**MINOR PROJECT REPORT ON**

A Fuzzy Preference Tree-Based Recommender System for Personalized Business-to-Business E-services

**Under guidance of Mr. Rahul Katarya**



**UNDERTAKEN BY:**

Anmol Mahajan (2K13/IT/014)

Bhrigu Srivastava (2K13/IT/027)

Dishti Aggarwal (2K13/IT/033)

Divyesh Kumar (2K13/IT/034)

**Department Of Computer Science and Engineering,**

**DELHI TECHNOLOGICAL UNIVERSITY**

***Abstract:***

As the World Wide Web continues to change rapidly, the size and complexity of many web sites change along with it. For the users of these web sites it becomes more and more difficult and time exhausting to find the information they are looking for. Some web sites provide personal information to the users by letting them choose from a set of predefined topics of interest. Users do not always know their interests in advance and their interests may change with time which would require them to change their selection at regular intervals. **Recommender systems** provide personalized information by studying the user’s interests from traces of communication with that user.

This project has three main aspects. 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 can be effectively used in a Web-based B2B recommender system.

We shall be developing a recommendation approach to recommending tree-structured items. The key technique in this project is an exhaustive tree matching method, which can correlate two tree-structured data and identify their corresponding parts by taking into account all the information on tree structures, node attributes, and weights. Importantly, the suggested fuzzy preference tree-based recommendation approach will be tested and justified using the MovieLens dataset to show that the proposed fuzzy tree-structured user preference proﬁle matches the user preferences effectively and the recommendation approach exhibits excellent performance for tree-structured items, especially in e-business applications.

We were able to achieve quite interesting results after implementing the above recommendation approach. Our modified approach has highest precision and recall out of all the algorithms tested and even more than the original approach. In addition to this, our algorithm has the second lowest MAE, slightly higher than the original, but this is only because of the time constraints.

***Key Terms****:* Recommender systems, E-business, fuzzy preferences, tree matching, web-based support system, collaborative filtering.

1. **INTRODUCTION:**

The rapidly increasing popularity of information and communication technologies in recent years strongly promotes the development of e-services in many application domains, such as e-commerce, e-business, e-learning and e-government. In the meantime, the rapid growth of web information has caused increasingly severe information overload problems whereby users are not able to locate relevant information to exactly meet their needs efficiently by using the current Internet search functions. The increasing amount of information available on the web usually involves a lot of irrelevant contents so that users have to spend a significant part of their navigation to search for more interesting contents. A possible solution to this problem lies in the use of recommender systems. **Recommender systems** are information filtering systems which provide users with information, they might be interested in. Driven by recommendation algorithms, these systems help consumers by selecting products they will probably like and might buy based on their browsing, searches, purchases, and preferences. Designed to help retailers boost sales, recommenders are a vast growing business. The so-called “information explosion” in a variety of web-based applications in e-commerce, e-learning and e-tourism has paved way for recommender systems that provide recommendations for new, movies, books, videos, resources and real estate. These systems tend to provide “personalized” suggestions because they track each user’s behavior- pages viewed, purchases, and ratings- to come up with recommendations. All such background data of users, and input data, such as item features and user ratings are used to instigate a recommendation. These two are then combined using models and algorithms thus, generating a recommendation.

A big hurdle in current recommender system research is: the items and user profiles in many recommender system applications nowadays, such as the e-business and e-learning recommender systems, are so complex that they can only be described in complicated tree structures. This is the reason that even though recommender systems tend to provide tremendous opportunities for the development of business-to-business (B2B) e-services such as finding partners online, their potential has rarely been exploited in the B2B environment. Owing to their high complexity items or user profiles in a B2B environment 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, the item or user similarity measure, as the core technique of the recommendation approach, becomes a tree similarity measure, which existing recommender systems cannot provide. Another challenge is that in many real life situations, online recommendations to customers in selecting the most suitable products/services are often made under incomplete and uncertain information, which needs fuzzy set theory and techniques to deal with. Thus, how to use fuzzy set techniques to handle data uncertainty issues in tree-structured items or user profiles needs to be investigated. Hence, 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 re-structured items or user profiles have not been considered to date. How to use fuzzy set techniques to handle these uncertainty issues in tree-structured items or user profiles in recommender system, clearly, needs to be investigated. **This study proposes a method for modeling fuzzy Tree-structured user preferences, presents a tree matching method, and, based on the proposed fuzzy preference tree model and the similarity measure on tree-structured data, develops an innovative fuzzy preference tree-based recommendation approach.** The developed new approach has been implemented and tested using a Movie lens dataset.

The remainder of the paper is organized as follows. In Section II, the related works in recommender systems, tree matching methods, and fuzzy set techniques are expatiated.

Section III presents the fuzzy tree-structured preference model. Section IV explains the preprocessing and similarity measure calculation required to set the environment for further computation. Section V proposes a comprehensive tree matching algorithm to identify the corresponding parts between two trees. The fuzzy preference tree construction algorithm is proposed in Section VI. A fuzzy preference tree-based recommendation approach for tree-structured items is presented in Section VII. The approach has been tested using the MovieLens dataset. The experimental evaluations and results are given in Section VIII. Finally, the conclusions and future study are given in Section IX.

**2. RELATED WORK:**

This section will analyze the information on tree matching methods, recommender systems, and fuzzy set techniques in recommender systems.

***2.1) Recommender Systems:***

Recommendation techniques have been in a spotlight and many recommendation approaches have been suggested. Some of the main techniques discussed are collaborative ﬁltering (CF), content-based (CB) and knowledge-based (KB) techniques [1]. The CF is the most successful and commonly used technique for the recommender systems [2][3] . It provides convenience to make choices based on the notions of other people who share identical interests. RS uses different sources of information for providing users with prediction and recommendation of items.

The CF technique can be further branched into user-based and item-based CF approaches [4]. The major limitations of CF methods are data sparsity and cold-start (CS) problems [1], [3], [4]. The data sparsity problem is encountered when the number of available items increases and the number of ratings in the rating matrix is insufﬁcient to generate accurate predictions. When the ratings obtained are very less compared to the number of ratings that need to be predicted, a recommender system becomes incapable of locating similar neighbors and produces poor recommendations. The CS problem consists of the CS user problem and the CS item problem. The CS user problem, also known as the new user problem, affects users who have a small number of ratings or none. When a CS user has rated lesser number of items, the CF-based approach cannot accurately ﬁnd user neighbors using rating similarity; therefore, it fails to generate accurate recommendations. The CS item problem, also known as the new item problem, affects items that have only a small number of ratings or none. With only a few ratings for CS items, CF-based approaches cannot aptly locate identical neighbors using rating similarity and will be unlikely to recommend them [1], [5].

***2.2) Tree Matching:***

Tree-structured data are widely used in many application ﬁelds [4]. The tree similarity measure [5], tree isomorphism [6], and subtree matching problems [4], [7] have been researched during the applications of tree-structured data. The tree edit distance model [7] is the most widely used method to compare the structures of ordered or unordered labeled trees. The model compares two trees with the minimum cost of the edit operation sequences that convert one tree into another. The edit operations give rise to an edit distance mapping which is a graphical speciﬁcation of which edit operations apply to each node in the two labeled trees [8]. The tree edit distance model is also used in tree isomorphism and subtree matching problems [7]. In the aforementioned researches, only tree structures and node labels are considered, which is insufﬁcient for use in the application of the B2B e-service recommendation. In our previous research, an edit distance mapping between two tree-structured data was constructed that considered tree structures, node attributes, and weights [9], [10]. A number of tree matching methods have been used in recommender systems. In [11], users’ behavior patterns were represented as tree structures, and tree matching was used to ﬁnd the correlation between different behavior patterns. In the telecom industry, products/services for business users usually have complex hierarchical structures; therefore, tree structured data have been used to represent items and user proﬁles [10]. Tree matching methods have been developed to evaluate the similarity between tree-structured items or users. However, these methods are unable to construct fuzzy tree-structured user preferences and match items to user preferences. In this study, to match tree-structured items and user preferences in a B2B e-service environment, a new tree matching method is developed based on tree edit distance mapping.

***2.3) Fuzzy Techniques:***

Fuzzy set theory and fuzzy logic gives us a way to measure the uncertainty, vagueness and imprecision. It is well suited to handle imprecise information, the unsharpness of classes of objects or situations, and the gradualness of preference proﬁles [12], [13]. In [14], an item is represented as a fuzzy set over an assertion set. The value of a feature or characteristic for an item is a fuzzy set over the subset of the assertions relevant to the feature. The user’s intentional preferences are shown as an elementary preference module, which is the ordered subjective averaging of components that can evaluate items. The user’s extensional preferences are expressed as a fuzzy set over the user’s experienced items whose membership degrees are the ratings. Grounded on the representation, the preference for an item by a user can be inferred. Zenebe et al. [12], [15] deﬁned a feature set for items and a set of values for each feature. The items are represented as the fuzzy subset over the values, denoted by a feature vector. Four kinds of fuzzy-set-based similarity measures: fuzzy set theoretic, cosine, proximity, and correlation-like, are presented. Cao and Li [16] used linguistic terms for domain experts to estimate the features of consumer electronic products and allow users to use linguistic terms to express their needs for item features. In [17], the user preferences are represented as two fuzzy relations, positive and negative feelings, from user set to item set. The item similarity is then calculated by integrating it with the CB similarity, which is a fuzzy relation within an itemed, and item-based CF similarity, which is computed on the basis of user preferences. The user similarity is created by fuzzy relational calculus from the preferences and item similarity relations. The ﬁnal recommendations, which are the positive and negative preferences, are generated by composing the aforementioned fuzzy relations. Porcel et al. developed a fuzzy linguistic-based recommender system based on both CB ﬁltering and the multi granular fuzzy linguistic modeling technique, which is convenient to assess different qualitative concepts. Fuzzy similarity measures and fuzzy matching methods have been used in previous research, but to the best of our knowledge, no research has focused on solving fuzziness problems in tree-structured data.

To make a recommendation to a user, the information about the user’s preferences must be known. The modeling method for user’s preferences is presented in this section. Information about user preferences can essentially be obtained in two different ways: extensionally and intentionally [10].The extensionally expressed preference information refers to information that is based on the actions or previous experiences of the user with respect to speciﬁc items. The intentionally expressed preference information refers to speciﬁcations by the user of what they desire in the items under consideration. In this paper, the user preference model covers both kinds of information. In the practice of recommender systems, a business user’s preferences are usually complex and vague. It might be difﬁcult to require a business user to express a crisp preference for an item or a feature of an item, and it is therefore difﬁcult to represent the user’s preferences with crisp numbers. In this study, fuzzy set techniques are used to describe users’ complex and vague preferences.

**3. PROPOSED WORK:**

There are two situations. First, the user can assign a preference value, such as Feature 1 in Figure 3-1. Second, the user can drill down to detail and express preferences for finer features under the macro feature, as shown for Feature 2. Therefore, users’ preference values, which are represented as fuzzy sets, can be expressed at different levels. For different features, the user can also specify various weights to express the different importance degrees of diverse features.

A user’s fuzzy preference tree Tu is constructed based on user input preferences. The tree has the same structure as the user preferences shown in Figure 3-1. The tree node attributes are the relevant features. If the user expresses the preference value for a feature, the value will be assigned to the relevant node accordingly. The node weights are also set according to the user’s input.

***3.1 EXTENSIONALLY EXPRESSED PREFERENCE***

The extensionally expressed preference of a user is constructed from the items experienced by the user. Let the items experienced by a user u be the set EIu = {*i1,i2,…… im*}. Each item *ij* ( j=1, 2, …, m) corresponds to an item tree Titem,j and a preference value given by the user p᷉uj = {f1,uj, f2,uj,…, fr,uj}. Let the user’s fuzzy preference tree be Tu. The construction process of Tu is presented as follows.

For each item *ij* experienced by user u with preference value ࣤpuj, add the item tree Titem,j into the fuzzy preference tree Tu. The add operation integrates the user’s preference for an item into Tu. When all the items experienced are added into Tu, the user’s fuzzy preference tree Tu will be constructed. The fuzzy preference tree construction algorithm is described in detail in the next section. During the process, the conceptual corresponding parts between two trees considering tree structures, node attributes and weights comprehensively must be identified. Therefore, the conceptual similarity computation algorithm is applied to construct the maximum conceptual similarity tree mapping between two trees.

***3.2 ORIGINAL APPROACH ‘vs’ OUR APPROACH***

The original fuzzy preference tree-based approach involves creating a user-specific user tree, Tu, consisting of all the items in the testing set that the user has rated, and a universal item tree, Ti, that contains all the items in the dataset. These two trees are processed and are run through the algorithms specified in the next section.

However, this approach is very slow.

In real situations of recommender system applications, the features of items and user behaviors are often subjective, vague and imprecise (Zenebe & Norcio 2009), and users’ preferences to items 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 can be effectively used for handling the fuzziness and uncertain issues in recommendation problems. User preferences and item features have been represented as fuzzy sets in previous research (Chen & Duh 2008; Yager 2003; Zenebe & Norcio 2009; Zenebe, Zhou & Norcio 2010), and recommendations to customers for the selection of the most suitable items are made with incomplete and uncertain information (Cornelis et al. 2007; Porcel, López-Herrera & Herrera-Viedma 2009). The previous research in recommender systems in handling users’ uncertainty preferences and items’ fuzzy representations only focuses on the single value or vector representations. However, in many application domains, such as business-to-business (B2B) e-services, items or user profiles are so complex that they can often be presented as tree structures.

To solve these challenges in personalization of recommendations – namely, tree structured items, tree-structured user preferences, and vague values of user preferences, a fuzzy preference tree can be constructed to model fuzzy tree-structured user preferences, and, based on the fuzzy preference tree model and the similarity measure on tree-structured data, we can develop an innovative fuzzy preference tree-based recommendation approach.

***3.1 FUZZY TREE-STRUCTURED PREFERENCE MODEL***

This section describes the representation of fuzzy tree-structured user preferences. Users’ fuzzy preferences are first modeled by use of fuzzy set techniques; and a fuzzy tree-structured user preference model is then presented by use of the tree-structured data model.

**3.1.1 USERS’ FUZZY PREFERENCES**

To make a recommendation to a user, the information about the user’s preferences must be presented. The representation method for user’s preferences is presented in this sub-section.

Information about user preferences can essentially be obtained in two different ways: extensionally and intentionally (Yager 2003). The extensionally expressed preference information refers to information that is based on the actions or past experiences of the user with respect to specific items. The intentionally expressed preference information refers to specifications by the user of what they desire in the items under consideration. In this study, the user preference model covers both kinds of information.

In the practice of recommender systems, a user’s preferences are usually complex and vague. It might be difficult to require a user to express a crisp preference for an item or a feature of an item, and it is therefore difficult to represent the user’s preferences with crisp numbers. In this study, fuzzy set techniques are used to describe users’ preferences.

To express users’ preferences, it is assumed that a totally ordered set R= {1, 2,…, r} is predefined to represent the crisp values of ratings. Based on the fuzzy set definition, a user u’s preference for an item (or feature) j is represented as a fuzzy set over R, {f1,uj/1, f2,uj/1,…, fr,uj/r} ,where each f1,uj ε [0,1], I ε R represents the membership degree of the rating *i*. p᷉uj will be expressed as {f1,uj, f2,uj,…, fr,uj} if there is no confusion.

For example, supposing that the crisp ratings are on a scale of 1 to 5, with 1 being the lowest rating and 5 the highest rating, a user’s preference for an item is represented as a fuzzy subset on {1, 2, 3, 4, 5} by membership degree [0, 1]. The preference value (0/1, 0/2, 0/3, 0.9/4, 1/5) indicates that a user likes an item very much by the high membership degree (1) on rating value “5” and also the very high membership degree (0.9) on “4”, while the preference value (1/1, 0/2, 0/3, 0/4, 0/5) indicates that the user does not like the item at all by the high membership degree (1) on rating value “1” and the low membership degree (0) on the other rating values.

***3.1.2 FUZZY TREE-STRUCTURED USER PREFERENCE***

The items considered in this study are presented as tree structures, and the features of items form a hierarchical structure. They are modeled as item trees. A user’s preferences concern a set of products/features, and user preference is therefore described as a tree structure which has fuzzy preference values.

**3.1.2.1 FUZZY PREFERENCE TREE**

To formally describe the tree-structured user preferences, the tree-structured data model is utilized here. To express the fuzzy preference values, a fuzzy tree-structured data model is defined by introducing the fuzzy features.

**Definition 3-1:** A fuzzy tree-structured data model is a tree-structured data whose node features, i.e. the node attributes, node values, node concept similarity and value similarity measures, or node weights, are represented as fuzzy sets.

A fuzzy preference tree is then defined as follows.

**Definition 3-2:** The fuzzy preference tree of a user is a tree-structured data whose node values are the user’s fuzzy preferences for the corresponding attributes.

A user’s preference is represented as a fuzzy preference tree. 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. The value of a node in an item tree represents the quantity or quality degree of the attribute associated with the node, while the value of a node in a fuzzy preference tree represents the user’s preference for the attribute represented by the node. The node value of the fuzzy preference tree contains two components. One is the preference which is expressed with a fuzzy set; the other is a count number count, which indicates the number of preference sources used to calculate the value. The count is used to incrementally update the user’s fuzzy preference tree, as shown below.

***3.1.2.2 INTENTIONALLY EXPRESSED PREFERENCE***

The intentionally expressed preference is acquired directly from users. This kind of information is especially important for new users to obtain recommendations.

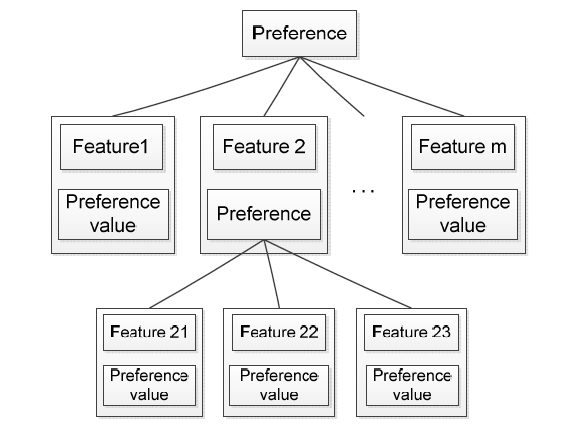


Figure 3-1. Intentionally expressed user preference [21]

Because the item features present tree structures, the preferences given by users are in tree structures, which is shown in Figure 3-1. To express preferences, a user selects several features. For example, Feature 1, Feature 2, … , Feature m are selected in Figure 5-1. For each feature, and gives way to a lot of redundant/unnecessary recursive calls and since, each call to the function implicitly involves running the hungarian algorithm(**O(n3)**), this makes the original approach too slow for practical application without some sort of optimization and/or pre-calculation.

Therefore, we proposed our own approach to tackle this problem. It involves trading-off a little accuracy for a substantial improvement in the runtime.

Our approach involves clustering the movies in the dataset into various genres.

This is done intelligently to minimize the inaccuracy inherently borne out of this approach:

1. We maintain a count of all the movies in the 19 genres in the dataset.
2. Each movie is added under the genre which has the least amount of movies in the current iteration. To reduce discrepancy, the same movie is put in the corresponding genre in the item tree.
3. The above step ensures that the tree formed is not skewed and all the cluster take approximately equal time to run, thereby, improving the average runtime.
4. To offset the approximations, the movie-genre pairs’ conceptual similarity measure value is adjusted so as to remain as true to the original structure as possible.
5. Finally, instead of processing the whole trees, we process corresponding genre-based clusters in both the trees and consolidate the results to display the most accurate result.

Since, we ignore some comparisons, this automatically leads to a slight increase in the Mean Absolute Error (MAE), but the balanced structure of the tree ensures a high precision and recall value, making this modified approach more accurate and faster than the original fuzzy preference tree-based approach.

**4. FRAMEWORK:**

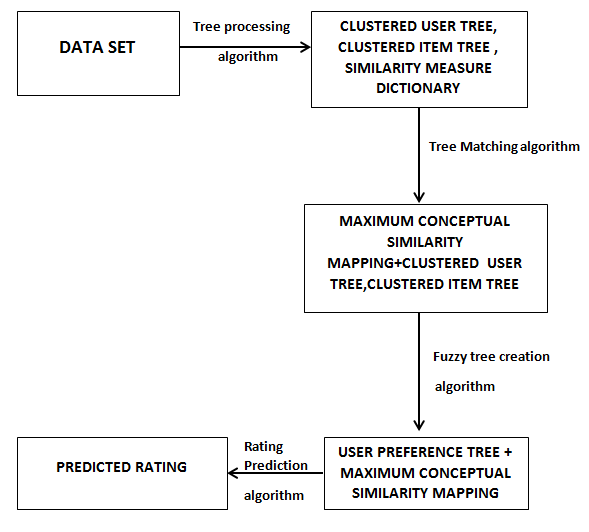
****

Fig- 4.1 Project Framework

**5. PRE-PROCESSING AND CALCULATING SIMILARITY MEASURES**

The first step is to collect the data-set and calculate pairwise similarity measures of the items present, and then to construct Item and User trees.

An attribute ‘*dictionary’* was created to store those similarity values of the pairs.

**Algorithm (A)**

*input:* The data-set input files

*output:* A dictionary of pairwise similarity measures

Make-Dictionary:

1. initialize empty dictionary
2. for each movie ‘a’ in file(‘item’):
3. for each movie ‘b’ in file(‘item’):
4. s = getSimilarityValue(a,b)
5. dictionary(a,b) = s

Where, the function getSimilarityValue(.,.) takes two inputs as movies and returns their corresponding similarity value. The functions calculates as follows: if the two movies are same, return 1; else , for each common genres the two movies have, increment the result by a weighted factor, and return the final result.

Now, we need to construct the User-Tree and the Item-Tree. Following our tree based structure, instead of adding Movie items as leaf nodes under a common ‘Movie’ root node, the movies were **clustered** according to the genres they were in, and the movies were added as leaf nodes under their respective genres. Finally, all the genres are added under the common ‘Movie’ root node.

Here, the problem arises as a movie can have multiple genres. To resolve this, a count for the number of movies under a genre is stored, and the genre having minimum number (among the genres of a particular movie), is made the parent of that movie. **This is done to ensure that the tree is balanced in the sense that all genres contain approximately the same number of movies, so as to equalize the recursive calls for each**. The inaccuracy introduced by this is dealt via intelligently adjusting the similarity measure value for Movie-Genre pairs.

**Algorithm (B)**

*input:* input files from data-set

*output:* corresponding Item or User tree

Create-Tree:

1. create root node(‘Movies’)
2. for each genre in genres:
3. add the genre under the root node
4. for each movie in item-file / data-file:
5. g ← genre with min stored movies, among genre of current movies
6. add the movie under the node corresponding to g
7. update preference value for the movie, from its actual rating

Finally, we update the preference values of genres for the user. To do this, the aggregate preference value is calculated from the preference values of the movies under a genre, which is stored as the preference value for that genre.

***6. TREE-STRUCTURED DATA MATCHING ALGORITHM***

Based on the tree-structured data model, we use a tree matching algorithm to construct a map to identify the parts of two trees that most correspond. The proposed tree matching algorithm will be used to construct the user’s fuzzy preference tree and match user preferences and items in the course of making recommendations. Let two trees to be matched be denoted as Tu and Ti. A maximum conceptual similarity tree mapping **[18]**, which is quite similar to of edit distance problem in dynamic programming, but here the two items to match are Trees, and a mapping is constructed to identify the parts of the two trees that most correspond. Tree structures, node attributes, and node weights are all taken into consideration, when constructing the mapping.

Certain symbols are used to represent the tree, their nodes, children, attributes, etc.. Suppose that we have a numbering for each tree node in a tree. Let t[j] be the jth node of the tree T in the given numbering. Let T[j] be the subtree rooted at t[j] and F [j] be the unordered forest obtained by deleting t[j] from T [j]. Let t[j1], t[j2], . . . , t[jnj] be the children of t[j].

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 between two trees is calculated as follows:

Given two trees Tu[j] and Ti[k] to be compared, according to the matching situations of their roots tu[j] and ti[k], three cases are considered:

* tu[j] and ti[k] are matched
* tu[j] is matched to t i [k]’s child
* ti[k] is matched to tu[j]’s child

The case with the maximum conceptual similarity value is the best match and the corresponding similarity value is taken as the conceptual similarity between the two trees.

*Case Ⅰ:* tu[j] and ti[k] are matched.

The conceptual similarity between the nodes Tu[j] and Ti[k] is calculated as (1) in the equations given below. Where a(tu[j]) and a(ti[k]) represent the attributes of tu[j] and ti[k], respectively, wkt and wjt are the weights of ti[kt] and tu[jt], respectively, and α is the influence factor of the parent node.

In situation S1, tu[j] and ti[k] are both leaves, and their conceptual similarity is equivalent to the conceptual similarity of their attributes. In situations S2 and S3, one node is a leaf and the other is an inner node. As the concept of a tree is dependent not only on it root’s attribute but also on its children’s attributes, the children of the inner node are also considered in the formula. In situation S4, both tu[j] and ti[k] have children. Their children construct two forests Fu[j] and Fi[k]. They are compared with the forest similarity measure scF(Fu[j], Fi[k]).

In the fourth situation in Formula (1), both tu[j] and ti[k] have children which construct two forests, denoted as Fu[j] and Fi[k]. To compute the conceptual similarity between Tu[j] and Ti[k], the conceptual similarity between Fu[j] and Fi[k] is required.

The roots of Fu[j] and Fi[k] construct a bi-partite graph. A maximum weighted bipartite mapping [18] denoted as MWBM is constructed. These maximum weighted bipartite mappings are recorded during the computation.

*Case II:* tu[j] is matched to ti[k]’s child.

The concept level of tu[j] is lower than the concept level of ti[k]. Tu[j] is mapped to one subtree of Ti[k] which has maximum conceptual similarity with Tu[j]. The conceptual similarity between Tu[j] and Ti[k] is represented as:

*Case III:* ti[k] is matched to tu[j]’s child

The concept level of ti[k] is lower than the concept level of tu[j]. Ti[k] is mapped to one subtree of T u [j] which has maximum conceptual similarity with T i [k]. The conceptual similarity between Tu[j] and Ti[k] is calculated as:

Considering the three cases given previously, the case with the maximum conceptual similarity is selected, and the conceptual similarity calculated in that case is taken as the conceptual similarity between Tu[j] and Ti[k]:

**Algorithm 2**: Tree Matching Algorithm

*input:* Two trees Tu[j], Ti[k] and empty mapping set M

*output:* Conceptual similarity between Tu[j] and Ti[k]

***7. FUZZY PREFERENCE TREE CONSTRUCTION ALGORITHM***

In this algorithm, a fuzzy preference tree is constructed, which includes both the intention and extensional preferences of the user. As explained earlier, the intentional preference of the user are already present in the form of the User Tree and the extensional preferences are obtained via the Tree Matching Algorithm.

The construction algorithm is presented in detail here.

**A. Fuzzy Preference Tree Construction Algorithm**

The fuzzy tree construction process is incremental,i.e., new items can be merged into it without any problems. It takes as input: the user’s fuzzy preference tree Tu , the item tree Ti, and the user’s preference value for the item p˜ui. It contains two steps.

**Step 1**: **Find Maximum Conceptual Similarity Mapping**

The maximum conceptual similarity tree mapping between the user tree,Tu, and the item tree, Ti is generated. A maximum conceptual similarity tree mapping between Tu and Ti, M is constructed to identify the corresponding parts between two trees and to determine the positions in Tu into which the relevant nodes in Ti can be merged.

**Step 2:** **Merge Ti into Tu**

Based on M ,the maximum conceptual similarity tree mapping, all the features in Ti are absorbed into Tu . A merge operation is defined which takes the mapping M, a user tree node Tu, an item tree node Ni and the fuzzy preference value for that node pui as inputs. According to the different mapping situations of ni, the merge operation is processed in the following five cases:

*Case I*: M is empty. This case is appropriate when Tu, the user tree is initially empty or Tu and the Ni’s subtree contain totally different items. Here, Tu is inserted as a subtree in a new Tu root. Ni’s subtree is inserted under the new root of Tu . Each leaf this subtree is initialized with count = 1 and preference value of Pui.

*Case II*: Ni is mapped to a node Np, but Ni’s and Np’s subtrees are not identical. Here, Ni’s subtree is inserted under Np’s parent. Each leaf of this subtree is initialized with preference value Pui and count = 1.

*Case III*: Ni is mapped to a node Np with identical attributes. We have two cases here.

*Case a*: Ni represents the finest feature, i.e, Ni has no children. The Pui is integrated into the preference value of node np which is denoted as Punp : fk,unp = fk,unp · count + fk,ui / (count + 1),k = 1, 2,...,r; count = count + 1.

*Case b:* Ni is not the finest feature and its child nodes are recursively merged.

*Case IV*: Not Ni but its parent is mapped to Np in M. Ni’s subtree is inserted under Np. Each leaf of the copied subtree is initialized with preference value of Pui and count = 1.

*Case V:* Not Ni, not Ni’s parent but Ni’s descendant is present in the mapping M. Thus, the features under Ni are partially represented by Tu. The root of Tu is mapped to Nt, Ni’s descendant. The tree under Ni apart from Nt’s subtree is taken as T’u . Each leaf of the copied subtree is initialized with Preference Value Pui and count = 1. Tu is inserted into T’u under Np’s corresponding node . T’u replaces Tu, and Nt is recursively merged.

The aforementioned process is shown in the following algorithm. Some of the functions are defined here:

After the merging operation is finished, the updated fuzzy preference tree’s(Tu) weights are normalized.

**Algorithm 3:** Fuzzy Tree Creation algorithm

*input:* User Tree Node, Item Tree Node, maximum conceptual similarity tree mapping M

*output*: Fuzzy User-Preference Tree

***8. FUZZY PREFERENCE TREE-BASED RECOMMENDATION APPROACH***

In this section, we propose a fuzzy preference method for recommendation. We take a user’s fuzzy preference tree Tu and the item tree Ti as input, and calculate the predicted rating of the user to the items in the Item Tree.

This approach is applied via two steps:

**Step 1: Identifying the Corresponding Parts of Tu and Ti**

Maximum conceptual similarity mapping M between the User Tree and Item Tree is calculated via the tree matching algorithm defined above,

**Step 2: A Fuzzy Preference Tree-Based Recommendation Approach**

Given Tu, a user’s fuzzy preference tree and , the item tree, the maximum conceptual similarity tree mapping between them, M, is constructed.

The function takes this mapping M and the fuzzy preference tree Tu and calculates the fuzzy preference rating based on them.

Let represent the value of node is not assigned a value. Let represent the child nodes of that are in the maximum conceptual similarity tree mapping.

According to whether in different cases are null or not, is calculated in the following four cases:

The subtree Tu[j] makes no contribution to the predicted rating

The node tu [j] is assigned a preference value. Let it be

*Case 3:* The predicted rating value of is calculated by taking the summation of the predicted ratings of its mapped children.

*Case 4*: Combination of second and third cases. Both the values of tu[j]’s children and itself should be considered.

**)** ----(8),

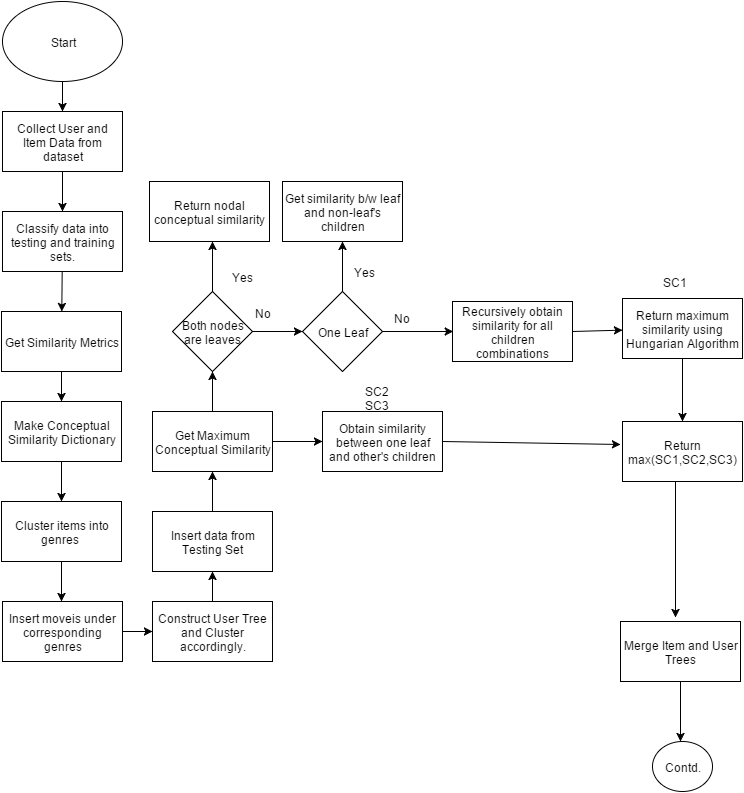
where and βj is the influence factor(taken arbitrarily for our purpose) of the parent node tu [j]. The calculation process of the predicted rating is shown in Algorithm 4.

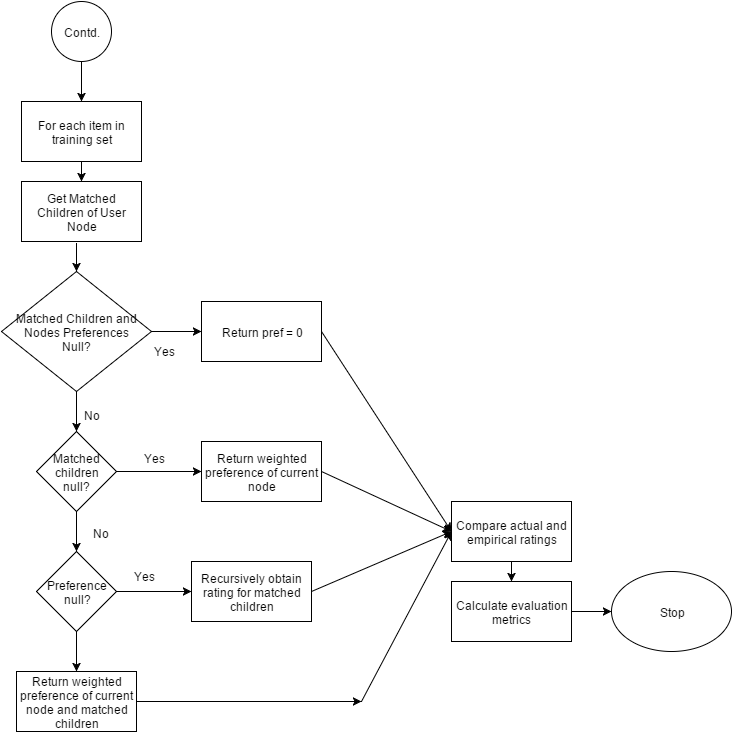
**Algorithm 4:** Rating prediction algorithm

*input:* fuzzy preference tree node, the maximum conceptual similarity tree mapping M

*output:* the predicted rating

***9. FLOWCHART:***

******

******

***Fig 9.1 – Flowchart and the activities of the Project***

***10. EXPERIMENTAL EVALUATION***

This section evaluates the performance of our modified fuzzy preference tree-based recommendation approach by comparing it with the original approach and other existing recommendation approaches. We also explain the various evaluation metrics used.

**1. Dataset:**

The MovieLens 100k dataset is used for evaluation. Since this dataset has been used to test many other approaches, it’s ideal for using it to compare the performance of our approach with other approaches.

In our experiment, each movie has one main attribute: the genres it is linked with. There are 945 users in the dataset. For the experiment, each user’s ratings are split into two parts, the training set and the testing set, and 20% ratings of each user make up the testing set. The movies in a user’s training set are used to construct that user’s preference profile tree and the predicted ratings are compared with the ratings in the testing set. The ratings of users on movies in the dataset are on a scale of 1 to 5 and thus is transformed into a fuzzy subset on {1, 2, 3, 4, 5}.

**2. Evaluation Metrics:** The following metrics have been used for comparison in our study.

* **Mean Absolute Error (MAE)**: The mean absolute error (MAE) is the most widely used statistical accuracy metric in recommendation systems [19]. MAE is calculated as the average of the absolute difference between actual and predicted ratings.

In particular, given the set of the actual/predicted rating pair (ra,rp) for all the n items in the test set, the MAE is computed by

* **Recall, Precision, and F1 Metrics:**

Recall is defined as the fraction of preferred items that are recommended.

Precision is defined as the fraction of recommended items preferred by the user.

The F1-measure, is the harmonic mean of precision and recall.

In this experiment, a preferred rating threshold is predefined (namely >= 3). The preferred movies are the movies in the test set whose actual ratings are greater than the preferred rating threshold**. The recommended movies are the movies whose predicted ratings are greater than the preferred rating threshold**.

The recall, precision, and F1 are defined as follows:

**3. Evaluation Results**

Figs. 10.1 – 10.4 show the MAE, precision, recall, and F1 of each recommendation approach. In these figures, the first approach is **our modified proposed fuzzy preference tree-based recommendation approach for the most experienced user** and the second is the **average of all metrics over 5 users using our approach**. The third approach is the original unmodified fuzzy preference tree-based recommendation approach.

Approaches 3 to 6 are the FTM methods with fuzzy set theoretic, cosine, proximity, and correlation-like similarity measures, respectively. The sixth approach is the crisp set-based method. The seventh approach is the fuzzy user preference model-based method [20].

**Evaluation Results on the MovieLens Dataset**: As we can see from the evaluation results on the MovieLens dataset, the original fuzzy preference tree-based approach has the lowest MAE.

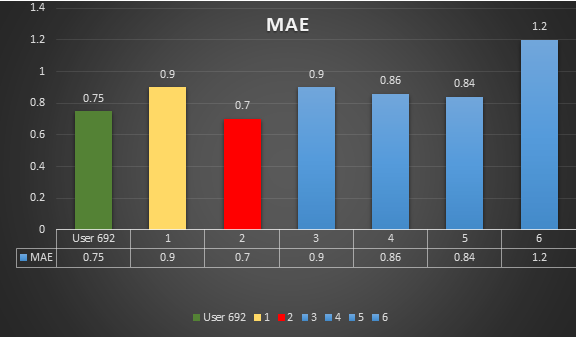
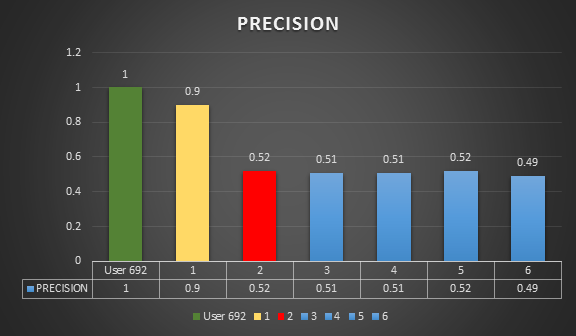


Fig 10.1:

Mean Absolute Error

Our modified approach has the second lowest MAE, though slightly higher than the original. This is because of the time-accuracy trade-off. To save multiple redundant recursions, we have clustered the movies in the dataset into various genres and applied the algorithm to each cluster, thereby reducing the time taken by a very large margin (~over 20 minutes to under 40 seconds).

Since, this approach cuts back on the various comparisons and bi-partite matching, the MAE increases slightly.

Fig 10.2:

Precision

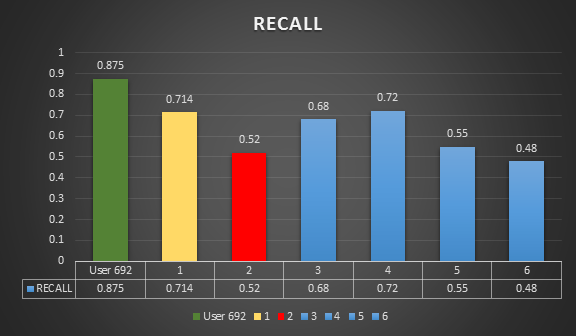


Fig 10.3:

Recall

However, our modified approach has the highest precision and recall out of all the algorithms tested and even more than the original approach.

Thus, experimental results show that high recommendation accuracy is obtained by representing the user preferences with our proposed fuzzy tree-structured preference mode. As a result, the effectiveness of the fuzzy preference tree-based model and the recommendation approach based on it is reflected quite clearly.

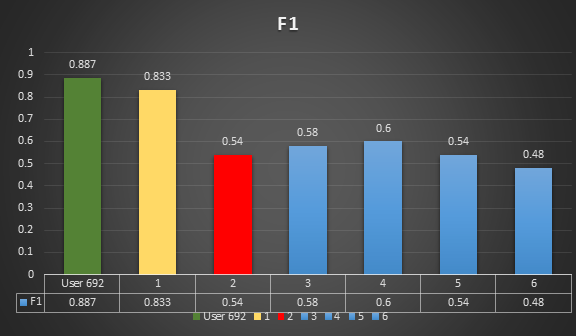


Fig 10.4: F1

**11. *FURTHER STUDIES AND CONCLUSION:***

This research paper proposes a method for modeling tree-structured user preferences and develops a new recommendation approach, which can recommend tree-structured items, by making a prediction on items ratings, based on the previously rated items by the user. The fuzzy tree-structured user preference modeling method integrates both the user’s extensionally and intentionally expressed preferences. During the construction process of the user’s fuzzy preference tree, the item tree and the matching process between the fuzzy preference tree and item trees, a comprehensive tree matching method for identifying the corresponding parts of two tree-structured data is presented, which comprehensively considers tree structures, nodes similarities, node attributes, and node weights. Experiment on the MovieLens dataset is conducted to evaluate the performance of the proposed recommendation approach. The results show that our approach makes valid recommendations and demonstrates that the fuzzy tree-structured user preference profile reflects user preferences effectively. This approach is also well-suited to the business application environment. The proposed recommendation approach is implemented in a business partner recommender system software.

At the present research stage, the inputs of the recommendation technique require tree-structured data and cannot deal with data in a network structure or a matrix. In addition, this approach requires that the tree-structured data must fulfil the requirement that the semantic meanings of the parent–child relations in different tree-structured data must be the same. However, this approach provides a new solution for improving recommender systems in general and it can therefore be used in e-learning, e-Government, e-Business, and so on when the data are described in a tree like structure.

In the future, we will consider the attributes and characteristics of groups of similar businesses or fields and will develop methods for identifying business groups and make group recommendations.

**REFERENCES:**

[1]-R.Burke,“Hybrid recommender systems:Survey and experiments,”User Modeling User-Adapted Interact., vol. 12, pp. 331–370, 2002.

[2] Z.Huang,D.Zeng,and H.Chen,“A comparison of collaborative-ﬁltering recommendation algorithms for e-commerce,” IEEE Intell. Syst., vol. 22, no. 5, pp. 68–78, Sep./Oct. 2007.

[3] J. 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. Berlin, Germany: Springer-Verlag, 2007, pp. 291–324.

[4] B.Sarwar, G.Karypis, J.Konstan,and J.Reidl,“Item-based collaborative ﬁltering recommendation algorithms,”in Proc.10th Int.Conf.World Wide Web, 2001, pp. 285–295

[5] R. M. Rodr´ıguez, M. Espinilla, P. J. S´anchez, and L. Mart´ınez-L´opez, “Usinglinguistic incomplete preference relations to cold start recommendations,” Internet Res., vol. 20, pp. 296–315, 2010.

[6] L.Yang,B.Sarker,V.Bhavsar,andH.Boley, “A weighted-treesimplicity algorithm for similarity matching of partial product descriptions,”inProc. Int. Soc. Comput. Appl. 14th Int. Conf. Intell. Adaptive Syst. Softw. Eng., Toronto, ON, Canada, 2005, pp. 55–60.

[7] R. Connor, F. Simeoni, M. Iakovos, and R. Moss, “A bounded distance metric for comparing tree structure,”Inf.Syst.,vol.36,pp.748–764,2011.

[8] F. Ricci and L. Senter, “Structured cases, trees and efﬁcient retrieval,” Adv. Case-Based Reason., vol. 1488, pp. 88–99, 1998. [37] P. Bille, “A survey on tree edit distance and related problems,” Theor. Comput. Sci., vol. 337, pp. 217–239, 2005.

[9] K. Zhang, “A new editing based distance between unordered labeled trees,” in Combinatorial Pattern Matching, A. Apostolico, M. Crochemore, Z. Galil, and U. Manber, Eds. Berlin, Germany: Springer-Verlag, 1993, pp. 254–265.

[10] D. Wu, J. Lu, and G. Zhang, “Similarity measure models and algorithms for hierarchical cases,” Expert Syst. Appl., vol. 38, pp. 15049–15056, 2011. [40] D. Wu, J. Lu, and G. Zhang, “A hybrid recommendation approach for hierarchical items,” in Proc. Int. Conf. Intell. Syst. Knowl. Eng., 2010, pp. 492–497.

[11] N. Biadsy, L. Rokach, and A. Shmilovici, “Transfer learning for content based recommender systems using tree matching,” in Availability, Reliability, and Security in Information Systems and HCI, A. Cuzzocrea, C. Kittl, D. Simos, E. Weippl, and L. Xu, Eds. Berlin, Germany: Springer, 2013, pp. 387–399.

[12] C.Cornelis,J.Lu,X.Guo,and G.Zhang,“One-and-only item recommendation with fuzzy logic techniques,” Inf. Sci., vol. 177, pp. 4906–4921, 2007.

[13] C.Porcel,A. G.L´opez-Herrera,and E.Herrera-Viedma,“A recommender system for research resources based on fuzzy linguistic modeling,”Expert Syst. Appl., vol. 36, pp. 5173–5183, 2009.

[14] J.Lu,Q.Shambour,Y.Xu,Q.Lin,and G.Zhang,“Biz Seeker:A hybrid semantic recommendation system for personalized government-to-business e-services,” Internet Res., vol. 20, pp. 342–365, 2010.

[15] Z.Huang,D.Zeng,and H.Chen,“A comparison of collaborative-ﬁltering recommendation algorithms for e-commerce,” IEEE Intell. Syst., vol. 22, no. 5, pp. 68–78, Sep./Oct. 2007.

[16] J. 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. Berlin, Germany: Springer-Verlag, 2007, pp. 291–324.

[17] L. P. Hung, “A personalized recommendation system based on product taxonomy for one-to-one marketing online,” Expert Syst. Appl., vol. 29, pp. 383–392, 2005.

[18] Graphs,Networks and Algorithms Volume 5 by Deiter Jungnickel.

[19] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, “GroupLens: An open architecture for collaborative filtering of netnews,” in Proc. ACM Conf. Comput. Supported Cooperative Work, Chapel Hill, North Carolina, USA, 1994, pp. 175–186.

[20] A. Zenebe, L. Zhou, and A. F. Norcio, “User preferences discovery using fuzzy models,” Fuzzy Sets Syst., vol. 161, pp. 3044–3063, 2010.

[21] A Fuzzy Preference Tree-Based Recommender System for Personalized Business-to-Business E-Services Dianshuang Wu, Guangquan Zhang, and Jie Lu, Senior Member, IEEE