

# Three-way recommender systems based on random forests

Heng-Ru Zhang, Fan Min\*

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



## ARTICLE INFO

### Article history:

Received 12 December 2014

Revised 26 March 2015

Accepted 25 June 2015

Available online 27 June 2015

### Keywords:

Cost sensitivity

Random forests

Recommender systems

Three-way decision

## ABSTRACT

Recommender systems attempt to guide users in decisions related to choosing items based on inferences about their personal opinions. Most existing systems implicitly assume the underlying classification is binary, that is, a candidate item is either recommended or not. Here we propose an alternate framework that integrates three-way decision and random forests to build recommender systems. First, we consider both misclassification cost and teacher cost. The former is paid for wrong recommender behaviors, while the latter is paid to actively consult the user for his or her preferences. With these costs, a three-way decision model is built, and rational settings for positive and negative threshold values  $\alpha^*$  and  $\beta^*$  are computed. We next construct a random forest to compute the probability  $P$  that a user will like an item. Finally,  $\alpha^*$ ,  $\beta^*$ , and  $P$  are used to determine the recommender's behavior. The performance of the recommender is evaluated on the basis of an average cost. Experimental results on the well-known MovieLens data set show that the  $(\alpha^*, \beta^*)$ -pair determined by three-way decision is optimal not only on the training set, but also on the testing set.

© 2015 Elsevier B.V. All rights reserved.

## 1. Introduction

Recommender systems (RSs) have been extensively studied to present items such as movies [20,21,40], music [15] to consumers. There are two main approaches to implementing RSs, memory based and model based [8]. Memory-based methods [13,51] employ the entire user-item database to generate a prediction. Model-based methods [22,41,71,72] use demographic and content information to create a model that generates recommendations. Demographic RSs [14,48] generate recommendations based on compiled user demographic profiles, whereas content-based systems [5,25,35,49] treat recommending as a user-specific classification problem and learn a classifier for the user's likes or dislikes based on product features. Most existing RSs implicitly assume that the underlying task is one of binary classification, and so consider only two actions (recommend or not) for candidate items.

In this paper, we propose a framework that integrates three-way decision and random forests to build recommender systems. Three-way decision is introduced to modify the recommender's user output [4], that is, for a given item, to recommend, not recommend, or consult the user actively for his or her preferences. A misclassification cost is incurred for wrong recommender behaviors, including recommending movies to users who dislike them or not recommending movies to users who would like them. Teacher cost is incurred for the consultation, e.g., distributing coupons to users.

Because two types of costs are considered, this approach essentially involves cost-sensitive learning [17,36,37,39,78]. There are various types of costs that can be considered [57,73]. In existing work, misclassification cost is the most widely addressed because classification is a major task of data mining (see, e.g., [28,60,77]). Test cost is often considered because data are not free (see, e.g., [58,61,69]). Delay cost is a major issue in the context of decision-theoretic rough sets [30,32,65,75]. Teacher cost is an essential consideration in active learning [6,43,54,55]. It is similar to test cost in that both “buy” data from the user. The major difference between the two lies in that test cost is incurred to obtain attribute values for data, whereas teacher cost is accrues to obtaining actual user decisions.

Three-way decision [33,34,67] is a methodological extension of decision-theoretic rough sets [31,59,62,64] that deals with situations where there are three possible decisions, namely, accept, reject, and wait and see [63,73,76]. It has been widely applied to situations such as filtering of spam email [76], determining biological matrices [18] and medical decision making [42]. The starting point is often a cost matrix with misclassification and delay costs. In this work, we consider teacher cost instead of delay cost. The purpose is also to compute two thresholds  $\alpha^*$  and  $\beta^*$ : if the probability that a user will like a movie is greater than  $\alpha^*$ , the movie is recommended, and if it is less than  $\beta^*$ , the movie is not recommended. Otherwise, we solicit his or her inclination, at the price of paying teacher cost.

To determine the recommender's behavior, we need to formulate the probability of the user's liking a movie. Decision-tree classification algorithms [50] are a natural approach. A decision tree

\* Corresponding author. Tel.: +86 135 4068 5200.  
E-mail address: [minfanphd@163.com](mailto:minfanphd@163.com) (F. Min).

describes graphically the decisions to be made, the events that may occur, and the outcomes associated with combinations of decisions and events [11]. However, a single decision tree can only take advantage of limited information of users and items, and hence no suggestion is generated for many new users and new items. To overcome this limitation, we employ a random forest instead for this purpose. A random forest is a collection of decision trees [10], and different trees in the forest may include different information, therefore the missing of prediction seldom happens.

Our approach is depicted in Fig. 1. It is divided into three parts: (1) An optimal threshold pair  $(\alpha^*, \beta^*)$  is computed based on the cost matrix, where  $\alpha^*$  determines the probability necessary to recommend an item and  $\beta^*$  determines that necessary to not recommend an item. (2) The user, item, and rating information are merged to construct an aggregate decision table in which the rating information serves as the decision attribute. A random forest is then built from the aggregate training set, where each leaf is assigned a distribution of discrete ratings. The probability of each testing object is computed from the random forest. (3) Finally the RS selects an action based on  $P$ ,  $\alpha^*$ , and  $\beta^*$ .

Experiments on the well-known MovieLens data set (<http://www.movielens.org/>) show that (1) the threshold pair  $(\alpha^*, \beta^*)$  determined with three-way decision is optimal not only on the training set, but also on the testing set and (2) our three-way RS outperforms the Pawlak, variable-precision, and probabilistic two-way models in terms of average cost.

The rest of the paper is organized as follows: Section 2 describes the three-way recommender problem and presents some background, including the rating system and cost-sensitive learning. An aggregate decision system is then built to mine the behavior of users on items. Section 3 discusses how to apply four rough-set models to our RSs. Section 4 constructs a random forest to predict the probability  $P$  and compute the average cost according to the three-way decision model. Section 5 presents experimental results from the MovieLens data set with the four rough-set models. How to set the threshold pair  $(\alpha, \beta)$  is discussed in detail. Concluding remarks are presented in Section 6.

## 2. Problem statement

Most existing RSs take a rating system as input, and the recommender's accuracy is regarded as a kind of evaluation metric. Our three-way RS considers misclassification and teacher costs, and through cost-sensitive learning, we build proper classifiers to find

a minimum average cost. For this purpose, the rating system is transformed into a decision system.

### 2.1. Rating system

We first revisit the definitions of information and rating systems [1,71].

**Definition 1.** An information system is a 2-tuple [71]

$$S = (U, A), \quad (1)$$

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  $x_i$  with respect to attribute  $a_j$  for  $i \in [1, n]$  and  $j \in [1, m]$ .

**Example 2.** Fig. 2 (a) shows an information system where  $U = \{u_1, u_2, u_3, u_4, u_5\}$  and  $A = \{\text{Age, Gender, Occupation}\}$ . UID is the user identification, which is usually not viewed as an attribute. Another example of an information system is given by Fig. 2(b).

**Definition 3.** Let  $U = \{u_1, u_2, \dots, u_n\}$  be the set of users of a RS and  $V = \{m_1, m_2, \dots, m_l\}$  be the set of all possible items that can be recommended to users. Then the rating function is defined as

(a) User table

UID	Age	Gender	Occupation
$u_1$	A25	F	technician
$u_2$	A50	M	writer
$u_3$	A18	F	student
$u_4$	A25	F	technician
$u_5$	A25	F	technician

(b) Item table

MID	Year	Comedy	Action	Drama
$m_1$	90s	0	0	1
$m_2$	90s	0	0	1
$m_3$	60s	1	0	0
$m_4$	70s,80s	1	1	1
$m_5$	60s	1	0	0

(c) Rating

	$m_1$	$m_2$	$m_3$	$m_4$	$m_5$
$u_1$	Y	N	0	Y	Y
$u_2$	N	Y	N	Y	0
$u_3$	N	N	0	N	N
$u_4$	Y	Y	N	Y	0
$u_5$	Y	N	0	N	Y

(c) Rating

Fig. 2. Rating system.

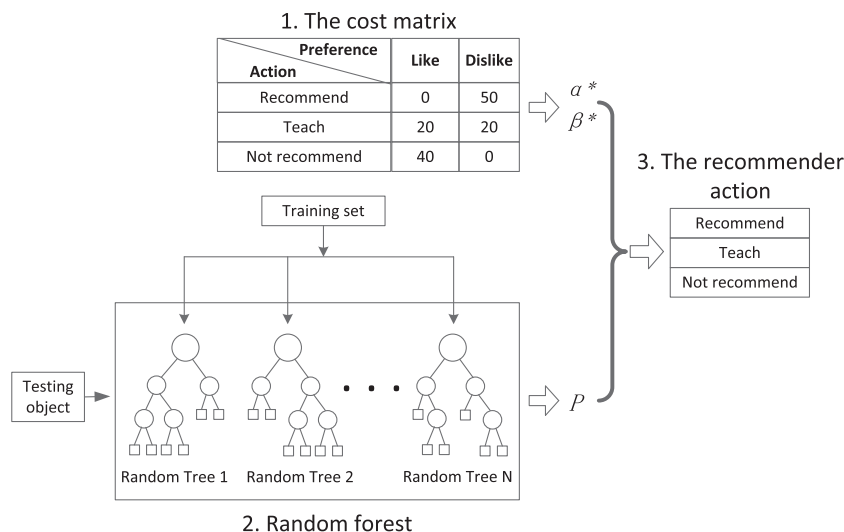


Fig. 1. Framework integrating three-way decision and a random forest.

$$R : U \times V \rightarrow V_R, \quad (2)$$

where  $V_R$  is the rating domain employed by the users to evaluate items.

In this study, we consider a situation in which  $V_R = \{0, Y, N\}$ .  $Y$  and  $N$  respectively indicate the user liked or disliked a movie, while  $0$  indicates that the user did not rate it. A rating function is often stored in the database as a table with two foreign keys for space savings. For the sake of illustration, here we represent it as an  $n \times l$  rating matrix.

With Definitions 1 and 3, the rating system is defined as follows:

**Definition 4.** A rating system is a 5-tuple [71]

$$RT = (U, A, V, B, R), \quad (3)$$

where  $(U, A)$  and  $(V, B)$  are two information systems and  $R : U \times V \rightarrow V_R$  is a rating function.

**Example 5.** Fig. 2 depicts a rating system in which Fig. 2(a) is the user table for  $(U, A)$ , Fig. 2(b) is the item table for  $(V, B)$ , and Fig. 2(c) is the rating table. User  $u_1$  likes movie  $m_1$  and dislikes movie  $m_2$ , while  $u_1$  never rated  $m_3$ .

## 2.2. Cost-sensitive learning

Most research in cost-sensitive learning [27,77] is an extension of machine learning approaches through the consideration of traditional misclassification cost. Hunt et al. [26] initially proposed two types of costs, (1) misclassification cost and (2) test cost. Yao [67] considered misclassification cost and delay cost. Turney [56] proposed a taxonomy of costs, including test cost, misclassification cost, delay cost, teacher cost, etc.

Test cost refers to the measurement cost of determining the value of an attribute for a given object. Min et al. introduced test costs for attribute reduction [36] and built a hierarchical model for test-cost-sensitive decision systems by considering common

test costs and sequential issues [38]. Zhao et al. [74] considered this attribute reduction problem for numeric data with measurement errors.

Misclassification cost is that incurred as a result of classification errors. AdaCost [16] uses the cost of misclassifications to update its training distribution on successive boosting rounds. An optimal Bayes classification rule that minimizes the expected misclassification cost has been constructed for various misclassification cost functions [9]. Pazzani et al. [47] introduced a reduced cost ordering algorithm to minimize classification costs. In real applications, how to evaluate misclassification cost is a very difficult problem. In some cases, relative values are assigned to represent misclassification cost [73].

Delay cost is incurred as a result of deferred decision making. Rough-set approximation or classification employs three pairwise disjoint regions that are defined by logical constructions, namely, positive, negative, and boundary regions [68]. Herbert and Yao [23,24] referred to the first two regions as representing immediate decisions and the third as containing delayed decisions.

Teacher cost comes from buying decision data from the user. Recommender action on the basis of paying teacher cost forms an active learning scenario. Active learning guides the acquisition of new knowledge suitable to updating recently rated or browsed information [29,53,55]. Turney [57] used inductive concept learning to create a taxonomy of the different cost types and discussed the expected cost of classifying a new object by itself versus the cost of asking a teacher to perform this task. Both teacher cost and test cost are forms of paying for data. The difference lies in that a teacher cost is paid to obtain a decision or result, while a test cost is paid to obtain an attribute value.

## 2.3. Data conversion

We next build an aggregate decision system to mine the behavior of users respect to individual items. A similar conversion has been discussed in [71]. We propose the concept of an aggregate decision system (ADS) as follows:

GID	Age	Gender	Occupation	Year	Comedy	Action	Drama	R
g <sub>1,1</sub>	A25	F	technician	90s	0	0	1	Y
g <sub>1,2</sub>	A25	F	technician	90s	0	0	1	N
g <sub>1,4</sub>	A25	F	technician	70s,80s	1	1	1	Y
g <sub>1,5</sub>	A25	F	technician	60s	1	0	0	Y
g <sub>2,1</sub>	A50	M	writer	90s	0	0	1	N
g <sub>2,2</sub>	A50	M	writer	90s	0	0	1	Y
g <sub>2,3</sub>	A50	M	writer	60s	1	0	0	N
g <sub>2,4</sub>	A50	M	writer	70s,80s	1	1	1	Y
g <sub>3,1</sub>	A18	F	student	90s	0	0	1	N
g <sub>3,2</sub>	A18	F	student	90s	0	0	1	N
g <sub>3,4</sub>	A18	F	student	70s,80s	1	1	1	N
g <sub>3,5</sub>	A18	F	student	60s	1	0	0	N
g <sub>4,1</sub>	A25	F	technician	90s	0	0	1	Y
g <sub>4,2</sub>	A25	F	technician	90s	0	0	1	Y
g <sub>4,3</sub>	A25	F	technician	60s	1	0	0	N
g <sub>4,4</sub>	A25	F	technician	70s,80s	1	1	1	Y
g <sub>5,1</sub>	A25	F	technician	90s	0	0	1	Y
g <sub>5,2</sub>	A25	F	technician	90s	0	0	1	N
g <sub>5,4</sub>	A25	F	technician	70s,80s	1	1	1	N
g <sub>5,5</sub>	A25	F	technician	60s	1	0	0	Y

GID	Age	Gender	Occupation	Year	Comedy	Action	Drama	L
mg <sub>1</sub>	A25	F	technician	90s	0	0	1	(3Y, 1N)
mg <sub>2</sub>	A25	F	technician	70s,80s	1	1	1	(2Y, 0N)
mg <sub>3</sub>	A50	M	writer	90s	0	0	1	(1Y, 1N)
mg <sub>4</sub>	A50	M	writer	60s	1	0	0	(0Y, 1N)
mg <sub>5</sub>	A50	M	writer	70s,80s	1	1	1	(1Y, 0N)
mg <sub>6</sub>	A18	F	student	90s	0	0	1	(0Y, 2N)
mg <sub>7</sub>	A18	F	student	70s,80s	1	1	1	(0Y, 1N)
mg <sub>8</sub>	A18	F	student	60s	1	0	0	(0Y, 1N)
mg <sub>9</sub>	A25	F	technician	60s	1	0	0	(1Y, 1N)

GID	Age	Gender	Occupation	Year	Comedy	Action	Drama	L
g <sub>1,2</sub>	A25	F	technician	90s	0	0	1	?
g <sub>1,5</sub>	A25	F	technician	60s	1	0	0	?
g <sub>4,2</sub>	A25	F	technician	90s	0	0	1	?
g <sub>5,4</sub>	A25	F	technician	70s,80s	1	1	1	?

Fig. 3. Aggregate decision system.

**Definition 6.** An ADS induced by a rating system  $RT = (U, A, V, B, R)$  is

$$DS(RT) = (G, C, d, \{E_a | a \in C \cup \{d\}, \{I_a | a \in C \cup \{d\}\}, \quad (4)$$

where  $G = \{(u_i, m_j) \in U \times V | R(u_i, m_j) \neq 0\}$ ,  $C = A \cup B$ , and  $\forall g_{i,j} = (u_i, m_j) \in G, d(g_{i,j}) = R(u_i, m_j)$ .

Because there may exist many identical objects in  $U$  and  $V$ , there are also many identical objects in  $G$ . We can merge these to save space and facilitate further analysis. Formally, an equivalence relation on  $G$  for the conditional attributes  $C$  can be defined as

$$IND(G, C) = \{(x, y) \in G^2 | \forall a \in C, I_a(x) = I_a(y)\}. \quad (5)$$

The equivalence class of  $x \in G$  on  $C$  is

$$[x] = [x]_{G,C} = \{y \in G | (x, y) \in IND(G, C)\}. \quad (6)$$

The partitioning of  $G$  induced by  $C$  is denoted

$$G/C = \{mg_1, mg_2, \dots, mg_e\}, \quad (7)$$

where  $\forall mg \in G/C$  is an equivalence class,  $\cup_{mg \in G/C} G = G$ , and  $\forall (mg_i, mg_j) \in G/C, mg_i \cap mg_j = \emptyset$ .

**Example 7.** Fig. 3(a) depicts an ADS with  $G = \{g_{1,1}, g_{1,2}, \dots, g_{5,4}, g_{5,5}\}$ ,  $C = \{\text{Age, Gender, Occupation, Year, Comedy, Action, Drama}\}$ , and  $d = \{R\}$ .

The equivalence class of  $g_{1,1}$  is  $[g_{1,1}] = \{g_{1,1}, g_{1,2}, g_{4,1}, g_{4,2}, g_{5,1}, g_{5,2}\}$ .

To test the performance of an RS, we need to divide  $DS(RT)$  into a training set and a testing set. This is done by randomly choosing a portion of  $DS(RT)$  as the training set and the remainder as the testing set. Formally, let  $G_r$  and  $G_t$  be the training set and the testing set respectively; we require  $G_r \cap G_t = \emptyset$  and  $G_r \cup G_t = G$ .

To facilitate the learning process, we merge the training set according to Eqs. (5)–(7). An example of a merged training set is shown in Fig. 3(b). Here the first object corresponds to  $3 + 1 = 4$  objects from the original set. Objects in the same equivalence class may have different decisions. Therefore, we use  $L$  instead of  $R$  to indicate the decision. Here  $L(mg_1) = (3Y, 1N)$ , indicating that there are three objects for decision  $Y$  and one for decision  $N$ .

For a merged object  $mg$ , the probability  $P$  that a user with the given demographic information will like a movie with the specified item information is denoted by  $P(mg)$ , which can easily be computed from  $L(mg)$ . For example,  $L(mg_1) = (3Y, 1N)$ , and therefore  $P(mg_1) = 3/(3 + 1) = 0.75$ . Formally,

$$P(mg) = |\{y \in mg | R(y) = Y\}| / |mg|. \quad (8)$$

### 3. Computational models

In this section, we discuss how to apply four rough-set models to our RSs. The models determine the action of the recommender according to the threshold pair  $(\alpha, \beta)$  and the probability  $P$  that a user will like a movie.

#### 3.1. Pawlak model

The Pawlak rough-set model [44] approximates a set by a pair of lower and upper approximations. The model has been used widely in instances such as attribute reduction [69], attribute-value reduction [45] and rule synthesis [7].

We apply the Pawlak model to the RS. For a set  $X \subseteq G$ , we construct the lower and upper approximations according to Eqs. (5) and (6):

$$\begin{aligned} \underline{\text{apr}}(X) &= \{x \in G | [x] \subseteq X\}, \\ \overline{\text{apr}}(X) &= \{x \in G | [x] \cap X \neq \emptyset\}. \end{aligned} \quad (9)$$

The nonempty set  $G$  is divided into three disjoint regions: (1) the lower approximation defines the positive region  $\text{POS}(X)$ , (2) the difference between the upper and lower approximations defines the boundary region  $\text{BND}(X)$ , and (3) the complement of the upper approximation defines the negative region  $\text{NEG}(X)$ . According to the lower and upper approximations, the positive, negative, and boundary regions of  $X$  are defined as [46]

$$\begin{aligned} \text{POS}(X) &= \underline{\text{apr}}(X), \\ \text{NEG}(X) &= \overline{\text{apr}}(X), \\ \text{BND}(X) &= \overline{\text{apr}}(X) - \underline{\text{apr}}(X). \end{aligned} \quad (10)$$

Furthermore, Pawlak and Skowron [45] suggested a rough membership function to redefine the two approximations, defined by

$$P(X|[x]) = |X \cap [x]| / |[x]|, \quad (11)$$

where  $|\cdot|$  indicates the cardinal number of objects in a set and  $P(X|[x])$  is the conditional probability of the classification.

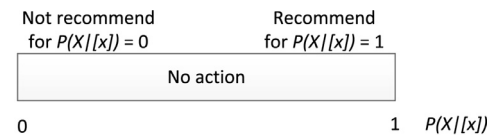
From the rough membership function, Pawlak approximations can be equivalently defined as follows:

$$\begin{aligned} \underline{\text{apr}}(X) &= \{x \in G | P(X|[x]) = 1\}, \\ \overline{\text{apr}}(X) &= \{x \in G | P(X|[x]) > 0\}. \end{aligned} \quad (12)$$

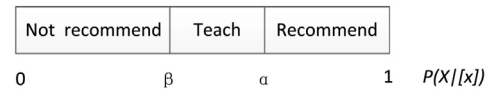
Based on Eq. (12), the Pawlak positive, negative, and boundary regions are

$$\begin{aligned} \text{POS}(X) &= \{x \in G | P(X|[x]) = 1\}, \\ \text{NEG}(X) &= \{x \in G | P(X|[x]) = 0\}, \\ \text{BND}(X) &= \{x \in G | 0 < P(X|[x]) < 1\}. \end{aligned} \quad (13)$$

In Fig. 4 (a), Pawlak's classical model takes no action in the boundary region. In this study, we consult the user actively to obtain his or her preference. For the Pawlak rough-set model as applied to our RS, the three regions' actions are shown in Table 1.



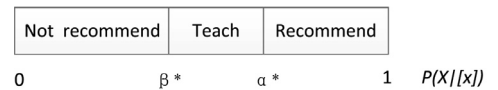
(a)



(b)



(c)



(d)

**Fig. 4.** Probabilistic decision models: (a) Pawlak, (b) variable precision, (c) two-way, (d) three-way.

**Table 1**  
Pawlak rough sets for recommender systems.

Region	Classical applications	Our RS
POS	Accept	Recommend
NEG	Reject	Not recommend
BND	No action	Teach

### 3.2. Variable-precision rough-set model

The variable-precision rough-set (VPRS) model [3,19,31,79] extends Pawlak rough sets to incorporate probabilistic information, substituting  $\alpha$  for 1 and  $\beta$  for 0 in Eq. (12). The condition is  $0 \leq \beta < \alpha \leq 1$ . The lower and upper approximations in VPRS are, respectively

$$\begin{aligned}\underline{\text{apr}}(X) &= \{x \in G | P(X|[x]) \geq \alpha\}, \\ \overline{\text{apr}}(X) &= \{x \in G | P(X|[x]) < \beta\}.\end{aligned}\quad (14)$$

Similarly, the positive, boundary, and negative regions with probabilistic variable precision can be defined as follows (see also Fig. 4(b)):

$$\begin{aligned}\text{POS}(X) &= \{x \in G | P(X|[x]) \geq \alpha\}, \\ \text{NEG}(X) &= \{x \in G | P(X|[x]) \leq \beta\}, \\ \text{BND}(X) &= \{x \in G | \beta < P(X|[x]) < \alpha\}.\end{aligned}\quad (15)$$

If we set  $\alpha + \beta = 1$  and  $\alpha > 0.5$ , this becomes symmetric VPRS. With  $\alpha + \beta \neq 1$ , it is asymmetric VPRS.

### 3.3. Probabilistic two-way decision model

The probabilistic two-way decision model [2] is based on inclusion of each equivalence class in either the positive or the negative region, as shown in Fig. 4(c). This model allows one to make useful decisions for all objects, i.e., all the objects are in a decision region.

If we set  $\gamma = \alpha = \beta$ , Eq. (15) can be rewritten as

$$\begin{aligned}\text{POS}(X) &= \{x \in G | P(X|[x]) \geq \gamma\}, \\ \text{NEG}(X) &= \{x \in G | P(X|[x]) < \gamma\}.\end{aligned}\quad (16)$$

This is a special case of the VPRS model with  $\alpha = \beta$ . If  $\gamma = 0$ , we simply recommend every movie. If  $\gamma = 1$ , we only recommend those movies with full probability  $P(X|[x]) = 1$ .

### 3.4. Three-way decision model

We apply the existing three-way decision model [66,67,70,76] to build our RSs. Our approach includes three parts: (1) two preferences and three actions are defined for our RSs; (2) a cost matrix is constructed according to the misclassification and teacher costs; (3) the threshold pair  $(\alpha^*, \beta^*)$  is computed based on the cost matrix. The three-way decision model is illustrated in Fig. 4(d).

With respect to a type of movie to be approximated, let  $X$  be a collection of movies liked by users and  $\tilde{X} = V - X$  be a collection of movies disliked by users. The recommended movies can be divided into the standard three regions, namely, positive (POS), boundary (BND), and negative (NEG).

The three recommender actions are given by  $R = \{r_P, r_B, r_N\}$  based on the three regions:  $r_P$  denotes “recommend” if a movie is classified into POS,  $r_B$  denotes “not recommend” if a movie is classified into BND, and  $r_N$  denotes “pay teacher cost” if a movie is classified into NEG.

The cost matrix for the actions in the different states is given by a  $3 \times 2$  matrix, shown in Table 2. If the movie belongs to  $X$ , then  $\lambda_{PP}$ ,  $\lambda_{BP}$ , and  $\lambda_{NP}$  denote the costs incurred for taking actions

**Table 2**  
A cost matrix.

Action	$X$	$\tilde{X}$
$r_P$	$\lambda_{PP}$	$\lambda_{PN}$
$r_B$	$\lambda_{BP}$	$\lambda_{BN}$
$r_N$	$\lambda_{NP}$	$\lambda_{NN}$

$r_P$ ,  $r_B$ , and  $r_N$ , respectively. If the movie belongs to  $\tilde{X}$ , then  $\lambda_{PN}$ ,  $\lambda_{BN}$ , and  $\lambda_{NN}$  are the costs incurred for taking these actions.

The expected costs associated with the different actions for a movie can be expressed as

$$\begin{aligned}T_P &= \lambda_{PP}P(X|[x]) + \lambda_{PN}P(\tilde{X}|[x]), \\ T_B &= \lambda_{BP}P(X|[x]) + \lambda_{BN}P(\tilde{X}|[x]), \\ T_N &= \lambda_{NP}P(X|[x]) + \lambda_{NN}P(\tilde{X}|[x]).\end{aligned}\quad (17)$$

The Bayesian decision procedure leads to the following minimum-risk decision rules:

- (P) If  $T_P \leq T_B$  and  $T_P \leq T_N$ , then decide  $x \in \text{POS}$ ;
- (B) If  $T_B \leq T_P$  and  $T_B \leq T_N$ , decide  $x \in \text{BND}$ ;
- (N) If  $T_N \leq T_P$  and  $T_N \leq T_B$ , decide  $x \in \text{NEG}$ .

To simplify the rules, some constraints should be added so that each movie is put into only one region. The first condition is

$$P(X|[x]) + P(\tilde{X}|[x]) = 1. \quad (18)$$

The cost of classifying a movie  $x$  in  $X$  into the positive region POS is no more than the cost of classifying  $x$  into the boundary region BND. Both costs are strictly less than the cost of classifying  $x$  into the negative region NEG. The reverse cost order is used for classifying a movie not in  $X$ . The second condition is

$$\begin{aligned}\lambda_{PP} &\leq \lambda_{BP} < \lambda_{NP}, \\ \lambda_{NN} &\leq \lambda_{BN} < \lambda_{PN}.\end{aligned}\quad (19)$$

We consider a special kind of cost function. The third condition is

$$\frac{\lambda_{NP} - \lambda_{BP}}{\lambda_{BN} - \lambda_{NN}} > \frac{\lambda_{BN} - \lambda_{NN}}{\lambda_{PN} - \lambda_{BN}}. \quad (20)$$

The final condition is

$$0 \leq \beta^* < \alpha^* \leq 1. \quad (21)$$

With these four conditions, we have the simplified decision rules as follows:

- (P1) If  $P(X|[x]) \geq \alpha^*$ , decide  $x \in \text{POS}$ ;
- (B1) If  $\beta^* < P(X|[x]) < \alpha^*$ , decide  $x \in \text{BND}$ ;
- (N1) If  $P(X|[x]) \leq \beta^*$ , decide  $x \in \text{NEG}$ .

Here  $\alpha^*$  and  $\beta^*$  are computed as follows:

$$\begin{aligned}\alpha^* &= \frac{\lambda_{PN} - \lambda_{BN}}{(\lambda_{PN} - \lambda_{BN}) + (\lambda_{BP} - \lambda_{PP})}, \\ \beta^* &= \frac{\lambda_{BN} - \lambda_{NN}}{(\lambda_{BN} - \lambda_{NN}) + (\lambda_{NP} - \lambda_{BP})}.\end{aligned}\quad (22)$$

### 3.5. Working example 1

Our RS takes different recommender actions with respect to objects with the same  $P$  when we use different probabilistic rough set models. We use the aggregate decision system depicted in Fig. 3 to illustrate this situation.



**Table 3**  
Different probabilistic rough set models.

$P$	Pawlak ( $\alpha = 1, \beta = 0$ )	VPRS ( $\alpha = 0.8, \beta = 0.6$ )	Two-way ( $\gamma = 0.6$ )	Three-way ( $\alpha^* = 0.6, \beta^* = 0.5$ )
0.75	Teach	Teach	Recommend	Recommend
0.5	Teach	Not recommend	Not recommend	Teach
1	Recommend	Recommend	Recommend	Recommend

$G_r$  and  $G_t$  are the training and the testing set, respectively. For any  $x \in G_r$  in Fig. 3(b) and any  $y \in G_t$  in Fig. 3(c), there are three steps to determine the recommender action. First, the probability  $P(y)$  is predicted. Before using the random-forest model for prediction, we match the conditional attributes between  $x$  and  $y$ . If the conditional attributes are the same,  $L(y)$  is assigned as the class attribute of  $x$ . Second, the probability  $P(y)$  is computed according to  $L(y)$  and Eq. (8). The probabilities are 0.75, 0.5, 0.75, and 1 for the four test objects illustrated in Fig. 3(c), respectively. Third, our RS makes different decisions, shown in Table 3, based on  $P(y)$  and different probabilistic rough set models.

Let  $MT_1 = \{v \in V | Drama(v) = 1\}$ . That is,  $MT_1$  is the type for dramatic movies. Similarly, let  $MT_2 = \{v \in V | Comedy(v) = 1\}$  and  $MT_3 = \{v \in V | (Drama(v) = 1) \wedge (Action(v) = 1) \wedge (Comedy(v) = 1)\}$ . Let  $UG_1 = \{u \in U | (Age(u) = 25) \wedge (Gender(u) = F) \wedge (Occupation(u) = technician)\}$ . That is,  $UG_1$  is the group of “(Age( $u$ ) = 25)  $\wedge$  (Gender( $u$ ) = F)  $\wedge$  (Occupation( $u$ ) = technician)” users. We take the following actions based on the different models:

1. With the Pawlak model, our RS pays teacher cost to  $UG_1$  for watching  $MT_1$  and  $MT_2$  but recommends  $MT_3$  to  $UG_1$ .
2. With the VPRS model, we assume that  $\alpha = 0.8$  and  $\beta = 0.6$ . Our RS pays teacher cost to  $UG_1$  for watching  $MT_1$ , does not recommend  $MT_2$ , and recommends  $MT_3$  to  $UG_1$ .
3. With the probabilistic two-way decision model, we assume that  $\gamma = 0.6$ . Our RS recommends  $MT_1$  and  $MT_3$  to  $UG_1$ , and does not recommend  $MT_2$ .
4. With the three-way decision model, the optimal thresholds are  $\alpha^* = 0.6$  and  $\beta^* = 0.5$ . Our RS recommends  $MT_1$  and  $MT_3$  to  $UG_1$  but pays teacher cost to  $UG_1$  for watching  $MT_2$ .

Let  $UG_2 = \{u \in U | (Age(u) = 25) \wedge (Gender(u) = F) \wedge (Occupation(u) = writer)\}$ . That is,  $UG_2$  is the group of “(Age( $u$ ) = 25)  $\wedge$  (Gender( $u$ ) = F)  $\wedge$  (Occupation( $u$ ) = writer)” users. For any  $ug_2 \in UG_2$ , if  $ug_2$  consults our RS, there is no matching object in Fig. 3(b) on the basis of matching conditional attributes. We can build a random forest to deal with this situation, as discussed in the next section.

#### 4. Algorithms

In the training set,  $P$  can be computed directly. However, in the testing set it should be predicted. In this section, we construct a random forest to predict  $P$  and compute the average cost according to the three-way decision model.

##### 4.1. Random-forest construction

In this subsection, the prediction of  $P$  based on a random forest is described in detail. First, random decision trees are built based on the aggregate training set. These trees are combined together to

form a random forest. Second, each random decision tree produces a predictive probability for each testing object. By averaging these predicted probabilities, we obtain the  $P$  of participation in the three-way decision.

Decision-tree learners build a tree by recursively partitioning training data. In the process of building a random decision tree, demographic and content information serve as conditional attributes and probability information serves as the decision attribute. For example, in Fig. 3(b) the conditional attributes are {Age, Gender, Occupation, Year, Comedy, Action, Drama} and the decision attribute is  $\{L\}$ . The probability  $P$  is easily computed according to  $L$  and Eq. (8). Each root-to-leaf path of the tree represents a rule for the probability that one group of users watches one type of movie. The process of building an unpruned random decision tree is described in Algorithm 1.

##### Algorithm 1. Build an unpruned random decision tree

**Input:** training set( $G_r$ ), condition attributes( $C$ ), probability( $P$ )

**Output:** An unpruned decision tree

**Method:** buildUnprunedRandomTree

```

1: if ( $C = \emptyset$  or  $G_r = \emptyset$ ) then
2:   return NULL;
3: end if
4:  $P_g = P(G_r)$ ;
5: currentNode = new buildUnprunedRandomTree( $G_r, C, P_g$ );
6: found = false;
7: for ( $a \in C$  is randomly selected) do
8:    $C = C - \{a\}$ ;
9:   if (informationGain( $a$ ) > 0) then
10:    found = true;
11:    break;
12:   end if
13: end for
14: if (not found) then
15:   currentNode.children = NULL;
16:   return currentNode;
17: end if
18: currentNode.splittingAttribute =  $a$ ;
19:  $|v_a|$  = the number of attribute values of  $a$ ;
20: currentNode.children = new
   buildUnprunedRandomTree[ $|v_a|$ ];
21: for ( $i = 1$  to  $|v_a|$ ) do
22:    $G_r^i = \{x \in G_r | a(x) = i\}$ ;
23:    $P_g^i = P(G_r^i)$ ;
24:   currentNode.children[i] = buildUnprunedRandomTree
     ( $G_r^i, C, P_g^i$ );
25: end for
26: return currentNode;

```

There are four steps to build the random decision tree:

- Step 1.** The probability  $P_g$  of the training set  $G_r$  is computed based on Eq. (8) (line 4).
- Step 2.** The current node is built based on the training set  $G_r$ , conditional attributes  $C$ , and probability  $P_g$  (line 5).
- Step 3.** An attribute  $a$  is randomly selected from the conditional ones and acts as the splitting attribute if its information gain is greater than 0 (lines 6–13).
- Step 4.** The original set is split into many subsets based on the values of the splitting attribute. The subsets are split recursively to construct subtrees. The probability  $P_g$  corresponding to subset  $G_r^i$  is computed according to Eq. (8). Each leaf is assigned the probability that one group of users likes one type of movie (lines 18–25).

A random decision tree only takes advantage of the limited information on users and items. Therefore, for many users and items a classical random decision tree may produce no prediction at all. In contrast, there is a probability  $P$  for each node in our tree. For many users and items, it may travel from the root to a branching node instead of a leaf node. In this way, the prediction is still rather imprecise. For example, the tree of Fig. 5(b) has no ( $\text{Age} = \text{A18}$ ) user information, so an imprecise prediction is produced. But the two trees of Fig. 5 can avoid this situation.

There are three steps to building the random forest: (1) aggregate training and testing sets are generated according to the different random-forest models; (2) condition attributes are randomized based on random seeds; (3)  $N$  random decision trees are built based on Algorithm 1.  $N$ , designated by the operator, is the size of the forest.

The random forest is used to predict  $P$  for test objects. Each tree uses a different random seed, so they may produce different results for the same test object. Consequently, all trees contribute to the final prediction. The process of predicting  $P$  based on random decision tree is described in Algorithm 2.

#### Algorithm 2. Predict based on random decision tree

---

**Input:** current node(currentNode), testing object( $y$ )  
**Output:** PR  
**Method:** predictBasedOnRandomTree

---

```

1:  $a = \text{currentNode.splittingAttribute};$ 
2:  $i = i + a(y);$ 
3: if (currentNode.children = NULL) then
4:   return currentNode. $P$ ;
5: else if (currentNode.children[i] = NULL) then
6:   return currentNode. $P$ ;
7: else
8:   return
     predictBasedOnRandomTree(currentNode.children[i],  $y$ );
9: end if

```

---

- Step 1.** We obtain the value of the splitting attribute  $a$  for test object  $y$  (lines 1–2).  
**Step 2.** If the current node has no children, the predictive value is the probability  $P$  of the current node (lines 3–4).  
**Step 3.** If the  $i$ th children of the current node are empty, the predictive value is the probability  $P$  of the current node (lines 5–6).  
**Step 4.** We obtain the predictive probability by recursively traversing the tree (line 8).

Finally, the average probability is equal to total probabilities divided by the size of the random forest:

$$P = \frac{1}{N} \sum_{i=1}^N \text{PR}_i, \quad (23)$$

where  $N$  is the size of the random forest.

#### 4.2. Average cost

In this subsection, we discuss the computation of the average cost of the training and testing sets. Throughout the study, the average-cost difference between training and testing sets is used to evaluate our RS.

$G_r$  and  $G_t$  are the training and testing sets, as shown in Fig. 3(b) and (c), respectively. Based on the three-way decision model, there are five steps to compute the average costs of  $G_r$  and  $G_t$ .

First, for any  $x \in G_r$ ,  $P(x)$  can be computed directly based on the aggregate decision table. However, for any  $y \in G_t$ ,  $P(y)$  should be predicted based on the random forest. Second, we decide that each object should belong to the positive, negative, or boundary region based on three-way decision. Third, we count the classification numbers of the different regions listed in Table 4. Fourth, the total cost of all regions is computed according to the cost matrix and the classification number of the respective region:

$$T_c = \lambda_{pp}N_{pp} + \lambda_{pn}N_{pn} + \lambda_{np}N_{np} + \lambda_{nn}N_{nn} + \lambda_{bp}N_{bp} + \lambda_{bn}N_{bn}. \quad (24)$$

Finally, the average cost is equal to the total cost divided by the number of training or testing objects. The average-cost computation for a training or testing set is described in Algorithm 3.

#### Algorithm 3. Average-cost computation

---

**Input:** aggregated set ( $E$ )  
**Output:** average cost ( $A_c$ )  
**Method:** costComputation

---

```

1: for ( $y \in E$ ) do
2:   if ( $E$  is training set) then
3:      $P(y)$  is computed directly based on Eq. (8);
4:   else
5:      $P(y)$  is predicted based on Algorithm 2;
6:   end if
7:    $R(y)$  is the original rating of  $y$ ;
8:   if ( $P(y) > \alpha^*$ ) then
9:     if ( $R(y) = \gamma$ ) then
10:       $N_{pp}++$ ; // classification number of positive–positive region
11:    else
12:       $N_{pn}++$ ; // classification number of positive–negative region
13:    end if
14:   else if ( $P(y) < \beta^*$ ) then
15:     if ( $R(y) = \gamma$ ) then
16:       $N_{np}++$ ; // classification number of negative–positive region
17:    else
18:       $N_{nn}++$ ; // classification number of negative–negative region
19:    end if
20:   else
21:     if ( $R(y) = \gamma$ ) then
22:       $N_{bp}++$ ; // classification number of boundary–positive region
23:    else
24:       $N_{bn}++$ ; // classification number of boundary–negative region
25:    end if
26:   end if
27: end for
28:  $T_c$  is computed based on Eq. (24);
29: return  $A_c = T_c/|E|$ ;

```

---

- Step 1.** Each object is classified into different regions based on  $P$ ,  $\alpha^*$ ,  $\beta^*$  and user's preference. This step corresponds to lines 8–26 of the algorithm.  
**Step 2.** The total cost is computed based on three-way decision. This step corresponds to line 27 of the algorithm.  
**Step 3.** The average cost is equal to the total cost divided by the number of aggregated objects. This step corresponds to line 29 of the algorithm.

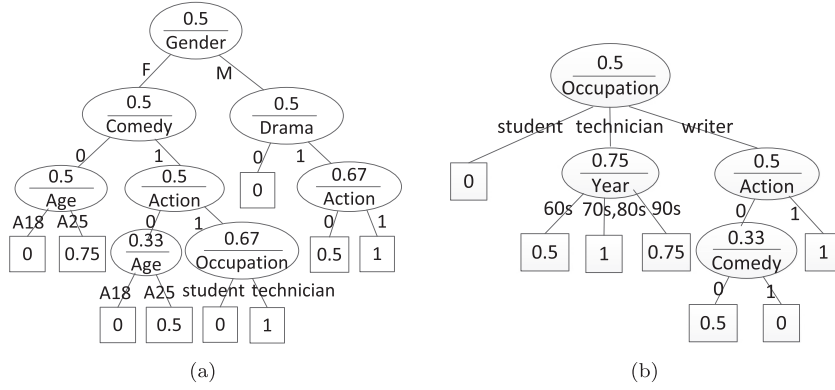


Fig. 5. Random trees.

**Table 4**  
Classification numbers.

	$\gamma$	$N$
Recommend	$N_{PP}$	$N_{PN}$
Teach	$N_{BP}$	$N_{BN}$
Not recommend	$N_{NP}$	$N_{NN}$

#### 4.3. Working example 2

In this subsection, we explain the cost computation for a training or testing object through a working example. First, we calculate the three-way threshold pair  $(\alpha^*, \beta^*)$  based on the cost matrix. Second, we compute the probability  $P(x)$  directly based on the training set or predict the probability  $P(y)$  based on the random forest. Third, we count the classification number based on the original rating,  $P$  and the pair  $(\alpha^*, \beta^*)$ . Finally, the expected cost is computed based on the classified counts and the cost matrix.

Based on the cost matrix of Fig. 1 and the three-way decision model, we calculate  $\alpha^*$  and  $\beta^*$  as

$$0 \times \alpha^* + 40(1 - \alpha^*) = 20\alpha^* + 20(1 - \alpha^*),$$

$$50\beta^* + 0 \times (1 - \beta^*) = 20\beta^* + 20(1 - \beta^*),$$

which implies that  $\alpha^* = 0.6$  and  $\beta^* = 0.5$ .

In the training set,  $P(x)$  can be computed directly. For example, we compute  $P(mg_1) = 1/(1 + 3) = 0.75$  in Fig. 3(b). Based on the original rating in Fig. 3(a), we find that  $N_{PP} = 3$  and  $N_{PN} = 1$ . Based on the cost matrix of Fig. 1, the expected cost is then computed as  $cost = \lambda_{PP}N_{PP} + \lambda_{PN}N_{PN} = 0 \times 4 + 50 \times 1 = 50$ .

In the testing set,  $P(y)$  should be predicted based on the random forest. Given the training set of Fig. 3(b), we construct two random trees as shown in Fig. 5.

In Section 3.5,  $ug_2$  consults our RS. For any  $ug_2 \in UG_2$ , there are three steps for our RS to decide whether or not recommend  $mt_2$  to  $ug_2$ . First, aggregated conditional attributes are constructed based on demographic and content information. Second, the user–movie combination (A25, F, writer, 90s, 1, 0, 0) does not match any object in the training set; therefore, we cannot arrive at the probability directly from the training set.  $P(y)$  should be predicted based on the random forest. Following the tree of Fig. 5(a), the probability is 0.5. Following the tree of Fig. 5(b), the probability is 0. Hence  $P(y) = \frac{1}{2}(0.5 + 0) = 0.25$ . Third, our RS makes one decision for  $mt_2$  to  $ug_2$  based on the three-way decision model. Because  $0.25 < \beta^*$ , our RS does not recommend  $mt_2$  to  $ug_2$ . Assume that the original rating of the object is  $\gamma$ , so we have  $N_{NP} = 1$ . Based on the cost matrix of Fig. 1, the expected cost of this decision is computed as  $cost = \lambda_{NP}N_{NP} = 40 \times 1 = 40$ .

## 5. Experiments

In this section, we report extensive computational tests designed to address the following questions:

1. Does the three-way decision model outperform other rough-set models?
2. Can random forests be replaced with other classifiers such as  $k$  nearest neighbors ( $kNN$ )?

Question 1 aims at a comparison among different probabilistic rough set models. Each model has its own method for setting the threshold pairs  $(\alpha, \beta)$ . The optimal threshold pair The setting is  $\alpha = 1$  and  $\beta = 0$  in the Pawlak rough-set model. The setting is  $\gamma = \alpha = \beta$  in the probabilistic two-way model. The setting is  $0 \leq \beta < \alpha \leq 1$  in the VPRS model. Question 2 examines the performance comparison using different classifiers to predict the probability  $P$  in our framework. We specifically contrast the performance between the  $kNN$  and random-forest classifiers.

### 5.1. Data set

We employ the well-known MovieLens data set, which is widely used to test recommender systems (see, e.g., [12,52]). The database schema is as follows:

- User (userID, age, gender, occupation)
- Movie (movieID, release-year, genre)
- Rating (userID, movieID)

We use the version with 943 users and 1682 movies. The original rating relation contains movie ratings with 5 gradations, while we only consider whether a user has rated a movie. All users have watched at least one movie, and the data set consists of approximately 100,000 ratings. However, the rating matrix is still sparse, because no one has watched more than 45% of the total movies, and only 20% of users have watched more than 10% of the movies.

### 5.2. Experimental design

We designed three sets of experiments to answer the questions raised at the beginning of this section.

*Exp1.* With the constraint  $\beta \leq \alpha$ , we compute each average cost as  $\alpha$  and  $\beta$  change based on three-way decision. According to the cost matrix, we compute the optimal  $\alpha^*$  and  $\beta^*$  of three-way decision and then validate whether  $\alpha^*$  and  $\beta^*$  are optimal on the training/testing sets.



*Exp2.* We examine several comparative experiments among four rough-set models. First, we consider the teacher cost for the constraint  $0 < P < 1$  in the Pawlak rough-set model and then compare the average cost of the Pawlak and the three-way decision models. Second, we compare the average cost of the probabilistic two-way

decision and VPRS models. Third, we compare the average cost of the VPRS and three-way decision models.

*Exp3.* We compare the average cost between the  $k$ NN and random-forest classifiers with an optimally set threshold pair  $(\alpha^*, \beta^*)$ .

**Table 5**

Average cost on the training set.

$\alpha$	$\beta$										
	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
0.0	22.34 ± 0.04										
0.1	20.76 ± 0.04	19.72 ± 0.05									
0.2	20.25 ± 0.04	19.21 ± 0.05	18.89 ± 0.05								
0.3	19.43 ± 0.06	18.38 ± 0.06	18.07 ± 0.06	17.59 ± 0.09							
0.4	18.32 ± 0.02	17.28 ± 0.03	16.96 ± 0.02	16.49 ± 0.04	15.96 ± 0.04						
0.5	17.20 ± 0.03	16.15 ± 0.04	15.84 ± 0.04	15.37 ± 0.03	14.84 ± 0.03	14.5 ± 0.03					
0.6	16.58 ± 0.02	15.53 ± 0.02	15.22 ± 0.02	14.75 ± 0.03	14.22 ± 0.02	<b>13.88 ± 0.02</b>	14.22 ± 0.03				
0.7	16.99 ± 0.02	15.94 ± 0.02	15.63 ± 0.02	15.16 ± 0.04	14.63 ± 0.03	14.29 ± 0.02	14.63 ± 0.03	15.73 ± 0.04			
0.8	17.74 ± 0.03	16.70 ± 0.03	16.38 ± 0.03	15.91 ± 0.04	15.38 ± 0.02	15.05 ± 0.02	15.38 ± 0.03	16.48 ± 0.03	17.51 ± 0.04		
0.9	18.47 ± 0.02	17.42 ± 0.03	17.10 ± 0.03	16.63 ± 0.04	16.10 ± 0.03	15.77 ± 0.02	16.10 ± 0.02	17.20 ± 0.04	18.23 ± 0.04	19.05 ± 0.04	
1.0	18.65 ± 0.02	17.60 ± 0.02	17.29 ± 0.02	16.81 ± 0.03	16.28 ± 0.024	15.95 ± 0.01	16.29 ± 0.02	17.39 ± 0.03	18.41 ± 0.03	19.24 ± 0.03	19.43 ± 0.02

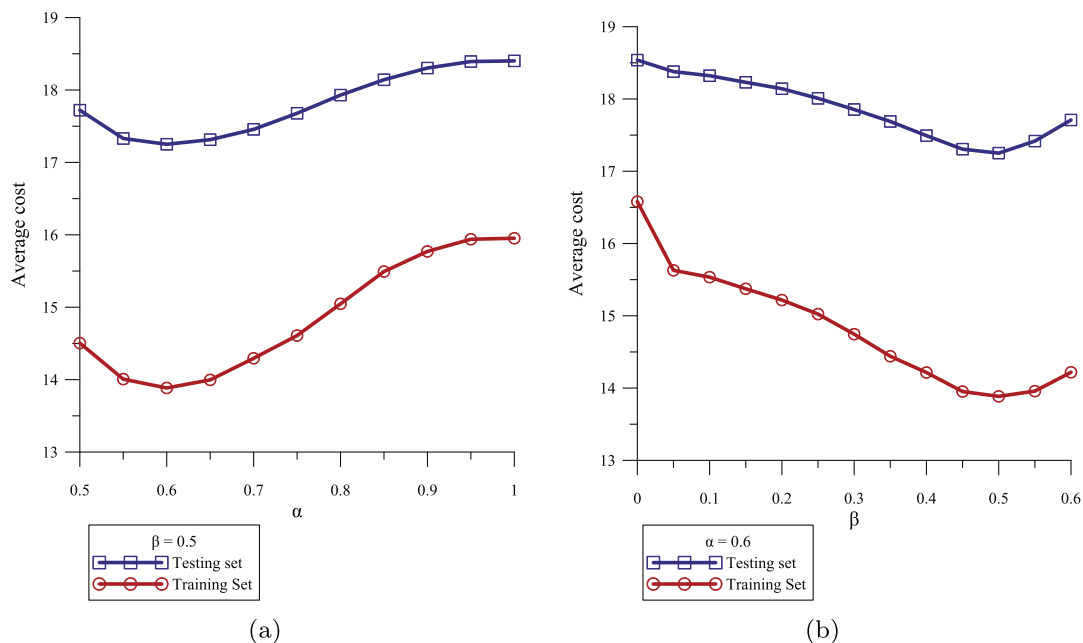
The bold values correspond to the setting  $(\alpha^*, \beta^*)$ .

**Table 6**

Average cost on the testing set.

$\alpha$	$\beta$										
	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
0.0	22.39 ± 0.18										
0.1	21.86 ± 0.19	21.65 ± 0.19									
0.2	21.54 ± 0.19	21.33 ± 0.19	21.15 ± 0.19								
0.3	20.96 ± 0.21	20.74 ± 0.21	20.56 ± 0.21	20.27 ± 0.23							
0.4	20.03 ± 0.21	19.81 ± 0.21	19.63 ± 0.21	19.35 ± 0.22	18.98 ± 0.23						
0.5	19.01 ± 0.15	18.79 ± 0.15	18.61 ± 0.15	18.33 ± 0.17	17.96 ± 0.17	17.72 ± 0.25					
0.6	18.54 ± 0.11	18.32 ± 0.12	18.14 ± 0.11	17.85 ± 0.13	17.49 ± 0.13	<b>17.25 ± 0.12</b>	17.71 ± 0.14				
0.7	18.74 ± 0.06	18.53 ± 0.07	18.35 ± 0.06	18.06 ± 0.06	17.70 ± 0.06	17.46 ± 0.08	17.91 ± 0.13	18.86 ± 0.17			
0.8	19.22 ± 0.06	19.00 ± 0.06	18.82 ± 0.06	18.53 ± 0.06	18.17 ± 0.06	17.93 ± 0.06	18.39 ± 0.11	19.33 ± 0.15	20.13 ± 0.14		
0.9	19.59 ± 0.04	19.37 ± 0.05	19.19 ± 0.04	18.91 ± 0.05	18.54 ± 0.05	18.30 ± 0.06	18.76 ± 0.11	19.71 ± 0.15	20.51 ± 0.14	21.05 ± 0.15	
1.0	19.69 ± 0.04	19.47 ± 0.05	19.29 ± 0.04	19.01 ± 0.05	18.64 ± 0.05	18.40 ± 0.06	18.86 ± 0.11	19.81 ± 0.15	20.61 ± 0.14	21.15 ± 0.14	21.27 ± 0.14

The bold values correspond to the setting  $(\alpha^*, \beta^*)$ .

**Fig. 6.** Average cost between training and testing sets: (a)  $\alpha$ , (b)  $\beta$ .

### 5.3. Results

The original set is repeated 10 times to divide into the training set and the testing set with different random partitionings (i.e.,  $10 \times$  cross-validation). We choose 50% of the original set as the training set and the other half as the testing set. Based on Section 4.3, we obtain the optimal threshold pair of three-way decision as ( $\alpha^* = 0.6$ ,  $\beta^* = 0.5$ ).

Tables 5 and 6 show the average cost changes as  $\alpha$  and  $\beta$  change with a step length of 0.1. Because of the constraint  $\beta \leq \alpha$ , the average-cost table is an oblique triangular matrix. Based on the 10-fold experiment, we compute the average cost and standard deviation for each  $\alpha$  and  $\beta$ . The optimal threshold pair is ( $\alpha^* = 0.6$ ,  $\beta^* = 0.5$ ) for both the training set and the testing set.

Fig. 6(a) compares the average cost between training and testing set of the VPRS model as  $\alpha$  increases. We set the teacher cost at 20. Thus the average cost of training and testing sets, obviously, are less than the teacher cost. When  $\beta = 0.5$ ,  $\alpha$  ranges from 0.5 to 1 with step length 0.05, the trend of the two curves is same and the optimal  $\alpha^*$  is 0.6.

Fig. 6(b) compares average cost between training and testing sets with the VPRS model as  $\beta$  increases. When  $\alpha = 0.6$ ,  $\beta$  ranges from 0 to 0.6 with step length 0.05, the trend of the two curves is same, and the optimal  $\beta^*$  is 0.5.

We calculate that the best predictive accuracy of the training set is 75.2% based on the random forest. Because of the high conflict rate of the aggregate decision table, accurate forecasting is unavailable. The average cost on the testing set is 17.25, or 24.3%

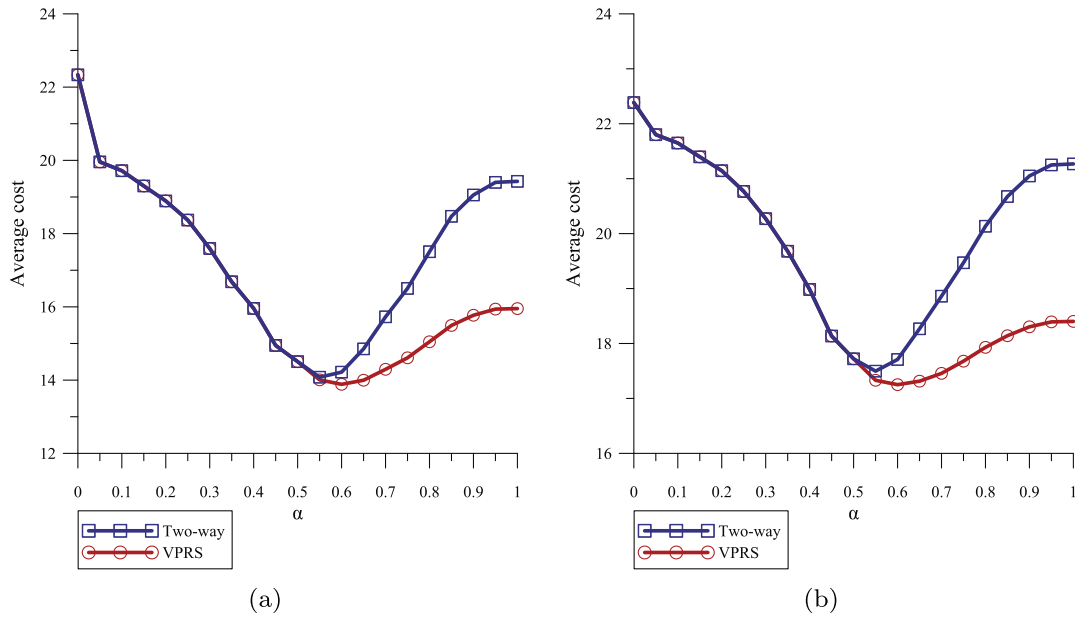


Fig. 7. Average cost between the probabilistic two-way decision and VPRS models: (a) training set, (b) testing set.

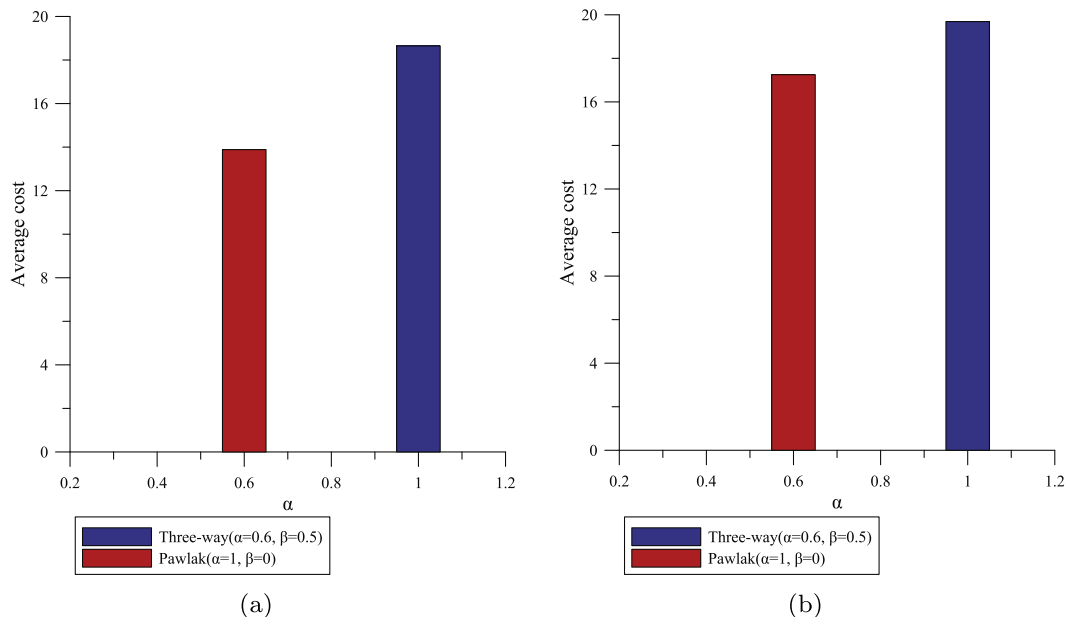


Fig. 8. Average cost between the Pawlak and three-way decision models: (a) training set, (b) testing set.

**Table 7**Average cost of random forest vs. kNN-based three-way decision ( $\alpha^* = 0.6$ ,  $\beta^* = 0.5$ ).

	Training set	Testing set
Random forest	13.88	17.25
kNN	13.81	17.19

more than the training set. Fortunately, we still observe that the trend of the average cost is the same for the training and testing sets based on Fig. 6.

In Fig. 7, the condition is  $\gamma = \alpha = \beta$  in the probabilistic two-way decision model. We draw an average-cost curve of the two-way model as  $\alpha$  changes from 0 to 1. For the VPRS model,  $\alpha$  ranges from 0 to 1 with step length 0.05, that is,  $\alpha = 0, 0.05, 0.1, \dots, 1$ . For any  $\alpha$ , we first compute all average costs as  $\beta$  changes from 0 to  $\alpha$  and find the minimum value. Then we draw an average-cost curve for the VPRS model formed by the minimum values as  $\alpha$  changes from 0 to 1.

Fig. 7(a) compares the average cost of the training set between the probabilistic two-way decision and VPRS models. When  $0 \leq \alpha \leq 0.5$ , there is little difference between the two. The average-cost gap between two models increases as  $\alpha$  changes from 0.5 to 1. That is, the VPRS model performs better than the two-way decision model when  $0.5 \leq \alpha \leq 1$ . Fig. 7(b) compares average cost of the testing set between the probabilistic two-way decision and VPRS models. The result is similar to that for the training set.

In Fig. 8, the threshold pair for the three-way decision model is assigned as ( $\alpha^* = 0.6$ ,  $\beta^* = 0.5$ ) and as ( $\alpha = 1$ ,  $\beta = 0$ ) for the Pawlak model. Fig. 8(a) compares average cost of the training set for the two models. For most objects, the Pawlak model will pay the teacher cost. The average cost of the Pawlak model pays more than the three-way decision model, i.e., the three-way decision model outperforms the two-way decision model. Fig. 8(b) compares average cost of testing set between the three-way decision and Pawlak rough set models. The result is similar to that for the training set.

Table 7 contrasts the performance between random-forest-based and kNN-based three-way. The average-cost gap between two classifiers is small not only on the training set, but also on the testing set.

Based on the above analysis, we can answer the questions raised at the beginning of this section:

1. The three-way decision model obtains a threshold pair ( $\alpha^*$ ,  $\beta^*$ ) that is optimal not only on the training set, but also on the testing set, thereby outperforming other rough-set models.
2. The kNN and random-forest models have approximately the same performance in terms of average cost, and therefore, the framework is meaningful. Because the number of instances in an aggregate decision table is often large, kNN is rather slow in the testing/recommending stage. For example, MovieLens consists of approximately 100,000 movie ratings, with a training set of 50%, and the time consumption of forecasting for each object is approximately two orders of magnitude times that of random-forest, making the latter a good choice.

## 6. Conclusions

We have proposed a framework integrating three-way decision and random forests to build recommender systems. In addition to recommending or not recommending a movie to a user, the third option of paying a teacher cost is considered for undecided cases. According to our experiments on MovieLens data set, the threshold pair ( $\alpha^*$ ,  $\beta^*$ ) determined with three-way decision is optimal not only on the training set, but also on testing set. Compared with the Pawlak, variable-precision, and probabilistic two-way rough

set models, the three-way decision model has lowest average cost. In the future, we will apply memory-based approaches to this problem.

## Acknowledgement

This work was supported in part by the National Natural Science Foundation of China (Grant 61379089).

## References

- [1] J.S. Armstrong, T.S. Overton, Estimating nonresponse bias in mail surveys, *J. Marketing Res. (JMR)* 14 (3) (1977).
- [2] N. Azam, J.T. Yao, Multiple criteria decision analysis with game-theoretic rough sets, *Rough Sets and Knowledge Technology*, Springer, 2012, pp. 399–408.
- [3] N. Azam, J.T. Yao, Analyzing uncertainties of probabilistic rough set regions with game-theoretic rough sets, *Int. J. Approx. Reason.* 55 (1) (2014) 142–155.
- [4] N. Azam, J.T. Yao, Game-theoretic rough sets for recommender systems, *Knowl.-Based Syst.* 72 (2014) 96–107.
- [5] M. Balabanović, Y. Shoham, Fab: content-based, collaborative recommendation, *Commun. ACM* 40 (3) (1997) 66–72.
- [6] Y. Baram, R. El-Yaniv, K. Luz, Online choice of active learning algorithms, *J. Mach. Learn. Res.* 5 (2004) 255–291.
- [7] J.G. Bazan, H.S. Nguyen, S.H. Nguyen, P. Synak, J. Wróblewski, *Rough set algorithms in classification problem*, *Rough Set Methods and Applications*, Springer, 2000, pp. 49–88.
- [8] J. Bobadilla, F. Ortega, A. Hernando, A. Gutiérrez, *Recommender systems survey*, *Knowl.-Based Syst.* 46 (2013) 109–132.
- [9] N. Boutros, H. Nasrallah, R. Leighty, M. Torello, P. Tuetting, S. Olson, Auditory evoked potentials, clinical vs. research applications, *Psychiat. Res.* 69 (2) (1997) 183–195.
- [10] L. Breiman, Random forests, *Mach. Learn.* 45 (1) (2001) 5–32.
- [11] Y.-H. Cho, J.-K. Kim, S.-H. Kim, A personalized recommender system based on web usage mining and decision tree induction, *Expert Syst. Appl.* 23 (3) (2002) 329–342.
- [12] P. Cremonesi, R. Turrin, F. Airolidi, Hybrid algorithms for recommending new items, in: *Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems*, 2011.
- [13] J. Delgado, N. Ishii, Memory-based weighted majority prediction, in: *SIGIR Workshop Recomm. Syst.* Citeseer, 1999.
- [14] H. Drachler, H.G. Hummel, R. Koper, Personal recommender systems for learners in lifelong learning networks: the requirements, techniques and model, *Int. J. Learn. Technol.* 3 (4) (2008) 404–423.
- [15] L. Duan, W.N. Street, E. Xu, Healthcare information systems: data mining methods in the creation of a clinical recommender system, *Enterprise Inf. Syst.* 5 (2) (2011) 169–181.
- [16] W. Fan, S. Stolfo, J. Zhang, P. Chan, Adacost: misclassification cost-sensitive boosting, in: *Proceedings of the 16th International Conference on Machine Learning*, 1999.
- [17] G. Fumera, F. Roli, Cost-sensitive learning in support vector machines, in: *Proceedings of VIII Convegno Associazione Italiana per L'Intelligenza Artificiale*, 2002.
- [18] I. García, L. Sarabia, M.C. Ortiz, J.M. Aldama, Three-way models and detection capability of a gas chromatography–mass spectrometry method for the determination of clenbuterol in several biological matrices: the 2002/657/EC European decision, *Anal. Chim. Acta* 515 (1) (2004) 55–63.
- [19] J.W. Grzymala-Busse, P.G. Clark, M. Kuehnhausen, Generalized probabilistic approximations of incomplete data, *Int. J. Approx. Reason.* 55 (1) (2014) 180–196.
- [20] X. He, F. Min, W. Zhu, A comparative study of discretization approaches for granular association rule mining, in: *Proceedings of the 2013 Canadian Conference on Electrical and Computer Engineering*, 2013.
- [21] X. He, F. Min, W. Zhu, Parametric rough sets with application to granular association rule mining, *Math. Probl. Eng.* 2013 (2013) 1–13.
- [22] X. He, F. Min, W. Zhu, Comparison of discretization approaches for granular association rule mining, *Can. J. Electr. Comput. Eng.* 37 (3) (2014) 157–167.
- [23] J.P. Herbert, J.T. Yao, Game-theoretic risk analysis in decision-theoretic rough sets, *Rough Sets and Knowledge Technology*, Springer, 2008, pp. 132–139.
- [24] J.P. Herbert, J.T. Yao, Criteria for choosing a rough set model, *Comput. Math. Appl.* 57 (6) (2009) 908–918.
- [25] Y. Ho, S. Fong, Z. Yan, A hybrid ga-based collaborative filtering model for online recommenders, in: *ICE-B*, 2007.
- [26] E.B. Hunt, J. Marin, P.J. Stone (Eds.), *Experiments in Induction*, Academic Press, New York, 1966.
- [27] X.-Y. Jia, W.-H. Liao, Z.-M. Tang, L. Shang, Minimum cost attribute reduction in decision-theoretic rough set models, *Inf. Sci.* 219 (2013) 151–167.
- [28] R. Kohavi, J.R. Quinlan, *Data mining tasks and methods: classification: decision-tree discovery*, *Handbook of Data Mining and Knowledge Discovery*, Oxford University Press, Inc., 2002.
- [29] A. Krogh, J. Vedelsby, et al., Neural network ensembles, cross validation, and active learning, *Adv. Neural Inf. Process. Syst.* (1995) 231–238.

- [30] H.-X. Li, X.-Z. Zhou, Risk decision making based on decision-theoretic rough set: a three-way view decision model, *Int. J. Comput. Intell. Syst.* 4 (1) (2011) 1–11.
- [31] T.-J. Li, X.-P. Yang, An axiomatic characterization of probabilistic rough sets, *Int. J. Approx. Reason.* 55 (1) (2014) 130–141.
- [32] D. Liu, T.-R. Li, H.-X. Li, A multiple-category classification approach with decision-theoretic rough sets, *Fundam. Inform.* 155 (2–3) (2012) 173–188.
- [33] D. Liu, T.-R. Li, D.-C. Liang, Three-way government decision analysis with decision-theoretic rough sets, *Int. J. Uncertain. Fuzz. Knowl.-Based Syst.* 20 (2012) 119–132.
- [34] D. Liu, Y.Y. Yao, T.-R. Li, Three-way investment decisions with decision-theoretic rough sets, *Int. J. Comput. Intell. Syst.* 4 (2011) 66–74.
- [35] P. Lops, M. De Gemmis, G. Semeraro, Content-based recommender systems: state of the art and trends, *Recommender Systems Handbook*, Springer, 2011, pp. 73–105.
- [36] F. Min, H.-P. He, Y.-H. Qian, W. Zhu, Test-cost-sensitive attribute reduction, *Inf. Sci.* 181 (2011) 4928–4942.
- [37] F. Min, Q.-H. Hu, W. Zhu, Feature selection with test cost constraint, *Int. J. Approx. Reason.* 55 (1) (2014) 167–179.
- [38] F. Min, Q.-H. Liu, A hierarchical model for test-cost-sensitive decision systems, *Inf. Sci.* 179 (2009) 2442–2452.
- [39] F. Min, W. Zhu, Optimal sub-reducts with test cost constraint, *Rough Sets Knowl. Technol.* (2011) 57–62.
- [40] F. Min, W. Zhu, Mining significant granular association rules for diverse recommendation, *Rough Sets and Current Trends in Computing*, Springer, 2014.
- [41] B. Mobasher, R. Burke, J.J. Sandvig, Model-based collaborative filtering as a defense against profile injection attacks, in: *AAAI*, vol. 6, 2006.
- [42] D. Mossman, Three-way rocs, *Med. Decis. Making* 19 (1) (1999) 78–89.
- [43] F. Olsson, A Literature Survey of Active Machine Learning in the Context of Natural Language Processing.
- [44] Z. Pawlak, Rough sets, *Int. J. Comput. Inf. Sci.* 11 (5) (1982) 341–356.
- [45] Z. Pawlak, A. Skowron, Rough membership functions, *Advances in the Dempster-Shafer theory of evidence*, John Wiley & Sons, Inc., 1994.
- [46] Z. Pawlak, et al., *Rough Sets: Theoretical Aspects of Reasoning About Data*, Kluwer Academic Publishers, Dordrecht & Boston, 1991.
- [47] M.J. Pazzani, C.J. Merz, P.M. Murphy, K. Ali, T. Hume, C. Brunk, Reducing misclassification cost, in: *Proceedings of the 11th International Conference of Machine Learning (ICML)*, Morgan Kaufmann, 1994.
- [48] M.J. Pazzani, A framework for collaborative, content-based and demographic filtering, *Artif. Intell. Rev.* 13 (5–6) (1999) 393–408.
- [49] M.J. Pazzani, D. Billsus, Content-based recommendation systems, in: *The adaptive web*, 2007, pp. 325–341.
- [50] J.R. Quinlan, Induction of decision trees, *Mach. Learn.* 1 (1) (1986) 81–106.
- [51] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Item-based collaborative filtering recommendation algorithms, in: *Proceedings of the 10th International Conference on World Wide Web*, ACM, 2001.
- [52] A.I. Schein, A. Popescul, L.H. Ungar, 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*, 2002.
- [53] B. Settles, Active learning literature survey, *Univ. Wisc. Madison* 52 (2010) 55–66.
- [54] J.-H. Su, B.-W. Wang, C.-Y. Hsiao, V.S. Tseng, Personalized rough-set-based recommendation by integrating multiple contents and collaborative information, *Inf. Sci.* 180 (1) (2010) 113–131.
- [55] S. Tong, D. Koller, Support vector machine active learning with applications to text classification, *J. Mach. Learn. Res.* 2 (2002) 45–66.
- [56] P.D. Turney, Cost-sensitive classification: empirical evaluation of a hybrid genetic decision tree induction algorithm, *J. Artif. Intell. Res.* 2 (1995) 369–409.
- [57] P.D. Turney, Types of Cost in Inductive Concept Learning.
- [58] C. Wan, Test-cost sensitive classification on data with missing values in the limited time, in: R. Setchi, I. Jordanov, R. Howlett, L. Jain (Eds.), *Proceedings of Knowledge-Based and Intelligent Information and Engineering Systems*, LNCS, vol. 6276, 2010, pp. 501–510.
- [59] W.-Z. Wu, J.-S. Mi, W.-X. Zhang, Generalized fuzzy rough sets, *Inf. Sci.* 151 (2003) 263–282.
- [60] X.-D. Wu, V. Kumar, J.R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G.J. McLachlan, A. Ng, B. Liu, S.Y. Philip, et al., Top 10 algorithms in data mining, *Knowl. Inf. Syst.* 14 (1) (2008) 1–37.
- [61] Q. Yang, C.-X. Ling, X.-Y. Chai, R. Pan, Test-cost sensitive classification on data with missing values, *IEEE Trans. Knowl. Data Eng.* 18 (5) (2006) 626–638.
- [62] X.-B. Yang, J.-Y. Yang, C. Wu, D.-J. Yu, Dominance-based rough set approach and knowledge reductions in incomplete ordered information system, *Inf. Sci.* 178 (4) (2008) 1219–1234.
- [63] X.-P. Yang, J.T. Yao, Modelling multi-agent three-way decisions with decision-theoretic Rough sets, *Fundam. Inf.* 115 (2–3) (2012) 157–171.
- [64] J.T. Yao, H.-X. Li, G. Peters, Decision-theoretic rough sets and beyond, *Int. J. Approx. Reason.* 55 (1) (2014) 99–100.
- [65] Y.Y. Yao, Probabilistic rough set approximations, *Int. J. Approx. Reason.* 49 (2) (2008) 255–271.
- [66] Y.Y. Yao, Three-way decision: an interpretation of rules in rough set theory, *Rough Sets and Knowledge Technology*, Springer, 2009, pp. 642–649.
- [67] Y.Y. Yao, Three-way decisions with probabilistic rough sets, *Inf. Sci.* 180 (3) (2010) 341–353.
- [68] Y.Y. Yao, The superiority of three-way decisions in probabilistic rough set models, *Inf. Sci.* 181 (6) (2011) 1080–1096.
- [69] Y.Y. Yao, Y. Zhao, Attribute reduction in decision-theoretic rough set models, *Inf. Sci.* 178 (17) (2008) 3356–3373.
- [70] H. Yu, T. Su, X.-H. Zeng, A three-way decisions clustering algorithm for incomplete data, *Rough Sets and Knowledge Technology*, Springer, 2014, pp. 765–776.
- [71] H.-R. Zhang, F. Min, X. He, Aggregated recommendation through random forests, *Sci. World J.* 2014 (2014) 1–11.
- [72] H.-R. Zhang, F. Min, S.-S. Wang, A random forest approach to model-based recommendation, *J. Inf. Comput. Sci.* (2014).
- [73] S.-C. Zhang, Cost-sensitive classification with respect to waiting cost, *Knowl.-Based Syst.* 23 (5) (2010) 369–378.
- [74] H. Zhao, F. Min, W. Zhu, Test-cost-sensitive attribute reduction of data with normal distribution measurement errors, *Math. Probl. Eng.* 2013 (2013) 1–12.
- [75] Y. Zhao, S.K.M. Wong, Y.Y. Yao, A note on attribute reduction in the decision-theoretic rough set model, *Transactions on Rough Sets XIII*, Springer, 2011, pp. 260–275.
- [76] B. Zhou, Y.Y. Yao, J.-G. Luo, A three-way decision approach to email spam filtering, *Advances in Artificial Intelligence*, Springer, 2010, pp. 28–39.
- [77] Z.-H. Zhou, X.-Y. Liu, Training cost-sensitive neural networks with methods addressing the class imbalance problem, *IEEE Trans. Knowl. Data Eng.* 18 (1) (2006) 63–77.
- [78] W. Zhu, F.-Y. Wang, Reduction and axiomization of covering generalized rough sets, *Inf. Sci.* 152 (1) (2003) 217–230.
- [79] W. Ziarko, Variable precision rough set model, *J. Comput. Syst. Sci.* 46 (1) (1993) 39–59.