**INTRODUCTION**

With the use of recommendation methods, recommender systems which are web-based support systems, actively suggest a set of limited and ranked items from all available items without the direct input of users. These systems are widely used to overcome the problems created by the so-called “information explosion” in a variety of web-based applications in e-commerce e-learning and e-tourism, as well as in such areas as the recommendation of news, movies, books, videos, resources and real estate. Prior to making a recommendation, recommender systems use background data, such as historical data consisting of ratings from users, and input data, such as features of items or user ratings, to initiate a recommendation; models and algorithms combine the two and generate a recommendation.

In real situations, the features of items and user behavior are often subjective, vague, and imprecise and users’ item preferences are frequently subjective and uncertain. It is difficult for a user to express his/her interest in an item with exact numbers. Fuzzy set theory and techniques lend themselves well to handling the fuzziness and uncertain issues in recommendation problems. User preferences and item features have been represented as fuzzy sets in previous research and recommendations to customers for the selection of the most suitable items are made with incomplete and uncertain information. Current research and recommender system applications focus mainly on making recommendations to personal users. Fuzzy user preference and item representations focus on vector representations accordingly. The abundance of information created and delivered via the Web provides excellent opportunities for the development of business-to-business (B2B) e-services, such as finding a business partner online. Excessive amounts of information on the Web create a severe information overload problem.An effective solution for this problem is the development of personalized recommender systems; however, recommendation techniques have been rarely used in the B2B environment. The main reason is that items or user profiles in a B2B environment are so complex that they can only be presented as complicated structures, such as tree structures. For example, a business in a B2B application environment may supply several product categories, each of which may contain a number of subcategories, under which there may be multiple specific products, which together form a tree structure. Therefore, tree-structured data modeling and tree matching methods are needed. However, an item is normally described as a single value or a vector in current research and tree-structured items or user profiles have not been considered to date. The fuzzy preference models mentioned previously, which are represented as vectors, are not suitable to dealing with the tree-structured data in a Web-based B2B environment.

To solve these challenges—namely, tree-structured items (products/services), tree-structured user preferences, vague values of user preferences, and personalization of recommendations—in B2B e-service recommendation problems, this study proposes a method for modeling fuzzy Tree-structured user preferences, presents a tree matching method, and, based on the previous methods, develops an innovative fuzzy preference tree-based recommendation approach. The developed new approach has been implemented and applied in a business partner recommender system. This paper has three main contributions. From the theoretical aspect, a tree matching method, which comprehensively considers tree structures, node attributes, and weights, is developed. From the technical aspect, a fuzzy tree-structured user preference modeling method is developed, as well as a fuzzy preference tree-based recommendation approach for tree-structured items. From the practical aspect, the proposed methods/approaches are used to develop a Web based B2B recommender system software known as Smart BizSeeker, with effective results. The remainder of the paper is organized as follows. The related works in recommender systems, tree matching methods, and fuzzy set techniques are expatiated.

**Fuzzy tree-structured preference model**

This section describes the representation of fuzzy tree structured user preferences. Fuzzy set techniques are used to model user preferences; a formal tree-structured data model is given, and a fuzzy tree-structured user preference model is then presented.

**Users’ Fuzzy Preferences**

To make a recommendation to a user, the information aboutthe user’s preferences must be known. The modeling methodfor user’s preferences is presented in this section.Information about user preferences can essentially be obtainedin two different ways: extensionally and intentionally*.*  The extensionally expressed preference information refersto information that is based on the actions or past experiencesof the user with respect to specific items. The intentionallyexpressed preference information refers to specifications bythe user of what they desire in the items under consideration.In this paper, the user preference model covers both kinds ofinformation.

**Tree-Structured Data Model**

A tree-structured data model is defined to represent treestructured items or user preferences. It is based on the basic tree definition.

**Fuzzy Tree-Structured User Preference**

A user’s preference is represented as the tree-structured data model defined previously. It is called a fuzzy preference tree and is defined as follows.

*Definition 2:* The fuzzy preference tree of a user is tree structured data whose node values are the user’s fuzzy preferences for the corresponding attributes. Each sub tree in a user’s fuzzy preference tree represents the user’s preference for one aspect of the features, and the sub trees of that aspect represent the user’s preferences for the finer features. The leaf nodes represent the preferences for the finest features. The fuzzy preference tree has a similar structure to the item tree except for the node value definition.



Fig 1-- Two tree-structured data examples.

Fig. 1 shows two tree-structured data in a business environment, which are called two trees for simplicity. Tree T1 is a user’s buying request. The user requires two kinds of product, wine and beer. For the wine request, two products are specified. Tree *T*2 is a product tree of a wine business. It provides two product categories, wine and cider. For each category, specific products are given. The two examples contain the four features listed in the aforementioned tree-structured data definition. For Feature 1—Node attribute, a domain attribute term set which includes product category names, product names, product feature names, and so on exists. For Feature 2—Attribute conceptual similarity, conceptual relations between these attributes exist, such as attribute conceptual similarity measures to express the similarity between the attributes. For Feature 3- Node value, the values are assigned to the trees. For example, the values of the product tree can be the quality or the quantity of the attributes. For Feature 4—Node weight, different nodes/attributes may have different importance degrees in real applications, which are reflected by the weights of the nodes. It can be seen from Fig. 1 that the two tree-structured data *T*1 and *T*2 have different structures and node attributes, which makes it difficult to identify the corresponding node pairs between them.