

Research Article

Aggregated Recommendation through Random Forests

Heng-Ru Zhang,¹ Fan Min,¹ and Xu He^{1,2}

¹ School of Computer Science, Southwest Petroleum University, Chengdu 610500, China

² Lab of Granular Computing, Minnan Normal University, Zhangzhou 363000, China

Correspondence should be addressed to Fan Min; minfanphd@163.com

Received 23 May 2014; Accepted 23 July 2014; Published 11 August 2014

Academic Editor: Marco Listanti

Copyright © 2014 Heng-Ru Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Aggregated recommendation refers to the process of suggesting one kind of items to a group of users. Compared to user-oriented or item-oriented approaches, it is more general and, therefore, more appropriate for cold-start recommendation. In this paper, we propose a random forest approach to create aggregated recommender systems. The approach is used to predict the rating of a group of users to a kind of items. In the preprocessing stage, we merge user, item, and rating information to construct an aggregated decision table, where rating information serves as the decision attribute. We also model the data conversion process corresponding to the new user, new item, and both new problems. In the training stage, a forest is built for the aggregated training set, where each leaf is assigned a distribution of discrete rating. In the testing stage, we present four predicting approaches to compute evaluation values based on the distribution of each tree. Experiments results on the well-known MovieLens dataset show that the aggregated approach maintains an acceptable level of accuracy.

1. Introduction

Recommender systems (RSs) [1–3] have been extensively studied to present items, such as movies, music, and books. They collect information on the preferences of their users for a set of items. The information is used to fulfill two main user tasks; one is predicting the rating [4], and the other is finding good items [5].

Model-based RSs apply demographic or content information to construct a model. Some algorithms, such as Bayesian classifiers [6] and decision trees [7], have been used to generate respective models. Model-based algorithms are suitable for cold-start recommendation for new item, new user, and new community [2, 8, 9].

Aggregated RS aims at recommending a kind of items to a group of users. A popular approach is to jointly recommend items to user groups [10] (e.g., a group of four friends who wish to choose a movie). Given the specific characteristics of the recommendation to groups, Jameson and Smyth [11] proposed to appropriately establish a consensus for different group semantics that formalize the agreements and disagreements among users.

In this paper, we propose a random forest (RF) approach to create aggregated RS through taking advantage of demographic, content, and rating information. This approach intends to predict the ratings of a group of users to a kind of items and deal with three cold-start recommendation problems, namely, new item (NI), new user (NU), and double cold-start (DOCS), where new items are recommended to new users. Decision tree [12] is a natural approach to these problems. However, one decision tree only takes advantage of limited information of users and items. Therefore, for many new users and new items, a decision tree may produce no predicting result at all. Our approach uses different information of users and items to construct an RF (a collection of decision trees [13]) to avoid this situation.

Our approach has three stages. In the preprocessing stage, the user, item, and rating tables of the original dataset are merged into an aggregated table. Then we construct training and testing sets through cross-validation. To the best of the authors' knowledge, little is known regarding the aggregated approach. In the training stage, an RF predictor is built to ensemble individual tree predictors. Each decision tree classifier is generated from the training set with each leaf

assigned a distribution of the class attribute. In the testing stage, the demographic and content information of each user-item pair are fed to all decision trees in the RF. Each instance can get the class distribution information through top-down search. We adopt four predicting approaches, called standard voting, weighted average, distribution aggregation based voting, and distribution aggregation based average, to compute evaluation values of the RF.

The contribution of the paper is fourfold. First, we propose a new aggregated approach to predict the rating of a group of users to a kind of items. Second, we merge demographic, content, and rating information into a new aggregated table and then adopt three kinds of strategies, namely, NI, NU, and DOCS, to split training and testing sets through cross-validation. Third, we build five kinds of aggregated RSs by RF. The first three RFs are NI, NU, and DOCS. The two other RFs are new item with average rating (NIAR) and new user with average rating (NUAR). DOCS RS can be used to recommend a kind of new items to a group of new users. Fourth, we develop four ensemble approaches to compute predicting ratings of these aggregated RSs.

Experiments are undertaken with five scenarios corresponding to five aggregated RSs. The abovementioned ensemble approaches are employed to find the appropriate setting of the forest size and compare the performance with respect to the mean absolute error (MAE) [14] in each scenario. MAE is a statistical accuracy metric that measures the deviation between real ratings and predictions generated by the RS. If $\{r_1, \dots, r_N\}$ are all the real values in the target set, $\{p_1, \dots, p_N\}$ are the predicted values for the same ratings, and $E = \{\varepsilon_1, \dots, \varepsilon_N\} = \{p_1 - r_1, \dots, p_N - r_N\}$, then the MAE is

$$|\bar{E}| = \frac{\sum_{i=1}^N (|\varepsilon_i|)}{N}. \quad (1)$$

The lower the MAE, the more accurate the approach.

Experimental results on the well-known MovieLens dataset show that (1) the size of the forest is not large to ensure that the performance in terms of MAE keeps stable; (2) the aggregated approach maintains an acceptable level of accuracy.

2. Data Models

In this section, the original data sets are converted to an aggregated decision table. Five kinds of aggregated data models are constructed through cross-validation.

2.1. Information Systems and Decision Systems. In this subsection, we revisit the definitions of information system [15] and decision systems.

Definition 1. $S = (U, A)$ is an information system, where $U = \{x_1, x_2, \dots, x_n\}$ is the set of all objects, $A = \{a_1, a_2, \dots, a_m\}$ is the set of all attributes, and $a_j(x_i)$ is the value of x_i on attribute a_j for $i \in [1, \dots, n]$ and $j \in [1, \dots, m]$.

Example 2. An example of information system is given by Table (a) of Figure 1, where $U = \{u_1, u_2, u_3, u_4\}$ and

$A = \{\text{UID, Age, Gender, Occupation}\}$. UID is a key. Another example of information system is given by Table (b) of Figure 1.

The decision system is a fundamental concept in data mining and machine learning and is often defined as follows [16].

Definition 3. A decision system S is the 5-tuple:

$$S = (U, C, D, \{V_a \mid a \in C \cup D\}, \{I_a \mid a \in C \cup D\}), \quad (2)$$

where U is a finite set of objects called the universe, C is the set of conditional attributes, D is the set of decision attributes, V_a is the set of values for each $a \in C \cup D$, and $I_a \mid U \rightarrow V_a$ is an information function for each $a \in C \cup D$.

2.2. Rating System. In this subsection, the rating system is defined.

Definition 4. Let $U = \{x_1, x_2, \dots, x_n\}$ and $V = \{y_1, y_2, \dots, y_l\}$ be two sets of objects.

Consider

$$R : U \times V \longrightarrow R_k, \quad (3)$$

where R_k is the domain of rating.

If R_k is boolean, R is a binary relation from U to V . If R_k is numeric, R is a rating function from U to V . In this paper, we discuss the numeric ratings. A rating function is more often stored in the database as a table with two foreign keys. In this way the storage is saved. For the convenience of illustration, here we represented it with an $n \times l$ rating matrix.

With Definitions 1 and 4, we propose the following definition.

Definition 5. A rating system is a 5-tuple $RS = (U, A, V, B, R)$, where (U, A) and (V, B) are two information systems, and $R : U \times V \rightarrow R_k$ is a rating function.

Example 6. An example of rating is given by Table (c) of Figure 1, where U is the set of users as indicated by Table (a) of Figure 1 and V is the set of items as indicated by Table (b) of Figure 1. Here $R_k = \{0, 1, 2, 3, 4, 5\}$.

An example of rating system includes Tables (a), (b), and (c) of Figure 1.

2.3. Aggregated Decision Systems. In this subsection, we build decision systems to mine the behavior of users on items. For this purpose, we propose the concept of aggregated decision system as follows.

Definition 7. An aggregated decision system induced by a rating system (RS) is

$$DS(RS) = (G, C, d), \quad (4)$$

where $G = U \times V$ and $C = A \cup B$, for all $g_{i,j} = (x_i, y_j) \in G$, $d(g_{i,j}) = R(x_i, y_j)$.

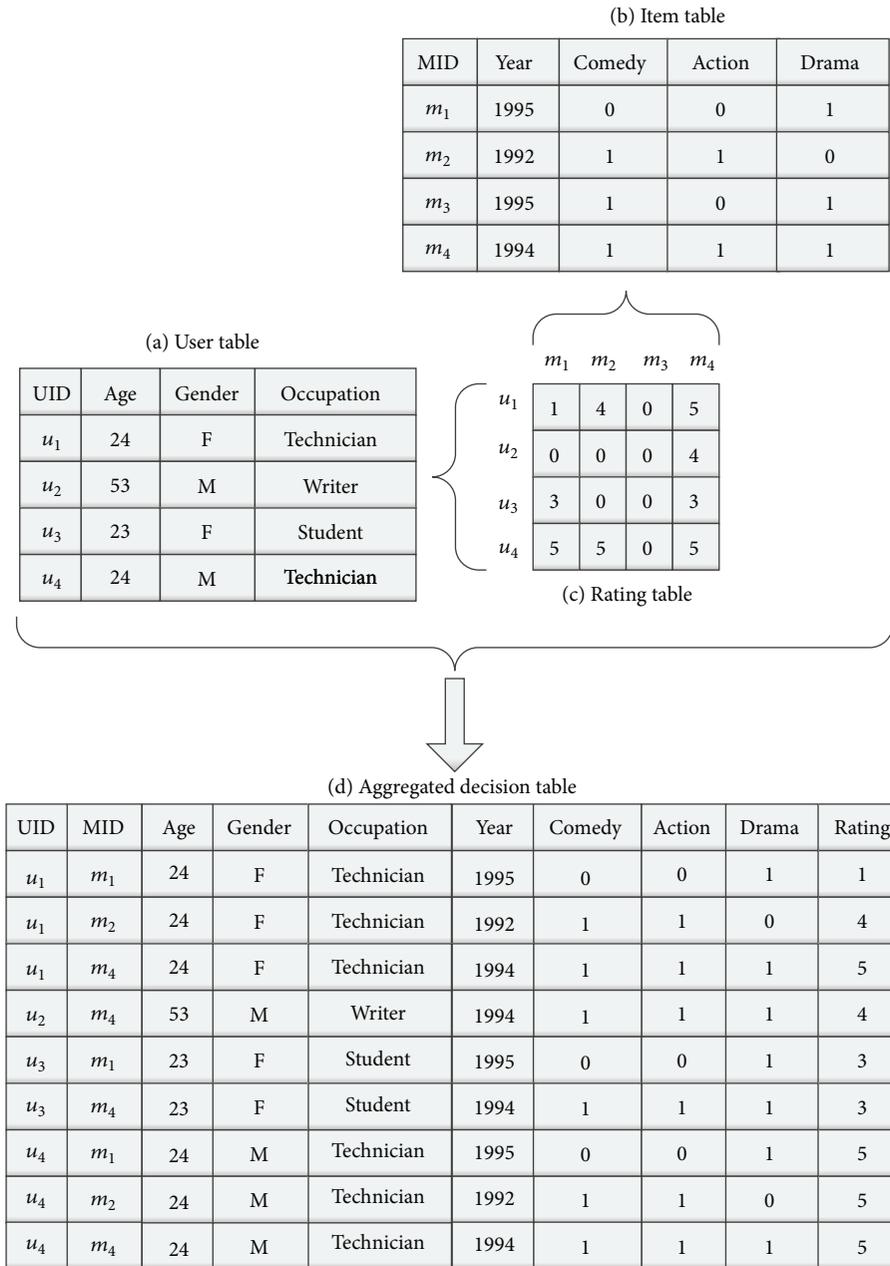


FIGURE 1: Aggregated decision system.

The number of objects in DS(RS) is $|U| \times |V|$. To distinguish this type of decision system from the other types discussed later, we refer to it as the *aggregated decision system* (ADS) or the *first-class decision system* (1-DS).

In Table (c) of Figure 1 some elements are 0, in which zero element means that a user has not rated the movie. We remove them and get a new decision system $DS^+(RS)$ as follows.

Definition 8. An aggregated decision system with positive rating induced by a rating system (RS) is

$$DS^+(RS) = (G^+, C, d), \tag{5}$$

where $G^+ = \{(x, y) \mid x \in U, y \in V, R(x, y) > 0\}$ and $C = A \cup B, \forall g_{i,j} = (x_i, y_j) \in G^+, d(g_{i,j}) = R(x_i, y_j)$.

The number of objects in $DS^+(RS)$ is $|G^+|$. We refer to it as the ADS^+ or the *second-class decision system* (2-DS).

Example 9. Table (d) of Figure 1 presents a ADS^+ decision system where $G^+ = \{(u_1, m_1), (u_1, m_2), (u_1, m_4), (u_2, m_4), (u_3, m_1), (u_3, m_4), (u_4, m_1), (u_4, m_2), (u_4, m_4)\}$, $C = \{Age, Gender, Occupation, Year, Comedy, Action, Drama\}$, and $d = \{Rating\}$. (UID, MID) is a key pair. The key pair does not participate in the establishment of RF model; therefore, it is not used in the mining work.

The attribute of average rating (AR) has been the focus of most empirical studies on product reviews [17]. There are two kinds of AR. One kind is AR of user (UAR), which reflects the rating habit for the user. The other is AR for item (IAR), which reflects the degree of item popularity. With Definition 8, we can define a new type of the aggregated decision system with AR as follows.

Definition 10. An aggregated decision system with AR(RDS⁺) or the *third-class decision system* (3-DS) is

$$DS_{AR}^+(RS) = (G^+, C', d), \quad (6)$$

where $G^+ = \{(x, y) \mid x \in U, y \in V, R(x, y) > 0\}$ and $C' = A \cup B \cup \{AR\}$, $\forall g_{i,j} = (x_i, y_j) \in G, d(g_{i,j}) = R(x_i, y_j)$.

Example 11. The movies m_1, m_2 , and m_4 are rated by user u_1 in Table (c) of Figure 1. Therefore, the AR of u_1 is $[(1+4+5)/3] = 3$. Similarly, the AR of u_4 is $[(5+5+5)/3] = 5$.

The movie m_1 is rated by the users u_1, u_3 , and u_4 in Table (c) of Figure 1. Therefore, the AR for m_1 is $[(1+3+5)/3] = 3$. Similarly, the AR for m_4 is $[(5+4+3+5)/4] = 4$.

In some situations we are interested in the aggregated decision system concerning a subset of users and items.

Definition 12. A subset of the aggregated decision system (SDS⁺) or the *fourth-class decision system* (4-DS) with respect to U' and V' is

$$DS^+(RS, U', V') = (G', C, d), \quad (7)$$

where $G' = \{(x, y) \mid x \in U', y \in V', R(x, y) > 0\}$, $U' \subseteq U$, $V' \subseteq V$, and $C = A \cup B \cup AR$, $\forall g_{i,j} = (x_i, y_j) \in G', d(g_{i,j}) = R(x_i, y_j)$.

Definition 13. A subset of the aggregated decision system with AR (SRDS⁺) or the *fifth-class decision system* (5-DS) with respect to U' and V' is

$$DS_{AR}^+(RS, U', V') = (G', C', d), \quad (8)$$

where $G' = \{(x, y) \mid x \in U', y \in V', R(x, y) > 0\}$, $U' \subseteq U$, $V' \subseteq V$, and $C' = A \cup B \cup AR$, $\forall g_{i,j} = (x_i, y_j) \in G', d(g_{i,j}) = R(x_i, y_j)$.

In this paper, we discuss the cold-start problem. In Definitions 12 and 13, the demographic or content information of the subsets is not independent. Therefore, the subsets are not used to solve cold-start problem.

2.4. Data Splitting. For proper estimation of the classification accuracy, the decision system is divided into training and testing sets. The training set is used to calculate a classifier, which is used to classify the testing set.

We adopt three kinds of splitting strategies based on ADS⁺, namely, NU, NI, and DOCS. Two kinds of splitting strategies are adopted based on RDS⁺, namely, NUAR and NIAR. NU and NUAR approaches split the user table into

two parts. Then each part is merged into item and rating information, respectively, to construct training and testing sets. A sample is that the number of training or testing sets is 1/2 of the original set. NI and NIAR approaches split the item table into two parts. Then each part is merged into user and rating information, respectively, to construct training and testing sets. A sample is that the number of training or testing sets is 1/2 of the original set.

DOCS approach splits user table and item table into two parts, respectively. The first part of user table, the first part of item table, and rating information are merged into training set. And the second part of user table, the second part of item table, and rating information are merged into testing set. A sample is that the number of training or testing sets is 1/4 of the original set.

Supposing a group of new users and the item model, the function predicts whether these users would be interested in a set of items. With Definitions 8 and 12, the training or testing set of NU is defined as follows.

Definition 14. A subset of the aggregated decision system with respect to user sampling NU is

$$DS_{NU}^+(RS, U', V) = (G', C, d), \quad (9)$$

where $U' \subseteq U$, $G' = \{(x, y) \mid x \in U', y \in V, R(x, y) > 0\}$.

Give $U'_r \neq \emptyset$ and $U'_t \neq \emptyset$, where $U'_r \cup U'_t = U$ and $U'_r \cap U'_t = \emptyset$. While $U' = U'_r$, $DS_{NU}^+(RS, U', V)$ is the training set. While $U' = U'_t$, $DS_{NU}^+(RS, U', V)$ is the testing set.

Given a set of new items and the user model, the function predicts whether a group of users would be interested in these items. With Definitions 8 and 12, the training or testing set of NI is defined as follows.

Definition 15. A subset of the aggregated decision system with respect to user sampling NI is

$$DS_{NI}^+(RS, U, V') = (G', C, d), \quad (10)$$

where $V' \subseteq V$, $G' = \{(x, y) \mid x \in U, y \in V', R(x, y) > 0\}$.

Give $V'_r \neq \emptyset$ and $V'_t \neq \emptyset$, where $V'_r \cup V'_t = V$ and $V'_r \cap V'_t = \emptyset$. While $V' = V'_r$, $DS_{NI}^+(RS, U, V')$ is the training set. While $V' = V'_t$, $DS_{NI}^+(RS, U, V')$ is the testing set.

With Definitions 12, 14, and 15, the training or testing set of DOCS is defined as follows.

Definition 16. A subset of the aggregated decision system with respect to user sampling DOCS is

$$DS_{DOCS}^+(RS, U', V') = (G', C, d), \quad (11)$$

where $U' \subseteq U$, $V' \subseteq V$, and $G' = \{(x, y) \mid x \in U', y \in V', R(x, y) > 0\}$.

Give $U'_r \neq \emptyset$ and $U'_t \neq \emptyset$, where $U'_r \cup U'_t = U$ and $U'_r \cap U'_t = \emptyset$. Give $V'_r \neq \emptyset$ and $V'_t \neq \emptyset$, where $V'_r \cup V'_t = V$ and $V'_r \cap V'_t = \emptyset$. While $U' = U'_r$ and $V' = V'_r$,

$DS_{DOCS}^+(RS, U', V')$ is the training set. While $U' = U'_t$ and $V' = V'_t$, $DS_{DOCS}^+(RS, U', V')$ is the testing set.

With Definitions 10 and 13, the training or testing set of NUAR is defined as follows.

Definition 17. A subset of the aggregated decision system with respect to user sampling NUAR is

$$DS_{NUAR}^+(RS, U', V) = (G', C', d), \quad (12)$$

where $U' \subseteq U$, $G' = \{(x, y) \mid x \in U', y \in V, R(x, y) > 0\}$, and $C' = A \cup B \cup \{IAR\}$.

Let $U'_r \cup U'_t = U$, $U'_r \cap U'_t = \emptyset$, $U'_r \neq \emptyset$, and $U'_t \neq \emptyset$. While $U' = U'_r$, $DS_{NUAR}^+(RS, U', V)$ is the training set. While $U' = U'_t$, $DS_{NUAR}^+(RS, U', V)$ is the testing set.

With Definitions 10 and 13, the training or testing set of NIAR is defined as follows.

Definition 18. A subset of the aggregated decision system with respect to user sampling NIAR is

$$DS_{NIAR}^+(RS, U, V') = (G', C', d), \quad (13)$$

where $V' \subseteq V$, $G' = \{(x, y) \mid x \in U, y \in V', R(x, y) > 0\}$, and $C' = A \cup B \cup \{UAR\}$.

Let $V'_r \cup V'_t = V$, $V'_r \cap V'_t = \emptyset$, $V'_r \neq \emptyset$, and $V'_t \neq \emptyset$. While $V' = V'_r$, $DS_{NIAR}^+(RS, U, V')$ is the training set. While $V' = V'_t$, $DS_{NIAR}^+(RS, U, V')$ is the testing set.

3. Random Forest Based Prediction

In this section, an RF for aggregated dataset is constructed. Four kinds of predicting approaches will be used to compute evaluation values.

3.1. Construct the Random Tree. There are two aspects to build an RF: (1) random decision trees are built based on the aggregated training set; (2) an RF is constructed based on these trees.

Decision tree learners build a decision tree by recursively partitioning training data. In the build process of random decision tree, demographic and content information serve as conditional attributes, and rating information serves as the decision attribute. Each root-to-leaf path of tree represents a rule for the ratings of one kind of movies by one group of users.

Example 19. In Table (d) of Figure 1, the conditional attributes are {Age, Gender, Occupation, Year, Comedy, Action, Drama} and the decision attribute is {Rating}.

There are four steps to build random decision tree: (1) an attribute is randomly selected from the conditional ones as the root node, when the information gain of the attribute is more than 0; (2) the original set will be split to many subsets based on values of the root node; (3) other splitting nodes are constructed based on algorithm of random decision tree, and

these subsets will be split recursively to construct subtrees; (4) the leaves are assigned the vector which indicates the distribution of the decision values. The building process of a random decision tree is described in Algorithm 1.

The following examples illustrate the selection process of root node and the way to get the distribution of leaf node.

Example 20. The conditional attribute is randomly selected as tree node. After a randomized selection, the root node of Figure 2(a) is {Occupation}, and the root node of Figure 2(b) is {Action}. Then we illustrate the subtree and leaf-node construction process of Figure 2(a). The training data is split according to three values {technician}, {writer}, and {student} of {Occupation}. {Drama} is randomly selected as subtree node corresponding to the value {technician}. The leaf node corresponding to the value {student} is the distribution of the decision values. There are two instances corresponding to the value {student}, and the decision value of the two instances is the same rating based on the aggregated decision table of Figure 1. Therefore, the distribution of the leaf node is {0, 0, 0, 2, 0, 0} corresponding to the rating rated from {student}. If standard voting to the distribution is used when the leaf node is built, the leaf of {student} is 3. In other words, the root-to-leaf path of the tree represents a rule that the rating of all movies rated by the *student* is 3.

A random decision tree only takes advantage of limited information of users and items. Therefore, for many new users and new items, a random decision tree may produce no predicting result at all. For example, the tree of Figure 2(a) has no {Age} information of users. No predicting result is produced if based on the classifier of {Age} information. But two trees of Figures 2(a) and 2(b) can avoid this situation.

3.2. Construct the Forest. Based on five kinds of splitting approaches mentioned in Section 2.4, we construct five kinds of RFs: (1) NU forest, (2) NUAR forest, (3) NI forest, (4) NIAR forest, and (5) DOCS forest. NU and NUAR forests are only used to solve new user problem. The two models can predict the rating of a group of new users to a kind of items. NI and NIAR forests are only used to solve new item problem. The two models can predict the rating of a group of users to a kind of new items. DOCS forest can be used to solve both new user and item problems. The model can predict the rating of a group of new users to a kind of new items.

There are three steps to build the RF. (1) Aggregated training and testing set are generated according to the different RF models. For NU forest, the original data is split based on Definition 14. For NUAR forest, the original data is split based on Definition 17. For NI forest, the original data is split based on Definition 15. For NIAR forest, the original data is split based on Definition 18. For DOCS forest, the original data is split based on Definition 16. (2) Condition attributes are randomized based on random seed. (3) Build N random decision trees based on Algorithm 1. N designated by the user is the size of forest. The building process of the random forest is described in Algorithm 2.

After the RFs are built, we can use them to predict. Figure 3 depicts the RF's building and predicting process.

```

Input: trainingSet ( $S_t$ ), condition attributes ( $C$ ), decision attribute ( $d$ )
Output: decisionTree
Method: buildRandomTree
(1) if ( $S_t = \emptyset$  or  $C = \emptyset$ ) then
(2)   leaf = the distribution of  $d$  in the  $S_t$ 
(3)   return
(4) end if
(5) if (all  $d$  of  $S_t$  are equal) then
(6)   leaf = the distribution of  $d$  in the  $S_t$ 
(7)   return
(8) end if
(9)  $C_t$  is a copy of  $C$ 
(10) findNode = false
(11) while (not findNode) do
(12)   att = an attribute randomly selected from  $C$ 
(13)   if (gain(att) > 0) then
(14)     findNode = true
(15)     break
(16)   end if
(17)    $C_t = C_t - att$ 
(18)   if  $C_t = \emptyset$  then
(19)     break
(20)   end if
(21) end while
(22) if (not findNode) then
(23)   leaf = the distribution of  $d$  in the  $S_t$ 
(24)   return
(25) end if
(26) Decision tree attribute for Root = att
(27) for (each possible value  $V_i$  of att) do
(28)   Add a new tree branch below Root, corresponding to  $V_i$ .
(29)   Let subSet( $V_i$ ) be the subset of  $S_t$  that have the value  $V_i$ 
(30)   if (subSet( $V_i$ ) =  $\emptyset$ ) then
(31)     leaf = the distribution of  $d$  in the  $S_t$ 
(32)     return
(33)   else
(34)     subTree = buildRandomTree(subSet( $V_i$ ),  $C - att$ ,  $d$ )
(35)   end if
(36) end for

```

ALGORITHM 1: Random decision tree.

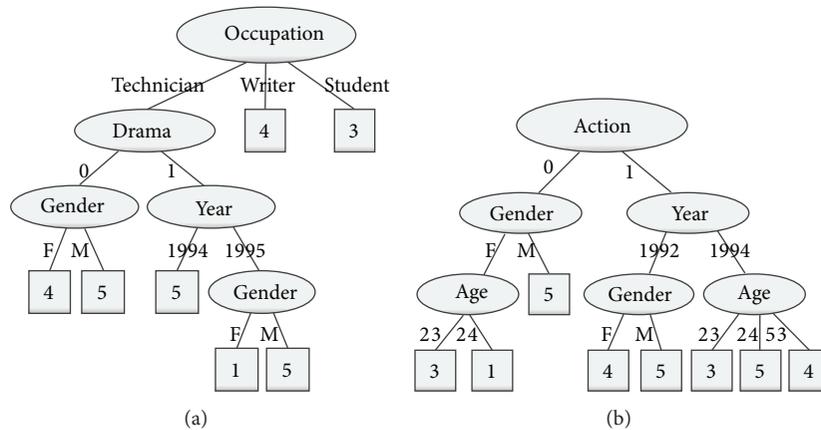


FIGURE 2: Samples of random decision tree.

```

Input: size of forest ( $N$ ), condition attributes ( $C$ ), decision attribute ( $d$ )
Output: randomForest( $RF$ )
Method: buildRandomForest
(1) generate aggregated training ( $S_i$ ) and testing ( $S_e$ ) set
(2)  $i = 0$ 
(3) while ( $i < N$ ) do
(4)   randomize  $C$ 
(5)    $RT_i = buildRandomTree(S_i, C, d)$ 
(6)    $i = i + 1$ 
(7) end while
(8)  $RF = \{RT_0, RT_1, \dots, RT_{N-1}\}$ 
(9) return  $RF$ 
    
```

ALGORITHM 2: Random forest.

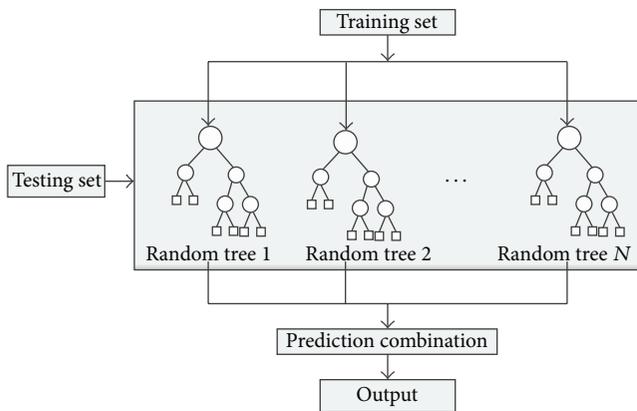


FIGURE 3: Random forest based prediction.

Multiple RFs are built through selecting different numbers of random trees. Each tree in Figure 3 uses a different random seed; therefore, each one significantly contributes to the prediction.

3.3. *Predicting Approaches.* For each RF, we design four prediction approaches: (1) standard voting, (2) weighted average, (3) distribution aggregation based voting, and (4) distribution aggregation based average.

By comparing approaches (1) and (2), we can know which is more precise between standard voting and weighted average. By comparing approaches (3) and (4), we can know which is more precise between distribution aggregation based voting and distribution aggregation based average. By comparing approaches (1) and (3), we can know which is more precise between standard voting and distribution aggregation based voting. By comparing approaches (2) and (4), we can know which is more precise between weighted average and distribution aggregation based average.

We describe the four combination algorithms as follows.

(1) *Standard Voting.* For each instance of testing set, there are three steps to get the predicting rating through standard voting. First, each predicting rating is computed in each decision tree of RF through standard voting. These predicting

ratings are discrete value. Second, the number of random trees is counted corresponding to each predicting rating. Third, the rating supported by the largest population of trees is used as the RF predicting value. This is given by

$$p = r_v^{\max}, \tag{14}$$

where v is count of random trees.

The following example illustrates the three steps.

(2) *Weighted Average.* For each instance of testing set, there are two steps through weighted average. The first step is the same as the first one of standard voting. Second, the weighted average of these predicting ratings is computed as the RF predicting value. This is given by

$$p = \frac{\sum_{i=1}^{r_{\max}} i * (\ln(v_i))}{\sum_{i=1}^{r_{\max}} (\ln(v_i))}, \tag{15}$$

where r_{\max} is the highest rating.

The following example illustrates the two steps.

Example 21. Based on Example 20 and Figure 2(a), the distribution of the leaf node is {0, 0, 0, 2, 0, 0} corresponding to the value of {student} if r_{\max} is 5. When an instance of testing set gets the distribution after traversing the random decision tree classifier, the predicting rating of the instance is 3 through standard voting.

Assume there are 10 trees of an RF. The number of random trees is 5 corresponding to the predicting rating 4. The number of random trees is 3 corresponding to the predicting rating 2. The number of random trees is 2 corresponding to the predicting rating 3.

The final predicting value of standard voting is 4, and then the final predicted value of weighted average is given by

$$p = \left\lfloor \frac{(5 * \ln(4) + 2 * \ln(3) + 3 * \ln(2))}{(\ln(4) + \ln(3) + \ln(2))} \right\rfloor = 5. \tag{16}$$

(3) *Distribution Aggregation Based Voting.* For each instance of testing set, there are three steps to get the predicting rating through distribution aggregation based voting. First, each predicting distribution is obtained through top-down search

of each decision tree of RF. Second, the cumulative vector is computed through accumulating the predicted distributions of all trees. Third, the final predicted value of the RF is computed based on the cumulative vector through standard voting.

(4) *Distribution Aggregation Based Average*. For each instance of testing set, there are three steps to get the predicting rating through distribution aggregation based average. The previous two steps of the algorithm are the same as distribution aggregation based voting. The third step is different between two algorithms. The final predicted value of the RF is computed based on the cumulative vector through average.

Example 22. Assume there are 3 trees of an RF. The predicted distribution of the first tree is {0, 0, 0, 3, 0, 0}. The predicted distribution of the second tree is {0, 0, 0, 1, 0, 4}. The predicted class distribution of the third tree is {0, 0, 0, 3, 1, 0}. The cumulative vector is {0, 0, 0, 7, 1, 4}.

The final predicted value of distribution aggregation based voting is 3, and then the final predicting value of distribution aggregation based average is given by

$$p = \left\lfloor \frac{(3 * 7 + 4 * 1 + 5 * 4)}{(7 + 1 + 4)} \right\rfloor = 4. \quad (17)$$

4. Experimental Results

In Section 3, we have designed five kinds of RFs and each kind has four predicting approaches. In this section, we finish a total of 20 forecast schemes. Each scheme is repeated 10 times with different random partitions into training and testing sets (i.e., $10 \times$ cross-validation).

We try to answer the following questions through experimentation.

- (1) How large is the size of RF when the precision in terms of MAE keeps stable?
- (2) Which is more precise, in terms of MAE, NU, NI, or DOCS?
- (3) Which is more precise, in terms of MAE, NU or NUAR?
- (4) Which is more precise, in terms of MAE, NI or NIAR?
- (5) Which is more precise, in terms of MAE, standard voting, weighted average, distribution aggregation based voting, or distribution aggregation based average?

4.1. Dataset. We experimented with a well-known MovieLens dataset (<http://www.movielens.org/>) assembled by the GroupLens project. It is widely used in recommender systems (see, e.g., [18]). The database schema is as follows:

- (i) user (userID, age, gender, and occupation),
- (ii) movie (movieID, release-year, and genre),
- (iii) rates (userID and movieID).

We use the version with 943 users and 1,682 movies. The original rate relation contains the rating of movies with 5 scales. The user age has 61 attributes as indicated by the data set. The user occupation has 21 attributes. Since there are 85 attributes of the movie release-year, the genre is a multivalued attribute. Therefore, we scale it to 18 boolean attributes, namely, action, adventure, animation, children, comedy, crime, documentary, drama, fantasy, FilmNoir, horror, musical, mystery, romance, scientific-fiction, thriller, war, and western. All users have watched at least one movie, and the dataset consists of approximately 100,000 movies ratings. But rating matrix is still sparse because no one has watched more than 45 percent of the total movies, and only the 20 percent users have watched more than 10 percent movies.

4.2. Results. The original dataset is partitioned into training set and testing set through cross-validation. The training set is 80% of the original one, and the testing set is 20% of the original one.

In order to know the size of the forest when the precision in terms of MAE keeps stable, the number of random trees defined by us is from 2 to 50. We undertake 20 sets of experiments to answer the questions raised at the beginning of the section one by one. Each experiment is repeated 10 times with different sampling of training and testing sets, and the average accuracy in terms of MAE [19] is computed. MAE has been used to evaluate recommender systems in several cases [20, 21].

Figure 4(a) compares four approaches of NI. MAE's range of four predicting approaches is between 0.88 and 0.91. Weighted average approach is the best in four ones. The precision of weighted average keeps stable when the size of the forest is greater than or equal to 20. However, the precision of three other approaches has kept stable.

Figure 4(b) compares four approaches of NU. MAE's range of four predicting approaches is between 0.92 and 0.99. Standard voting approach is the best in four ones. The precision of standard voting has kept stable. However, the precision of three other approaches keeps stable when the size of the forest reaches a certain value.

Figure 4(c) compares four approaches of DOCS. MAE's range of four predicting approaches is between 0.92 and 1.07. Standard voting approach is the best in four ones. The precision of standard voting has kept stable. However, the precision of three other approaches keeps stable when the size of the forest reaches a certain value.

Figure 4(d) compares four approaches of NIAR. MAE's range of four predicting approaches is between 0.88 and 0.91. Weighted average approach is the best in four ones. The precision of standard voting has kept stable. However, the precision of three other approaches keeps stable when the size of the forest reaches a certain value.

Figure 4(e) compares four approaches of NUAR. MAE's range of four predicting approaches is between 0.91 and 0.96. Distribution aggregation based voting approach is the best in four ones. The precision of standard voting has kept stable. However, the precision of three other approaches keeps stable when the size of the forest reaches a certain value.

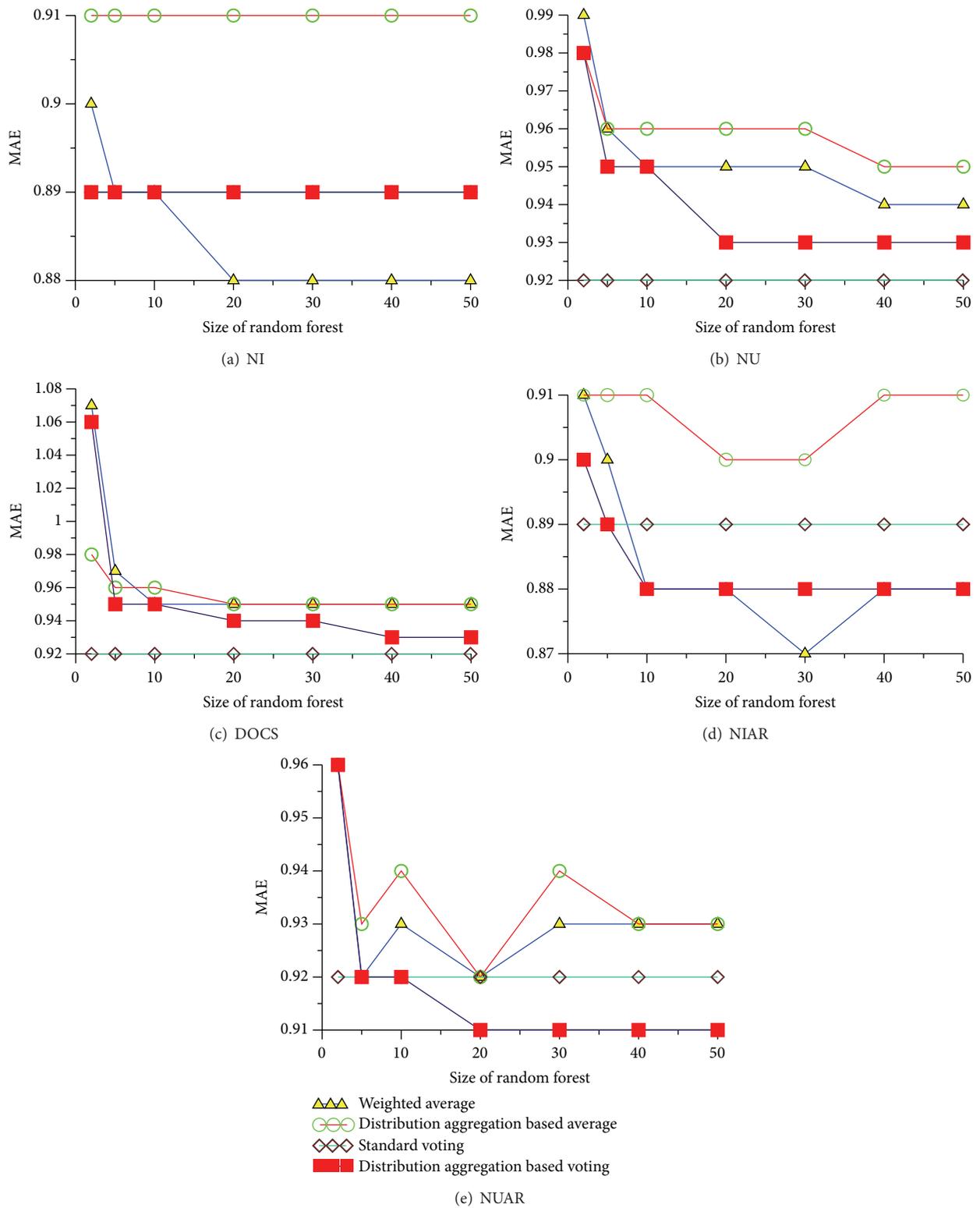


FIGURE 4: Comparisons of different prediction approaches.

In summary, we know from Figure 4 that (1) the precision in terms of MAE is stable on the whole when size of random forest is 2 to 20. Among these approaches, the precision of standard voting has kept stable; (2) NI approach is more precise than NU one. NU approach is more precise than DOCS one; (3) aggregated-based algorithms with AR are more slightly precise than without one. NIAR approach is almost the same as NI. The two approaches are the most precise in all prediction ones. One reason is that it is based on user rating history which forms his/her preference. They yield a MAE of 0.88 (on a five-point rating scale) on movie rating datasets.

5. Discussions

To the best of the authors' knowledge, little is known in previously published studies of aggregate in recommender systems. The work is related to previously published works on model-based RSs and group recommender.

Model-based RSs use the demographic, item information, and collection of ratings to create a model that generates the recommendations [8]. Model-building methods work by creating a model offline and then running the model online. Among the most widely used models, there are Bayesian classifiers [6], neural networks [22], and decision tree [23]. These models have been used to solve three kinds of cold-start problems [8]: new community, new item, and new user. The new community problem refers to the difficulty in obtaining a sufficient amount of data (ratings) for making reliable recommendations. The new item problem [8] arises because the new items entered in RS do not usually have initial ratings. The new user problem represents one of the great difficulties faced by the RS in operation. Since new user has not yet provided any rating in the RS, he/she cannot receive any personalized recommendations based on memory-based RS.

The random forest [13] is composed of many decision trees. Decision tree is a general computational model represented as a set of if-then rules [24]. Each tree is built based on a different set of training data and grown to the largest extent possible without pruning. Each splitting or decision node is acted by the best splitting attribute from randomly selected subset of the conditional attributes. To classify a new object, each tree in the forest gives a classification. The final classification of the object is determined by majority of votes among the classes decided by the forest of trees.

As defined in [10, 25], there are two strategies commonly adopted for generating group recommendations: the aggregated models and aggregated predictions. The former combines individual user models, that is, individual user profiles that capture the preferences of a group member into a group model from where items are recommended for the group are identified, whereas the latter generates predictions for individual group members and then aggregates the predictions to suggest items for the group. In this paper we present our proposed aggregated recommender, which predicts the rating of a group of users to a kind of items.

6. Conclusions

In this paper, we proposed random forest approaches to aggregated recommendation. By comparing several predicting approaches, we may draw the following conclusions: (1) MAE is stable on the whole when size of random forest is 5 to 20; (2) aggregated recommender can be used to solve NI, NU, and DOCS problems; (3) the precision of the NI approach is the highest; (4) the attribute of AR can improve the predicting accuracy; (5) the precision of DOCS approach maintains an acceptable level.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is in part supported by National Science Foundation of China under Grant nos. 61379089 and 61379049 and Scientific Research Starting Project of SWPU no. 2014QHZ025.

References

- [1] L. Duan, W. N. Street, and E. Xu, "Healthcare information systems: data mining methods in the creation of a clinical recommender system," *Enterprise Information Systems*, vol. 5, no. 2, pp. 169–181, 2011.
- [2] F. Min, Q. Hu, and W. Zhu, "Granular association rules with four subtypes," in *Proceeding of the IEEE International Conference on Granular Computing (GrC '12)*, pp. 353–358, Hangzhou, China, August 2012.
- [3] X. He, F. Min, and W. Zhu, "A comparative study of discretization approaches for granular association rule mining," in *Proceedings of the 26th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE '13)*, pp. 725–729, May 2013.
- [4] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen, "Collaborative filtering recommender systems," in *The Adaptive Web. Iem plus 0.5em minus 0.4em*, pp. 291–324, Springer, 2007.
- [5] F. Min and W. Zhu, "Mining top-k granular association rules for recommendation," in *PROceedings of the FSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS '13)*, pp. 1372–1376, IEEE, 2013.
- [6] M.-H. Park, J.-H. Hong, and S.-B. Cho, "Location-based recommendation system using bayesian user's preference model in mobile devices," in *Ubiquitous Intelligence and Computing*, pp. 1130–1139, 2007.
- [7] K. Zhou, S. Yang, and H. Zha, "Functional matrix factorizations for cold-start recommendation," in *Proceedings of the 34th International ACM Conference on Research and Development in Information Retrieval (SIGIR '11)*, pp. 315–324, July 2011.
- [8] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowledge-Based Systems*, vol. 46, pp. 109–132, 2013.
- [9] X. He, F. Min, and W. Zhu, "Parametric rough sets with application to granular association rule mining," *Mathematical Problems in Engineering*, vol. 2013, Article ID 461363, 13 pages, 2013.

- [10] J. Masthoff, "Group recommender systems: combining individual models," in *Recommender Systems Handbook*, pp. 677–702, Springer, New York, NY, US, 2011.
- [11] A. Jameson and B. Smyth, "Recommendation to groups," in *The Adaptive Web*, pp. 596–627, Springer, 2007.
- [12] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.
- [13] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [14] U. Shardanand and P. Maes, "Social information filtering: algorithms for automating "word of mouth"," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '95)*, pp. 210–217, ACM Press/Addison-Wesley, 1995.
- [15] J. S. Armstrong and T. S. Overton, "Estimating nonresponse bias in mail surveys," *Journal of Marketing Research*, vol. 14, no. 3, pp. 396–402, 1977.
- [16] C. P. Bonini, *Simulation of Information and Decision System in the Firm*, Prentice-Hall, New York, NY, USA, 1963.
- [17] F. Zhu and X. Zhang, "Impact of online consumer reviews on Sales: the moderating role of product and consumer characteristics," *Journal of Marketing*, vol. 74, no. 2, pp. 133–148, 2010.
- [18] P. Cremonesi, R. Turrin, and F. Airoldi, "Hybrid algorithms for recommending new items," in *Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems*, pp. 33–40, October 2011.
- [19] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock, "Methods and metrics for cold-start recommendations," in *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 253–260, 2002.
- [20] 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.
- [21] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, pp. 43–52, 1998.
- [22] B. J. Dahlen, J. A. Konstan, J. Herlocker, N. Good, A. Borchers, and J. Riedl, "Jump-starting movielens: user benefits of starting a collaborative filtering system with "dead-data"," TR 98-017, University of Minnesota, Minnesota, Minn, USA, 1998.
- [23] T. H. Roh, K. J. Oh, and I. Han, "The collaborative filtering recommendation based on SOM cluster-indexing CBR," *Expert Systems with Applications*, vol. 25, no. 3, pp. 413–423, 2003.
- [24] M. Sun, F. Li, J. Lee, K. Zhou, G. Lebanon, and H. Zha, "Learning multiple-question decision trees for cold-start recommendation," in *Proceedings of the 6th ACM International Conference on Web Search and Data Mining (WSDM '13)*, pp. 445–454, February 2013.
- [25] M. Montaner, B. López, and J. L. de la Rosa, "A taxonomy of recommender agents on the internet," *Artificial Intelligence Review*, vol. 19, no. 4, pp. 285–330, 2003.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

