A Fuzzy Content Matching-based e-Commerce Recommendation Approach

Mingsong Mao, Jie Lu, *Senior Member, IEEE,* Guangquan Zhang, *Member, IEEE,* Jinlong Zhang,

***Abstract*—E-Commerce products often come with rich and tree-structured content information describing the attributes. To well utilize the content information, this study proposed a fuzzy content matching-based recommendation approach to assist e-Commerce customers to choose their truly interested items. In this paper, users' ratings and preferences are represented using fuzzy numbers to remain uncertainties. Tree-structured content information is transformed to a set of descriptors, and users' preferences on these descriptors are derived from fuzzy ratings by using fuzzy number operations. A kind of preference dependence relations is established between descriptors to explore the relations of different content features, and as a base to sketch the complete profile of users. While the extended preference profile of a user is established, given a new item, the fuzzy match degree of the user preference and the item content information is carried out, and then a fuzzy Topsis ranking method is proposed to able to rank all candidate items according to the fuzzy match degrees, and the highest ranked items are recommended to the target user. We conduct empirical experiments on Yelp and MovieLens datasets. The results indicate that the proposed approach improve recommendation performance in terms of both coverage and accuracy.**

***Index Terms*—Recommender systems ,Fuzzy preference, E-commerce, Fuzzy ranking**

I. Introduction

Recommender systems are emerging as effective shopping assistants for online e-Commerce sites with increasingly large-scale-data where it is difficult for users to navigate all candidate items and discover what potentially interests them. After decades of development, recommender systems have been able to well study the users' preference profiles in terms of the content attributes of items. For example, from the user-item rating records and movie genre information, the profile of an individual user can be represented by his average ratings to each type of movies. This is a very simple example, and the content information of movies is "flat", meaning the content features (in this case the movie genres) can be mapped to a vector space. However, despite such non-structural content information, e-Commerce products are often associated by complex structural content information, which is usually called as the taxonomy tree of one (or multiple) domain of items. In some ontology-based recommender systems, item taxonomy has been well utilized to improve the accuracy of user profile molding and recommendation making [1]–[3]. It is accepted that the taxonomy information provides a means of discovering the relations between item content and user profile [4].

To handle the tree-structured taxonomy information, previous studies mainly employ tree-matching techniques such as tree similarity measure, tree isomorphism, and sub-tree matching to establish a kinds of semantic similarities between item to items or user to items. For example, in the food recommender system for diabetes patients developed by Arwan *et al*. [5], both items (the food menus) and users (the patients) are represented by weighted trees with the same structure to denoting the nutrition supply and demand of foods and patients, respectively. Based on the construct ontology of users, foods, and nutrition taxonomy, the similarities of patient-to-patient and food-to-food are calculable so that new diet suggestions can be produced based on previous successful cases. Biadsy *et al*.propose a transfer learning model for content-based recommender systems, in which the ratings on a pacific domain are modeled as tree-structured patterns of users as the base of transfer leaning on other domains [6]. In the work of Wu et.al [7], the semantical (structural) comparison of user-to-user profile trees, item-to-item content trees and user-to-item preference trees are comprehensive discussed and resolved by using tree and subtree matching techniques.

New issues also arise for modern content-based recommender systems. First, the semantic tree matching models usually assume the content features in a same and deeper branch of the taxonomy tree are having high semantic similarities such that people will have similar preferences on them. These models are thereby hard to discover the potential relations of features from different taxonomy categories. Actually, the features from different aspects may be also related closely. For example, an action movie actor is high related the the movie genre *Action*. A second issue is the uncertainty of data such as the fuzzy membership of an item and taxonomy features. Besides, the ratings and preferences of users are actually subjective and vague [8]. Motivated by the two issues, this paper propose a fuzzy tree model to handle the uncertainties of users and items, and develop an fuzzy inference model to discover possible relations of different taxonomy features. The contributions of this study can be concluded as three aspects as follows. 1) We propose the fuzzy dependence relations between content features, based on which we can greatly enrich the initial preference profile of a user. 2) We propose a fuzzy prediction model to infer users' preferences on unknown items in the form of fuzzy numbers rather than crisp values. 3) We propose a fuzzy Topsis ranking method to be able to rank the candidate items with fuzzy predictions to generate

Mingsong Mao (mingsong.mao@student.uts.edu.au), Jie Lu (jie.lu@uts.edu.au), Guanquan Zhang (Guangquan.Zhang@uts.edu.au) are with Decision Systems and e-Service Intelligence Lab, Centre for Quantum Computation and Intelligent Systems, University of Technology Sydney (UTS), Australia

Mingsong Mao and Jinlong Zhang (lzhang@mail.hust.edu.cn) are with Institute of Management Information, School of Management, Huazhong University of Science and Technology (HUST), Wuhan, China

recommendations.

The remainder of this paper is organized as follows. Section II presents a fuzzy data model where each item or user can be represented as a fuzzy weighted subtree of the complete content taxonomy. Besides, the preference tree of user can be expand according to the proposed dependence relationships between content features. In Section III, a fuzzy content matching-based recommendation approach is elaborated step by step. In Section IV, empirical evaluation of proposed approach is conducted with experiments on two real-wold datasets representing different e-Commerce environments. Finally, a discussion of the proposed approach and future directions are discussed in Section V.

II. Modeling Item Content and User Preference

In this section, we introduce the item representation and user representation using fuzzy tree model. The techniques of fuzzy number and fuzzy operations are used to explore inherent similarities between different content features. As a result, a user's original preferences on a small number of features can be expanded to more features so that an extended fuzzy preference tree is constructed for every user.

A. Preliminary of Fuzzy Numbers and Its Operations

The definition of fuzzy sets and fuzzy numbers (here the triangular fuzzy number is used) are imported from [9]:

Definition 1. ***(Fuzzy set)*** *If X is a collection of objects denoted generically by x then a fuzzy set $\tilde{A}$ is a set of ordered pairs:*

$$\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) | x \in X\} \tag{1}$$

$\mu_{\tilde{A}}(x)$ is called the membership function of x in $\tilde{A}$ which maps X to the closed interval $[0, 1]$ that characterizes the degree of membership of x in $\tilde{A}$.

Definition 2. ***(Triangular fuzzy number)*** *A triangular fuzzy number (TFN) is denoted by $\tilde{A} = (a, b, c), a \leq b \leq c$, if its membership function is*

$$\mu_{\tilde{a}}(x) = \begin{cases} \frac{x-a}{b-a}, & \text{if } a \leq x \leq b \\ \frac{x-b}{c-b}, & \text{if } b \leq x \leq c \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

With the extension principle put forward by Zadeh [10], the operational laws of two TFNs $\tilde{A}_1 = (a_1, b_1, c_1)$ and $\tilde{A}_2 = (a_2, b_2, c_2)$ are defined as follows:

- Addition of two fuzzy numbers

$$(a_1, b_1, c_1) \oplus (a_2, b_2, c_2) = (a_1 + a_2, b_1 + b_2, c_1 + c_2)$$

- Subtraction of two fuzzy numbers

$$(a_1, b_1, c_1) \ominus (a_2, b_2, c_2) = (a_1 - a_2, b_1 - b_2, c_1 - c_2)$$

- Multiplication of any real number k and a fuzzy number

$$k \otimes (a, b, c) = (ka, kb, kc)$$

B. Notations

At first, some key concepts and entities of this study are given as follows.

1) Users and items: A set of users $U = \{u_1, u_2, \ldots\}$ are the participants of an e-Commerce recommender system. The item set $T = \{t_1, t_2, \ldots\}$ includes all products or services that are provided for users to choose in a recommender system.

2) User fuzzy ratings: A rating set $R \in \mathbb{R}^{|U| \times |T|}$ is the set of ratings assigned by users to items, where an element R_{ut} denotes the rating given by a particular user u to an item t. In this study, a rating is treated as a fuzzy number $\tilde{r}(u, t)$ as that ratings are usually given actually in the form of graded vague linguistic evaluations. For instance, in Amazon, users rating items in the range of 1 star to 5 stars, representing that "not satisfied at all" to "bets satisfied". We use triangle fuzzy numbers to represent ratings. Initially, the fuzziness (the base of a TFN) of an initial rating is simply assumed to be ± 1, as denoted in Fig. 1. Therefore, a rating of "3 stars" refers to a TFN of $\tilde{r} = (2, 3, 4)$ in our study.

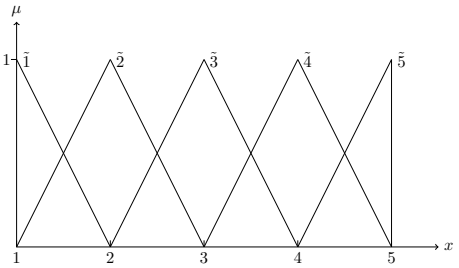


Fig. 1: Initial representation of fuzzy ratings

3) Complete content tree and descriptor set: In e-Commerce environments, items are usually described with rich, hierarchical content features [7]. The content features of all items are called as the taxonomy tree of the whole item set, denoted as $\Theta = (C, \hookrightarrow)$, where $C = \{c_1, c_2, \ldots, c_{|C|}\}$ is a finite set of nodes (features or subfeatures of items), and $\hookrightarrow$ is a "parent-child (feature-subfeature) relationship. For two nodes $c_i, c_j \in C$, if $c_i \hookrightarrow c_j$, then c_j is a child (subfeature) of c_j. Note that there is no cycles and there is a distinguished root node of Θ, denoted as $c_\top$.

Clearly, there is one and only one path from the root to any other nodes in the tree, and we say, a path from the root node $c_\top$ to every leaf node $c_\perp$ is a *descriptor*, marked as $d_\perp$. Suppose there are in total m leaf nodes in the tree Θ, then a complete descriptor set is constructed as $D = \{d_i\}, i = 1, \ldots, m$, where a single descriptor is denoted as $d_i = \{c_{i1}, c_{i2}, ..., \ldots, c_{i|d_i|}\}$ satisfying $c_{i1} \hookrightarrow c_{i2} \hookrightarrow \ldots \hookrightarrow c_{i|d_i|}$. Essentially, the descriptor set D can be seen as the "flat" form of the tree Θ, and inversely, Θ is the hierarchical form of D. An example is given in Fig. 2. In the case, the content tree Θ is a three level tree with 11 nodes denoting the content features and subfeatures. There are in total seven descriptors $d_1, \ldots, d_7$ collected, and the whole set of descriptors represents the flat form of the content tree.

4) Item fuzzy content tree: The content information of a specific item $t \in T$ consists of two parts: the qualitative part

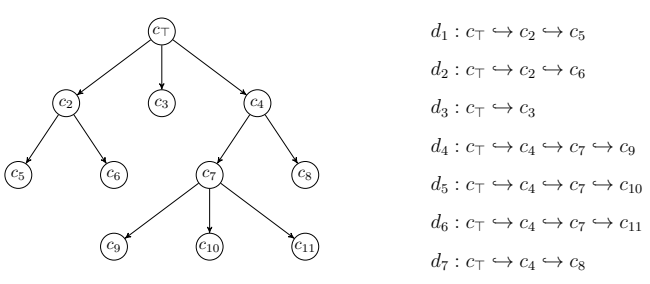


Fig. 2: A three-level complete content tree with 11 nodes (features) and the corresponding descriptors set.

and the quantitative part. The qualitative part is the structure of the features associated with this item, can be seen as a part of the complete content tree, denoted as $\Theta_t = \{C^t, \hookrightarrow\}$, in which $C^t \in C$. The quantitative part indicates the degree of fuzzy "membership" of this item to each tree node (a feature), denoted by a membership degree $x \in [0, 1]$. The fuzziness of item content information is very common in the real world. For example, a book t is thought to be relevant to a content feature "c_1: Computer Science" with a degree of $x(t, c_1) = 1$, while it is also considered to relate to "c_2: Management" with a degree of $x(t, c_2) = 0.8$. Consequently, for each item, a fuzzy tree $\Theta_t = \{C^t, \hookrightarrow, X\}$ can be built as the representation of itself, in which, C^t is the nodes set, $\hookrightarrow$ is the "feature-subfeature" relationships, and $X = \{x(c)\}, c \in C^t$ is an extra set denoting the membership degrees. Fig. 3 shows an example item and its content tree Θ_t. We can find that the structure of Θ_t is a part of the complete content tree Θ, but it associates a decimal of each node, announcing the membership degree of this item to this feature.

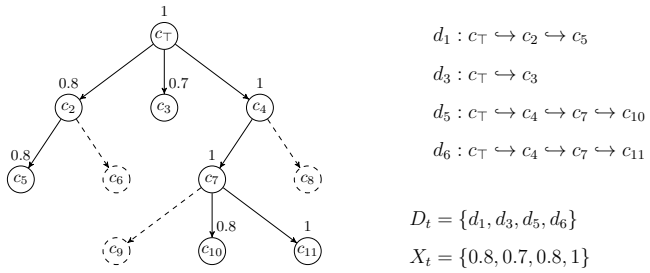


Fig. 3: For a single item, its content information is a part of the complete content tree Θ, and its descriptor set D_t is a subset of the complete descriptor set D. Additionally, for each node or descriptor, a decimal is associated to represent the "membership" degree of this item.

5) Item descriptor set: For an item t with its fuzzy content tree Θ_t, assuming it has m_t leaf nodes in the tree, we can then find m_t paths from the root node to each leaf node, which are called as descriptors, denoted as $D_t = \{d_1^t, \ldots, d_{m_t}^t\}$. We have known that the qualitative part of an item content tree Θ_t is a part of the complete content tree Θ, but note that the item descriptor set D_t is not guaranteed to be a subset of D, because a leaf node of Θ_t may be not a leaf node in Θ. That is to say, we allowed the "incomplete" description of items, *e.g.*, we have a three levels taxonomy tree being used to describe books, but for a particular book, it can be detailed only to two levels. It is easily to obtain following relationships:

$$D \subseteq \cup_{t \in T} D_t \tag{3}$$

For each single descriptor $d \in D_t$, a membership degree is also allocated, represented by the membership of the last (leaf) node, that is, $x(t, d) = x(t, c_\perp), c_\perp \in d$. In Fig. 3, the descriptor set of the example item contains four descriptors, and a membership set X is established associating with each descriptor.

C. User fuzzy preference

Before discussing user preferences, an elaboration of "how a personalized rating is assigned by a user to a specific item" is needed. Keeping in mind that users' personalities lead to their different ratings (higher or lower) to a same item. The gap between an individual user's rating against the average rating (indicating the inherent qualities of items) can be seen as his/her explicit expression personality. Inversely, we can obtain the basic assumption of our study: when a user face an item, there are two factors that impact his subjective rating to this item: a) the inherent quality of this item, and b) the personality of him/her. For the first part, the inherent quality can be seen as a constant for every item, and simply, can be represented by the average ratings. For the second part, a user's personality for a specific item arises from his/her unique preferences on the content information of the item.

We establish the whole profile a user by summarizing his/her (fuzzy) preference on each single descriptor in the complete descriptor set D, which is defined as follwos.

Definition 3. *Fuzzy preference on a single descriptor*
For a given user $u \in U$, denoting T_u the set of items that have been rated him/her, for a single descriptor $d \in D$, only if $\exists t \in T_u : d \in D_t$, the fuzzy preference of this user to the descriptor is calculable, and being calculated by:

$$\tilde{y}(u, d) = \frac{\sum_{t \in T_u} \left(\tilde{r}(u, t) - \bar{r}_t \right) x(t, d)}{\sum_{t \in T_u} x(t, d)} \tag{4}$$

where $\bar{r}_t$ is the average rating of item t. Let U_t denote the users who have rated t , we have:

$$\bar{r}_t = \frac{\sum_{u \in U_t} \tilde{r}(u, t)}{|U_t|} \tag{5}$$

With our initialization, the value range of a fuzzy rating is $(1, 1, 5) \precsim r \precsim (4, 5, 5)$. According to Eq. 4, we can have the value range of a fuzzy preference score is $(-4, -4, -2) \precsim \tilde{y} \precsim (2, 4, 4)$, so that the definition domain of $\tilde{y}$ is [-4,4], where a positive score indicates a positive preference to the descriptor, while a negative score indicates a negative preference such as "disliking","rejecting", *etc.*. Comparing to crisp values, the fuzzy preference can preserve the uncertainties derived from fuzzy ratings.

It can be seen that Eq. (4) is only practicable for the descriptors that have been 'reached' by the target user, *i.e.*, the user has rated the items that contain the certain descriptors. Due to the fact that an individual user commonly reviews only a small part of millions of items in an e-Commerce site, the initialized preferences of him/her may be only available for a few descriptors. To handle this issue, we infer the missing preference information of the not-reached descriptors by first exploring the descriptor-to-descriptor relationships. Differing from conventional tree matching models that only expand user preference to semantically similar features (with near distance in the content tree), we transfer the tree structure to flat descriptor set and propose a kind of cross-dependence relationships between any descriptors, so that we can expand user preference to any other content features even if they look not relevant in the taxonomy tree.

D. Extended User Preference

With the proposed data model, we assume the existence of cross-dependence relationships between different descriptors, for instance, in the form of "if a user likes/dislikes d_1 then he/she also likes/dislikes d_2". Therefore, we compare two descriptors to see whether they are consistently preferred by users, and those descriptors that always share similar preferences are considered to have strong dependence relationship.

Because the preferences are represented using fuzzy numbers, a fuzzy number closeness/distance calculation method is needed to compare users' preferences, we hence an area-based fuzzy number comparison method. Similar to [11], we compare fuzzy numbers based on the proportion of overlap area of their membership functions.

Definition 4. *Closeness and distance of fuzzy numbers*
Given two triangle fuzzy number $\tilde{A}$ and $\tilde{B}$, whose membership function are $\mu_{\tilde{A}}(x)$ and $\mu_{\tilde{B}}(x)$, respectively, on the same domain of $x \in \mathbb{R}$. The closeness of $\tilde{A}$ and $\tilde{B}$ is:

$$\ell(\tilde{A}, \tilde{B}) = \frac{\int \mu_{\tilde{A}}(x) \wedge \mu_{\tilde{B}}(x) dx}{\int \mu_{\tilde{A}}(x) \vee \mu_{\tilde{B}}(x) dx} \in [0, 1] \tag{6}$$

and their distance is:

$$\delta(\tilde{A}, \tilde{B}) = 1 - \ell(\tilde{A}, \tilde{B}) \tag{7}$$

Noticing that, not only for TFNs, Eq. (6) can be applied to fuzzy numbers with general membership functions. Figure 4 illustrates the calculation of Eq. (6): the subfigure (a), (b) and (c) show three possible situations of the overlapping area of two TFNs $\tilde{A}$ and $\tilde{B}$. The subfigure (d) shows that for two general fuzzy numbers, their overlapping area is enclosed by the lower bound of the two fuzzy numbers and the x-axis, while their union area is enclosed by their upper bound and the x-axis. Therefore, Eq. (6) works for general fuzzy numbers as well as TFNs.

Consequently, users' preferences on different descriptors becomes comparable, and the definition of preference dependence of descriptors is given as follows.

Definition 5. *Preference dependence of descriptors*
For two descriptors $d_i, d_j \in D, d_i \neq d_j$, first, we define the situation of "a user has similar preferences on the two descriptors" is equivalent to:

$$\delta\Big(\tilde{y}(u, d_i), \tilde{y}(u, d_j)\Big) \leq \epsilon \tag{8}$$

where ϵ is a small positive threshold, for example, in this study, we let $\epsilon = 0.1$.

Next, the preference dependence degree of d_i to d_j is:

$$p(d_j|d_i) = \frac{\#\text{users satisfying } Eq.(8)}{\#\text{users that } \tilde{y}(u, d_i) \text{ is available}} \tag{9}$$

Now we extend the user preferences by fulfilling the missing preferences to not-reached descriptors. For a user $u \in U$, given a descriptor $d_j \in D$ that has no directed preference information, the preference of this user to this descriptor is estimated by:

$$\tilde{y}(u, d_j) = \frac{\sum_{d_i \neq d_j} \tilde{y}(u, d_i) p(d_j|d_i)}{\sum_{d_i \neq d_j} p(d_j|d_i)} \tag{10}$$

Comparing to semantic similarities, the descriptor preference dependence relationships can discover potential dependencies of two features that look not relevant in the structure of content tree.

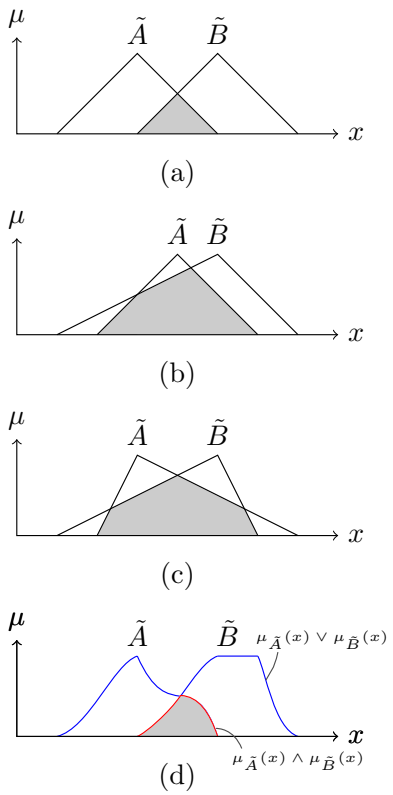


Fig. 4: Different situations of area-based closeness calculation of two fuzzy numbers. (a), (b) and (c) are the three possible overlapping areas of two TFNs; and (d) shows the situation of two general fuzzy numbers.

III. Fuzzy Content Matching and Recommendation

In this section, a fuzzy content matching-based recommendation approach is developed based on the proposed fuzzy

data model. In summary, following input information shall be obtained prior to recommendation making: 1) for a user $u \in U$, there is a TFNs vector $\tilde{\mathbf{Y}}_{\mathbf{u}} = \{\tilde{y}(u,d), d \in D\}^{|D|}$ representing the fuzzy preference of him/her on each descriptor.r; 2) for an item $t \in T$, there is a decimal vector $\mathbf{X_t} = \{x(t,d), d \in D\}^{|D|}$ representing the membership degree of this item to each descriptor. When a user is fixed as the target user, the goal of our approach is to identify the a certain number (*e.g.*top-k) from the whole item set as recommendations for the user.

A. Fuzzy Rating Prediction

As discussed, a personalized rating is affected by two aspects: 1) the inherent quality of item and 2) the user's preference on the content features of the item. Therefore, when given a pair of user and item as a single prediction task, we predict the fuzzy rating by considering the both factors.

Easily, the average fuzzy rating issued by different users is treated as the inherent quality of an item $t \in T$, denoting as $\bar{r}_t$, referring to Eq. (5).

For the second aspect, the overall preference of the given user on the content information of the given item can be generated by a content matching process, which is calculated by:

$$\tilde{P}_c(u,t) = \frac{\tilde{\mathbf{Y}}_{\mathbf{u}} {\mathbf{X}_{\mathbf{t}}}^T}{\sum \mathbf{X_t}} \tag{11}$$

Accordingly, the prediction of the fuzzy rating of a pair of user u and item t is calculated as follows:

$$\begin{aligned} \hat{r}(u,t) &= \bar{r}_t + \lambda \tilde{P}_c(u,t) \\ &= \bar{r}_t + \lambda \frac{\tilde{\mathbf{Y}}_{\mathbf{u}} {\mathbf{X}_{\mathbf{t}}}^T}{\sum \mathbf{X_t}} \end{aligned} \tag{12}$$

where λ is a nonnegative parameter adjusting the weight of user personalization. For example, setting a high level of λ can be understood as users' ratings are influenced more by their personalized preferences of the content information, and be less impacted by the quality of items, that is, for this domain of items, people are more personalized.

B. Fuzzy Topsis Ranking

We propose a fuzzy Topsis ranking method to rank the items w.r.t the fuzzy predictions, as known as TFNs. For a target user $u \in U$, assuming there are m alternative items and predictions have been generated as $\tilde{p}_i = (a_i, b_i, c_i), i = 1, 2, \ldots, m$. The fuzzy Topsis ranking process is carried out in following steps:

Step 1: Determine the worst and best conditions. Defining the minimum left bound as $a_{min} = \max_{i=1}^{m} a_i$, and the maximum right bound is $c_{max} = \max_{i=1}^{m} c_i$. The worst condition is determined as a TFN $\tilde{p}^- = (a_{min}, a_{min}, c_{max})$, and the best condition is $\tilde{p}^* = (a_{min}, c_{max}, c_{max})$.

Step 2: Calculate the distance to the worst and best conditions. The distance of the prediction of an item to the worst condition is:

$$d_i^- = \delta(\tilde{p}_i, \tilde{p}^-) \tag{13}$$

The distance of the prediction of an item to the best condition is:

$$d_i^* = \delta(\tilde{p}_i, \tilde{p}^*) \tag{14}$$

Step 3: Ranking score. The alternative items are ranked according to the score of $f_i, i = 1, \ldots, m$, calculated as follows:

$$f_i = \frac{\delta(\tilde{p}_i, \tilde{p}^-)}{\delta(\tilde{p}_i, \tilde{p}^-) + \delta(\tilde{p}_i, \tilde{p}^*)}, f_i \in [0,1] \tag{15}$$

Ultimately, the top k items with highest ranking scores are recommended to the target user and the whole recommendation precess is completed.

IV. Experiments

A. Data sets

To evaluation our approach in different scenarios, we select two datasets of different e-Commerce environments:

1) Yelp dataset: Generally, for real recommender systems, the content dimension is usually "fixed" while the item dimension is "incremental". Take the Yelp.com (footnote) for instance, the taxonomies structure (for example, types of restaurant, types of cuisine, etc.) is not changed frequently and can be used to classify new-entered businesses. However, the population of users and businesses are increasing everyday. The Yelp dataset [12] is used for evaluating our approach, in which, there are over 45k users and 11k items, categorized to 570 descriptors. The rating sparsity is 99.96%, and averagely a user only rated 5 items. In contrast, the content information is rich, as for each descriptor, there are about 74.4 relevant items.

2) MovieLens dataset: The MovieLens datasets with rich rating data are the ideal test pool for CF approaches [13]. After cleaning, there are 2112 users and 4856 items associated with 551 descriptors. The rating sparsity is comparatively low (97.42%) such that averagely one user has rated about 166 items. Each content descriptor is associated with about 60.6 items in average.

B. Compared approaches

The proposed fuzzy content matching recommendation approach (shorted as FCM) is compared with following baseline models.

1) Standard CF: We compare our approach with standard CF to see if the cold-start problem of CF is alleviated by importing item content information. The prediction formula of [14] is used.

2) Semantic content analysis: Most of existing content-based recommendation approaches reveal the relationships between content features by analyzing their "semantic" similarities. As one of the start-of-the-art approaches, the tree matching model of [7] is compared, shorted as "TreeSim".

Besides, we also propose a variant of FCM, by replacing the descriptor dependence of Eq. (6) with semantic similarity to extend user preferences. We mark this variant as "Semantic" for short.

3) Crisp content filtering: A "crisp" version of our approach is tested, *i.e.*, ratings and user preferences are replaced with crisp values, and CF-like approach is applied using the preferences scores rather than ratings scores. This variant is titled as "Crisp" for short.

We can summarize CF, TreeSim and Crisp as the neighborhood-based approaches, and FCM and Semantic as the type of user-item marching approaches

C. Evaluation metrics

1) Coverage: As we know, given a pair of user and item, due to the cold-start problem, CF approach may fail to generate a prediction. The metric Coverage is hence used to evaluate the successful rate of a recommender system:

$$\text{Coverage} = \frac{\#\text{ successful predictions}}{\#\text{ test records}} \tag{16}$$

2) Ranking accuracy: The final recommendation list is generated according to a ranking order of the candidate items. The consistence of the ranking order is then needed to be compared with the actual test data. The metric nDCG (Normalized Discounted Cumulative Gain) is selected to evaluate the ranking accuracy. First, the DCG (Discounted Cumulative Gain) metric at a particular rank position p is defined as:

$$\text{DCG}_p = rel_1 + \sum_{i=1}^{p} \frac{rel_i}{log_2(i)} \tag{17}$$

where rel_i denotes the score at position i.

The nDCG is then calculated as:

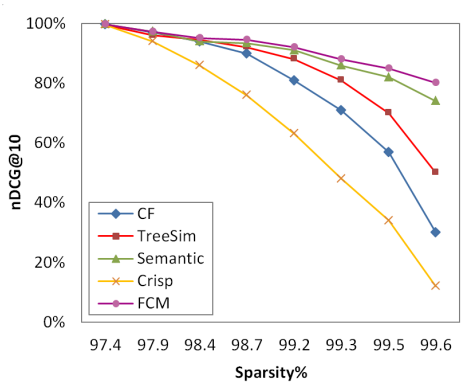
$$\text{nDCG}_p = \frac{\text{DCG}_p}{\text{IDCG}_p} \tag{18}$$

where IDCG_p is the DCG of "ideal ranking order", i.e., the actual rankings of test set.

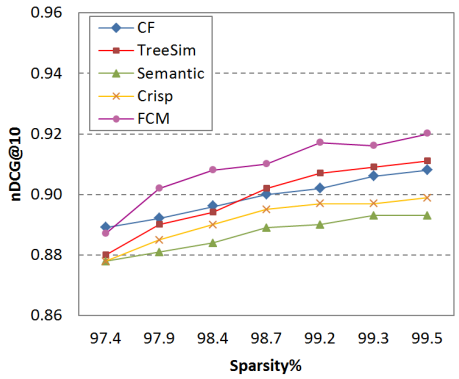
D. Result

1) Yelp dataset: The experiment results on Yelp dataset are collected in Table I. As we know that the ratings of Yelp dataset is very sparse (99.96%), such that the standard CF only completes 50.5% predictions. Comparing the metric of coverage, the sparsity problem also ruin the results of content-based approaches (TreeSim, Semantic, Crisp), especially for Crisp, only 8.3% is predictable. Generally, TreeSim, Semantic have higher coverage than CF, maybe because neighbor users are more easily found resorting to their preference on content information (with lower dimension) than resorting to their ratings on items (with higher dimension). The significant improvement between Crisp (0.083) and FCM (0.888) indicates the advantage of using fuzzy techniques to represent user preference. Comparing to these neighborhood-based approaches, FCM performs the best result in terms of coverage: about 88.8% test data has been successful predicted. According to these comparisons, FCM can significantly alleviate the sparsity problem by directly matching user preferences and item content information in the sparse environment.

In terms of ranking accuracy, represented by nDCG with $p = 10$ as default, CF (0.935), TreeSim (0.94) and Semantic (0.936) perform closely. The best performance is still achieved by FCM with 0.987. It indicates that though FCM does not generate crisp ratings it can rank the items accurately by using the fuzzy Topsis ranking method.



(a) Coverage



(b) Coverage

Fig. 5: Result comparison on MovieLens dataset

TABLE I: Result comparison on Yelp dataset.

	(a) CF	(b) TreeSim	(c) Semantic	(d) Crisp	(e) FCM
Coverage	0.505	0.530	0.633	0.083	0.888
NDCG(p=10)	0.935	0.940	0.936	-	0.987

2) MovieLens dataset: As the ratings of MovieLens is very dense, we can dilute the rating data to test the performance of each approach under different levels of sparsity. It should be mentioned that even after nine times of dilution, the sparsity of Movielens (99.57%) is still lower than the sparsity of Yelp dataset (99.96%).

First, Fig.5a demonstrates the trend of coverage of each approach with increasingly sparsity level. In overall, recommendation coverage is decreasing with the increase of rating sparsity. At beginning with relatively dense data (for example, sparsity under 98%), all recommendation approaches performs well and closely (almost 100%), but when the data becomes sparser, the gaps between them become more significant. It can be found that FCM and Semantic still maintains high coverage, but CF, TreeSim and Crisp lost their coverage sharply. Particularly, for Crisp approach, the coverage decreases sharply

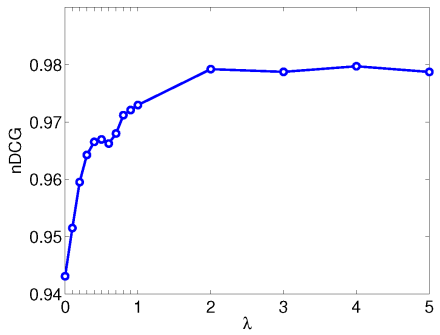
that only 12% at final. TreeSim and CF also reduce quickly when the data become sparse. In contrast, FCM and Semantic do not suffer the sparsity problem significantly. Even at the last round, they can predict over 80%. The proposed approach FCM maitains highest coverage all the time.

The results of nDCG reflecting the accuracy of ranking order are plotted in Fig. 5b. It evidently shows that FCM performs better than compared approaches in terms of ranking accuracy. Noticing that only the test tasks that are successfully predictable by all approaches are used to compare the nDCG metric. In particular, for the sparsest test set (sparsity is 99.6%), only 12% is comparable (determined by the worst approach: Crisp) so that this test set is ignored for comparing nDCG as there are insufficient test data.. As the result shows, we can find that the neighborhood-based approaches, CF, TreeSim and Crisp performs worse than FCM, but better than Semantic, a variant of FCM that uses semantic similarities to extend user preferences.

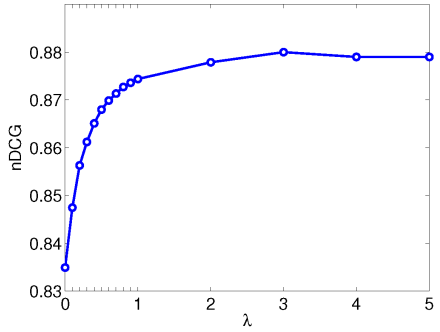
Summing up the comparisons on the two datasets, the advantages of proposed fuzzy content-matching approach are demonstrated well. The proposed recommendation approach archives a better performance in terms of both recommendation coverage and accuracy compared to standard CF and the latest tree matching-based approach [7]. Two variants of proposed approach, are also evaluated as comparison and the results show the importance of utilizing fuzzy techniques in our approach.

3) Sensitive of parameter λ: In Eq. (12), parameter λ is set to adjust the weight of item inherent quality versus user personality. To elaborate how it impacts the performance, we test the nDCG with different values of λ, on both Yelp dataset and MovieLens dataset. The results are reported in Fig. 6.

From Fig.6, the best performance is reached at $\lambda = 2$ and $\lambda = 3$ for Yelp and MovieLens, respectively. It illustrates that MovieLens users are of more personalization than Yelp dataset. In other words, consumers' flavors of electronic products like movies are highly personalized. In contrast, when people choose real businesses such as restaurants and hotels, they are more easy to follow the choices of other ones, *i.e.*, trust the average word-of-mouth of the businesses.



(a) Yelp



(b) MovieLens

Fig. 6: Parameter setting of λ

V. Conclusion and Future Study

This paper proposes a method of modeling user preferences to complex structured content information and develop a fuzzy content matching recommendation approach for e-Commerce environments. At first, content information is collected as a complete content tree and be transformed to a flat form using a set of descriptors. To handle the uncertainties, ratings and user preferences are represented using triangle fuzzy numbers, and user preferences on descriptors can be derived from their deviations of ratings on items. By comparing user initial preferences we can determine which content descriptors often share similar preferences by people and a kind of cross-dependence relationships among descriptors are established. The dependence relationships between content features can be used as a new clue to extend user preferences to more unrelated features. Therefore, given a new item that usually comes with rich content information, the proposed approach can match the item's content information with the target user fuzzy preferences. A fuzzy Topsis method is also proposed to compare the matching results in the form of triangular fuzzy numbers, so that candidate items can be ranked and the best matched items are recommended to the user. The experiments conducted with two different datasets indicate the good empirical performance of proposed approach in terms of both recommendation coverage and recommendation accuracy.

In the era of Web 2.0, despite item taxonomies that are created by system managers, there are also plenty of user-created content information such as social tags and comments. These new emerging information can provide more content information of items. In the future, we shall integrate such more diverse of content information to establish more rich and accurate profile of users.

References

[1] Y. H. Cho and J. K. Kim, "Application of web usage mining and product taxonomy to collaborative recommendations in e-commerce," *Expert Systems with Applications*, vol. 26, no. 2, pp. 233–246, 2004.

[2] L.-p. Hung, "A personalized recommendation system based on product taxonomy for one-to-one marketing online," *Expert Systems with Applications*, vol. 29, no. 2, pp. 383–392, 2005-08.

[3] A. Albadvi and M. Shahbazi, "A hybrid recommendation technique based on product category attributes," *Expert Systems with Applications*, vol. 36, no. 9, pp. 11 480–11 488, 2009-11.

[4] M. Ruiz-Montiel and J. F. Aldana-Montes, "Semantically enhanced recommender systems," in *On the Move to Meaningful Internet Systems: OTM 2009 Workshops*, ser. Lecture Notes in Computer Science, R. Meersman, P. Herrero, and T. Dillon, Eds. Springer Berlin Heidelberg, 2009-01-01, no. 5872, pp. 604–609.

[5] A. Arwan, B. Priyambadha, R. Sarno, M. Sidiq, and H. Kristianto, "Ontology and semantic matching for diabetic food recommendations," in *2013 International Conference on Information Technology and Electrical Engineering (ICITEE)*, 2013-10, pp. 170–175.

[6] 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*, ser. Lecture Notes in Computer Science, A. Cuzzocrea, C. Kittl, D. E. Simos, E. Weippl, and L. Xu, Eds. Springer Berlin Heidelberg, 2013-01-01, no. 8127, pp. 387–399.

[7] 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. 22, no. 99, pp. 1–1, 2014.

[8] 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, no. 1, pp. 76–94, 2009.

[9] L. A. Zadeh, "Probability measures of fuzzy events," *Journal of Mathematical Analysis and Applications*, vol. 23, no. 2, pp. 421–427, 1968-08.

[10] ——, *The concept of a linguistic variable and its application to approximate reasoning*. Springer, 1974.

[11] J. H. Purba, J. Lu, G. Zhang, and D. Ruan, "An area defuzzification technique to assess nuclear event reliability data from failure possibilities," *International Journal of Computational Intelligence and Applications*, vol. 11, no. 4, 2012.

[12] J. Blomo, M. Ester, and M. Field, "Recsys challenge 2013," in *Proceedings of the 7th ACM conference on Recommender systems*. ACM, 2013, pp. 489–490.

[13] I. Cantador, P. Brusilovsky, and T. Kuflik, "Second workshop on information heterogeneity and fusion in recommender systems (HetRec2011)." in *RecSys*, 2011, pp. 387–388.

[14] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: an open architecture for collaborative filtering of netnews," in *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, 1994, pp. 175–186.