A Fuzzy Tree Matching-based Personalized e-Learning Recommender System

Dianshuang Wu, Jie Lu, *Senior Member*, *IEEE*, and Guangquan Zhang

Abstract*—The rapid development of e-learning systems provides learners with great opportunities to access learning activities online, and this greatly supports and enhances learning practices. However, an issue reduces the success of application of e-learning systems: too many learning activities (such as various leaning materials, subjects, and learning resources) are emerging in an e-learning system, making it difficult for individual learners to select proper activities for their particular situations/requirements because there is no personalized service function. Recommender systems, which aim to provide personalized recommendations for products or services, can be used to solve this issue. However, e-learning systems need to be able to handle certain special requirements: 1) leaning activities and learners' profiles often present tree structures; 2) learning activities contain vague and uncertain data, such as the uncertain categories that the learning activities belong to; 3) there are pedagogical issues, such as the precedence relations between learning activities. To deal with the three requirements, this study first proposes a *fuzzy tree-structured learning activity model* and a *learner profile model* to comprehensively describe the complex learning activities and learner profiles. In the two models, fuzzy category trees and related similarity measures are presented to infer the semantic relations between learning activities or learner requirements. Since it is impossible to have two completely same trees in practice, a fuzzy tree matching method is carefully discussed. A fuzzy tree matching-based hybrid learning activity recommendation approach is then developed. This approach takes advantage of both the knowledge-based and collaborative filtering-based recommendation approaches, and considers both the semantic and collaborative filtering similarities between learners. Finally, an e-learning recommender system prototype is well designed and developed based on the proposed models and recommendation approach. Experiments are done to evaluate the proposed recommendation approach, and the experimental results demonstrate good accuracy performance of the proposed approach. A comprehensive case study about learning activity recommendation further demonstrates the effectiveness of the fuzzy tree matching-based personalized e-learning recommender system in practice.

***Index Terms*—E-learning, fuzzy sets, knowledge-based recommendation, recommender systems, tree matching.**

Manuscript received April 07, 2014. This work was supported in part by the Australian Research Council (ARC) under discovery grant DP110103733. Thanks to Prof. Ramon López de Mántaras for his valuable suggestions for this paper.

D. Wu, J. Lu and G. Zhang are with the Decision Systems & e-Service Intelligence Lab, Centre for Quantum Computation & Intelligent Systems, Faculty of Engineering and Information Technology, University of Technology Sydney, Australia (e-mail: Dianshuang.Wu@uts.edu.au; Jie.Lu@uts.edu.au; Guangquan.Zhang@uts.edu.au).

I. Introduction

E-LEARNING systems are becoming increasingly popular in educational establishments due to the development of web-based information and communication technologies. The rapid growth of e-learning systems has changed traditional learning behavior and presents a new situation to learners (students), which greatly supports and enhances learning practices online. Due to the emergence of numerous kinds of learning activities (unit of learning [1], which can be subjects, learning materials, resources and so on) in the e-learning environment, learners find it difficult to select the learning activities that best meet their criteria. The information overload problem is increasingly severe in the big data era. It is imperative for an e-learning system to automatically generate personalized recommendations to guide a learner's activities [2], and as demonstrated by Lu [3], an e-learning recommender system is necessary to make personalized recommendations. The motivation of this study is to develop a recommendation approach to support learners in the selection of the most appropriate learning activities in an e-learning environment.

E-learning systems can be divided into two types according to their application environments: a formal setting and an informal setting [4]. A formal setting e-learning system includes learning offers from educational institutions (e.g. universities, schools) within a curriculum or syllabus framework. An informal setting is described in the literature as a learning phase of so-called lifelong learners who are not participating in any formal learning and are responsible for their own learning pace and path [5]. The learning process depends, to a large extent, on individual preferences or choices, and is often self-directed [6]. Different to the formal setting, the informal setting may provide numerous learning activities from different providers, where learners are also from different backgrounds. There is not usually a curriculum or syllabus framework. Therefore, it is very difficult for students to choose proper learning activities in the informal setting. This can cause high drop-out rates and low completion rates [7, 8]. This study focuses on supporting learners in the informal setting through the development of a new personalized recommendation approach.

Recommender systems [9], as one of the most popular applications of personalization techniques, is first proposed and applied in the e-commerce area for product purchase. Recommender systems can be defined as programs that attempt to recommend items to users 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. The basic idea of recommender systems is that similar users like similar items. Therefore, the similarity measure for users or items is vital in the application of recommender systems. Recommender systems have been widely used in various web-based applications in e-commerce, e-business [10, 11], e-tourism [4], e-government [12], but very few in e-learning. The main reason is that e-learning activities have special features and demands that are different to commercial products [1] in e-commerce and e-business, which involve special requirements for recommendation approaches and similarity measures.

1) 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 presents a hierarchical (tree) structure.

2) In a real life situation, learning activities and learner profiles always contain vague and uncertain data. One learning activity may be under several categories with different degrees. For example, the subject *Business Intelligence* is mainly in information technology area but also involves business. A learner's requirements are usually described in linguistic terms such as "highly required" or "very important". Fuzzy set techniques are suitable to deal with these uncertain category data [13-15] and linguistic terms [16, 17]. The tree-structured learning activities and learner profiles are therefore represented as fuzzy trees.

3) The pedagogical issues must be considered in the learning activity recommendation. Some learning activities require prerequisite courses. For example, studying the subject *Data Mining* requires the pre-knowledge about database and algorithms. Additionally, learners always want to learn something new or with higher (more advanced) difficulty levels, so these types of precedence relations among learning activities must be considered.

4) It is not feasible to differentiate between two learning activities just from their IDs or names, because learning activities provided from different schools may have different names, such as one subject is called *Java* and another is called *Program Fundamental*, but the same or similar content.

To deal with the above special requirements in e-learning recommender systems, this study proposes a fuzzy tree matching-based hybrid recommendation approach. Based on our previous research on the fuzzy preference tree-based recommender system [18], a fuzzy tree-structured data model is proposed to describe learner profiles and learning activities. To handle the uncertain issues, fuzzy set techniques are applied. As the similarity measure is the core technique in recommendation approaches, the relevant fuzzy tree similarity measures are developed. The recommendation approach takes advantage of both the knowledge-based and collaborative filtering-based recommendation approaches, and considers both the semantic and collaborative filtering similarities between learners. The learning activity precedence relations are also handled through analyzing the learning sequences and modeling the prerequisite learning activities.

The study presented in this paper is innovative since it is the first to use fuzzy tree-structured data model to model learning activities and learner profiles. It makes contributions to both theoretical and practical issues in the fields of e-learning and recommender systems. At the theoretical level, fuzzy tree-structured data models and related fuzzy tree similarity measures are developed. At the practical level, a fuzzy tree matching-based hybrid recommendation approach for e-learning systems is developed.

The remainder of the paper is organized as follows. Section II reviews the related works on recommendation approaches and e-learning recommender systems. In Section III, the preliminaries on fuzzy tree-structured data models and tree matching method are provided. The fuzzy tree-structured learning activity model and learner profile model are presented in Sections IV and V, respectively. Development of the fuzzy tree matching-based hybrid recommendation approach for learning activities is presented in Section VI. The experimental evaluations and result analysis are given in Section VII. Section VIII outlines the application of the proposed approach to an e-learning recommender system prototype, and a comprehensive case study is given to show its effectiveness. Finally, the proposed approach is summarized in Section IX, and the directions for future study are outlined.

II. Related Works

In this section, the related works on recommendation approaches and e-learning recommender systems are reviewed.

A. *Recommendation Approaches*

Recommendation techniques have attracted much attention and many recommendation approaches have been proposed. In general, the most commonly used three recommendation approaches are collaborative filtering (CF), content-based (CB) and knowledge-based (KB) techniques [19]. The CF technique helps people make choices based on the opinions of other people who share similar interests [20]. It can be further divided into user-based and item-based CF approaches. CB techniques recommend items that are similar to those previously preferred by a specific user [21]. KB techniques offer items to users based on knowledge about the users and items [22]. Each technique has its limitations, such as the item content dependency problem and over-specialization problem for CB [9, 21]; and the cold start and sparsity problems for CF [9]. To gain higher performance and avoid the drawbacks of the typical recommendation approaches, a hybrid recommendation approach can be developed by combining the best features of two or more recommendation approaches into one hybrid approach [23]. A variety of recommendation techniques, such as data mining [24, 25], agents [26] and reasoning, have been developed and applied into recommender systems [27, 28]. Many advanced recommendation approaches, such as social network-based recommender systems [29], fuzzy recommender systems [11, 30], context aware-based recommender systems [31] and group recommender systems [32], have also been proposed recently.

B. E-learning Recommender Systems

Recommender systems have been applied in the e-learning area recently. Zaiane [33] proposed an approach that uses data mining techniques such as association rule mining to model user behaviors and suggest activities or shortcuts. A personalized e-learning material recommender system framework was proposed in [3]. Under the framework, a multi-criteria student requirement analysis model is developed to identify a student's requirements; a fuzzy matching method is used to deal with the uncertain criteria values in real life situation. The CF recommendation approach was adapted to be used in an e-learning context by considering the learners' knowledge levels in [34]. Attributes of materials are considered in the e-learning material recommendations in [5], and CB, CF and some hybrid approaches are used to generate recommendations. To alleviate the stability vs. plasticity problem of technology enhanced learning recommender systems, a recommendation approach that combines a fuzzy collaborative filtering algorithm with a content based one, using learners' preferences and importance of knowledge was proposed in [6]. In order to improve the quality of learning material recommendations, the multi-dimensional attributes of material, rating of learners, and the sequential patterns of the learner's accessed material are considered in [4], where a sequential-based recommendation module was developed to discover the latent patterns of accessing materials, and a learner preference tree was introduced to describe the learner profiles. However, to the best of our knowledge, there has been no research focusing on comprehensively solving the fuzzy tree-structured data in e-learning recommendations.

III. Preliminaries on the Fuzzy Tree-Structured Data Model and Tree Matching Method

This section will define a fuzzy tree-structured data model, which is used to represent tree-structured learning activities or learner profiles. 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. A tree is defined as a directed graph $T = (V, E)$ where the underlying undirected graph has no cycles and there is a distinguished root node in V, denoted by $root(T)$, so that for any node $v \in V$, there is a path in T from $root(T)$ to node v.

The definition only defines the hierarchical relations between 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 Δ can be specified according to specific applications. Two commonly defined similarity measures are concept similarity and value similarity, which are used 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 real applications, the data are usually vague and uncertain. For example, a subject in an 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 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 the tree T in the given numbering, $T[i]$ be the sub-tree rooted at $t[i]$ and $F[i]$ be the unordered forest obtained by deleting $t[i]$ from $T[i]$. Let $t[i_1]$, $t[i_2]$, …, $t[i_{n_i}]$ be the children of $t[i]$.

B. A Tree-Structured Data Matching Method

A tree-structured data matching method is summarized in this sub-section based on our previous research [35-38]. To identify the parts of the two trees that most conceptually correspond, a maximum conceptual similarity tree mapping [38] is constructed. When constructing the mapping, tree structures, node concepts and node weights are all taken into consideration.

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. Another example is matching a sub-tree to a target tree to find out whether the target tree includes the sub-tree, where 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 the latter is called asymmetric matching. The maximum conceptual similarity tree 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 $sc_{T_{sym}}$ and $sc_{T_{asym}}$ when the matching type needs to be specified.

Given two trees $T_1[i]$ and $T_2[j]$ to be compared, their conceptual similarity is calculated as Formula (1). 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.

$$sc_T(T_1[i], T_2[j]) =$$
$$\begin{cases} sc(t_1[i], t_2[j]), & F_1[i]=\phi, F_2[j]=\phi \\ \alpha \cdot sc(t_1[i], t_2[j]) & \\ +(1-\alpha)\cdot \sum_{t=1}^{n_j} w_{j_t} \cdot sc_T(T_1[i], T_2[j_t]), & F_1[i]=\phi, F_2[j]\neq\phi \\ \alpha \cdot sc(t_1[i], t_2[j]) & \\ +(1-\alpha)\cdot \sum_{t=1}^{n_i} w_{i_t} \cdot sc_T(T_1[i_t], T_2[j]), & F_1[i]\neq\phi, F_2[j]=\phi \\ \alpha \cdot sc(t_1[i], t_2[j]) & \\ +(1-\alpha)\cdot sc_F(F_1[i], F_2[j]), & F_1[i]\neq\phi, F_2[j]\neq\phi \end{cases} \quad (1)$$

where w_{j_t} and w_{i_t} are the normalized weights of $t_2[j_t]$ and $t_1[i_t]$, respectively, and α is the influence factor of the parent node. According to the condition of whether $t_1[i]$ and $t_2[j]$ are leaves, four situations are listed in Formula (1). From the formula, the conceptual similarity between two trees considers the attribute conceptual similarity of two roots and also the attributes of any children. 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 calculate $sc_F(F_1[i], F_2[j])$, the conceptual corresponding sub-trees are first identified based on both their concepts and structures, and are then compared separately. These local similarities are then weight aggregated. To find the node pairs that most correspond between the roots of the two forests, a maximum weighted bipartite matching (MWBM) problem [39] 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: $sc_F(F_1[i], F_2[j]) = \sum_{(t_1[i_p], t_2[j_q]) \in M_{ij}} w_{i_p, j_q} \cdot sc_T(T_1[i_p], T_2[j_q])$, where w_{i_p, j_q} is the weight of the matching node pair. If the matching is a symmetric matching, both 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 between two trees can be identified. The roots of two trees are corresponding node pairs. The corresponding nodes in the children of two roots are then identified based on two roots' children's maximum weighted bipartite matching. Other corresponding nodes are identified in the same way.

The computation complexity of the tree-structured data matching method is analyzed. When computing $sc_F(F_1[i], F_2[j])$, the maximum weighted bipartite matching method in [39] is applied, whose complexity is bounded by $O(n_i \times n_j \times \min\{n_i, n_j\})$. The complexity of computing $sc_T(T_1[i], T_2[j])$ for any node pair $t_1[i]$ and $t_2[j]$ is then obtained, which is bounded by $O(n_i \times n_j \times \min\{n_i, n_j\})$. Therefore, the complexity of the whole method is $\sum_{i=1}^{|T_1|} \sum_{j=1}^{|T_2|} O(n_i \times n_j \times \min\{n_i, n_j\}) \leq O(|T_1| \times |T_2| \times \sqrt{\deg(T_1) \cdot \deg(T_2)})$. In our system, the tree matching method is used to match two tree-structured learner profiles or learning activities. Because the node number of each learner profile or learning activity is limited, the complexity is acceptable.

IV. Fuzzy Tree-Structured Learning Activities

In this section, the data structure of learning activities in the e-learning recommender system is presented. A learning activity can be described from several perspectives, such as the prerequisite courses, the categories, the content, the lecture, and so on, and some features may be described from several sub-features, which construct a hierarchical tree structure. Some features of a learning activity are uncertain in real applications. For example, the subject *Business Intelligence* mainly belongs to the category of information technology, but also belongs to the category of business to some degree. To deal with the tree-structured data and fuzzy category data in learning activities, the fuzzy tree-structured data model is, therefore, used to model the learning activities in our system. The structure of a learning activity is illustrated in Fig. 1. In the learning activity tree, each node is assigned a *label* attribute, as shown in the figure. Some nodes are assigned a *category* attribute. The node concept similarity is calculated based on the two attributes. If two nodes are assigned *category*, the category similarity will be taken as the node concept. Otherwise, their labels are compared.

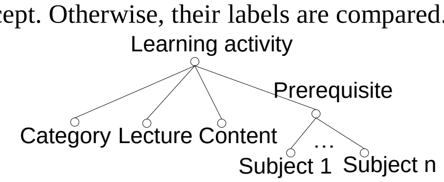

Fig. 1. The structure of a learning activity.

The *category* of a learning activity is an important attribute to infer the semantic relations between different learning activities. In a real life situation, one learning activity may belong to several categories with different degrees. Therefore, the value of a *category* is a fuzzy category tree in our system. The fuzzy category trees and their similarity measure are presented in detail as follows.

A. Fuzzy Category Tree and the Fuzzy Category Similarity

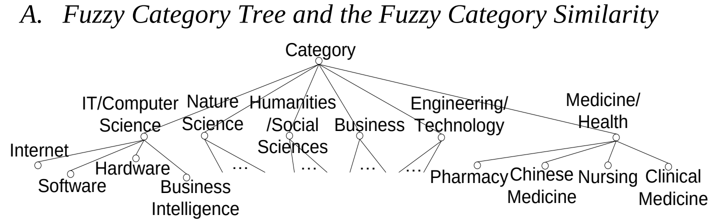

Fig. 2. The learning activity category tree.

To divide the learning activities, a learning activity category is usually introduced in the e-learning system. The learning activity category defined in our system is shown in Fig. 2. It 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, the "IT/Computer Science" category can be divided into four sub-categories, which are "Internet", "Software", "Hardware", and "Business Intelligence".

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 that belongs to the sub-category. The sub-categories and corresponding membership degrees are specified by the learning activity providers when they insert the learning activities into the system. To represent the categories of a learning activity, a fuzzy category tree is defined.

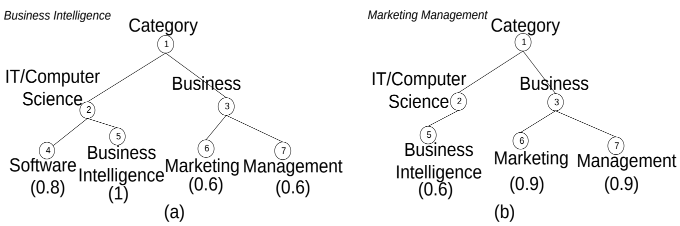

Fig. 3. Two fuzzy category trees of two learning activities: (a) is the fuzzy category tree of the subject *Business Intelligence*. (b) is the fuzzy category tree of the subject *Marketing Management*.

Definition 4. A fuzzy category tree of a learning activity represents the categories the learning activity belongs to, which is a sub-tree of the learning activity category tree. The nodes of the fuzzy category tree are assigned category values, which represent the membership degrees of the learning activity belonging to the relevant sub-categories.

Two examples of fuzzy category tree are shown in Fig. 3.

Let $v_c(t[i])$ represent the category value of node $t[i]$. If a learning activity does not belong to the sub-category represented by node $t[i]$, $v_c(t[i]) = 0$. The category value of $T[i]$, the sub-tree under the node $t[i]$, can be inferred from the category values of nodes in the sub-tree $T[i]$, which is calculated by Formula (2).

$$v_c(T[i]) = \begin{cases} v_c(t[i]), & F[i] = \phi \\ (\bigvee_{j=1}^{n_i} v_c(T[i_j])) \vee v_c(t[i]), & F[i] \neq \phi \end{cases}. \quad (2)$$

Similarly, the category value of the forest $F[i]$ can be defined, and calculated by

$$v_c(F[i]) = \begin{cases} 0, & F[i] = \phi \\ \bigvee_{j=1}^{n_i} v_c(T[i_j]), & F[i] \neq \phi \end{cases}. \quad (3)$$

The category value of the sub-tree $T[i]$ or the forest $F[i]$ will be 0, if the learning activity is not relevant to the categories under the sub-tree $T[i]$ or the forest $F[i]$.

1) Fuzzy Category Similarity

The similarity measure between categories is necessary to evaluate the semantic similarity between learning activities, which is vital to make recommendations. Because the category for each learning activity is represented as a fuzzy category tree, the traditional node distance based method cannot be used. A fuzzy category tree similarity measure is developed in this sub-section.

As the fuzzy category trees are all based on the learning activity category tree shown in Fig. 2, the numbering of the learning activity category tree is used to represent tree nodes. Let $T_1[i]$ and $T_2[i]$ represent two fuzzy category trees of two learning activities a_1 and a_2, respectively. To evaluate the similarity between two fuzzy category trees, the values of all nodes must be taken into account. According to the fuzzy category tree definition, four properties of the fuzzy category trees can be discovered: 1) the structures of $T_1[i]$ and $T_2[i]$ are the same as they are based on the same category tree; 2) only the sub-trees with positive category values need to be considered when calculating the similarity as the sub-trees with zero category values are not relevant; 3) the category values may be assigned to nodes at different levels; 4) category values in different levels present different weights. When calculating the similarity between two category trees, all these properties must be considered. According to the conditions whether the children of $t_1[i]$ and $t_2[i]$ are assigned positive values or zero, four situations are considered in the formula. The fuzzy category similarity between a_1 and a_2 is calculated as:

$$s_{fc}(T_1[i], T_2[i]) = \begin{cases} v_c(t_1[i]) \wedge v_c(t_2[i]), & v_c(F_1[i]) = 0, v_c(F_2[i]) = 0 \\ v_c(t_1[i]) \wedge v_c(T_2[i]), & v_c(F_1[i]) = 0, v_c(F_2[i]) \neq 0 \\ v_c(T_1[i]) \wedge v_c(t_2[i]), & v_c(F_1[i]) \neq 0, v_c(F_2[i]) = 0 \\ (\alpha^{h-d_i} - \alpha^h) \cdot (v_c(T_1[i]) \wedge v_c(T_2[i])) & \\ + (1 - \alpha^{h-d_i} + \alpha^h) \cdot (\bigvee_{j=1}^{n_i} s_{fc}(T_1[i_j], T_2[i_j])), & v_c(F_1[i]) \neq 0, v_c(F_2[i]) \neq 0 \end{cases}, \quad (4)$$

where α is the influence factor of the parent node, h is the height of the learning category tree, and d_i is the depth of node i in the category tree. In the first situation, $v_c(F_1[i]) = 0$ and $v_c(F_2[i]) = 0$, which means that $t_1[i]$ and $t_2[i]$ have no children nodes or their children nodes are not assigned positive values. Therefore, only the values of $t_1[i]$ and $t_2[i]$ are considered. In the second situation, $t_1[i]$ has no children or its children nodes are not assigned positive values. Thus, the two trees $T_1[i]$ and $T_2[i]$ can only be compared at the level of $t_1[i]$. The third situation is similar to the second one. In the fourth situation, the children of both $t_1[i]$ and $t_2[i]$ are assigned positive values. Therefore, the lower levels of $t_1[i]$ and $t_2[i]$ should also be compared. As the categories in the lower level are more specific, the lower level should gain more weight in the similarity measure. The coefficient α^{h-d_i} in Formula (4) reflects the point. To guarantee that the similarity between different general categories be 0, α^h is subtracted from α^{h-d_i} in the formula.

Take two subjects, *Business Intelligence* and *Marketing Management*, which are illustrated in Fig. 3, as examples. Let α be 0.5. In the example, h=2. $s_{fc}(T_a[4], T_b[4]) = 0$; $s_{fc}(T_a[5], T_b[5]) = 0.6$; $v_c(T_a[2]) = v_c(T_a[4]) \vee v_c(T_a[5]) = 1$, $v_c(T_b[2]) = 0.6$, $s_{fc}(T_a[2], T_b[2]) = (\alpha^{h-1} - \alpha^h) \cdot (v_c(T_a[2]) \wedge v_c(T_b[2])) + (1 - \alpha^{h-1} + \alpha^h) \cdot (s_{fc}(T_a[4], T_b[4]) \vee s_{fc}(T_a[5], T_b[5])) = 0.6$; similarly, $s_{fc}(T_a[3], T_b[3])$ =0.6; the fuzzy category similarity between these two subjects is calculated as $s_{fc}(T_a[1], T_b[1]) = s_{fc}(T_a[2], T_b[2]) \vee s_{fc}(T_a[3], T_b[3])$=0.6.

2) Fuzzy Category Tree Combination

In practice, there are times when the fuzzy category trees need to be combined. For example, a learner has completed several learning activities. To examine the categories learned by the learner comprehensively, the categories of all the learning activities learned by the user should be combined. A fuzzy category tree combination procedure $combine(\cdot)$ is presented in this sub-section.

Definition 5. Let $S_{T_c} = \{T_1[i], T_2[i], \ldots, T_m[i]\}$ represent a set of fuzzy category trees. The combination of the fuzzy category trees in S_{T_c} is denoted as $T_c[i] = combine(S_{T_c})$. For each node $t_c[j]$ in $T_c[i]$, $v_c(t_c[j]) = \vee_{k=1}^{m} v_c(t_k[j])$.

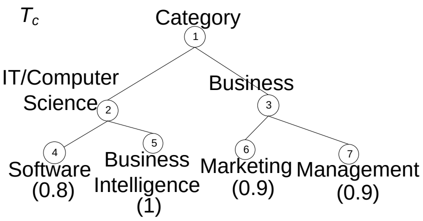

Fig. 4. The combination of two fuzzy category trees in Fig. 3.

For example, the combination of two fuzzy category trees in Fig. 3 is shown in Fig. 4.

B. The Pedagogical Relations between Learning Activities

In the learning activity recommendation, the learning process, which is concerned with repeatability, periodicity and some dependency relations, must be considered [4]. Recommended learning activities must be new, or have a level slightly above the learners' current competence level [4, 40]. For some learning activities with similar content, or under the similar categories, it is not reasonable to recommend the elementary activities to a learner if he/she has already learned some advanced activities. Our system considers two kinds of precedence relations between learning activities.

The first kind of precedence relations are derived from the prerequisites of learning activities. Many learning activities have prerequisite courses. These prerequisite learning activities are specified for the learning activity and described in the fuzzy tree-structured learning activity profiles. The second kind of precedence relations are derived from learning sequences in learners' learning history. These learning sequences can be used to infer the advanced levels of learning activities, which are difficult to identify due to the open environment in the informal learning setting. Some sequential feature factors are defined as follows to identify the sequential relations between learning activities from the learning sequences.

1) The Sequential Relation between Learning Activities

For a learning activity a learned by a learner, there is a starting time t_s and a finishing time t_f. Obviously, $t_s(a) < t_f(a)$. Let a_1 and a_2 be two learning activities which are both learned by a learner. According to Allen's interval algebra [41], there are thirteen temporal relations between a_1 and a_2. In this study, only the precedence relations are concerned. The following three sequential relations are considered: 1) a_1 is prior to a_2, denoted as $a_1 \to a_2$, if $t_f(a_1) \le t_s(a_2)$; 2) a_2 is prior to a_1, denoted as $a_2 \to a_1$, if $t_f(a_2) \le t_s(a_1)$; 3) a_1 and a_2 are concurrent, if $t_s(a_1) < t_f(a_2)$ and $t_s(a_2) < t_f(a_1)$. In the learning history, the relevant learning times of the learning activities for each learner are recorded.

To analyze the sequential relations between learning activities from the whole learners' learning histories, the following coefficients are defined. Let the support of a learning activity set L, $support(L)$, be defined as the percentage of the learners who learned all the activities in L in all learners. $support(\{a_1, a_2\})$ represents the proportion of learners who learned both a_1 and a_2. $support(\{a_1 \to a_2\})$ represents the proportion of learners who learned a_1 before a_2. A prior relation confidence coefficient is defined as:

$$priorc(a_1 \to a_2) = \frac{support(\{a_1 \to a_2\})}{support(\{a_1, a_2\})}. \tag{5}$$

When learning activities a_1 and a_2 satisfy a minimum prior relation confidence and a minimum support threshold, i.e., $priorc(a_1 \to a_2) > priorc_{thres}$ and $support(\{a_1, a_2\}) > support_{thres}$, it indicates that there is a dependency relation between a_1 and a_2, and a_1 is usually learned before a_2. If a learner has learned a_2, it will not be suitable to recommend a_1 to him/her. A sequence set S_{prior} is used to record these relations. S_{prior} is constructed offline periodically.

V. Fuzzy Tree-Structured Learner Profiles

When a learner selects a learning activity, various kinds of information, such as the learner's background, learning goal, required learning categories, and learned learning activities, will influence the learner to make the decision. In our recommender system, all these aspects of information are taken into consideration when making recommendations. Each aspect usually contains several sub-aspects. This constructs a tree structure. The structure of a fuzzy tree-structured learner profile is illustrated in Fig. 5. In real applications, learners are more likely to express their requirements with linguistic terms, such as "very high required". Fuzzy set techniques are suitable to handle these linguistic terms. Therefore, the fuzzy tree-structured data model is applied to model the learner profiles.

This section explains the fuzzy tree-structured learner profile model. Similar to the learning activity tree, the learner profile tree nodes are assigned a *label* attribute and a *category* attribute, which are used to calculate the node concept similarity.

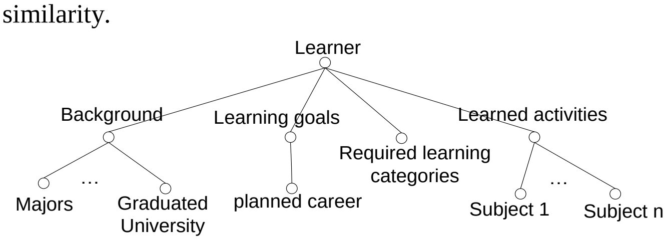

Fig. 5. The structure of a learner profile.

In the learner profile, the learned activities are recorded by the system during the learning process. Other information, such as the learner's background, planned career and required learning categories, is specified by the learner when the learner registered. In particular, the planned career is selected from a predefined career list, such as *Software Engineer*, *Developer Programmer*, *Accountant*, etc. The required learning categories are selected by learners from the learning activity category tree, which is illustrated in Section IV, and the requirement levels of the sub-categories are specified. In a real life situation, the requirements are usually uncertain and described by linguistic terms. Thus, the category requirements are represented as fuzzy required category trees.

Our recommender system defines a linguistic term set *R*={*Very low required* (*VLR*), *Low required* (*LR*), *Medium required* (*MR*), *High required* (*HR*), *Very high required* (*HR*)} for learners to express their requirements for a specific learning category. To handle these linguistic terms in the recommendation calculation process, fuzzy set technology is, therefore, used [42]. A set of triangular fuzzy numbers is applied to deal with these linguistic terms [43, 44]. The related fuzzy numbers to these linguistic terms are shown in Table I.

TABLE I. LINGUISTIC TERMS AND RELATED TRIANGULAR FUZZY NUMBERS FOR LEARNER REQUIREMENT

VLR	*LR*	*MR*	*HR*	*VHR*
(0,0,0.25)	(0,0.25,0.5)	(0.25,0.5,0.75)	(0.5,0.75,1)	(0.75,1,1)

Two examples of a fuzzy required category tree are shown in Fig. 6. The linguistic terms under the nodes represent the learner's requirement. It can be seen from the examples that learners' requirements can be specified at different levels. For each branch of the tree, only one node is assigned to the user's requirement.

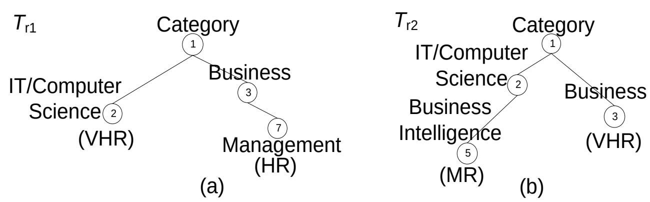

Fig. 6. Two fuzzy required category trees.

A. The Similarity Measures Related to the Fuzzy Required Category Tree

In the recommendation generation process, the similarity measure is necessary to find similar users or items and to match suitable items to users' requirements. In this section, the similarity between two learners' fuzzy required category trees is presented to assist to compare the two learners, and the matching similarity of a learning activity's fuzzy category tree to a learner's fuzzy required category tree is given to help select proper learning activities.

1) Fuzzy Required Category Similarity

Let T_{r1} and T_{r2} be two fuzzy required category trees. The similarity measure between T_{r1} and T_{r2} is given in this sub-section. As learners' fuzzy required category trees are based on the learning activity category tree, T_{r1} and T_{r2} have the same base structure and labels. We use the numbering of the learning activity category tree to represent the nodes in T_{r1} and T_{r2}. The fuzzy required category similarity between T_{r1} and T_{r2} is calculated by

$$s_{frc}(T_{r1}[i],T_{r2}[i]) = \begin{cases} s_v(v_r(t_{r1}[i]),v_r(t_{r2}[i])), & v_r(F_{r1}[i])=0, v_r(F_{r2}[i])=0 \\ s_v(v_r(t_{r1}[i]),v_r(T_{r2}[i])), & v_r(F_{r1}[i])=0, v_r(F_{r2}[i])\neq 0 \\ s_v(v_r(T_{r1}[i]),v_r(t_{r2}[i])), & v_r(F_{r1}[i])\neq 0, v_r(F_{r2}[i])=0 \\ (\alpha^{h-d_i}-\alpha^h)\cdot s_v(v_r(T_{r1}[i]),v_r(T_{r2}[i]))+ & \\ (1-\alpha^{h-d_i}+\alpha^h)\cdot\sum_{j=1}^{n_i} w_j\cdot s_{frc}(T_{r1}[i_j],T_{r2}[i_j]), & v_r(F_{r1}[i])\neq 0, v_r(F_{r2}[i])\neq 0 \end{cases},\quad (6)$$

where α is the influence factor of the parent node, h is the height of the learning category tree, and d_i is the depth of node i in the category tree; $w_j = (w(t_{r1}[i_j]) + w(t_{r2}[i_j]))/2$; $v_r(t_{r1}[i])$ represents the value of node $t_{r1}[i]$, which is a fuzzy number; $v_r(F_{r1}[i])$ represents the value of forest $F_{r1}[i]$, which is 0 if $F_{r1}[i]$ is null or none of its nodes are assigned values; $v_r(T_{r1}[i])$ represents the value of the sub-tree $T_{r1}[i]$, which is calculated by Formula (7); $s_v(\cdot)$ is the similarity measure for two fuzzy numbers.

$$v_r(T_r[i]) = \begin{cases} v_r(t_r[i]), & v_r(F_r[i])=0 \\ \sum_{j=1}^{n_i} w(t_r[i_j])\cdot v_r(T_r[i_j]), & v_r(F_r[i])\neq 0 \end{cases}. \quad (7)$$

For two fuzzy numbers $\tilde{a}$, $\tilde{b}$,

$$s_v(\tilde{a},\tilde{b}) = 1 - d(\tilde{a},\tilde{b})/d_{\max}, \quad (8)$$

where $d(\tilde{a},\tilde{b}) = \left(\int_0^1 \frac{1}{2}[(a_\lambda^L - b_\lambda^L)^2 + (a_\lambda^R - b_\lambda^R)^2]\, d\lambda\right)^{\frac{1}{2}}$ is the distance between fuzzy numbers $\tilde{a}$ and $\tilde{b}$, $d_{\max}$ is the maximum distance between fuzzy numbers in the domain.

Let α be 0.5. Taking the two learner requirement trees in Fig. 6 as an example, the fuzzy required category similarity between them is computed by $s_{frc}(T_{r1}[1], T_{r2}[1]) = w_2 \cdot s_v(v_r(t_{r1}[2]), v_r(T_{r2}[2])) + w_3 \cdot s_v(v_r(T_{r1}[3]), v_r(t_{r2}[3]))$, and calculated as 0.675.

2) Fuzzy Category Matching Similarity

Let T_r be a learner's fuzzy required category tree, and T_c represent the fuzzy category tree of a learning activity. The fuzzy category matching similarity measure of T_c to T_r is calculated by Formula (9).

$$s_{fcm}(T_r[i],T_c[i]) = \begin{cases} s_v(v_r(t_r[i]),v_c(t_c[i])), & v_r(F_r[i])=0, v_c(F_c[i])=0 \\ s_v(v_r(t_r[i]),v_c(T_c[i])), & v_r(F_r[i])=0, v_c(F_c[i])\neq 0 \\ s_v(v_r(T_r[i]),v_c(t_c[i])), & v_r(F_r[i])\neq 0, v_c(F_c[i])=0 \\ (\alpha^{h-d_i}-\alpha^h)\cdot s_v(v_r(T_r[i]),v_c(T_c[i]))+ & \\ (1-\alpha^{h-d_i}+\alpha^h)\cdot\sum_{j=1}^{n_i} w(t_r[i_j])\cdot s_{fcm}(T_r[i_j],T_c[i_j]), & v_r(F_r[i])\neq 0, v_c(F_c[i])\neq 0 \end{cases}. \quad (9)$$

In respect of the fuzzy category matching similarity measure, first, the category values of nodes in T_c are real numbers, which will be seen as special fuzzy numbers in the similarity measure $s_v(\cdot)$. Second, the similarity between T_r and T_c is asymmetric, and only the weights of T_r are considered.

Taking the fuzzy required category tree T_{r1} in Fig. 6 and the two fuzzy category trees of *Business Intelligence* and *Marketing Management* illustrated in Fig. 3 as examples, the matching similarity of *Business Intelligence* to T_{r1} is computed by $s_{fcm}(T_{r1}[1], T_a[1]) = w_2 \cdot s_v(v_r(t_{r1}[2]), v_c(T_a[2])) + w_3 \cdot s_{fcm}(T_{r1}[3], T_a[3])$, and calculated as 0.845. Similarly, the matching similarity of *Marketing Management* to T_{r1} is calculated as 0.722. Because T_{r1} expresses very high requirement on "IT" category, and the degree of *Business Intelligence* belonging to "IT" is higher than that of *Marketing Management*, the calculated matching similarity degrees reflect the requirement.

VI. A Fuzzy Tree Matching-Based Hybrid Recommendation Approach

This section outlines the development of a fuzzy tree matching-based hybrid recommendation approach for learning activities. For a target learner u_t, the recommendation process is described in seven steps, as follows.

A. Step 1: Determine the Recommendation Alternatives

There are numerous learning activities under various categories in an e-learning system, but for a specific target learner, only the learning activities under certain relevant categories are suitable for recommendation. To improve the recommendation efficiency, the relevant categories of the target learner are first identified, and the learning activities under the categories are then selected.

The relevant learning categories of the target learner u_t are identified in two ways: the learning activities that have been learned by u_t and other learners with the same learning goals; and the fuzzy required category tree T_{frc} of u_t. Let the learning goal of u_t be g_t. The learners whose learning goal is g_t are selected to constitute a set U_{g_t}. For each learner $u_i \in U_{g_t}$, the learned activities are $\{a_{i,1}, a_{i,2}, \dots, a_{i,n_i}\}$, and the corresponding fuzzy category trees are $\{T_{i,1}, T_{i,2}, \dots, T_{i,n_i}\}$. The learned category tree of u_i, denoted as T_i, can be calculated as $T_i = combine(\{T_{i,1}, T_{i,2}, \dots, T_{i,n_i}\})$. The learned category trees of all the users in U_{g_t} are combined, and the learned category tree for the learning goal g_t is obtained and denoted as T_{g_t}. A fuzzy category tree T_{cr} is derived from the learner's fuzzy required category tree by setting the membership degrees of leaf nodes in T_{frc} as 1. The relevant learning category tree is obtained by combing T_{g_t} and T_{cr}, as $T_{cr} = combine(\{T_{cr}, T_{g_t}\})$. For any learning activity a with fuzzy category tree T_{ca}, if $s_{fc}(T_{ca}, T_{cr}) > 0$, it is preselected.

The pedagogical constraints are considered when preselecting the learning activities. 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 $T_{t,l}$. The learned activities are $\{a_{t,1}, a_{t,2}, \dots, a_{t,n_t}\}$. For a learning activity a, the sequential and prerequisite constraints are verified separately. For the sequential constraints, if $\exists (a \rightarrow a_{t,i}) \in S_{prior}$, $1 \le i \le n_t$, a will not be suitable for recommendation. For the prerequisite constraints, let the learning activity's prerequisite sub-tree be denoted as $T_{a,p}$. 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. A sub-tree match is calculated as $s_a = sc_{T_{asym}}(T_{a,p}, T_{t,l})$. A matching similarity threshold s_{thres} is predefined. If $s_a > s_{thres}$, then learning activity a can be selected as a recommendation alternative.

By using this step, a set of recommendation alternatives are chosen. For each alternative learning activity a, the following steps are taken to predict its rating.

B. Step 2: Calculate the Matching Degree of the Learning Activity a to the Learner's Requirement

The learner u_t's fuzzy required category tree is T_{req}, and the learning activity a's fuzzy category tree is T_{ca}. The matching degree of a to u_t is calculated by Formula (10):

$$s_m(u_t, a) = s_{fcm}(T_{req}, T_{ca}) . \quad (10)$$

C. Step 3: Calculate the Semantic Similarity between Users

The users who have rated a 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:

$$s_{sem}(u_t, u_i) = sc_{T\,sym}(T_t, T_i) . \quad (11)$$

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 learning activities rated by u_t and u_i, respectively.

D. Step 4: Calculate the CF Similarity between Users

A learning activity similarity threshold as_t is predefined. For any learning activity pair (p,q), T_p and T_q are the tree-structured profiles. p and q will be shown to be irrelevant if the similarity between T_p and T_q, $s_{sem}(p,q) = sc_{T\,sym}(T_p, T_q)$ is less than as_t. Given the matched learning activity set $M_{t,i}$ of u_t and u_i, a sub-set $M'_{t,i} = \{(p,q): (p,q) \in M_{t,i}, s_{sem}(p,q) > as_t\}$ is selected. Based on $M'_{t,i}$, the CF similarity between u_t and u_i is calculated as:

$$s_{CF}(u_t, u_i) = \frac{\sum_{(p,q)\in M'_{t,i}} r_{t,p} \times r_{i,q}}{\sqrt{\sum_{(p,q)\in M'_{t,i}} {r_{t,p}}^2} \times \sqrt{\sum_{(p,q)\in M'_{t,i}} {r_{i,q}}^2}} , \quad (12)$$

where $r_{t,p}$ is the rating of item p from user u_t.

E. Step 5: 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.

$$s_u(u_t, u_i) = \beta \times s_{sem}(u_t, u_i) + (1-\beta) \times s_{CF}(u_t, u_i) , \quad (13)$$

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 sorted according to the total similarity. The top-*N* most similar users are selected as neighbors to predict ratings.

F. Step 6: Calculate the Predicted Rating

The predicted rating to learning activity a of learner u_t is calculated as:

$$pr_{t,a} = \theta \times s_m(u_t, a) \times r_{\max} + (1-\theta) \times \frac{\sum_{i=1}^{N} r_{i,a} \times s_u(u_t, u_i)}{\sum_{i=1}^{N} s_u(u_t, u_i)} , \quad (14)$$

where $\theta \in [0,1]$, $r_{\max}$ represents the maximum value of ratings. The formula contains two parts. $s_m(u_t, a) \times r_{\max}$ is the requirement matching-based predicted rating. If the target learning activity is exactly matched to the user's requirement, the target item should achieve the highest rating. $\sum_{i=1}^{N} r_{i,a} \times s_u(u_t, u_i) / \sum_{i=1}^{N} s_u(u_t, u_i)$ is the traditional item-based CF-based predicted rating. θ is a parameter that combines the two parts.

G. *Step 7: 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.

VII. Experimental Evaluation

This section presents the performance evaluation results of the proposed fuzzy tree matching-based hybrid recommendation approach, which includes the experiment design and the result analysis.

A. *Experiment Design*

Due to the lack of any well-known dataset publicly accessible for research in e-learning recommendation area, we used a well-known recommender system dataset, the MovieLens dataset (http://www.grouplens.org/node/73). Although it is different from e-learning data, other researchers in e-learning recommender systems, such as Bobadilla et al. [34], also used this dataset. In the dataset, each movie is described from several aspects, such as genres, directors and actors, and each aspect contains several sub-aspects of information. The data construct tree structures naturally, which is suitable to be handled with our approach. In this study, we treat movies as learning activities, and treat the movie users as learners. Therefore, without loss of generality, the MovieLens dataset was used in this experiment.

There are 2113 users in the dataset, and each user rated at least 20 movies. There are 20 movie genres in the MovieLens dataset, including Action, Adventure, Animation, Children, Comedy, Crime, Documentary, Drama, Fantasy, Film-Noir, Horror, IMAX, Musical, Mystery, Romance, Sci-Fi, Short, Thriller, War and Western. Each movie may belong to several genre categories. Thus, the category of a movie is represented as a category tree as illustrated in Section IV. A user's profile contains the rated movies and the required category of the user, which construct the tree-structured user profile. The tree structures of the movie representation and the user profile are illustrated in Fig. 7.

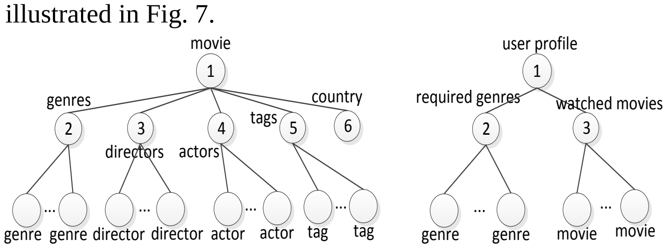

Fig. 7. The tree structures of the movie and user profile.

In this experiment, the ratings of each user are sorted according to the chronological order, and the most recent ratings of each user make up the testing set. Three groups of training and testing sets, which take the 20%, 40% and 50% most recent ratings of each user as testing sets respectively, are constructed.

Since the recommendation approach for e-learning recommender systems proposed by Bobadilla et al. [34] was also evaluated by use of the MovieLens dataset, to show the effectiveness of our proposed approach, it is compared with Bobadilla's approach.

B. *Experiment Results*

In this experiment, the accuracy performance of the approaches, which is assessed with the mean absolute error (MAE), was assessed and compared. The MAE of Bobadilla's approach and our proposed approach on the three testing sets are illustrated in Fig. 8. It can be seen from the results that our proposed approach has better and more stable accuracy performance. The accuracy performance of our proposed approach is improved compared to Bobadilla's approach by 25.9%, 23.9% and 21.3% on the 50%, 40% and 20% testing sets respectively. The reason to have the results is that our proposed approach fully utilizes the semantic information and requirement matching knowledge.

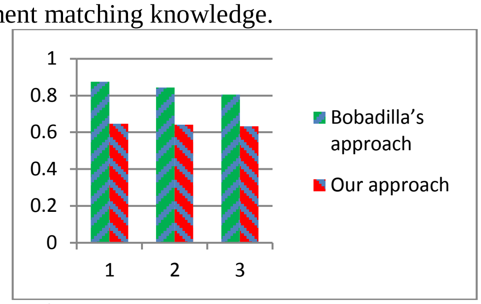

Fig. 8. MAE of the recommendation approaches: (1) on the 50% testing set; (2) on the 40% testing set; (3) on the 20% testing set.

VIII. An E-learning Recommender System Prototype and a Case Study

This section outlines the design and implementation of the fuzzy tree matching-based e-learning recommender system (TeLRS) prototype according to the proposed recommendation approach. A case study is presented to show the effectiveness of the system.

A. *System Architecture*

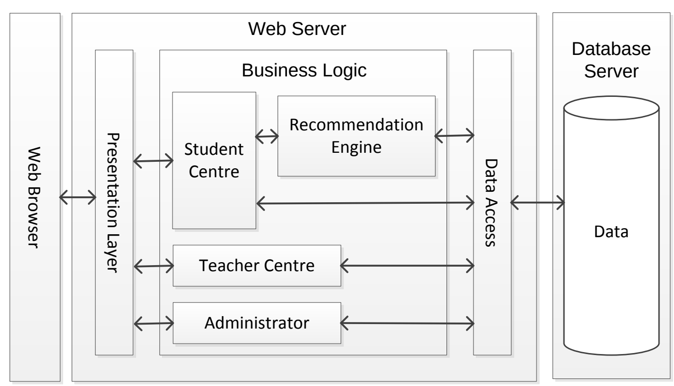

Fig. 9. The architecture of the e-learning recommender system.

The e-learning recommender system TeLRS is designed to have three types of users: system administrators, teachers and students. The roles of the users are described as follows.

The role of the system administrator is to maintain the structure of the learning activity category and the career list of learners, which are used to support the operation of the system.

The teachers are responsible for managing the learning activities. They input the learning activities with detailed descriptions into the system. When a learning activity is input, its categories and the related membership degrees are specified by the teacher. During the operation of the system, teachers obtain feedback from students about their learning activities and interact with the students.

The students are searching the appropriate learning activities and want to receive recommendations. They provide their background information and learning requirements when registering in the system. After finishing a learning activity, the student can provide feedback and rate the learning activity.

The architecture of the e-learning activity recommender system is depicted in Fig. 9. As a web-based online system, the e-learning recommender system has a standard multi-tier architecture, which includes web browser, web server, and database server. The main components of the system are described as follows. The database stores all the data of the system, which includes the data of user profiles, learning activities, learning activity categories, user ratings, and so on. The application in the web server contains three layers: the presentation layer, business logic layer and data access layer. The presentation layer is responsible for generating the requested web pages and handling the user interface logic and events for the three kinds of users. The business logic layer realizes the business services and the core recommendation algorithm. It contains four main parts: the student centre, the teacher centre, the administrator centre, and the recommendation engine. The student centre collects the user's profile and requirements, tracks the user's learning behavior, and provides the search and recommendations of learning activities. The recommendation engine implements the proposed recommendation approach and generates recommendations for student users. Teachers input and manage the learning activities in the teacher centre. The administrator centre is used by administrators to manage the users and common data. The data access layer deals with the data operations of the database.

B. System Implementation

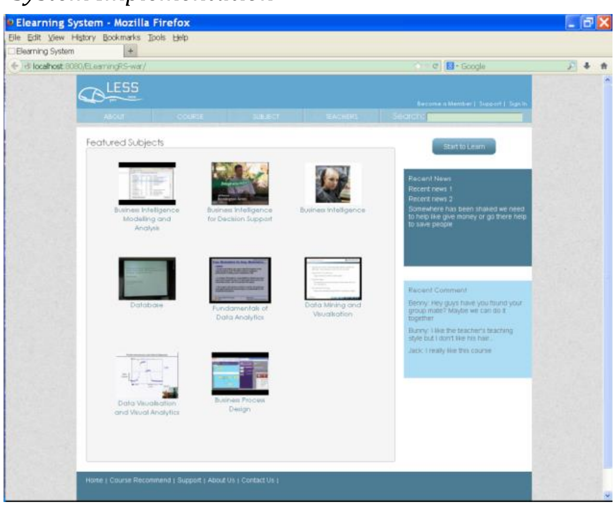

Fig. 10. The homepage of the e-learning recommender system.

The system is developed and implemented using the Netbeans development platform. JSF, EJB and JPA frameworks are used in the implementation of the presentation layer, business logic layer and data access layer. The database is designed and implemented in the PostgreSQL database server. To test the recommender system, it is deployed in the Glassfish web server. Fig. 10 shows the home page of the e-learning recommender system.

C. A Case Study

In the e-learning recommender system, there are five learners (Leaner 1, …, Leaner 5) and eight subjects (S1-*Business Intelligence*, …, S8-*Business Process Design*). The fuzzy tree-structured learner profiles are described in Fig. 11, and the fuzzy category trees of the subjects are shown in Fig. 12.

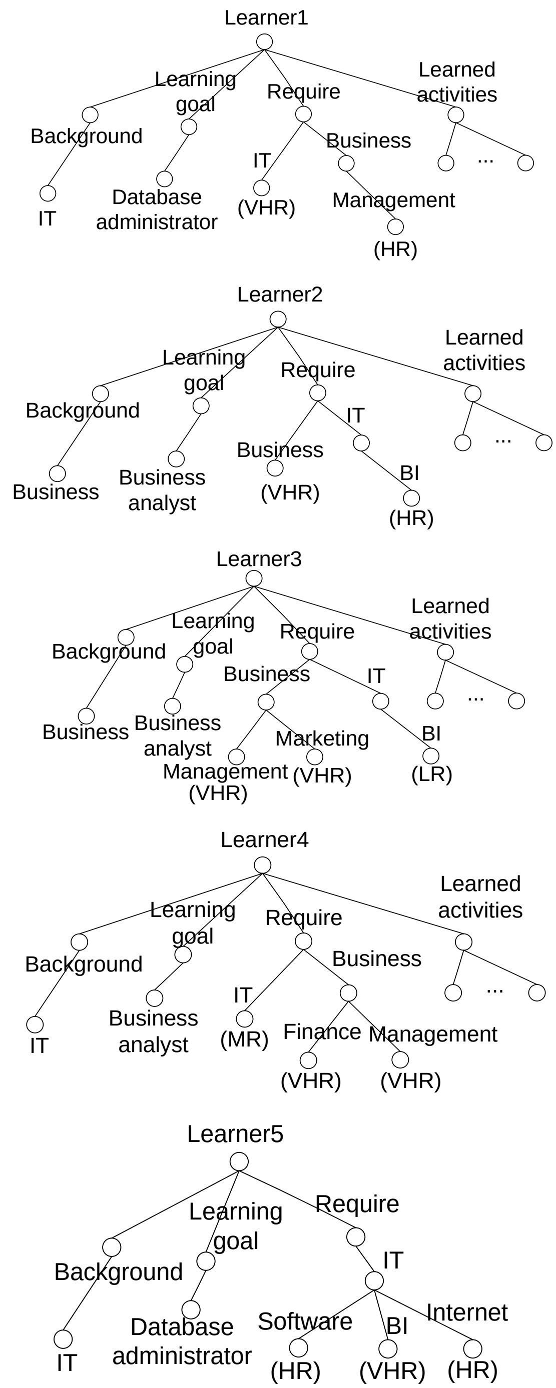

Fig. 11. Five learner profiles.

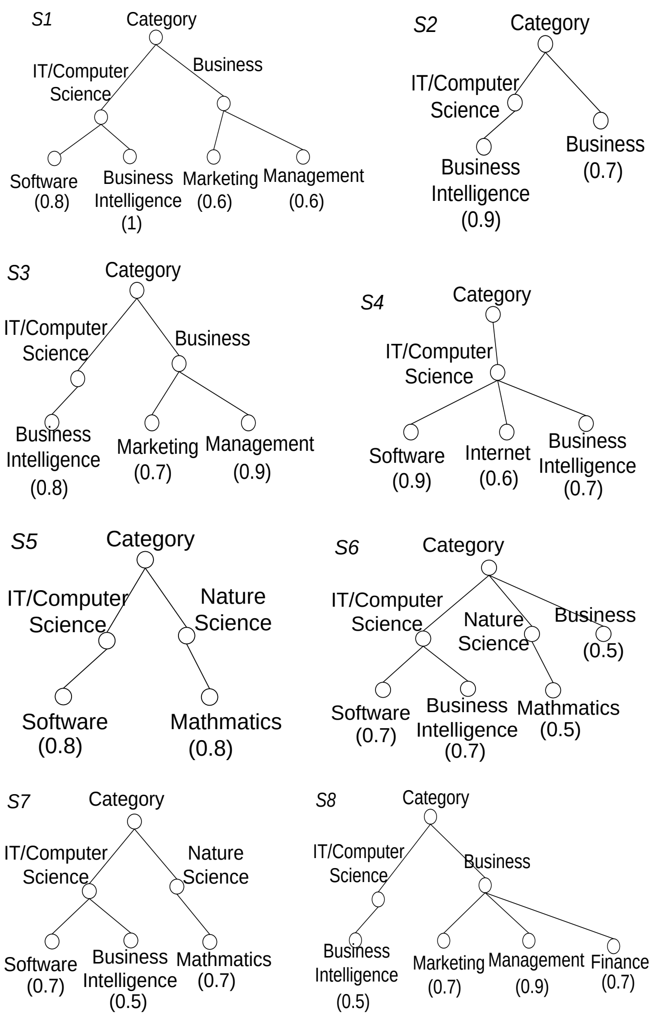

Fig. 12. The fuzzy category trees of the subjects.

When a learner signs into the system, he/she can edit the profile. The learner's background, learning goal, and preferred learning categories can be specified. For example, Fig. 13 shows the profile editing page of Learner 4, in which the learner's required categories construct a tree structure and the required levels are expressed by linguistic terms.

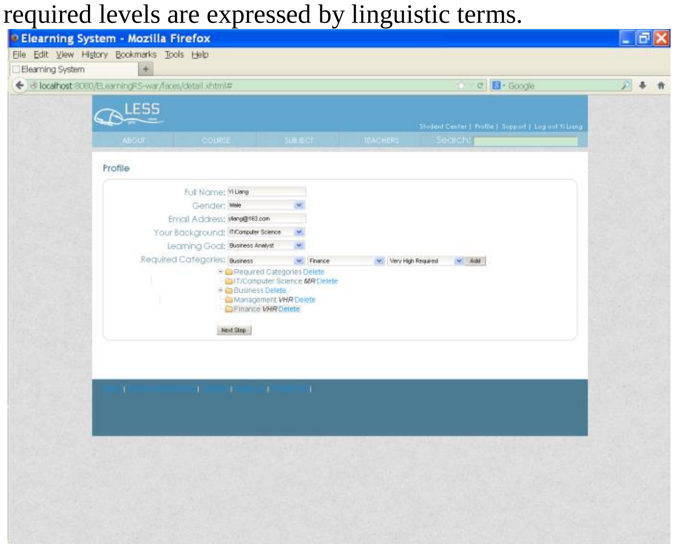

Fig. 13. The student profile page.

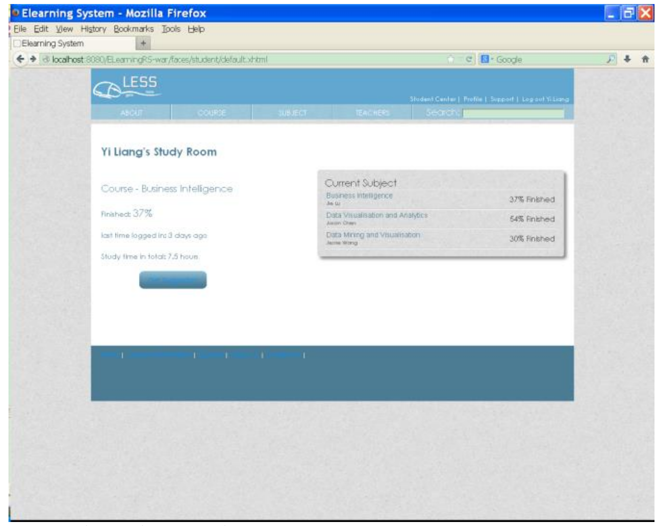

Fig. 14. The student's study room.

The study room in the student center presents the learner's learned activities and current learning progress, which is used for learners to manage their current learning activities. For example, the study room of Learner 4 is shown in Fig. 14. A learner can also provide ratings and comments for a learning activity. This recommender system provides ratings on a scale of 1 to 5. Fig. 15 provides an example.

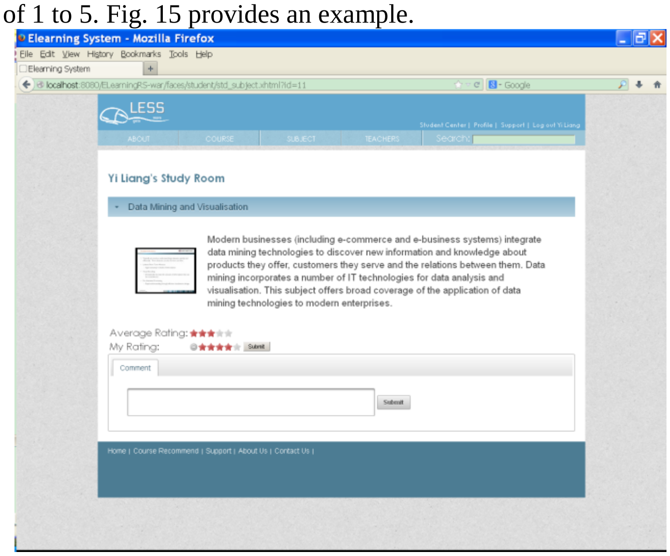

Fig. 15. Student rating and comment input page.

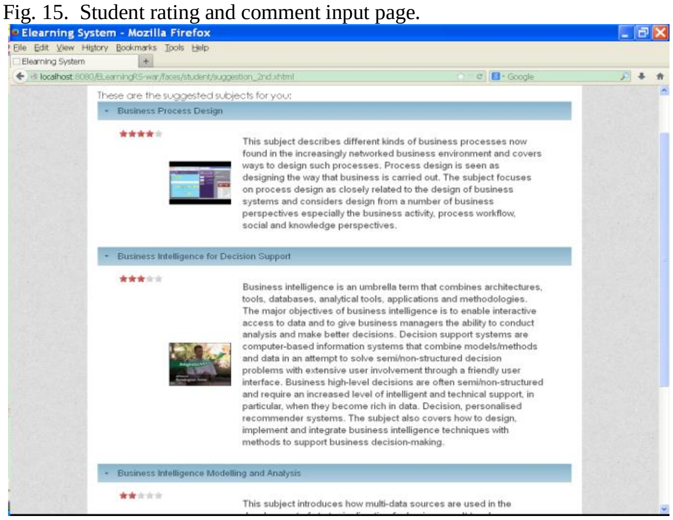

Fig. 16. Learning activity recommendation results.

The existing learner-subject rating matrix in the case study is depicted in Table II. It can be seen that Learner 5 is a new registered learner, and the subject S8 (*Business Process Design*) is a new item. In this case study, subjects recommended to Learner 4 and Learner 5 will be generated.

TABLE II. LEARNER-SUBJECT RATING MATRIX

Subjects	L1	L2	L3	L4	L5
S1: Business Intelligence		4	4	5	
S2: BI Modelling and Analysis	3		2		
S3: BI for Decision Support		3	3		
S4: Database	3	4			
S5: Fundamentals of Data Analytics	4		3		
S6: Data Visualisation and Analytics	5			2	
S7: Data Mining and Visualisation		2		4	
S8: Business Process Design					

The recommendation process is as follows:

1) The recommendation alternatives are selected for Learner 4 and Learner 5 according to the Step 1 in Section VI. The potential learning activities for Learner 4 are {S2, S3, S4, S5, S8}, and for Learner 5 are {S1, S2, S3, S4, S5, S6, S7, S8}.

2) The matching degrees of the alternative learning activities to the learners are calculated, as shown in Table III.

3) The semantic similarity degrees between learners are calculated, as shown in Table IV.

4) The CF similarity degrees between learners are calculated, as shown in Table V.

5) The total similarity degrees between learners are calculated, as shown in Table VI.

6) The predicted ratings of the alternative learning activities by the learners are calculated, as shown in Table VII.

7) The alternative learning activities are ranked according to their predicted ratings. The learning activities with the highest ratings are recommended to the learners.

Fig. 16 shows the recommendation result for Learner 4.

TABLE III. THE MATCHING DEGREES OF THE LEARNING ACTIVITIES TO THE LEARNERS

	S1	S2	S3	S4	S5	S6	S7	S8
Learner 4		0.66	0.62	0.29	0.33			0.85
Learner 5	0.63	0.44	0.42	0.78	0.43	0.61	0.56	0.31

TABLE IV. THE SEMANTIC SIMILARITY BETWEEN LEARNERS

	Learner 1	Learner 2	Learner 3	Learner 4
Learner 4	0.59	0.67	0.62	1
Learner 5	0.77	0.14	0.06	0.44

TABLE V. THE CF SIMILARITY BETWEEN LEARNERS

	Learner 1	Learner 2	Learner 3	Learner 4
Learner 4	0.84	0.94	0.78	
Learner 5	N/A	N/A	N/A	N/A

TABLE VI. THE TOTAL SIMILARITY BETWEEN LEARNERS

	Learner 1	Learner 2	Learner 3	Learner 4
Learner 4	0.72	0.80	0.70	
Learner 5	0.77	0.14	0.06	0.44

TABLE VII. THE PREDICTED RATINGS

	S1	S2	S3	S4	S5	S6	S7	S8
Learner 4		2.91	3.05	2.48	2.59			4.26
Learner 5	3.92	2.56	2.57	3.54	3.04	3.47	3.16	1.53

In the case study, Learner 4 expresses very high requirement on "Business" category and medium requirement on "IT" category, while Learner 5 requires very highly on "IT" category. Their recommendation results reflect their requirements. Additionally, the similar learners, selected according to both the CF similarity and semantic similarity, also take effects in the recommendation process. Even though Learner 5 is a new learner, his/her similar learners can also be identified with the semantic similarity. The new subject *Business Process Design* can also be recommended through the requirement matching knowledge. It is seen from the case study that the semantic information of learning activities and learners, and the requirement matching knowledge are fully utilized in the recommendation process, and the system can recommend learning activities to learners effectively.

D. Comparison with Other Approaches

The developed fuzzy tree matching-based personalized e-learning recommender system has the following six features: (1) tree structured data -- it deals with tree structured learning activities and learner profiles; (2) fuzzy learning activity -- it handles the fuzzy categories of learning activities; (3) fuzzy learner requirement -- it handles learners' fuzzy requirement; (4) pedagogical constraint -- it considers the pedagogical constraints; (5) matching knowledge -- it utilizes the learning requirement matching knowledge; (6) semantic and CF similarity -- it utilizes both the semantic similarity and CF similarity between learners. We compare our method with other e-learning recommender systems from these six aspects. We take into account the e-learning recommender systems of Lu's [3], Salehi and Kamalabadi's [4, 5], Maâtallah and Seridi's [6], Zaiane's [33] and Bobadilla et al.'s [34], as they can represent the typical e-learning recommender systems. The comparison results are illustrated in Table VIII, where "√" represents that the e-learning recommender system has the related feature.

TABLE VIII. COMPARISON BETWEEN OUR DEVELOPED SYSTEM AND THE OTHER E-LEARNING RECOMMENDER SYSTEMS

e-Learning Recommender System	Tree structured data	Fuzzy learning activity	Fuzzy learner requirement	Pedagogical constraint	Matching knowledge	Semantic and CF similarity
Our system: TeLRS	√	√	√	√	√	√
Lu's [3]			√		√	
Salehi and Kamalabadi's [4, 5]	√			√	√	√
Maâtallah and Seridi's [6]			√		√	√
Zaiane's [33]				√		
Bobadilla et al.'s [34]						√

It can be seen from Table VIII that compared with other e-learning recommender systems, the developed TeLRS can deal with more complex data in real world e-learning applications, such as tree-structured data and fuzzy data, and it fully utilizes the domain knowledge in e-learning area. Other e-learning recommender systems only realize parts of the features.

IX. CONCLUSIONS AND FURTHER STUDIES

This paper has outlined the development of a fuzzy tree matching-based hybrid recommendation approach for an e-learning system. The approach develops both a fuzzy tree-structured learning activity model and a fuzzy tree-structured learner profile model. A fuzzy tree similarity measure is presented to evaluate the similarity between learning activities or learners. In the fuzzy tree-structured learning activity model, a fuzzy category tree is defined to specify the categories that each learning activity roughly belongs to, and the fuzzy category similarity measure is developed to evaluate the semantic similarity between learning activities. The precedence relations between learning activities are also handled through analyzing the learning sequences and modeling the prerequisite learning activities. In the fuzzy tree-structured learner profile model, a fuzzy required category tree is defined for learners to express their requirements. The recommendation approach takes advantage of both the CF and KB recommendation approach. When finding similar learners, the proposed system draws strength from both the semantic and CF similarities. 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. The experimental results demonstrate good accuracy performance of the proposed recommendation approach. The case study shows the effectiveness of the proposed system in practice, in which both new learner and new learning activity can be recommended.

The proposed e-learning recommender system will be further tested and compared with existing recommender systems which don't use fuzzy tree-structure data models in a future study. In addition, the features and characteristics of groups of similar learners will be considered, and the methods to identify learner groups and make group recommendations will be exploited to improve the recommendation performance.

REFERENCES

[1] H. Drachsler, 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, pp. 404-423, 2008.

[2] O. R. Zaıane and J. Luo, "Web usage mining for a better web-based learning environment," in *Proceedings of Conference on Advanced Technology for Education*, 2001, pp. 60-64.

[3] J. Lu, "Personalized e-learning material recommender system," in *Proceedings of the 2nd International Conference on Information Technology for Application (ICITA 2004)*, 2004, pp. 374-379.

[4] M. Salehi and I. N. Kamalabadi, "Hybrid recommendation approach for learning material based on sequential pattern of the accessed material and the learner's preference tree," *Knowledge-Based Systems*, vol. 48, pp. 57-69, 2013.

[5] M. Salehi and I. N. Kmalabadi, "A hybrid attribute–based recommender system for e–learning material recommendation," *IERI Procedia*, vol. 2, pp. 565-570, 2012.

[6] M. Maâtallah and H. Seridi, "Enhanced collaborative filtering to recommender systems of technology enhanced learning," in *ICWIT 2012*, 2012, pp. 129-138.

[7] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, pp. 83-97, 1955.

[8] S. Fankhauser, K. Riesen, and H. Bunke, "Speeding up graph edit distance computation through fast bipartite matching," in *Graph-Based Representations in Pattern Recognition*. vol. 6658, X. Jiang, M. Ferrer, and A. Torsello, Eds., ed: Springer Berlin Heidelberg, 2011, pp. 102-111.

[9] 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, pp. 734-749, 2005.

[10] 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, pp. 342-365, 2010.

[11] 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, pp. 37-69, 2013.

[12] X. Guo and J. Lu, "Intelligent e-government services with personalized recommendation techniques," *International Journal of Intelligent Systems*, vol. 22, pp. 401-417, 2007.

[13] A. Zenebe, L. Zhou, and A. F. Norcio, "User preferences discovery using fuzzy models," *Fuzzy Sets and Systems*, vol. 161, pp. 3044-3063, 2010.

[14] A. Zenebe and A. F. Norcio, "Representation, similarity measures and aggregation methods using fuzzy sets for content-based recommender systems," *Fuzzy Sets and Systems*, vol. 160, pp. 76-94, 2009.

[15] R. R. Yager, "Fuzzy logic methods in recommender systems," *Fuzzy Sets and Systems*, vol. 136, pp. 133-149, 2003.

[16] J. Lu, Y. Zhu, X. Zeng, L. Koehl, J. Ma, and G. Zhang, "A linguistic multi-criteria group decision support system for fabric hand evaluation," *Fuzzy Optimization and Decision Making*, vol. 8, pp. 395-413, 2009/12/01 2009.

[17] J. Ma, J. Lu, and G. Zhang, "Decider: A fuzzy multi-criteria group decision support system," *Knowledge-Based Systems*, vol. 23, pp. 23-31, 2010.

[18] D. Wu, G. Zhang, and J. Lu, "A fuzzy preference tree-based recommender system for personalized business-to-business e-services," *IEEE Transactions on Fuzzy Systems*, vol. 23, pp. 29-43, 2015.

[19] R. Burke, "Hybrid recommender systems: survey and experiments," *User Modeling and User-Adapted Interaction*, vol. 12, pp. 331-370, 2002.

[20] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen, "Collaborative filtering recommender systems," in *The Adaptive Web*. vol. 4321, P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds., ed: Springer Berlin Heidelberg, 2007, pp. 291-324.

[21] M. Pazzani and D. Billsus, "Content-based recommendation systems," in *The Adaptive Web*. vol. 4321, P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds., ed: Springer Berlin Heidelberg, 2007, pp. 325-341.

[22] R. Burke, "Knowledge-based recommender systems," *Encyclopedia of Library and Information Systems*, vol. 69, pp. 175-186, 2000.

[23] R. Burke, "Hybrid web recommender systems," in *The Adaptive Web*. vol. 4321, P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds., ed: Springer-Verlag, Berlin Heidelberg, 2007, pp. 377-408.

[24] 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, pp. 231-243, 2003.

[25] 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, pp. 325-335, 2001.

[26] 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, pp. 275-284, 2002.

[27] O. Byung Kwon, ""I know what you need to buy": context-aware multimedia-based recommendation system," *Expert Systems with Applications*, vol. 25, pp. 387-400, 2003.

[28] R. D. Lawrence, G. S. Almasi, V. Kotlyar, M. Viveros, and S. S. Duri, *Personalization of supermarket product recommendations*: Springer, 2001.

[29] J. He and W. Chu, "A social network-based recommender system (SNRS)," in *Data Mining for Social Network Data*. vol. 12, N. Memon, J. J. Xu, D. L. Hicks, and H. Chen, Eds., ed: Springer US, 2010, pp. 47-74.

[30] Z. Zhang, H. Lin, K. Liu, D. Wu, G. Zhang, and J. Lu, "A hybrid fuzzy-based personalized recommender system for telecom products/services," *Information Sciences*, vol. 235, pp. 117-129, 2013.

[31] G. Adomavicius and A. Tuzhilin, "Context-aware recommender systems," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds., ed: Springer US, 2011, pp. 217-253.

[32] J. Masthoff, "Group recommender systems: combining individual models," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds., ed: Springer US, 2011, pp. 677-702.

[33] O. R. Zaiane, "Building a recommender agent for e-learning systems," in *Proceedings of 2002 International Conference on Computers in Education*, 2002, pp. 55-59

[34] J. Bobadilla, F. Serradilla, and A. Hernando, "Collaborative filtering adapted to recommender systems of e-learning," *Knowledge-Based Systems*, vol. 22, pp. 261-265, 2009.

[35] D. Wu, G. Zhang, J. Lu, and W. A. Halang, "A similarity measure on tree structured business data," in *ACIS 2012: Location, location, location: Proceedings of the 23rd Australasian Conference on Information Systems 2012*, 2012, pp. 1-10.

[36] D. Wu, G. Zhang, and J. Lu, "A fuzzy tree similarity measure and its application in telecom product recommendation," in *2013 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Manchester, 2013, pp. 3483-3488.

[37] D. Wu, G. Zhang, and J. Lu, "A fuzzy tree similarity based recommendation approach for telecom products," in *2013 Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS)*, Edmonton, Canada, 2013, pp. 813-818.

[38] 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.

[39] D. Jungnickel, *Graphs, Networks, and Algorithms*, 3rd ed.: Springer-Verlag, Berlin Heidelberg, 2008.

[40] K. Riesen and H. Bunke, "Approximate graph edit distance computation by means of bipartite graph matching," *Image and Vision Computing*, vol. 27, pp. 950-959, 2009.

[41] J. F. Allen, "Maintaining knowledge about temporal intervals," *Communications of the ACM*, vol. 26, pp. 832-843, 1983.

[42] G. Zhang and J. Lu, "Using general fuzzy number to handle uncertainty and imprecision in group decision-making," *Intelligent Sensory Evaluation: Methodologies and Applications*, pp. 51-70, 2004.

[43] G. Zhang and J. Lu, "An integrated group decision-making method dealing with fuzzy preferences for alternatives and individual judgments for selection criteria," *Group Decision and Negotiation*, vol. 12, pp. 501-515, 2003.

[44] G. Zhang and J. Lu, "A linguistic intelligent user guide for method selection in multi-objective decision support systems," *Information Sciences*, vol. 179, pp. 2299-2308, 2009.

Dr. Dianshuang Wu is a member of the Decision Systems and e-Service Intelligence Research Lab in the Centre for Quantum Computation & Intelligent Systems in Faculty of Engineering and Information Technology at University of Technology Sydney. His research interests include tree similarity measure, recommender systems, and business intelligence.

Professor Jie Lu is the Associate Dean Research (Acting) of Faculty of Engineering and Information Technology, and the Director of the Decision Systems and e-Service Intelligence Research Laboratory in the Centre for Quantum Computation & Intelligent Systems at the University of Technology, Sydney (UTS). She received her PhD from the Curtin University of Technology in 2000. Her main research interests lie in the area of decision making modeling, decision support system tools, uncertain information processing, recommender systems and e-Government and e-Service intelligence. She has published five research books and 270 papers in refereed journals and conference proceedings. She has won five Australian Research Council (ARC) discovery grants. She received the first UTS Research Excellent Medal for Teaching and Research Integration in 2010. She serves as Editor-In-Chief for Knowledge-Based Systems (Elsevier), and editor for book series on Intelligent Information Systems (World Scientific).

Dr. Guangquan Zhang is an Associate Professor and Co-Director of the Decision Systems an e-Service Intelligence (DeSI) Lab within the Centre for Quantum Computation and Intelligent Systems (QCIS) in Faculty of Engineering and Information Technology at University of Technology Sydney. He has authored and co-authored 246 publications including four monographs, four text books, 13 book chapters and 119 refereed international journal papers. Dr. Zhang has served as an advisory board member, or a member of the editorial board for several international journals, co-chaired several international conferences/workshops in the area of fuzzy decision-making and knowledge engineering, such as FLINS08, FLINS10, ISKE09.