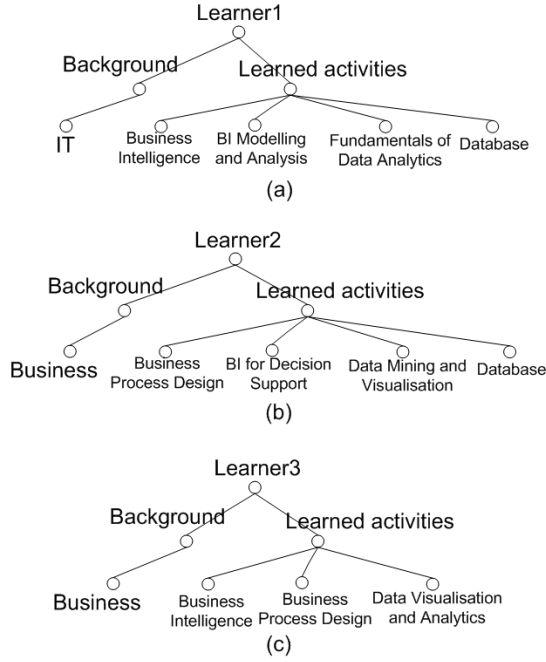


Fig. 4. Tree-structured learner profiles.

5) *Step 5: Calculate the predicted rating:* The predicted rating of learning activity  $a$  from learn  $u_t$  is calculated as:

$$pr_{u,a} = \frac{\sum_{i=1}^N r_{i,a} \times s_u(u_t, u_i)}{\sum_{i=1}^N s_u(u_t, u_i)} \quad (12)$$

6) *Step 6: Generate the recommendations:* The predicted ratings of all the alternative learning activities of learner  $u_t$  are calculated. The alternatives are ranked according to the predicted rating, and the top- $K$  are recommended to the learner.

## V. AN ILLUSTRATIVE EXAMPLE

An illustrative example is given in this section to show how the predicted rating is calculated. Suppose there are three learners and their tree-structured profiles are described in Fig. 4. The learner-subject rating matrix is depicted in Table I.

TABLE I  
LEARNER-SUBJECT RATING MATRIX

learners	Learner1	Learner2	Learner3
Business Intelligence	2		1
BI Modelling and Analysis	4		
Fundamentals of Data Analytics	3		
Business Process Design		3	2
Data Visualisation and Analytics			5
BI for Decision Support		2	
Data Mining and Visualisation		4	
Database	3	5	

In this example, the predicted preference (rating) of Learner 3 to the subject *Database* will be calculated.

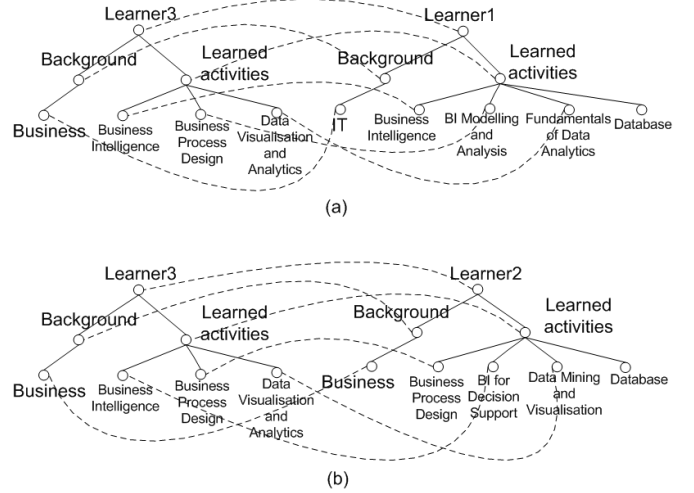


Fig. 5. The maximum conceptual similarity tree mapping between Learner 3 and Learner 1 (a) and that between Learner 3 and Learner 2 (b).

Two learners, Learner 1 and Learner 2, have rated the subject *Database*. Therefore, the semantic similarity between Learner 3 and Learner 1 and that between Learner 3 and Learner 2 are calculated by use of Equation 9.  $s_{sem}(learner3, learner1) = 0.572$ ;  $s_{sem}(learner3, learner2) = 0.896$ . During the computation, the maximum conceptual similarity tree mappings are constructed, which are shown in Fig. 5.

Based on the maximum conceptual similarity tree mappings, the CF similarity between Learner 3 and Learner 1 and that between Learner 3 and Learner 2 are calculated by use of Equation 10.  $s_{CF}(learner3, learner1) = 0.24$ ;  $s_{CF}(learner3, learner2) = 0.96$ . In this step, the sparsity problem of the rating matrix is alleviated by using the maximum conceptual similarity tree mappings.

The total similarity between learners is calculated by use of Equation 11. Let  $\beta = 0.5$ ,  $s_u(learner3, learner1) = 0.406$ ;  $s_u(learner3, learner2) = 0.928$ .

The predicted rating of Learner 3 to the subject *Database* is finally calculated by use of Equation 12, which is 4.39.

The predicted ratings to other unlearned subjects of Learner 3 can also be calculated. These unlearned subjects are ranked according to the predicted ratings, and the subjects with the highest predicted ratings are recommended.

## VI. DISCUSSION AND FURTHER STUDY

In this paper, a fuzzy tree-based hybrid recommendation approach for e-learning system is developed. The approach has the following merits. First, the proposed approach can deal with the complex tree-structured learner profiles and learning activities. A fuzzy tree-structured data model is used to describe the learners and learning activities comprehensively and flexibly. A tree-structured data matching method is developed, which can effectively match and compare the tree-structured data in e-learning environment. Second, it can handle the uncertain issues in e-learning systems by use

of fuzzy set techniques. A fuzzy category tree is defined to specify the categories that each learning activity roughly belongs to, and the category similarity measure method is developed to evaluate the conceptual similarity between them. Third, the proposed approach can handle the precedence relationships among learning activities to some extent. In Step 1 of the recommendation process which is illustrated in Section IV, only the learning activities whose prerequisites have been learned by the target user can be selected as the recommendation alternatives. Fourth, the proposed approach draws strength from both the semantic similarity and CF similarity. As the learner profile contains several aspects of a user, such as the background and learning goals of the user, the similar existing users can also be searched for a new user by the semantic similarity. Thus, it can make recommendations to new users. When calculating the CF similarity, the ratings of the matched learning activities rather than the exactly common learning activities between two users are used, which alleviates the sparsity problem caused by the sparse user-item rating matrix. On the other hand, there are some limitations in the current recommendation approach. For instance, the relations between learning activities have not been fully utilised.

The proposed recommendation approach is being implemented in an online e-learning recommender system which will be tested and compared with existing systems in a future study. The e-learning recommender system will greatly support and enhance learners to use e-learning systems, especially in the big data era. In addition, the semantic relations between learning activities and the matching between user requirements and learning activities will be exploited to improve the recommendation performance.

## REFERENCES

- [1] *The Economist*, "The data deluge," Special Report on Managing Information, Technology Section, February 25 (<http://www.economist.com/node/15579717>), 2010.
- [2] *The Economist*, "All too much," Special Report on Managing Information, Technology Section, February 25 (<http://www.economist.com/node/15557421>), 2010.
- [3] O.R. Zaane and J. Luo, "Web usage mining for a better web-based learning environment," in *Proceedings of Conference on Advanced Technology for Education*, pp. 60-64, 2001.
- [4] J. Lu, "A personalized e-learning material recommender system," in *Proceedings of the 2nd International Conference on Information Technology and Applications*, Harbin, China, CDROM, 2004.
- [5] G. Adomavicius and 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*, vol. 17, no. 6, pp. 734-749, 2005.
- [6] J. Lu, Q. Shambour, Y. Xu, Q. Lin and G. Zhang, "BizSeeker: A hybrid semantic recommendation system for personalized government-to-business e-services," *Internet Research*, vol. 20, no. 3, pp. 342-365, 2010.
- [7] J. Lu, Q. Shambour, Y. Xu, Q. Lin and G. Zhang, "A Web-based personalized business partner recommendation system using fuzzy semantic techniques," *Computational Intelligence*, vol. 29, no. 1, pp. 37-69, 2013.
- [8] M. Al-hassan, H. Lu and J. Lu, "Personalized e-Government services: Tourism recommender system framework," in J. Filipe and J. Cordeiro (eds), *Web Information Systems and Technologies*, vol. 75, Springer Berlin Heidelberg, pp. 173-187, 2011.
- [9] X. Guo and J. Lu, "Intelligent e-government services with personalized recommendation techniques," *International Journal of Intelligent Systems*, vol. 22, no. 5, pp. 401-417, 2007.
- [10] S.W. Changchien and T.C. Lu, "Mining association rules procedure to support on-line recommendation by customers and products fragmentation," *Expert Systems with Applications*, vol. 20, no. 4, pp. 325-35, 2001.
- [11] K.W. Cheung, J.T. Kwok, M.H. Law, and K.C. Tsui, "Mining customer product ratings for personalized marketing," *Decision Support Systems*, vol. 35, no. 2, pp. 231-243, 2003.
- [12] W.P. Lee, C.H. Liu and C.C. Lu, "Intelligent agent-based systems for personalized recommendations in Internet commerce," *Expert Systems with Applications*, vol. 22, no. 4, pp. 275-284, 2002.
- [13] O. Byung Kwon, "I know what you need to buy: Context-aware multimedia-based recommendation system," *Expert Systems with Applications*, vol. 25, no. 3, pp. 387-400, 2003.
- [14] R.D. Lawrence, G.S. Almasi, V. Kotlyar, M. Viveros and S.S. Duri, *Personalization of Supermarket Product Recommendations*, Springer, 2001.
- [15] R. Burke, "Hybrid recommender systems: survey and experiments," *User Modeling and User-Adapted Interaction*, vol. 12, no. 4, pp. 331-370, 2002.
- [16] J.B. Schafer, D. Frankowski, J. Herlocker and S. Sen, "Collaborative filtering recommender systems," *The Adaptive Web*, vol. 4321, pp. 291-324, P. Brusilovsky, A. Kobsa and W. Nejdl, eds., Springer Berlin Heidelberg, 2007.
- [17] M. Pazzani and D. Billsus, "Content-based recommendation systems," *The Adaptive Web*, vol. 4321, pp. 325-341, P. Brusilovsky, A. Kobsa and W. Nejdl, eds., Springer Berlin Heidelberg, 2007.
- [18] R. Burke, "Hybrid Web recommender systems," *The Adaptive Web*, vol. 4321, pp. 377-408, P. Brusilovsky, A. Kobsa and W. Nejdl, eds., Springer Berlin Heidelberg, 2007.
- [19] H. Drachler, H.G.K. Hummel and R. Koper, "Personal recommender systems for learners in lifelong learning networks: the requirements, techniques and model," *International Journal of Learning Technology*, vol. 3, no. 4, pp. 404-423, 2008.
- [20] G. Valiente, *Algorithms on Trees and Graphs*, Springer-Verlag, New York, 2002.
- [21] D. Wu, J. Lu and G. Zhang, "Similarity measure models and algorithms for hierarchical cases," *Expert Systems with Applications*, vol. 38, pp. 15049-15056, 2011.
- [22] D. Jungnickel, *Graphs, Networks, and Algorithms*, 3rd ed., Springer-Verlag, Berlin/Heidelberg, 2008.