Learning User Preference Models under Uncertainty for Personalized Recommendation

Azene Zenebe, Lina Zhou, Anthony F. Norcio

Abstract-- **Preference modeling has a crucial role in customer relationship management systems. Traditional approaches to preference modeling are based on decision and utility theory by explicitly querying users about the behavior of value function, or utility of every outcome with regard to each decision criterion. They are error-prone and labor intensive. To address these limitations, computer based implicit elicitation approaches have been proposed. However, the extant approaches to implicit elicitation in preference modeling have failed to: (i) integrate user feedbacks and item attributes; (ii) take into account of the subjective, incomplete, imprecise, and vague nature of features of an item, and features of a user preference; (iii) quantify how much a user likes, dislikes, or be indifferent to a given item; and (iv) provide a complete preference model. We propose a novel knowledge representation method for item and user preference that accounts for uncertainty due to the subjectivity, vagueness and imprecision using concepts from the fuzzy set and logic theory. A comprehensive preference model that accounts for positive, negative, neutral and in-deterministic categories of user preferences is defined. Furthermore, algorithms are developed for learning user preferences, and for prediction and recommendation. An evaluation with a benchmark dataset on movies shows that the accuracy in predicting user preference is found to be nearly twice that of random prediction. Additionally, the proposed approach outperformed the state-of-the-art approaches in terms of precision, recall, and F1-measure. The findings of this study have significant implications for preference modeling, recommender systems and customer relationships management systems.**

Index Terms-- **user preference, learning algorithm, uncertainty representation, recommender system**

I. INTRODUCTION

Internet technology enables personalization and customization in various sectors such as business, education, and services. Personalized services including recommender systems are an emerging strategy for developing customer-centered services, which have significant roles in customer relationship management (CRM). Recommender systems are systems that use information about a user's preferences, demographic, and cognitive ability and attempt to predict items such as movies, music, books, news, web pages, and restaurants that the user may be interested in.

Preference modeling can be viewed as an activity of preference elicitation from users about items. In human-to-human interaction we learn others’ preferences to various items by asking them (explicit elicitation) or observing their actions such as purchases and ratings to infer about what they like/dislike (implicit elicitation). Traditional approaches to preference modeling are based on decision and utility theory by explicitly querying users about the behavior of value function, or utility of every outcome with regard to each decision criterion. They are error-prone and labor intensive[1],[2]. Compared with explicit elicitation, implicit elicitation requires less involvement of users and tends to be more efficient. Moreover, preference modeling is studied in various fields such as information filtering, information retrieval, recommender systems, user modeling and personalized agents. This paper focuses on implicit preference modeling for recommender systems.

Adomavicius and Tuzhilin [3] suggest various areas of improvements for current recommender systems, among which two of them are: user feature and item feature representation methods, and recommendation modeling methods. The features of users and items, which are commonly used in preference modeling, raise a number of challenging issues for building recommender systems. Particularly, descriptions about items are subjective, vague and imprecise; and user preferences are vague and imprecise and may change with the context and time. These in turn induce uncertainty during representing and reasoning about item features, user features and user preferences.

The uncertainty considered here refers to epistemic uncertainty, also called non-stochastic uncertainty, induced from imperfection of information including imprecision, vagueness, non-specificity and conflict [4]. It is also worth mentioning that stochastic uncertainty that occurs due to the inherent variation associated with the system or the environment that can be modeled using probability theory [4] is not considered in this research. Moreover, in current recommender systems research, there is a lack of insight from combining user behavior and item features. Either item features or user features are employed separately in previous work on preference modeling. As a result of these problems and others described in [3], the performance of the state-of-the-art approaches to preference modeling in recommender systems is far from satisfactory and requires further improvements.

We propose an approach to preference modeling by accounting for the induced uncertainty in item features, and user features and preferences. Moreover, it integrates item features and user feedback in discovering knowledge about user preferences. The proposed approach is the first that models uncertainty based on a novel knowledge representation framework. The uncertainty is of non-stochastic type that is induced from subjectivity, vagueness and imprecision and can be successfully modeled using fuzzy set and logic. In relation to items, the uncertainty concerns to what extent (e.g. example low, medium or high) the items have some features. For instance, given a movie, to what extent the movie has drama content or is highly drama? In relation to preference, the uncertainty is associated to what extent a user likes, dislikes, or be indifferent to an item or a feature of an item.

The representation framework allows us to perform automatic discovery of user preferences and make items recommendations from data. Based on the representation framework, we have developed new algorithms for modeling items and user preferences, and predicting user preferences on new items for content-based recommendation. The algorithms are implemented and evaluated with a benchmark dataset. The experimental results indicate that the accuracy in the prediction of user preference is nearly twice of that of random prediction. In addition, the precision, recall, and F1-measure of movie recommendation are 62%, 57%, and 60% respectively, which are higher than the performance of the state-of-the-art approaches.

This study opens a new venue for discovering the knowledge of user preferences under uncertainty from a combination of user behavior and item features. The proposed algorithms for preference modeling not only model and reason on uncertainties due to subjectivity, vagueness and imprecision but also offer computational efficiency in learning preference models. The superior performance of the proposed approach suggests that it is valuable to model and reason on uncertainty as well as integrate user behavior and item features in preference modeling for recommender systems.

The proposed approach and algorithms, and findings of this research can be used to enhance CRM systems. They strengthen strategic marketing for attracting new customers and retaining existing customers in various ways. First, the proposed approach and new algorithms provide in-depth insights into customer preferences. Second, the discovered customer preferences can be used to drive the supply/production of items. Third, the preference models of customers can be used to improve the quality of recommendations, which further help foster user's trust in CRM systems. Recommender systems have been adopted as an effective marketing strategy and antecedent of trust for B2C e-commerce [5],[6]. Fourth, in terms of marketing research and practice, the study shows how to effectively integrate item attributes and user feedback, which are subjective, incomplete, imprecise, and vague, in gaining insights into customer preferences. Fifth, the approach to user preference modeling can be used for dimensionality reduction to address the scalability problem in collaborative filtering [7].

Throughout the paper, we use the term item as a synonym to product, the term user as a synonym to customer and client, and the term feature as a synonym to attribute, characteristic, and variable. The remainder of the paper is organized as follows. Section 2 provides a review of related literature. Section 3 presents a methodology for item representation, preference elicitation, and preference representation. Section 4 introduces new algorithms for discovering preferences, and for the predication and recommendation of items. Section 5 describes dataset, experimental settings, and evaluation metrics, as well as illustrates how the recommendation algorithms work with an example. Section 6 reports the results and discusses the findings and potential application of this research. Finally, Section 7 describes conclusion and future work.

II. Related Literature

Based on decision and utility theory, two methods for preference elicitation have been proposed: utility function and analytical hierarchy process [1]. These traditional methods query users about the behavior of value function, or utility of every outcome with respect to each decision criterion. They are time-consuming, error-prone, and labor intensive. To address the limitations of explicit elicitation methods, various computer based implicit elicitation methods such as content-based, collaborative filtering and hybrid approaches are developed [8],[3]. The implicit methods learn user preferences on items from user's past behavior such as visiting a web site, purchasing an item, and rating a product or service; and the preference model is used in recommender systems. A recent survey of state-of-the-art recommender systems is found in [3].

Content-based recommender systems [9] model user preferences based on the features presented in rated items using a learning-based approach. Instance similarity-based and clustering are two types of content-based approaches. The similarity-based models user preferences with a feature vector obtained from examples or constraints initialized by users (e.g., [8]). The clustering forms clusters of user preferences based on a sufficient number of features indicating user preferences. The clusters to which the user likely belongs become the user's initial preferences (e.g., [10]). However, the content-based requires a user to provide not only input values to a number of attributes but also weights of each of these attributes (e.g., [11]). Such a practice not only creates cognitive burdens on users but also may result in inaccurate preferences.

In a collaborative filtering (CF) approach, users express their ratings on items, which are in turn used to approximate user preferences. CF based recommender systems match the ratings of a user to those of other users' to find the "most similar" users, and recommend items that similar users liked but the user has not rated. Typical examples of CF system are MovieLens [12],[13], personalized recommender systems of Amazon and CDNow. The CF approach is faced with scalability problem due to a large-sized rating matrix, first rater problem, and sparsity problem [7]. In addition, the CF approach ignores the features of items such as genre and directors of movies. Furthermore, both conventional content-based and CF methods consider and represent user preferences and item features as deterministic – presence (1) or absent (0), which does not conform to the uncertainty nature of user preferences and item attributes.

To address the limitations of extant approaches to preference modeling, we proposed a fuzzy theoretic approach. Unlike CF, our approach is based on individual user behavior and knowledge about items for preference elicitation. In view of the non-stochastic nature of features of items and user preferences over items, we account for uncertainties due to subjectivity, imprecision and vagueness in preference modeling using fuzzy set and logic. Compared with related literature (e.g., [14],[15]), our model not only focuses on non-stochastic nature of item features and user preferences over items, but also contains neutral and unknown categories in addition to positive and negative categories in space of possible categories of preferences. The novel uncertainty representation approach for an item and user preferences enables the automatic discovery of knowledge about user preferences with the minimum amount of involvement from a user as well as predicting user preferences for new items. Moreover, the proposed approach improves the performance of recommender systems over those of state-of-the-art methods.

III. A Fuzzy Theoretic Approach to User Preference Modeling and Recommendation

Preference indicates a liking of one thing more than another thing. A user's preference to an item falls into at least one of the following categories: Preferred, Not Preferred, Indifferent, and Unknown. Moreover, user's preference for an item can belong to two or more categories with varying degrees of memberships. For instance, for a movie containing drama and action content, a user may like it to some degree due to its drama content but dislike it to some degree due to its action content. Since both preference and features of an item are vague, subjective and measured with imprecision, we propose to use fuzzy set theory as the representation method and fuzzy logic as the reasoning approach in user preferences modeling.

A. Fuzzy Set and Fuzzy Logic in Preference Modeling

Fuzzy set theory consists of mathematical approaches that are flexible and well suited to handle incomplete information, the un-sharpness of classes of objects or situations, or the gradualness of preference profile. Further, fuzzy set theory and logic provide a way to quantify the uncertainty due to vagueness and imprecision [16], [17]. Therefore, the use of fuzzy sets provides opportunities for modeling items and user preferences for recommender systems under uncertainty induced by vagueness and imprecision features of item and user feedbacks. Membership functions, a building block of fuzzy sets, have possiblistic interpretation, which assumes the presence of a property and compares its strength in relative to other members of the set [18].

A fuzzy set A in X is characterized by its membership, which is defined as [19]: $\mu_A(x): x \in X \to [0,1]$, where X is a domain space or universe of discourse. Alternatively, A can be characterized by a set of pairs: $\mathbf{A} = \{(\mathbf{x}, \mu_{\mathbf{A}}(\mathbf{x})), \mathbf{x} \in \mathbf{X}\}$. According to the context in which X is used and the concept to be represented, the fuzzy membership function, $\mu_A(x)$, can have different interpretations [20]. As degree of similarity, it represents the proximity between different pieces of information. For example, a user's movie interest to the fuzzy set of "drama movies lover" can be estimated by the degree of similarity. As degree of preference, the membership degree represents the intensity of preference in favor of x, or the possibility of selecting x as a value of X. For instance, a movie rating of 4 stars out of 5 indicates the degree of a user's satisfaction or liking with x based on certain criteria, say movie attributes such as content-intensity of action, drama, and humor.

The most commonly used membership functions are triangular, trapezoid, Gaussian function, S-function and exponential-like functions. The selection of membership function can only be determined in the application context [21]. Fuzzy set operators are the result of substitution or extension of crisp set operators. The triangular norm (t-norm) and a triangular co-norm (t-conorm or s-norm) are the general classes of intersection and union operators [21].

Compared with traditional statistical or probabilistic methods, using fuzzy set theory and related mathematics brings the following benefits [22]: (i) the membership function in fuzzy set theory is deliberately designed to treat the vagueness and imprecision in the context of the application. Therefore, it is more reliable and accurate to use fuzzy set theory to assess subjectivity and vagueness in attributes; (ii) the membership function can be continuous and overlapping, which are more accurate in representing the attributes of items; and (iii) the fuzzy mathematical method is easier to perform once the membership functions of attributes are defined.

B. Items Representation Using Fuzzy Set

User preferences for items can be associated to different attributes of the items. For an item described with multiple attributes, more than one attribute can be used for user preference modeling. Moreover, some attributes can be multi-valued involving overlapping or non-mutually exclusive possible values. For example, movies are multi-genres [23]. As a result, this induces uncertainty in the determination of the genres of movies. Fuzzy set allows us to represent uncertainty induced due to the vagueness in terms of attributes associated with items. In other words, the value of attributes in an item can be represented more accurately with fuzzy set framework than with crisp set framework.

Let an item I_j ($j = 1 \ldots M$) is defined in the space of an attribute $X = \{x_1, x_2, x_3, \ldots. X_N\}$, then I_j can take multiple values x_1, x_2, …, and/or x_N. These values of X can be sorted in the decreasing order of their presence in item I_j expressed by their degrees of memberships. The membership function of item I_j to value x_k ($k = 1 \ldots N$) is denoted by $\mu_{x_k}(I_j)$, which can be obtained either heuristically from analysis of domain or empirically from the data. Hence, a vector $X_j=\{(x_k, \mu_{x_k}(I_j)), k= 1\ldots N\}$ is formed for I_j. $\mu_{x_k}(I_j)$ can be interpreted as the degree of similarity of I_j to a hypothetical (or prototype) pure x_k type of item; or as the degree of presence of value x_k in item I_j.

The next step is to determine the form of the membership function. Based on the heuristics that the possibility for item I_j to take different values of X varies, the membership function should meet the following three criteria: 1) assigning higher degree of membership to major values than minor values; 2) assigning 0 to values that are not associated with the item; and 3) degrees of membership should be normalized to the range of [0, 1]. Thus, we propose a Gaussian-like or exponential-like function to compute the fuzzy set membership, as shown in (1).

$$\mu_{x_k}(I_j) = r_k / 2^{\sqrt{\alpha * |L_j|(r_k - 1)}} \qquad (1)$$

where N=$|L_j|$ is the number of values of X associated with I_j and r_k ($1 \le r_k \le |L_j|$) is the rank position of value x_k, and $\alpha > 1$ is a constant threshold to control the difference between consecutive values of X in I_j. For example, with α set to 1.2, movie 'Muppet Treasure Island (1996)' is represented in terms of genres as follows. The representation for any movies is shown in Figure 1.

$|L_j|$ =6 and x_j = {(Family, 1), (Action, 0.31), (Adventure, 0.22), (Comedy, 0.16), (Musical, 0.12), (Thriller, 0.09)}.

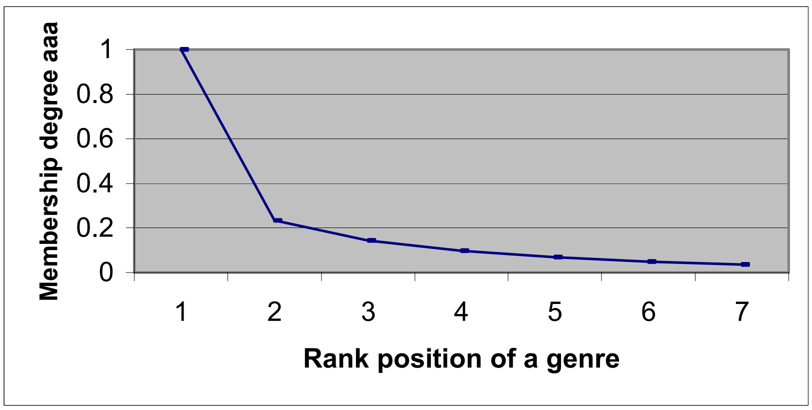


Figure 1: Possibility distribution of seven genres in a movie

It is noted from (1) that the same values of X at same rank positions between different items can have varying degrees of membership values if the numbers of values of X associated with the items are different.

The representation scheme for items can be generalized to any item with multi-valued attributes that can be ordered. Consequently, the representation can be easily extended to recommender systems based on a combination of multiple attributes. For example, we can use movie genre describing the content of movies as the first attribute and actresses/actors as the second attribute to model users' preference for favorite genres and actors, and then to movies. The actors in a movie can be represented in a vector $A=\{a_1, a_2, \ldots a_k\}$ for K actors. The degree of role or importance of an actor a_k in a movie m_i can be represented by degree of membership associated with the fuzzy set degree of role or importance. That is, $A_j = \{(a_k, \mu_{a_k}(m_j))$, for k=1 to K}, where $\mu_{a_k}(m_j)$ can be defined in the same way as that for genres. Preference modeling by integrating a single attribute of an item and user feedback on the item is the unique feature and focus of this paper.

C. Preferences Elicitation and Representation

The ideal scenario for eliciting users' preferences is to ask users to express their preference of the various features of items. In practice this not only has limited use but also is not always practical. Alternatively, user feedbacks become a promising source for inferring user's preference to the item. The next question is: how can we infer user preferences from the user's feedback and attributes of an item? A related question is: how can we represent the inferred user preferences?

Assume X is a feature vector consisting of *n* discrete values; I_j is an item with an assignment or instantiation of X; B is a set of user feedbacks on items similar to I_j. The preference of user U_i to item I_j given B is defined as:

$$p((U_i, I_j)|B) = f(X/I_j, \bar{\mathbf{p}}(X/B)) \quad (2)$$

where f is an inference function, and $\bar{\mathbf{p}}(X/B)$ denotes preference to X inferred from B.

Based on a user's feedback such as ratings, items are categorized into three groups: disliked items (NI), liked items (PI), and indifferent items (II). If R={PI, NI, II} denotes a set of rated items as a user's feedback (B) and P_X={PX, NX, IX, UX} represents a set of user preferences for different values of attribute X where PX is preferred attribute values, NX is not preferred attribute values, IX is indifferent attribute values, and UX is unknown attribute values, then component $\bar{\mathbf{p}}$(X/B) in (2) can be inferred with (3).

$$\bar{\mathbf{p}}(X/B) = \bar{\mathbf{p}}(P_X/R) = \bar{\mathbf{p}}((PX,NX,IX, UX)/R) = f(X/PI, X/NI, X/II) \qquad (3)$$

where f is an inference function, which is described in detail in Section IV.

IV. Learning Algorithms

The proposed representation framework presents an opportunity for automatic discovery of user preferences on items. Based on the proposed representation framework, as discussed above, we have developed algorithms for preference modeling and preference prediction in this section.

A. User Preferences Modeling

Generally, domain analysis on an item I_j of interest can help verify the soundness of preference modeling based on feature X of I. As shown in (2), user preferences to X can be inferred from user's feedback on similar items and their features. The remaining issue is how to determine degrees of memberships of each value of X to different types of preferences (i.e., PX, NX, IX, and UX). The proposed algorithm is motivated by the following ideas:

1. Dominant values of X in items that have received positive feedback from a user can be considered as members of preferred attribute values (PX).
2. Dominant values of X in those items with negative feedback from a user can be considered as members of not preferred attribute values (NX).
3. Dominant values of X in those items with neutral feedback from a user can be considered as members of indifferent attribute values (IX).

4. Dominant X's values that do not exist in those items for which a user has provided feedbacks can be considered as members of unknown attribute values (UX). That is, $UX=X-(PX \cup NX \cup IX)$.

It needs to be noted that user's preference to a value of X can belong to one or more of the preference types (PX, NX, IX and UX) with varying degree of memberships. Figure 2 presents pseudo code of the proposed algorithm. The inputs to the algorithm are user feedbacks such as ratings and item values of attribute X. The outputs are vectors PX, NX, IX and UX comprising of user's preferences for values of X. Each of the output is represented as a vector consisting of elements of (value of X, membership degree). An illustration of how the algorithm works for movies recommendation is presented in Section V.

B. Preferences Prediction and Items Recommendation

The inferred user preferences can be used to predicate whether an item would be liked or preferred, disliked or not preferred, indifferent, or unknown due to lack of information to categorize as one of the former three preference classes. The preference predication function for an item, $\widehat{p}\,(I_j / P_X)$, is defined as:

$$\widehat{p}\,(I_j / P_X) = f(I_j, P_X) \qquad (3)$$

where $P_X=\{PX, NX, IX, UX\}$ is the user preferences that are inferred using the algorithm in Figure 2. f is a prediction function that can be implemented using various techniques such as similarity-based nearest-neighbor, weighted sum, and regression [12].

Let R=rated items from a user as user feedbacks (B)
Let TPX, TNX, TIX are temporary arrays to store degree of memberships to attribute X { $x_{k,}$ k= 1… L}
Let P_X={PX, NX, IX and UX} is a vector consist of array to store the final average degree of memberships to values of attribute X
Let X_j={(x_k, $\mu_{x_k}(I_j)$), k= 1… N} vector of attribute X's values for an item I_j

For each user DO {
Load user feedbacks, e.g. ratings on items, say Matrix R={item_id, I, rating}
Split R into three sets or segments: liked items (PI), disliked items (NI), and indifferent items (II)
For $I_j \in$ PI (j=1…|PI|), add x_j to TPX
For $I_j \in$ NI (j=1…|NI|), add x_j to TNX
For $I_j \in$ II (j=1…|II|) add x_j to TIX
For each $x_k \in$ TPX and $I_j \in$LI DO

$$\mathbf{x_k^{PI}} = \frac{1}{|\mathbf{PI}|}\sum_{\mathbf{j}=1}^{|\mathbf{PI}|}\mu_{\mathbf{x_k}}(\mathbf{I_j})$$

For each $x_k \in$ TNX and $I_j \in$NI DO

$$\mathbf{x_k^{NI}} = \frac{1}{|\mathbf{NI}|}\sum_{\mathbf{j}=1}^{|\mathbf{NI}|}\mu_{\mathbf{x_k}}(\mathbf{I_j})$$

For each $x_k \in$ TIX and $I_j \in$II DO

$$\mathbf{x_k^{II}} = \frac{1}{|\mathbf{II}|}\sum_{\mathbf{j}=1}^{|\mathbf{II}|}\mu_{\mathbf{x_k}}(\mathbf{I_j})$$

For each x_k in X DO {
$\mathbf{x_k^{max}}$ =maximum{ $\mathbf{x_k^{PI}}$, $\mathbf{x_k^{NI}}$, $\mathbf{x_k^{II}}$ }
if $\mathbf{x_k^{max}} > 0$ then {
If $\mathbf{x_k^{max}} = \mathbf{x_k^{PI}}$, insert ($x_k$, $\mathbf{x_k^{PI}}$) into PX
else if $\mathbf{x_k^{max}} = \mathbf{x_k^{NI}}$, insert ($x_k$, $\mathbf{x_k^{NI}}$) into NX
else if $\mathbf{x_k^{max}} = \mathbf{x_k^{II}}$, insert ($x_k$, $\mathbf{x_k^{II}}$) into IX
}
else if $\mathbf{x_k^{max}} = 0$ then insert ($x_{k,}$ 1) into UX
}next x_k
Sort PX in the descending order of $\mathbf{x_k^{LI}}$
Sort NX in the descending of $\mathbf{x_k^{NI}}$
Sort IX in the descending order of $\mathbf{x_k^{II}}$

}next user

Figure 2: An algorithm for learning user preference model

An algorithm is developed to predict a user's preference to items based on values of attribute X of the items and inferred user preferences, as shown in Figure 3. In particular, two approaches are employed in this algorithm. The first approach was based on Yager's suggestion [24] that considers a single value of an attribute. The rule is: If a user likes a value $x \in X$ with degree of membership μ_x and if a given un-experienced item I_j has the value x with a degree of membership $\mu_x(I_j)$, the confidence score of recommending I_j to the user is a function of μ_x and $\mu_x(I_j)$. For example, if genre drama is the most preferred category of movies with mean degree of membership of 0.40 by a user, and a movie M is not rated by the user and has a membership degree of 0.683 to genre drama, then the confidence score for recommending this movie to the user would be either 0.40 using minimum operator or 0.27 using the product operator.

The first approach only considers a single attribute value as well as user preferences for that value. To address these limitations, we propose a new approach to predicting user preferences. The proposed approach takes a comprehensive view of items and values of attributes. It predicts preferences in the targeted item based on the similarity between the inferred preferences of a user to all values of the item's attribute and the degree of the attribute presence in an item for which prediction is to be made.

The algorithm is preference model-based and shown in Figure 3. The inputs of the algorithm are inferred preferences of a user to values of an attribute of the item - PX, NX, IX, and UX (from Figure 2); and the target item (potential item for recommendation) represented using the membership function in the attribute space using (1). The output of the algorithm is the predicated preference class for the targeted item: Liked (PI), Disliked (NI), Indifferent (II), or Unknown (UI) along with their corresponding degrees of membership – indicating how much a user likes, dislikes, or be indifferent to the item.

The preferences of a user on item attribute values, and attributes of targeted items are represented using fuzzy set with possiblistic interpretation, which allows for applying various fuzzy theoretic similarity measures, including extension of Jaccard [25], cosine-based and correlation-based measures [26]. The cosine-based similarity measure appropriately judges the difference in shape or quality between two n-dimensional vectors from a common origin [26]. Moreover, simulation studies [26],[27] show that the cosine similarity measure within fuzzy theory framework is found to be effective. Also, the measure is most widely used and an accurate similarity measure in recommender system research in general [28],[29].

Using fuzzy theoretic cosine measure, a user's preference to an item can be estimated by computing the similarity between two vectors: preferences vector of a user to values of attribute X (i.e., P_X= PX, NX, IX and UX), and vector values of the attribute of the targeted item in space of X computed using (1). The similarity indices of the targeted item to the four different preferences vector PX, NX, IX and UX are computed separately, as shown in (4). These similarity indices can be considered as confidence scores or degree of support for an item to be in the four preference classes: PI, NI, II, or UI. Finally, the targeted item is classified into the class with the maximum similarity index among the four. The algorithm is presented in Figure 3. An illustration of how the algorithm works for movies recommendation is presented in Section V.

$$\mathrm{Cos}\,(P_X(U_i, X), X(I_j)) = \frac{\sum_k \mu_{x_k}(U_i) * \mu_{x_k}(I_j)}{\sqrt{\sum_k (\mu_{x_k}(U_i))^2}\sqrt{\sum_k (\mu_{x_k}(I_j))^2}} \qquad (4)$$

where $P_X(U_i, X)=\{(x_k, \mu_{x_k}(U_i|PX \text{ or } NX \text{ or } IX \text{ or } UX)), k=1\ldots N\}$ is the inferred preference of user U_i for values x_k of attribute X with respect to PX, NX, IX, or UX; and $X(I_j)=\{(x_k, \mu_{x_k}(I_j)), k=1\ldots N\}$ is the possibility distribution of the targeted item I_j in the space of X.

Let PI = predicated set of liked items for a user

Let NI = predicated set of disliked items for a user

Let II = predicated set of indifferent or neutral items for a user

Let UI = predicated set of unknown or undetermined items for a user

For each user u do {

//Compute the degree of compatibilities or similarities between the

//targeted item I_j is represented by $X(I_j) =\{(x_k, \mu_{x_k}(I_j)), k= 1 \ldots N\}$

//preference of a user u is represented by $P_X(u)=\{PX(u), NX(u), IX(u), UX(u)$

For each targeted item I_j for recommendation do {

$simP_j$ = similarity($X(I_j)$, PX(u))

$simN_j$ = similarity($X(I_j)$, NX(u))

$simI_j$ = similarity($X(I_j)$, IX(u))

$simU_j$ = similarity($X(I_j)$, UX(u))

max = maximum { $simP_j$, $simN_j$, $simI_j$, $simU_j$}

if max = $simP_j$ then predicted class is Liked/Preferred, and add (I_j, $simP_j$) into PI

else if max = $simN_j$ then predicted class is Disliked, and add (I_j, $simN_j$) into NI

else if max = $simI_j$ then predicted class is Indifferent, and add (I_j, $simI_j$) to II

else if max = $simU_i$ then predicted class is Unknown, and add (I_j, $simU_j$) to UI

} next item

//TOP-N Recommendation
//Input: PI={(I_j, $simP_j$) , for all j; and Output: Top N preferred items
Sort PI by $simP_j$ in descending order
Select Top N items
Recommend the Top N items
} next user

Figure 3: Algorithm for Prediction and recommendation

C. Computational Time Complexity of the Algorithms

We analyze time complexity of the proposed algorithms for constructing user preferences, and for making recommendations. For the preference modeling:

1. Computation of degree of memberships: For r rated items with respect to an attribute space of size *n*, it requires r * n operations.

2. Segmentation and computation of mean degrees of membership associated to four classes of user preferences: For r rated items in R with respect to an attribute space of size *n*, it requires r* (n + 4n) = r* 5n.

Consequently the complexity of the algorithm for preferences modeling is in the order of O(rn), where both r and n are small numbers. Therefore, the computational complexity of the proposed algorithm is greatly reduced compared with those of both user-user CF and item-item CF algorithms. The latter are in the orders of $O(l^2m)$ and $O(m^2l)$ respectively for *m* items rated by *l* different users [29], where both *l* and *m* are usually large numbers.

Recommendation of an item requires representation of the item using (1). This requires n operations. The recommender algorithm also computes similarity between the four user preferences classes and the item. This requires 4 * n operations. Therefore, the complexity of the algorithm is 5n, which is O(n). Consequently for recommending q items, the complexity would be O(qn). It is significantly lower than time complexities of user-user CF and item-item CF algorithms. The latter are in the orders of O(qsr), where r is the number of items for which the user provides feedback and s is the number of most similar items (or users) to each of the q items (or users) [29].

V. Experimental Evaluations

The performance of the proposed algorithms for preference modeling and recommendation is evaluated in this section.

A. Dataset

We select movie as the test domain because movies have multi-valued attributes with vague and subjective features. For example, movies are multi-genres, and multi-actors [23]. Moreover, the following features of movies support discovery of preferences based on movie genres and user past movies watching behavior [30]: (i) As one of the experiential products, mostly movies are selected for pleasure and expenditures of time. Thus, consumers choose movies based on what they like and enjoy; and (ii) Consumers are more likely to use subjective features such as "funny" and "romantic" to select movies than objective features such as the director, actresses/actors, location, and price.

The benchmark dataset from MovieLens at GroupLens research project of University of Minnesota (http://movielens.umn.edu) is employed in this study. The dataset includes movie attributes, user ratings, and simple user demographic information. The dataset was collected for a seven-month period from 1997 to April 22nd, 1998. It consists of 100,000 ratings (1-5) from 943 users on 1682 movies; each user has rated at least 20 movies. In the dataset, movies are described with: movie id, movie title, release date, video release date, IMDb URL, and 20 genres including action, adventure, animation, children's, comedy, crime, documentary, drama, fantasy, film-noir, horror, musical, mystery, romance, sci-fi, thriller, war, western, family, and unknown.

Genres in the MovieLens dataset are represented as binary values, which do not reflect the true content of movies in the genre space. Therefore, we use the proposed representation scheme that is described in Section III.C by incorporating information about movie genres from the Internet Movie Data Base (imdb.com[1]). The IMDB is a large database consisting of comprehensive information about past, present and future movies. Moreover, in (1) α is set to 1.2 after analyzing genres of movies and various experiments.

The dataset is pre-processed to segment items. Based on a user's ratings, movies are categorized into three groups: disliked (NI) with ratings of 1 and 2, liked (PI) with ratings of 4

1 "IMDb History," vol. 2004: Internet Movie Database Inc., n.d.http://www.imdb.com/

and 5, and indifferent (II) with rating of 3. Similarly, given a set of movie genres: animation, adventure, romance, thriller, action, comedy, drama, crime, documentary, fantasy, film-noir, horror, musical, mystery, science fiction, war, western, family, and others, we categorize the genres into three groups: preferred (PX), non-preferred (NX), and indifferent (IX) genres.

The procedures of the algorithms for movies recommendation are illustrated with a real example as follows. For user 5 and genres x_1='Drama' and x_3='Action', the algorithm groups the 134 movies rated by user 5 into the following categories: (i) NI and the mean degrees of membership of these movies to x_1 and x_3 are 0.203 and 0.167 respectively; (ii) PI and the mean degree of membership of these movies to x_1 and x_3 are 0.139 and 0.309 respectively; and (iii) II and the mean degree of membership of these movies to x_1 and x_3 are 0.242 and 0.358, respectively. For user 5, execution of the preference modeling algorithm (Figure 2) produces vectors consisting of mean degrees of membership of each genre to PX, NX, IX, and UX denoted as (genre(x_k), $\mu_{PX}(x_k), \mu_{NX}(x_k), \mu_{IX}(x_k), \mu_{UX}(x_k)$):

> (Drama, 0.139, 0.203, 0.242, 0); (Comedy, 0.389, 0.393, 0.237, 0); (Action, 0.309, 0.167, 0.358, 0); (Thriller, 0.067, 0.112, 0.174, 0); (Romance, 0.054, 0.093, 0.038, 0); (Adventure, 0.174, 0.113, 0.078, 0); (Animation, 0.066, 0.018, 0.072, 0);
>
> (Children's, 0, 0, 0, 1); (Crime, 0.102, 0.005, 0.055, 0); (Documentary, 0, 0, 0, 1); (Fantasy, 0.100, 0.070, 0.042, 0); (Film-nor, 0, 0, 0, 1); (Horror, 0.098, 0.198, 0.087, 0); (Musical, 0.036, 0.015, 0.057, 0); (Mystery, 0.016, 0.046, 0.010, 0); (Science Fiction, 0.231, 0.068, 0.099, 0); (War, 0, 0.023, 0, 0); (Western, 0.024, 0.032, 0, 0); and (Family, 0.077, 0.171, 0.221, 0).

Based on the proposed preference modeling algorithm (Figure 2) and maximum fuzzy logic operator, ordered lists of genre preferences of user 5 are inferred as:

- PX={(Science Fiction, 0.231), (Adventure, 0.174), (Crime, 0.102), (Fantasy, 0.100)},
- NX={(Comedy, 0.393), (Horror, 0.198), (Romance, 093), (Mystery, 0.046), (Western, 0.032), (War, 0.023)},
- IX={(Action, 0.358), (Drama, 0.242), (Family, 0.221), (Thriller, 0.174), (Animation, 0.072), (Musical, 0.057)}, and
- UX={(Children's, 1), (Documentary, 1), (Film-nor, 1), (Others, 1)}.

For movies 2 and 222 that were not rated by user 5, they are represented as: {Movie 2, (Action, 1.00), (Thriller, 0.35), (Adventure, 0.29), (Crime, 0.44)}; and {Movie 222, (Action, 0.44), (Thriller, 0.29), (Adventure, 1.00), and (Science Fiction, 0.35)}. Degrees of preferences of user 5 for these movies are computed using the proposed algorithm (Figure 3): (i) the similarities between movie 2 and PX, NX and IX are 0.82, 0.85, and 0.93, respectively; and (ii) the similarities between movie 222 and PX, NX and IX are 0.79, 0.85, and 0.64, respectively. In addition, the similarity between UX and the two movies are zero. Therefore, using the maximum operator the system predicts that user 5 would dislike movie 222, and be indifferent to movie 2.

B. Experimental Settings

The proposed algorithms are evaluated in two settings: 1) use the entire dataset as both training and testing data; and 2) randomly split the data set into 3:1 as training and testing cases. In the second setting, 10 runs were performed to reduce the sampling bias.

The distribution of user ratings over the entire dataset is positively skewed, and the minimum and maximum numbers of ratings are 20 and 737, respectively. As the result, median instead of mean of the average number of ratings is used to compare the results of the proposed approach with results of traditional approaches in Section VI.C. In the first setting the average ratings are 65 for both testing and training sizes. In the second setting, the average ratings are 16 and 48 for testing size and training size, respectively.

C. Evaluation Metrics

Accuracy is a commonly used metrics for a recommender system based on user tasks or goals [31]. The accuracy metrics includes predictive and recommendation accuracy measures. Predictive accuracy measures the percentage of correct predictions. Predictive accuracy metrics such as mean absolute error and mean square error are found to be less appropriate when the user task is to find 'good' items and when the granularity of true value is small because predicting a 4 as 5 or a 3 as 2 makes no difference to the user [32],[31]. Instead, recommendation accuracy metrics including recall, precision and F-measures are appropriate.

Precision measures the ratio of correct recommendations being made. Recall reflects the coverage or hit rate of recommendations. F1-measure is the harmonic mean of the precision and recall, which are inversely related to each other as the number of recommendations increases. We have developed an algorithm for constructing confusion matrix (see Table 1). The horizontal dimension indicates actual classes of that an item should belong to and the vertical dimension indicates the predicted class classes. P, N, I, and U denote Liked, Disliked, Indifferent, and Unknown classes, respectively. Each element in the matrix represents the co-occurrence frequencies between pairs of actual and predicted classes. Based on information in the confusion matrix, we compute the values of four accuracy metrics using (5)-(8).

TABLE 1: CONFUSION MATRIX OF THE PREDICATIONS

Actual Class / Predicated Class	N	I	P
N	NN	NI	NP
I	IN	II	IP
P	PN	PI	PP
U	UN	UI	UP

accuracy = (NN + II + PP) / (NN+NI+NP+IN+II+IP+PN+PI+PP+UN+UI+UP) (5)

precision = PP/(PN + PI + PP) (6)

recall = PP/(NP + IP + PP + UP) (7)

F1-measure = (2*Precision * Recall)/(Precision + Recall)) (8)

VI. RESULTS AND DISCUSSION

The algorithms are implemented with Java and evaluated via various simulation runs. All the ratings data from 943 users are used in the first test setting and the means of the 943 results are reported. Moreover, for each user, using ten different random 3:1 splits of the dataset, the performance of ten predications and recommendations are computed and the average is reported. Given four targeted classes (P, N, I, and U), the accuracy of a baseline random classification is 25%.

A. Preference Predication Accuracy

In the evaluation setting with all the data used as both training and testing cases, the mean of predication accuracy is 64%, and the minimum and maximum accuracies are 26% and 100% respectively. Figure 4 presents the percentile distribution of mean accuracies, which reveals that, in the first evaluation setting, the accuracies are higher than 72% for 25% of users, higher than 52% for 75% of users, and higher than 62% for 50% of users.

In the 3:1 split evaluation setting, the mean of prediction accuracy is 48.49%. Additionally, as shown in Figure 4, the accuracies are higher than 56% for 25% of the users, larger than 39% for 75% of users, and larger than 46% for 50% of the users. Overall, the accuracies in both evaluation settings are better than the baseline of 25%.

B. Recommendation Accuracies

The average precisions, recalls and F1-measures from the two types of test settings are reported in Table 2. The percentile distributions of precisions, recalls and F1-measures of 10 runs of the second test setting are shown in Figure 5.

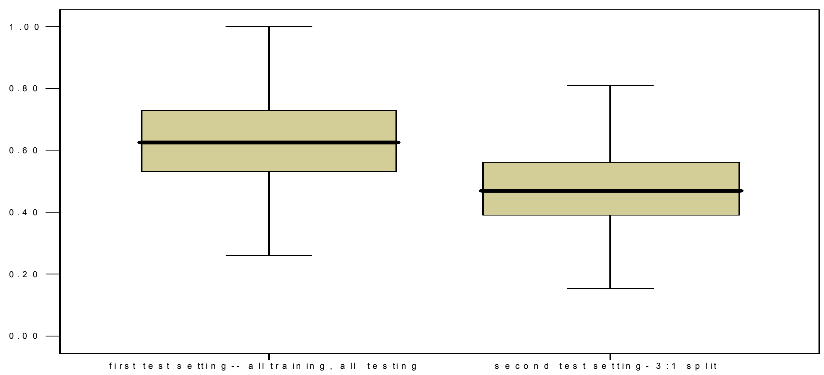
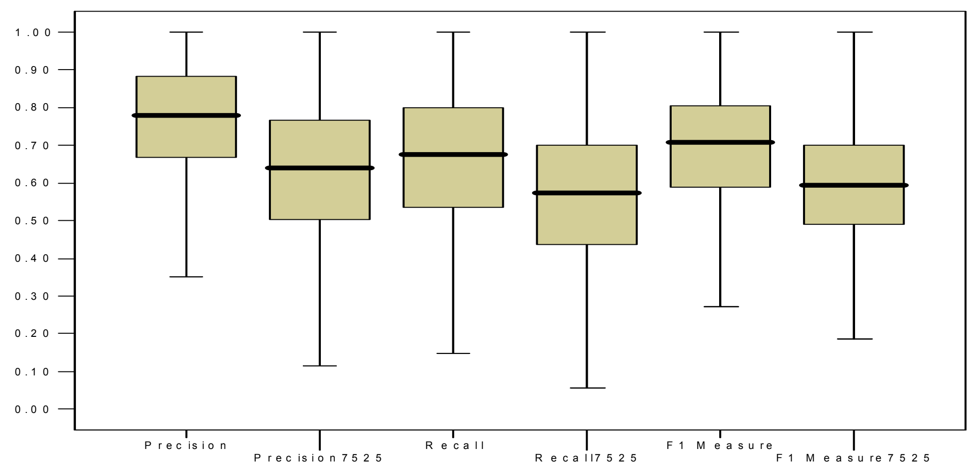


Figure 4: Box-plots for predication accuracy

In the first test setting, the mean of precisions is around 76%. Specifically, as shown in Figure 5, the precisions are equal or greater than 88% for 25% of the users, equal or greater than 67% for 75% of users, and equal or greater than 78% for 50% of the users. The mean of recalls is around 69%. As shown in Figure 5, the recalls are equal or greater than 80% for 25% of users, and equal or greater than 54% for 75% of users. The mean of F1-measures is around 71%. Specifically, the F1-measures are equal or greater than 81% for 25% of the users and equal or greater than 59% for 75% of users. In the second test setting, the mean of precisions is approximately 62%, and the precisions ranged from 50% to 77% for 75% of the users. The mean of recalls is approximately 57%, and the recalls ranged from 44% to 70% for 75% of the user. The mean of F1-measure is approximately 60%, and the F1-measures ranged from 49% to 70% for 75% of the user.

TABLE 2: AVERAGES AND PERCENTILES DISTRIBUTION OF THE RECOMMENDATION ACCURACY MEASURES

		PRECISION	RECALL	F1 MEASURE	PRECISION7525	RECALL7525	F1 MEASURE7525	TRAINING SIZE	TESTING SIZE
PERCENTILES	5	.4824	.3220	.4139	.2539	.2425	.3486	16	5
	10	.5468	.4046	.4767	.3657	.3131	.4052	17	6
	25	.6667	.5357	.5882	.5020	.4372	.4912	25	8
	50	.7803	.6753	.7069	.6398	.5731	.5942	49	16
	75	.8824	.8000	.8055	.7679	.7016	.7018	111	37
	90	.9523	.9091	.8837	.8573	.8155	.8016	184	61
	95	1.0000	.9698	.9365	.9002	.8845	.8559	233	78



F1 Measure, Recall and Precision are computed using first test setting – all training, all testing;

F1 Measure7525, Recall7525 and Precision7525 are computed using second test setting – 3:1 split

Figure 5: Box-plots for recommendation accuracies

C. Comparisons with Existing Approaches

MovieLens is a benchmark dataset that has been widely used in recommendation research. Among other related successful studies using MovieLens, most of them reported results for top-5, top-10, top-15, and top-20 [12],[29],[33]. As shown in Sections 5.2, the median number of training cases and testing cases are 48 and 16 for the 3:1 split setting, respectively. Thus, the results reported in Section VI. B can be considered as approximation of those for top-15. This creates an equivalent base for a fair comparison between our study and existing successful studies on movie recommendation using the MovieLens dataset. The results are summarized in Table 3.

As shown in Table 3, the best precision, recall, and F1-measure of CF approaches to movies recommendation reported in the extant literature are 25%, 28%, and 23% respectively. Our approach significantly outperformed existing CF approaches by large margins. Additionally, for users whose recommendation sizes are between 7 and 15 (241 users, mean recommendation size =10, mean model size =32), the mean precision, recall and F1 measure are 64%, 60%, and 63% respectively. They are notably higher than those of CF approaches with comparable experimental setups and model parameters.

Probabilistic memory-based collaborative filtering [15] is another approach to modeling uncertainties due to stochastic nature of preference. Evaluations of the approach showed that on EACHMOVIE (ratings on movies) dataset, the mean precision, recall, and F1 measure of top-10 are 66%, 51%, and 57% respectively; and on JESTER (ratings on jokes) dataset, the mean precision, recall, and F1 measure of top-10 are 40%, 47%, and 43% respectively. Other studies that used both EACHMOVIE and MOVIELENS reported greater performance on EACHMOVIE [34],[29]**.** For example, Deshpande & Karypis [29] reported 40% of top-10 hit rate for EACHMOVIE compared to 26% for MovieLens dataset. Therefore, we expect that the performance of the proposed algorithms should be better than probabilistic CF approaches on these datasets.

TABLE 3: SUMMARY OF PERFORMANCE COMPARISON

Approaches	Recommendation Size	Model size	Performance Metrics	Results
Propose herein	Top-16	3:1 split of the data set, i.e., average 48	Accuracy	64%
			Precision	62%
			Recall	57%
			F1-measure	60%
Item-Based CF [29] User-Based CF [29]	Top-10 Top-10	20 20	Recall Recall	27% 28%
User-Based CF [12]	Top-10	50	F1-measure	23%
User-based [33]	Top-15	3:1 split	Precision	23%

Conventional recommender systems, e.g. CF, are computationally expensive and not scalable, and require high main memory during online adaptation and recommendation [7],[35]. Nasraoui and Petenes [35] empirically compared fuzzy inference engine with those of CF and nearest-profile based approaches in Web pages recommendation. The fuzzy method was found to provide very low computational cost, very faster and much lower main memory, and very intuitive in dealing with the natural lack of clear boundaries in user preferences. The complexity of the proposed algorithms is lower than conventional algorithms. Hence, similar advantages to [35] are expected to hold true for the proposed approach from empirical study.

D. Potential Applications

Discovered preference models in this study have a number of promising applications. First, they can be integrated into CRM systems to provide personalized items recommendations to users as well as provide in-depth insight to business professionals. Second, they can be used to identify customer or user segments with similar preferences using clustering techniques. Third, they can serve as the foundation for developing user-to-user CF systems (e.g., [12]). Fourth, they can be used to address the well-known scalability problem in CF algorithms.

1) Application to CRM

CRM systems include programs that allow business and its employees to provide fast, convenient, dependable, and consistent service to its customers. One of the emerging categories of CRM systems is its analytical CRM component that attempts to: (i) extract in-depth customer preferences, history, and profitability information from data warehouse and databases; (ii) analyze, predict, and infer customer behavior, preferences and demand; and (iii) offer recommendation of relevant items and offers that are personalized to individual customer preferences and needs [36]. The proposed approach and new algorithms can be integrated into CRM systems to support the various functions of the analytical CRM component. As shown in Figure 7, customer information and item attributes are first represented with inherent uncertainty. Then, customer preferences are discovered using the algorithm shown in Figure 2. Next, based on the discovered customer preferences, a ranked list of item recommendations is generated for the customer. Finally, customer feedback is recorded to support dynamic learning and updating of customer preferences.

The proposed approach, algorithms, and findings of this research offer a number of advantages to CRM. First, they provide in-depth insights into customer preferences to an item. This is because they are able to tell not only whether the customer likes or dislikes the item but also to what extent and why a customer prefers the item. Second, the discovered customer preferences can drive the production and supply of items. Third, preference models of customers can be incorporated into making recommendations for personalized targeted marketing. Fourth, the study shows how to integrate item attributes and user feedback in effective marketing.

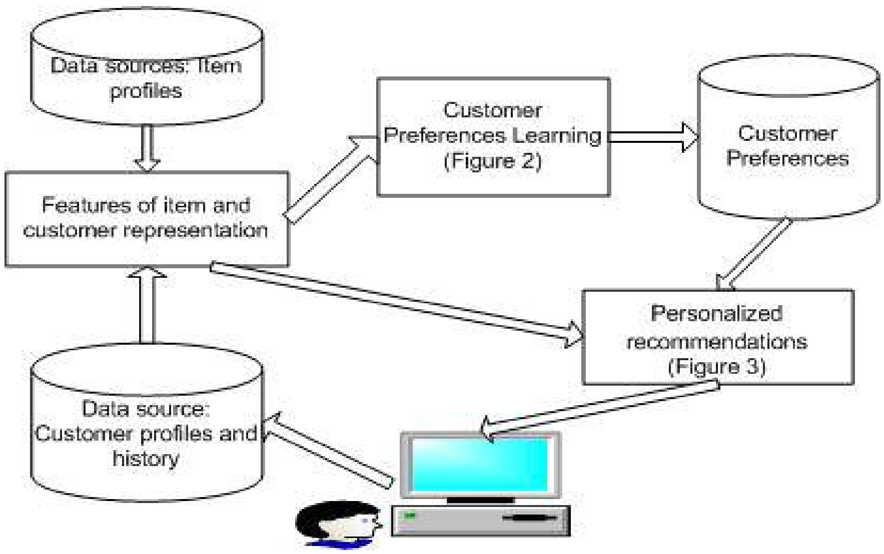


Figure 7: A Personalized Recommender System Architecture for Enhancing a CRM System

2) Application to Dimensionality Reduction in CF

In real-world e-commerce applications, which involve millions of users and a large number of items, the input rating-matrix is very large. It is also sparse because a user only rates a few items. The scalability problem emerges in searching tens of millions of potential neighbors. Some encouraging results have been obtained by using singular value decomposition to produce a low-dimensional representation of the original customer-product ratings matrix [7].This study provides a promising alternative approach to achieving the same goal. As shown in Figure 6, the discovered user preferences can be used to form user neighborhoods using the nearest neighbor algorithm based on preference similarities. Thus, recommendation scores of items can be computed using CF techniques. It is noted that the number of possible distinct values of attribute of an item (p) is much less than the number of items in traditional user-by-item or item-by-item rating matrix.

$$
\begin{array}{cccccc}
 & x_1 & & x_k & .. & x_p \\
u_1 & x_{i1} & .. & x_{i1} & .. & x_{1p} \\
.. & .. & .. & .. & .. & .. \\
u_n & x_{n1} & .. & x_{nk} & .. & x_{np}
\end{array}
$$

where x_{ij} is the degree of preferences of user u_i to a feature value x_j

Figure 6: Reduced matrix in space of preferences

VII. Conclusions and Future Work

User preferences can be learned from user behavior manifested during user-system interactions. This paper presents a novel approach to representing items and user preferences under uncertainty. The approach uses fuzzy set to represent item attributes and user preferences, which is more accurate and appropriate than traditional approaches using binary or crisp set. This in turn improves the reasoning about user's preference on items. It also provides an opportunity for learning user's preferences from the user past behavior expressed by user feedbacks (ratings) on items.

Based on the proposed representation framework and fuzzy logic, an algorithm is developed to determine user preference on values of an attribute of items. In addition, another algorithm was developed to predict user preferences and make recommendation of items by utilizing the learned user preferences and the values of the attribute in targeted items. Compared to conventional preference modeling and recommendation algorithms, the proposed algorithms are unique in the following perspectives:

1. They integrate information from both user feedbacks such as ratings and item attributes rather than only considering user feedbacks in making recommendations.
2. They are based on a single user's feedbacks and item attributes rather than other users'

feedbacks. Hence, they can provide individualized recommendations without suffering from rating sparsity and new item problems that are common in collaborative filtering.

3. They use fuzzy set and logic theory to represent and reason about uncertainty due to subjectivity, imprecision and vagueness in item attributes and user preferences. Therefore, they provide how much a user likes, dislikes or be indifferent to a given item and its attributes.
4. They use Fuzzy theoretic based extension of the cosine similarity measure.

The effectiveness of the proposed algorithms is empirically evaluated in the movie recommendation domain using the dataset extracted from MovieLens and Internet movie databases. The results show an over 100% increase in precision, recall and F1-measure in recommendation accuracy compared with traditional recommender systems. Moreover, the proposed approach, and algorithms can be used for any item or service recommendation that is represented with uncertainty in space of values of attributes as defined in Section III. Items such as movies, books, music, web pages, and restaurants are specific examples. Furthermore, the proposed approach and algorithms are scalable.

The proposed approach and results of this study advance the theory and practice in preference modeling and recommender systems in several ways. First, this study contributes to the theory of knowledge representation for preference modeling by identifying the types of subjectivity, vagueness and associated uncertainty that exist in user preferences and item features. Second, this study shows a formalism to quantify how much a user likes, dislikes or be indifferent to a given item and its features based upon fuzzy set and logic theory. Third, this study enhances the completeness of preference modeling by including positive, negative, neutral and unknown classes of preferences. Finally, this study improves the effectiveness of recommendation systems by inferring about users' preferences to items using item attributes and user attributes, the complete preference model, and fuzzy theoretic based extension of the cosine similarity measure. Therefore, integrating item attributes and user feedbacks along with modeling uncertainty can contribute to the effectiveness and efficiency of recommender systems.

The proposed approach, algorithms and findings of this research can be integrated into CRM systems to improve CRM by helping businesses create effective marketing strategies, and targeted production and promotions. Ultimately, they can lead to increased customer retention and loyalty, and increase sales and advertising profits. Moreover, they can be used to identify customer segments with similar preferences using clustering techniques, and to enhance collaborative filtering algorithms.

Further studies are planned to address some of the limitations of this study, as well as to extend the proposed approach in several directions. First, additional attributes e.g. actors/actresses, and directors in movies, can be included in preference modeling to further improve the accuracy in predicting user preferences. Second, the relationship between user characteristics (e.g., demographic information) and preference to item attributes can be investigated to enhance targeted marketing effort. Third, the effectiveness of discovered preferences in reducing the scalability problem and providing quality recommendation in CF needs to be evaluated empirically. Furthermore, based on the proposed approach, a hybrid of the content-based (Figure 3) and collaborative recommendation (Figure 6) approaches can be developed and evaluated. Fourth, genetic algorithm can be used for learning and tuning the fuzzy sets parameters of the membership function defined in (1). Fifth, user preferences may change over time. Hence, it is crucial for recommender systems to develop mechanism for dynamically updating user preference based on the distribution and recency of user feedbacks.

REFERENCES

[1] V. Ha and P. Haddawy, "Similarity of Personal Preferences: Theoretical Foundations and Empirical Analysis," *Artificial Intelligence*, vol. 146, no.2, pp. 149-173, 2003.

[2] Q. Chen and A. F. Norcio, "Knowledge Engineering in Adaptive Interface and User Modeling," in *Human Computer Interaction: Issues and Challenges*, Q. Chen, Ed. PA: Idea Group Publishing, 2001, pp. 113-133.

[3] G. Adomavicius and A. Tuzhilin, "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," *IEEE Transactions of Knowledge and Data Engineering*, vol. 17, no.6, pp. 734-749, 2005.

[4] C. Joslyn and L. Rocha, "Towards a formal taxonomy of hybrid uncertainty representations," *Information Sciences*, vol. 110, pp. 255-277, 1998.

[5] A. Salam, L. Iyer, P. Palvia, and R. Singh, "Trust in e-commerce," *Communication of ACM*, vol. 48, no.2, pp. 72-77, Feb. 2005.

[6] C. M. Serino, C. P. Furner, and C. Smatt, "Making it personal: How personalization affects trust over time," Proc. 38th Hawaii International Conference on System Sciences, 2005.

[7] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl, "Application of Dimensionality Reduction in Recommender Systems -- A Case Study," ACM WebKDD 2000, Web Mining for E-Commerce Workshop, 2000.

[8] R. Burke, "Hybrid Recommender Systems: Survey and Experiments," *User modeling and user-adapted interaction*, vol. 12, no.4, pp. 331-370, 2002.

[9] J. B. Schafer, K. J, and J. Riedl, "Electronic Commerce Recommender Applications," *Journal of Data Mining and Knowledge Discovery*, vol. 5, no.1/2, pp. 115-152, 2001.

[10] H. T. Nguyen and P. Haddawy, "The Decision-Theoretic Video Advisor," Proc. 15th Conference on Uncertainty in Artificial Intelligence, pp. 494-501, 1999.

[11] R. Mukherjee, P. S. Dutta, and S. Sen, "MOVIES2GO: A new approach to online movie recommendation," Seventeenth International Joint Conference on Artificial Intelligence (IJCAI 2001) Workshop on Intelligent Techniques for Web Personalization, 2001.

[12] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl, "Analysis of Recommender Algorithms for E-Commerce," Proc. Proceedings of the ACM E-Commerce 2000 Conference, pp. 158-167, 2000.

[13] A. Rashid, I. Albert, D. Cosley, S. Lam, S. Mcnee, J. Konstan, and J. Riedl, "Getting to know you: Learning new user preferences in recommender systems," Proc. 2002 International Conference on Intelligent User Interfaces (IUI'02), pp. 127-134, 2004.

[14] S. Y. Jung, J.-H. Hong, and T.-S. Kim, "A Statistical Model for User Preference," *IEEE Transactions of Knowledge and Data Engineering*, vol. 17, no.6, pp. 834-843, 2005.

[15] K. Yu, A. Schwaighofer, V. Tresp, X. Xu, and H.-P. Kriegel, "Probabilistic Memory-Based Collaborative Filtering," *IEEE Transactions of Knowledge and Data Engineering*, vol. 16, no.1, pp. 56-69, 2004.

[16] D. Dubois, H. T. Nguyen, and H. Prade, "Possibility Theory, Probability and Fuzzy Sets," in *Fundamentals of Fuzzy Sets*, *the Handbooks of Fuzzy Sets*, D. Dubois and H. Prade, Eds. Boston: Kluwer Academic Publishers, 2000, pp. 343-438.

[17] L. A. Zadeh, "Fuzzy Logic, Neural Networks, and Soft Computing," *Comm. ACM*, vol. 37, no.3, pp. 77-84, 1994.

[18] E. Turban and J. E. Aronson, *Decision Support Systems and Intelligent Systems*. New Jersey: Prentice Hall, 2001.

[19] L. A. Zadeh, "Fuzzy Sets," *Information Control*, vol. 8, pp. 338-353, 1965.

[20] T. Bilgiç and I. B. Turksen, "Measurement of Membership Functions: Theoretical and Empirical Work," in *Fundamentals of Fuzzy Sets*, *Handbook of Fuzzy Sets and Systems*, D. Dubois and H. Prade, Eds. Boston: Kluwer, 2000, pp. 195-232.

[21] W. Pedrycz and F. Gomide, *An Introduction to Fuzzy Sets*. Cambridge, Massachusetts: The MIT Press, 1998.

[22] S.-M. Hsu, C. Wu, and T.-W. Tien, "A Fuzzy Mathematical Approach for Measuring Multi-facet Consumer Involvement in the Product Category Evaluation," *Marketing Research On-Line*, vol. 3, pp. 1-19, 1998.

[23] R. Altman, *Film/Genre*. London: British Film Institute, 1999.

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

[25] R. A. M. Gregson, *Psychometrics of Similarity*. New York: Academic Press, 1975.

[26] V. V. Cross and T. A. Sudkamp, *Similarity and Compatibility in Fuzzy Set Theory: Assessment and Applications*. Heidelberg; New York: Physica-Verlag, 2002.

[27] A. Zenebe, "Uncertainty Identification, Representation and Inference in a User Adaptive Information System," Dissertation Thesis, *Department of Information Systems*. Baltimore, MD: University of Maryland Baltimore County, 2005.

[28] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," Proc. 10th International World Wide Web Conference (WWW10), pp. 285-295, 2001.

[29] M. Deshpande and G. Karypis, "Item-Based Top-N Recommendation algorithms," *ACM Transactions on Information Systems*, vol. 22, no.1, pp. 143-177, 2004.

[30] E. Cooper-Martin, "Consumers and Movies: Some Findings on Experiential Products," *Advances in Consumer Research*, vol. 18, pp. 372-378, 1991.

[31] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating Collaborative Filtering Recommender Systems," *ACM Transactions on Information Systems*, vol. 22, no.1, pp. 5-53, 2004.

[32] D. Billsus and M. J. Pazzani, "User Modeling for Adaptive News Access," *User modeling and user-adapted interaction*, vol. 10, no. 2-3, pp. 147-180, 2000.

[33] M. R. McLaughlin and J. L. Herlocker, "A Collaborative Filtering Algorithm and Evaluation Metric that Accurately Model the User Experience," Proc. 27th Annual ACM Conference on Research and Development in Information Retrieval, pp. 329-336, 2004.

[34] D. O. Sullivan, B. Smyth, and D. Wilson, "Preserving Recommender Accuracy and Diversity in Sparse Datasets," *International Journal on Artificial Intelligence Tools*, vol. 13, no.1, pp. 219-235, 2004.

[35] O. Nasraoui and C. Petenes, "Combining Web Usage Mining and Fuzzy Inference for Website Personalization," Proc. Fifth WebKDD Workshop: Web mining as a Premise to effective and Intelligent Web Applications (WebKDD'2003), in conjunction with the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 37-45, 2003.

[36] J. A. O'Brien, *Management information systems: managing information technology in business enterprise*, 6th ed. Boston: McGraw-Hill Inc., 2004.