Information Sciences 370–371 (2016) 159–183

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Multi-attribute search framework for optimizing extended belief rule-based systems

Long-Hao Yang[a], Ying-Ming Wang[a], Qun Su[b], Yang-Geng Fu[b,*], Kwai-Sang Chin[c]

[a] Decision Sciences Institute, Fuzhou University, Fuzhou 350116, P.R. China
[b] School of Mathematics and Computer Science, Fuzhou University, Fuzhou, 350116, P.R. China
[c] Department of Systems Engineering and Engineering Management, City University of Hong Kong, Kowloon Tong, Hong Kong

ARTICLE INFO

Article history:
Received 14 September 2015
Revised 3 July 2016
Accepted 25 July 2016
Available online 29 July 2016

Keywords:
Best activated rule set
Efficiency
Extended belief rule base
K-neighbor search
Multi-attribute search framework

ABSTRACT

The advantages and applications of rule-based systems have caused them to be widely recognized as one of the most popular systems in human decision-making, due to their accuracy and efficiency. To improve the performance of rule-based systems, there are several issues proposed to be focused. First, it is unnecessary to take the entire rule base into consideration during each decision-making process. Second, there is no need to visit the entire rule base to search for the key rules. Last, the key rules for each decision-making process should be different. This paper focuses on an advanced extended belief rule base (EBRB) system and proposes a multi-attribute search framework (MaSF) to reconstruct the relationship between rules in the EBRB to form the MaSF-based EBRB. MaSFs can be divided into *k*-dimensional tree (KDT)-based MaSFs and Burkhard–Keller (BKT)-based MaSFs. The former is targeted at decision-making problems with small-scale attribute datasets, while the latter is for those with large-scale attribute datasets. Based on the MaSF-based EBRB, the *k*-neighbor search and the best activated rule set algorithms are further proposed to find both the unique and the desired rules for each decision-making process without visiting the entire EBRB, especially when handling classification problems with large attribute datasets. Two sets of experiments based on benchmark datasets with different numbers of attributes are performed to analyze the difference between KDT-based MaSFs and BKT-based MaSFs, and to demonstrate how to use MaSFs to improve the accuracy and efficiency of EBRB systems. MaSFs and their corresponding algorithms are also regarded as a general optimization framework that can be used with other rule-based systems.

1. Introduction

Among all knowledge representation forms that can be used in artificial intelligence [43], rules are the most common scheme for expressing numerous types of knowledge, and rule-based systems has been the fastest growing branch of decision-support systems [3,42]. In many cases, a rule-based system consists of two essential components; namely, a large number of IF-THEN rules and a suitable reasoning approach. For IF-THEN rules, it is inevitably necessary to represent information that contains different varieties of uncertainty, as fuzzy, incomplete, or incorrect information, as a result of imprecision or incompleteness of human knowledge and the vagueness intrinsic to human knowledge [36,56]. In the development and implementation of knowledge reasoning mechanisms, uncertainty has already drawn considerable attention.

* Corresponding author.
E-mail address: ygfu@qq.com (Y.-G. Fu).

http://dx.doi.org/10.1016/j.ins.2016.07.067
0020-0255/

Acronyms

BKT	Burkhard-Keller tree
BKTSF	algorithm of constructing BKT-based MaSF
BRB	belief rule base
CRR	combination rule rate
DRA	method of dynamic rule activation
DST	Dempster-Shafer theory
EBRB	extended belief rule base
ER	evidential reasoning
KDT	k-dimensional tree
KDTSF	algorithm of constructing KDT-based MaSF
KSKDT	k-neighbor search in KDT-based MaSF
KSBKT	k-neighbor search in BKT-based MaSF
MaSF	multi-attribute search framework
RIMER	belief rule-base inference methodology using the evidential reasoning
SBR	algorithm of searching for best activated rule set
VRR	visitation rule rate

Notations

$A_{i,j}$	j^{th} referential value of the i^{th} antecedent attribute
$a_{i,j}$	j^{th} quantitative referential value in the i^{th} antecedent attribute
$\bar{a}_{i,j}$	j^{th} non-dimensional referential value in the i^{th} antecedent attribute
C_i^k	common form of referential value in the i^{th} antecedent attribute of thek^{th} rule
$C(Q)$	assessment of the best activated rule set
D_j	j^{th} consequent in rule base
$d(\bullet,\bullet)$	distance of rule
node	index-unit in the KDT/BKT-based MaSF
node.cns	set of sub-index-unit in *node*
node.lsn	index of left sub-index-unit in *node*
node.r	index of extended belief rule in *node*
node.rsn	index of right sub-index-unit in *node*
node.sa	splitting attribute in *node*
node.sv	splitting value in *node*
p	percentage to tune η
Q	set of activated rule
R_k	k^{th} rule in rule base
$SRA(\bullet,\bullet)$	similarity of rule antecedent
$SRC(\bullet,\bullet)$	similarity of rule consequent
U_i	i^{th} antecedent attribute in rule base
$x_q = (x_{q,1}, x_{q,2}, ..., x_{q,T})$	q^{th} input data vector
$y_q = (y_1^q, y_2^q, ..., y_T^q)$	non-dimensional value of $x_q = (x_{q,1}, x_{q,2}, ..., x_{q,T})$
$\alpha_{i,j}$	individual matching degree of the j^{th} referential value in the i^{th} antecedent attribute
$\beta_{j,k}$	belief degree assessed to D_j in the k^{th} rule
δ_i	antecedent attribute weight of the i^{th} antecedent attribute
η	number of rules need to be activated
θ_k	rule weight of the k^{th} belief rule
ξ	maximum distance between query y_q and R_k
$\tau(j,k)$	relationship between the j^{th} consequent and the k^{th} belief rule
Φ	set of extended belief rule

Three of the most widely applied approaches for reasoning with uncertain information include Bayesian probability theory [6], the Dempster–Shafer theory (DST) of evidence [18,40], and fuzzy set theory [57]. However, each of these theories, due to its own features and limitations, is only suitable for specific applications [17,22,24,32,35,67]. In the example of DST, researchers were plagued by the Zadeh paradox [58]. Suppose that three people A, B, and C were murder suspects with guilt probabilities of 1%, 99%, and 0% from witness D, respectively; according to witness E, the guilt probabilities for A, B, and C are 1%, 0% and 99%, respectively. It is inconceivable that A is the murderer with guilt probability 100% using DST's combination rule. Hence, there is a need to develop a powerful knowledge reasoning mechanism to handle different kinds of information under uncertainty.

Some studies in recent years have attempted to represent and combine several methods of handling uncertain information using rule-based systems. Yang et al. in [51] proposed a belief rule-base inference methodology using the evidential reasoning (RIMER) approach to handle hybrid information using a belief structure and evidential reasoning (ER) approach, in which an IF-THEN rule is designed with the distribution of belief degrees embedded in the consequent (the THEN part of the rule), called the belief rule base (BRB). Owing to the combination of uncertain information under the belief structure using the ER approach, RIMER approach has proved to be highly effective in solving uncertain multiple-attribute decision-making problems [12,14,41,54,59,63]. A similar belief rule-base methodology was proposed by Jiao et al. [25], which extends fuzzy rules in fuzzy rule-based classification systems with a belief structure and then develops a belief rule-based classification system to address imprecise or incomplete information in complex classification problems. Its referential value in the antecedent (the IF part of the rule) depends on the fuzzy membership function and each rule is generated from a numerical training dataset without parameter learning. However, these approaches are combinatorial schemes to determine the number of rules in the initial rule-based system. In other word, the rule-based system may turn out to be a combinatorial explosion problem resulting from covering all combinations of each referential value for each antecedent attribute.

Several methods for optimizing the structure of rule-based systems were proposed under the premise of small-scale attributes. For examples, Chang et al. [11] and Wang et al. [46] proposed some structure learning approaches using grey target, multidimensional scaling, isomap, principle component analysis, and rough set, those approaches can efficiently downsize the number of antecedent attributes while its number is five. Zhou et al. [64] proposed the concept of "statistical utility" which can determine whether a rule should be kept or reduced by calculating its utility value when there are two antecedent attributes. Wang et al. [47] introduced error analysis and density-based spatial clustering of applications with noise (DBSCAN) for adjusting structure, leading to the establishment of a complete, instead of incomplete or over-complete, structure based on the same experiment data [64]. Ken et al. [28] utilized the number of training datasets in each fuzzy partition to reduce unused rules when the fuzzy partition is divided into six partitions. Chang et al. [10] proposed another approach that restricting the number of rules in a rule-based system, is proposed to avoid the combinatorial explosion problem. But the failed rule problem, meaning that none of the rules would be used to combine the final result, is more likely to emerge from the rule-based system of [10]. The above approaches have the common drawback that the number of attributes in the decision-making problem cannot be large-scale, because the number of rules in the initial rule base would be overlarge due to the exponential relationship between attributes and rules. A data-driven approach was proposed in [31], in which the number of rules in a rule-based system relies on the datasets rather than antecedent attributes; simultaneously, the belief structure is also embedded in the antecedent attribute, forming what is called the extended belief rule base (EBRB). This system is able to target large-scale qualitative and quantitative attribute datasets, overcoming some of the main drawbacks of other state-of-the-art approaches to rule-based systems.

The aim of this research is to enhance the performance of rule-based systems regarding data-driven schemes, such as the EBRB system [31]. Since the EBRB system has been proven its efficiency in dealing with the oil pipeline leak detection and software defect detection problems [26,31], Yu et al. [55] introduced the 80/20 principle, a novel method of activating rules in the EBRB system, in which only the top 20% of activated rules should be used to infer the final result during each decision-making process. Another activation method proposed by Calzada et al. [9], named the dynamic rule activation (DRA) method, is designed to find a suitable set of activated rules for each input. Its main idea is to tackle the incompleteness and the inconsistency found in the EBRB by automatically adjusting the size of the set of activated rules. It is an excellent method for searching in the smaller set of activated rules to make the most suitable final decisions. However, DRA is a time consuming method that needs to search for the entire EBRB during its each iteration. In reviewing the methods for EBRB, it is clear that several issues need to be recognized for optimizing EBRB systems. First, it is unnecessary to take the entire EBRB into consideration during each decision-making process, only the key rules are indispensable. For example, there are many parameters that may be used to determine the performance of a fighter jet or a submarine, of which only certain key performance parameters [33], are critical to its final assessment. Second, it is unnecessary to visit the entire EBRB to search for the key rules. The EBRB system is efficient due to its lack of parameter learning. However, a full search is likely to be inefficient because of the large number of rules in the EBRB. Last, the key rules for each decision-making process should be different. In comparison with the existing approaches, such as the evidential k-nearest neighbor classification rule [19] and the traditional belief rule-based methodology [51], among others [1,16,45,66], these approaches have properly activated rules for any decision-making process.

To overcome the aforementioned issues in the EBRB system and the drawbacks of traditional approaches, this study proposes a multi-attribute search framework (MaSF) to optimize the EBRB system. For EBRBs with small- and large- scale attributes, the k-dimensional tree (KDT) [27] and the Burkhard–Keller tree (BKT) [8] are applied to construct MaSF, called KDT-based MaSF and BKT-based MaSF, respectively. The main aim of the MaSF is to tackle the disorder of rules in the EBRB system using the relationship between rules, such as Euclidean distance, variance, and average, as the foundation for improving the accuracy and the efficiency of EBRB systems. A new EBRB, MaSF-based EBRB, is proposed in this study. Despite the extra time needed to construct a MaSF-based EBRB, the new EBRB system will be more efficient in terms of the number of rules and the increased frequency of decision-making in return. The main focus of this paper is to propose k-neighbor search algorithms based on the MaSF-based EBRBs. The k-neighbor search [2] finds the nearest k neighbors for a query from given datapoints. These searches appear in many scientific and engineering applications including spatial databases, robotics, computer graphics, pattern recognition, data clustering, and function approximation [13,23,30,37,39]. In the traditional query-retrieval process, visiting the entire dataset to search for desired information is unavoidable. The

construction of a MaSF changes this situation and guarantees that each decision-making process only needs to search a subset of rules in the EBRB, which allows EBRB systems work in real time. For determining k in the k-neighbor search algorithm, which draws inspiration from the DRA method [9], a search for the best activated rule set is proposed. The new standard of the best activated rule set not only considers the number of rules that have been found, but also takes belief values and distance into consideration. To evaluate effectiveness of the two kinds of MaSFs and the proposed algorithms in this study, two experiments are carried out using benchmark datasets. First, KDT-based MaSF and BKT-based MaSF will be analyzed using datasets with small and large numbers of attributes. Second, the performance of the MaSF-based EBRBs is compared with that of existing approaches.

The rest of this paper is organized as follows. Section 2 briefly summarizes the rule representation scheme, such as combinatorial scheme and data-driven scheme, in the rule-based system and rule reasoning methodology using the ER algorithm. Section 3 reviews the rule generation in an EBRB and introduces how the MaSF-based EBRB is constructed in different scale antecedent attributes. Section 4 introduces k-neighbor searches in the MaSF-based EBRBs and how the search for a suitable set of activated rules is performed. Section 5 demonstrates and analyzes the performance of MaSF-based EBRB by presenting several case studies using benchmark datasets. The experiment results suggest that two kinds of MaSF-based EBRB systems have different advantages and both of them can improve the accuracy and efficiency of EBRB systems. Finally, this paper is concluded in Section 6.

2. Problem demonstration

2.1. Rule base with belief structure

Yang's BRB [51] employed a new knowledge representation scheme in which belief structure, referred to as a belief rule, is embedded in the consequent of a traditional IF-THEN rule. This innovative form of rule can be applied widely and intuitively to reflect uncertainty, such as the incomplete, fuzzy, and incorrect information. For example, a belief structure D in the safety analysis [4,7,62] can be expressed as:

$$D = (D_1, D_2, D_3, D_4) = (good,\ average,\ fair,\ poor). \tag{1}$$

And the k^{th} belief rule with a belief distribution representation can be written as:

$$R_k : IF\ conditions, THEN\ safety\ \ estimate\ \ is\ \{(D_1, 0.0), (D_2, 0.0), (D_3, 0.7), (D_4, 0.2)\}. \tag{2}$$

This specific belief rule means that the consequent (safety estimate) is described as 70% sure that the safety level is D_3 (fair) and 20% sure that the safety level is D_4 (poor) under several conditions. Notably, this belief rule is incomplete, with 10% uncertainty under belief structure D. For conditions of the belief rule, two patterns are introduced to explain its formation. Assume that three antecedent attributes are applied to construct conditions of the belief rule in the safety analysis, and these attributes can be expressed as

$$U = (U_1, U_2, U_3) = (failure\ \ rate,\ \ consequenc\ e\ severity,\ \ failure\ \ consequenc\ e\ probabilit\ \ y). \tag{3}$$

For simplicity, each antecedent attribute have the same set of referential values, and the qualitative referential value of the i^{th} attribute can be described as

$$A_i = (A_{i,1}, A_{i,2}, A_{i,3}) = \{frequent,\ critical,\ unlikely\}, i = 1, 2, 3; \tag{4}$$

According to [51], all referential values from each antecedent attribute should be used to construct the conditions. Therefore, the k^{th} belief rule in the form of a conjunctive IF-THEN can be written as

$$R_k : IF\ \ (U_1\ is\ A_{1,1})\ AND\ \ (U_2\ is\ A_{2,2})\ AND\ \ (U_3\ is\ A_{3,3}),\ \ THEN\ \ consequent\ \ \ s. \tag{5}$$

To capture uncertain information for the antecedent and consequent attributes as similar as possible, the entire set of referential values for each antecedent attribute are treated as a belief structure to describe conditions of the belief rule, which is called an extended belief rule [31]. In the safety analysis, the belief rule of Eq. (5) can be extended to be 100% sure that U_1 (failure rate) is $A_{1,1}$ (frequent), 100% sure that U_2 (consequence severity) is $A_{2,2}$ (critical) and 100% sure that U_3 (failure consequence probability) is $A_{3,3}$ (unlikely), and an extended belief rule in the form of a conjunctive IF-THEN can be written as

$$\begin{aligned} R_k : IF\ (U_1\ is\ \{(A_{1,1}, 1.0), (A_{1,2}, 0.0), (A_{1,3}, 0.0)\})\ AND\ (U_2\ is\ \{(A_{2,1}, 0.0), (A_{2,2}, 1.0), (A_{2,3}, 0.0)\}) \\ AND\ (U_3\ is\ \{(A_{3,1}, 0.0), (A_{3,2}, 0.0), (A_{3,3}, 1.0)\}),\ THEN\ \ consequent\ \ s \end{aligned}. \tag{6}$$

Obviously, the belief rule is a special instance of the extended belief rule through comparing Eqs. (5) and (6). When a BRB consists of extended belief rules, it is called the extended belief rule base (EBRB). With rule conditions and consequents defined, a complete rule is introduced. The k^{th} rule in a BRB or an EBRB is exhaustively written as

$$\begin{aligned} R_k : IF\ U_1\ is\ C_1^k\ AND\ U_2\ is\ C_2^k\ AND \cdots AND\ U_T\ is\ C_T^k, THEN\{(D_1, \beta_{1,k}), (D_2, \beta_{2,k}), \cdots, (D_N, \beta_{N,k})\}, \\ with\ rule\ \text{weight}\ \theta_k\ and\ attribute\ weight\ \delta_1, \delta_2, ..., \delta_T \end{aligned} \tag{7}$$

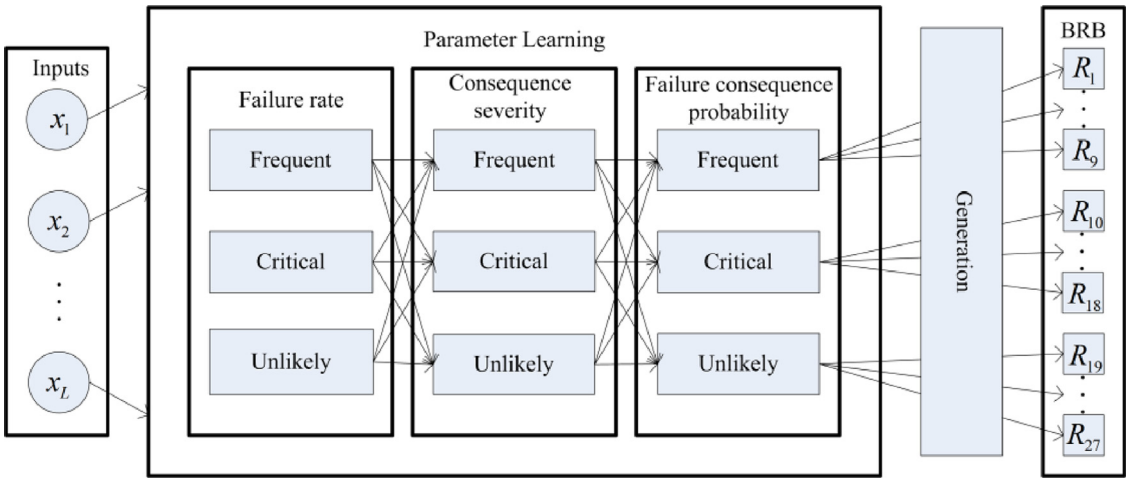

Fig. 1. Illustration of combinational scheme.

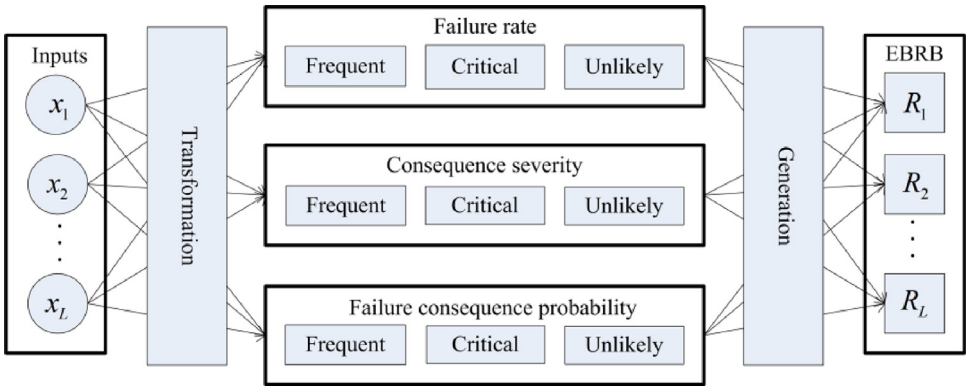

Fig. 2. Illustration of data-driven scheme.

Where $U_i(i = 1, ..., T)$ represents the i^{th} antecedent attribute, such as failure rate, T is the number of attributes in the antecedent attribute set U; $C_i^k(i = 1, ..., T; k = 1, ..., L)$ represents the referential value of the i^{th} antecedent attribute for the k^{th} rule, such as $A_{i,j}(j = 1, ...J_i)$ in a BRB or $\{(A_{i,j}, \alpha_{i,j}^k); k = 1, ..., L; j = 1, ..., J_i\}$ in an EBRB, L is the number of rules in a rule base; J_i is the number of referential values of the i^{th} antecedent attribute; $D_j(j = 1, ..., N)$ represents the j^{th} consequent; N is the number of consequents in a rule base; and $\beta_{j,k}(j = 1, ..., N; k = 1, ..., L)$ represents the belief or probability assessed to the j^{th} consequent in the k^{th} rule. In addition, the antecedent attribute weight $\delta_i(i = 1, ..., T; k = 1, ..., L)$ and rule weight $\theta_k(k = 1, ..., L)$ are also key parameters for a rule base.

2.2. Rules generation from distinctive schemes

In the process of rule generation, the aforementioned rule bases have distinctive schemes in accordance with their rule formulation. Based on [51], the BRB belongs to a combinatorial scheme. When a BRB is constructed, the conditions of the belief rule are required to cover all combinations of each referential value for each antecedent attribute. Fig. 1 describes the process of constructing a BRB for the above safety analysis. It is worth noting that the number of rules is related to the number of antecedent attributes and referential values instead of the number of inputs. In addition, the inputs are applied to train the BRB's parameters by parameter learning, and the trained parameters are used to update belief rules.

According to [31], the EBRB belongs to a data-driven scheme, which means that the extended belief rules are transformed from a priori information (i.e. a historical data set). As shown in Fig. 2, the inputs are transformed to a belief distribution for each antecedent attribute, and then an extended belief rule is constructed. Therefore, the number of rules is related to the number of inputs instead of the number of antecedent attributes or referential values.

2.3. Rules inference using the ER algorithm

Suppose the input vector $x_p = (x_{p,1}, x_{p,2}, ..., x_{p,T})$ is given for all antecedent attributes in a rule base, where $x_{p,i}$ is the p^{th} input for the i^{th} antecedent attribute. To obtain the predicted value of the input vector from the rule base, two ways to activate rules are described as follows.

For the combinatorial scheme, the input $x_{p,i}$ should be transformed to an individual matching degree using reasonable techniques, such as the rule- or utility-based equivalence transformation techniques [50]. It is supposed that the quantitative referential value $a_{i,j}$ of antecedent attribute U_i is equivalent to the qualitative referential value $A_{i,j}$ or

$$a_{i,j} \text{ means } A_{i,j} \ (i = 1, ..., T; j = 1, ..., J_i), \tag{8}$$

where T is the number of antecedent attributes and J_i is the number of referential values in the i^{th} antecedent attribute. Without loss of generality, assume that the quantitative referential value $a_{i,j+1}$ is greater than $a_{i,j}$, with $a_{i,1}$ being the smallest quantitative value and a_{i,J_i} being the largest. Then, an input x_p can be represented by the following equivalent expectation.

$$E(x_p) = \{\text{the input of } U_i \text{ is } \{(A_{i,j}, \alpha_{i,j}); j = 1, ..., J_i\}; i = 1, ..., T\}, \tag{9}$$

where

$$\begin{aligned} &\alpha_{i,j} = \frac{x_{p,i} - a_{i,j}}{a_{i,j+1} - a_{i,j}}, \alpha_{i,j+1} = 1 - \alpha_{i,j}; \ \text{if } a_{i,j+1} \geq x_{p,i} \geq a_{i,j} \\ &\alpha_{i,t} = 0; \ \text{while } t = 1, 2, ..., J_i, \ \text{and } t \neq j, j+1 \end{aligned}, \tag{10}$$

where $\alpha_{i,j}$ denotes the individual matching degree of the j^{th} referential value of the i^{th} antecedent attribute. When the referential value exists in the k^{th} rule, the individual matching degree can also be denoted as $\alpha_i^k = \alpha_{i,j}$.

For the data-driven scheme, the individual matching degree can be computed by measuring the similarity between each input $x_{p,i}$ and its corresponding antecedent attribute U_i in the k^{th} rule. The formula is denoted as

$$\alpha_i^k = 1 - \sqrt{\frac{\sum_{j=1}^{Ji} (\alpha_{i,j} - \alpha_{i,j}^k)^2}{2}}. \tag{11}$$

Thereafter, in both the combinatorial and data-driven schemes, the activated weight for the k^{th} rule is calculated by

$$\omega_k = \frac{\theta_k \prod_{i=1}^{T_k} \left(\alpha_i^k\right)^{\bar{\delta}_{k,i}}}{\sum_{l=1}^{L} \left(\theta_l \prod_{i=1}^{T_k} \left(\alpha_i^l\right)^{\bar{\delta}_{l,i}}\right)}, \ \bar{\delta}_{k,i} = \frac{\delta_{k,i}}{\max_{i=1,...,T_k}\{\delta_{k,i}\}}. \tag{12}$$

In the next process, the rules need to be combined using the ER algorithm. The analytic calculation process of ER [44,52,53] is introduced as follows:

$$\beta_n = \frac{\prod_{k=1}^{L} \left(\omega_k \beta_{n,k} + 1 - \omega_k + \omega_k \left(1 - \sum_{n=1}^{N} \beta_{n,k}\right)\right) - \prod_{k=1}^{L} \left(1 - \omega_k + \omega_k \left(1 - \sum_{n=1}^{N} \beta_{n,k}\right)\right)}{\sum_{n=1}^{N} \prod_{k=1}^{L} \left(\omega_k \beta_{n,k} + 1 - \omega_k + \omega_k \left(1 - \sum_{n=1}^{N} \beta_{n,k}\right)\right) - (N-1) \prod_{k=1}^{L} \left(1 - \omega_k + \omega_k \left(1 - \sum_{n=1}^{N} \beta_{n,k}\right)\right) - \prod_{k=1}^{L} (1 - \omega_k)}. \tag{13}$$

Finally, the output of the rule base is provided as a belief distribution in the form $\{(D_n, \beta_n); n = 1, ..., N\}$.

2.4. Problem demonstration

According to the combinatorial scheme, when a BRB is constructed, it is required to cover all combinations of each referential value for each antecedent attribute. This challenge is demonstrated in the following situation. Supposing there are T antecedent attributes and J_i alternative values for the i^{th} antecedent attribute, there, intuitively, would be $\prod_{i=1}^{T} J_i$ rules in the BRB. This is a combinatorial explosion problem that the size of the BRB would grow exponentially along with the increase in attributes. It is well known that parameter learning is targeted at maintaining the BRB. However, parameter-learning algorithms, including the optimization tool FMINCON in MATLAB [15,16], the Clonal Selection algorithm [60] and the Differential Evolution algorithm [10], iteratively update parameters until reaching a pre-defined stop point. The more rules there are in a BRB, the more time is needed for the process of parameter learning. Therefore, the BRB is hard to construct and maintain in multiple-attribute decision-making problems. Although some structure-learning approaches [11,28,46,47,64] have been proposed to solve the combinatorial explosion problem, the main idea of these is to sacrifice the accuracy of RIMER approach through reducing antecedent attributes.

Therefore, the data-driven scheme is better suited for solving multiple-attribute decision-making problems. There two main advantages of this scheme, namely, the number of rules in an EBRB is only related to the current dataset and the parameter learning of an EBRB does not require a time-consuming iterative algorithm. However, some challenges remain. When calculating an individual matching degree, almost all rules in the EBRB need to be activated. This often impacts the accuracy of EBRB systems due to inconsistency. Hence, Alberto *et al* [9] proposed a DRA method to address this issue,

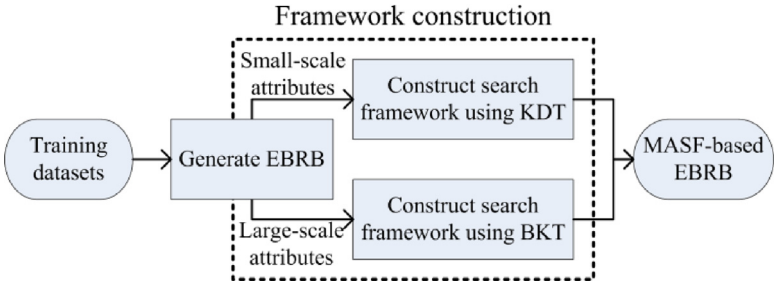

Fig. 3. Construction of search framework.

in which activated rules are selected dynamically and are based on input data. Unfortunately, when an EBRB is constructed based on the data-driven scheme, the rules in the EBRB are out of order. Therefore, no matter which methods are employed, including traditional rule activation methods and the DRA method, the entire EBRB must be visited to activate rules. This is an especially time-consuming process due to the number of rules in an EBRB.

As previously mentioned, the EBRB from a data-driven scheme is suitable to construct or maintain for multiple-attribute decision-making problems in comparison with the BRB from a combinatorial scheme, but this kind of EBRB faces time-consuming problems due to the large-scale rules in an EBRB. In fact, it is unnecessary to take all rules into consideration when activating a rule. The key to time-consumption problem can be attributed to unorganized structure of EBRB. Therefore, a multi-attribute search framework in accordance with KDT and BKT can be used to optimize the structure of the EBRB. As a result, only a few rules are searched in an organized EBRB.

3. Search framework construction for EBRB systems

The MaSF of EBRB systems plays a significant role in the rule activation process and is influenced by the number of antecedent attributes. Therefore, KDT and BKT are applied in this study as MaSF to optimize the EBRB system's unorganized structure, where the framework construction is as shown in Fig. 3.

3.1. Generating EBRB from classification dataset

According to the rule generation methodology presented in [31], the core step of generating an EBRB is to initialize the value of parameters using a dataset. Hence, the definitions of basic parameters such as referential value, consequent value, and weight value are given as follows. Definitions 1.1 to 1.3 are mainly based on previously published work [31].

Definition 1.1 (Initialization of referential value)**.** Suppose $x_p = (x_{p,1}, x_{p,2}, ..., x_{p,T})$ is the $p(p = 1, ..., P)^{\text{th}}$ datapoint of a classification dataset. The $j(j = 1, ..., J_i)^{\text{th}}$ referential value of the $i(i = 1, ..., T)^{\text{th}}$ antecedent attribute is then defined as follows.

$$a_{i,j} = \min_{p=1,...,P}\{x_{p,i}\} + \frac{(\max_{p=1,...,P}\{x_{p,i}\} - \min_{p=1,...,P}\{x_{p,i}\}) * (j-1)}{J_i - 1}, \tag{14}$$

Definition 1.2 (Initialization of consequent value)**.** Let $D = \{D_1, ..., D_N\}$ be the set of classes and $R = \{R_1, ..., R_L\}$ be the set of rules. The number of consequents for each rule is defined as N, and the $n(n = 1, ..., N)^{\text{th}}$ consequent value of the $k(k = 1, ..., L)^{\text{th}}$ rule is defined as follows.

$$\beta_{n,k} = \frac{\tau(n,k)}{\sum_{i=1}^{N}\tau(i,k)}, \tau(n,k) = \begin{cases} 1, & if\ R_k \in D_n \\ 0, & otherwise \end{cases}, \tag{15}$$

Definition 1.3 (Initialization of weight value)**.** Let $\{\alpha_{i,1}^k, ..., \alpha_{i,J_i}^k\}$ be the referential values of the $i(i = 1, ..., T)^{\text{th}}$ antecedent attribute in the $k(k = 1, ..., L)^{\text{th}}$ rule and $\{\beta_1^k, ..., \beta_N^k\}$ be the consequent values of the k^{th} rule. The rule weight of the k^{th} rule is then defined as follows.

$$\theta_k = 1 - \frac{\sum_{l=1,l\neq k}^{L}[1.0 - \exp\{-(SRA(l,k)/SRC(l,k) - 1)^2/(1/SRA(l,k))^2\}]}{\sum_{j=1}^{L}\sum_{t=1,t\neq j}^{L}[1.0 - \exp\{-(SRA(t,j)/SRC(t,j) - 1)^2/(1/SRA(t,j))^2\}]}, \tag{16a}$$

where

$$SRA(s,k) = 1 - \max_{t=1,...,T}\left\{\sqrt{\frac{\sum_{j=1}^{J_t}(\alpha_{t,j}^s - \alpha_{t,j}^k)^2}{2}}\right\}, SRC(s,k) = 1 - \sqrt{\frac{\sum_{n=1}^{N}(\beta_n^s - \beta_n^k)^2}{2}}. \tag{16b}$$

Table 1
Training dataset for diagnosis of lymph node metastasis.

Data number	Lymph node number (U_1)	Lymph node size (U_2)	Consequents (D)
1	1800	0.022	D_2, D_3
2	2000	0.020	D_2
3	1000	0.030	D_1
4	1600	0.024	D_3
5	3000	0.010	D_4

Table 2
Extended belief rules for diagnosis of lymph node metastasis.

R_k	θ_k	$U_1(\delta_1 = 1.0)$	$U_2(\delta_2 = 1.0)$	D
R_1	0.938	$\{(a_{1,1}, 0.2), (a_{1,2}, 0.8), (a_{1,3}, 0.0)\}$	$\{(a_{2,1}, 0.0), (a_{2,2}, 0.8), (a_{2,3}, 0.2)\}$	$\{(D_1, 0.0), (D_2, 0.5), (D_3, 0.5)(D_3, 0.0)\}$
R_2	0.824	$\{(a_{1,1}, 0.0), (a_{1,2}, 1.0), (a_{1,3}, 0.0)\}$	$\{(a_{2,1}, 0.0), (a_{2,2}, 1.0), (a_{2,3}, 0.0)\}$	$\{(D_1, 0.0), (D_2, 1.0), (D_3, 0.0)(D_3, 0.0)\}$
R_3	0.852	$\{(a_{1,1}, 1.0), (a_{1,2}, 0.0), (a_{1,3}, 0.0)\}$	$\{(a_{2,1}, 0.0), (a_{2,2}, 0.0), (a_{2,3}, 1.0)\}$	$\{(D_1, 1.0), (D_2, 0.0), (D_3, 0.0)(D_3, 0.0)\}$
R_4	0.532	$\{(a_{1,1}, 0.4), (a_{1,2}, 0.6), (a_{1,3}, 0.0)\}$	$\{(a_{2,1}, 0.0), (a_{2,2}, 0.6), (a_{2,3}, 0.4)\}$	$\{(D_1, 0.0), (D_2, 0.0), (D_3, 1.0)(D_3, 0.0)\}$
R_5	0.854	$\{(a_{1,1}, 0.0), (a_{1,2}, 0.0), (a_{1,3}, 1.0)\}$	$\{(a_{2,1}, 1.0), (a_{2,2}, 0.0), (a_{2,3}, 0.0)\}$	$\{(D_1, 0.0), (D_2, 0.0), (D_3, 0.0)(D_3, 1.0)\}$

In these definitions, all referential values increase in equal proportion for each antecedent attribute, in which the first referential value is the smallest input value and the last referential value is the largest input value. The consequent value is actually a probability value, and the final classification is determined by the class with the highest consequent value. The weight value is incorporated with a measurement of EBRB inconsistency. More information about generating EBRB is available in the literature [31]. In this study, a simple classification dataset is used to illustrate how an EBRB can be constructed using the above definitions.

Example 1 (Construction of EBRB)**.** A classification dataset regarding the diagnosis of lymph node metastasis [60,61] consists of two antecedent attributes, lymph node number (U_1) and lymph node size (U_2), and four consequent classes, stage 0 (D_1), stage 1 (D_2), stage 2 (D_3) and stage 3 (D_4). For simplicity, suppose there are five datapoints as shown in Table 1.

To construct the EBRB using Definitions 1.1–1.3, the quantitative referential values of U_1 and U_2 are initialized as $\{a_{1,1}, a_{1,2}, a_{1,3}\} = \{1000, 2000, 3000\}$ and $\{a_{2,1}, a_{2,2}, a_{2,3}\} = \{0.01, 0.02, 0.03\}$. All rules of the EBRB are expressed in Table 2.

3.2. Small-scale attributes search framework based on k*-dimensional trees*

The k-dimensional tree (KDT) [27] is a space-partitioning data structure for organizing datapoints in a k-dimensional space, which is also a binary tree where each node is a k-dimensional datapoint. Every non-leaf node can be regarded as implicitly generating a splitting hyper-plane that divides the space into two parts based on the splitting attribute and the splitting value, known as half-spaces. Datapoints to the left of this hyper-plane are represented by the left sub-tree of that node and points on the right, by the right sub-tree. A significant advantage of KDT is that a single structure can be applied efficiently to k-neighbor searches. However, one deficiency of KDT is that it fails to find the desired datapoints in high-dimensional spaces. As a general rule, if the dimensionality is k, the number of datapoints S should be $S >> 2^k$. Otherwise, when KDT are used with high-dimensional datasets, most of the nodes in the tree will be evaluated and the efficiency is no better than that of an exhaustive search [21].

On the basis of the KDT's features, KDT are introduced to optimize EBRB with small-scale attributes (KDT-based MaSF, for short). For constructing a KDT-based MaSF, the splitting attribute and the splitting value first need to be expressed in the form of an extended belief rule. Because the splitting value requires a single value and the belief distribution is embedded in antecedent attributes, it is necessary to transform the distributed value into a single value.

Definition 2.1 (Single attribute value)**.** Let $\{\bar{a}_{i,1}, ..., \bar{a}_{i,J_i}\}$ be the non-dimensional referential values of the $i(i = 1, ..., T)^{\text{th}}$ antecedent attribute, $\{(A_{i,j}, \alpha_{i,j}^k); j = 1, ..., J_i\}$ be the belief distribution of the i^{th} antecedent attribute in the $k(k = 1, ..., L)^{\text{th}}$ rule. The single attribute value for the i^{th} antecedent attribute of the k^{th} rule is then defined as follow.

$$y_i^k = \sum_{j=1}^{J_i} (\alpha_{i,j}^k \bar{a}_{i,j}) + (\bar{a}_{i,1} + \bar{a}_{i,J_i}) \left(1 - \sum_{j=1}^{J_i} \alpha_{i,j}^k\right), \tag{17}$$

Definition 2.2 (Splitting attribute)**.** The splitting attribute is the most recognizable antecedent attribute in the EBRB, which indicates that referential values of this attribute are very spread out around the mean and from each other.

Definition 2.3 (Splitting value)**.** The splitting value is the center single attribute value of the splitting attribute, which indicates that the rules can be equally divided into two sets according to the splitting value.

Table 3
Matrix of numerical value.

R_k	U_1	U_2
R_1	0.4	0.7
R_2	0.5	0.6
R_3	0.0	1.0
R_4	0.3	0.8
R_5	1.0	0.0

With these basic definitions given, the detailed algorithm for constructing KDT-based MaSF is introduced as follows.

KDTSF algorithm: an algorithm to construct a KDT-based MaSF using the entire EBRB.

Input: *node* is an index-unit of KDT-based MaSF, which contain an index of left sub-index-unit (*node.lsn*), an index of right sub-index-unit (*node.rsn*), an index of extended belief rule (*node.r*), the splitting attribute (*node.sa*) and the splitting value (*node.sv*); Φ is an extended belief rule set to store the entire EBRB initially.

Output: A node set of KDT-based MaSF can be used to index rule in the EBRB.

```
function KDTSF (node, Φ)
    for each antecedent attribute U_i in attribute set U do
        calculate variance of antecedent attribute v_i = 1/(L-1) Σ_{k=1}^{L} (y_i^k - 1/L Σ_{l=1}^{L} y_i^l)^2
    end for
    initialize splitting attribute node.sa = arg max_{i=1,...,T}{v_i}.
    find the ⌊(|Φ| + 1)/2⌋ th rule R_m in the ordered Φ with respect to y^k_{node.sa} (k = 1, ..., |Φ|).
    initialize rule index node.r = m and splitting value node.sv = y^m_{node.sa}.
    for each extended belief rule R_k in rule set Φ do
        if k == m then
            continue.
        else if k < (|R| + 1)/2 then
            Φ^left = Φ^left ∪ {R_k}.
        else if y^k_{node.sa} ≥ y^m_{node.sa} then
            Φ^right = Φ^right ∪ {R_k}.
        end if
    end for
    initialize a new index-unit lcnode and node.lsn = lcnode.
    KDTSF (lcnode, Φ^left).
    initialize a new index-unit rcnode and node.rsn = rcnode.
    KDTSF (rcnode, Φ^right).
End
```

The core of constructing a KDT-based MaSF is to find the splitting attribute and the splitting value. The splitting attribute is the attribute with the greatest variance, which indicates that the referential values of this attribute are very spread out around the mean and from each other. The splitting value is the center single attribute value. Based on the splitting attribute and the splitting value, a rule that has a center single attribute value as the most recognizable attribute can be found in the current rule set and then applied to construct one node of the KDT-based MaSF; the other rules are divided into two sets according to the center referential value and continue to be used to construct the KDT-based MaSF. An example for constructing a KDT-based MaSF is introduced as follows.

Example 2 (Construction of KDT-based MaSF)**.** An EBRB is shown in Table 2. The non-dimensional referential values are $\{\bar{a}_{1,1}, \bar{a}_{1,1}, \bar{a}_{1,3}\} = \{0.0, 0.5, 1.0\}$ and $\{\bar{a}_{2,1}, \bar{a}_{2,1}, \bar{a}_{2,3}\} = \{0.0, 0.6, 1.0\}$.

According to Definition 2.1, all rules of the EBRB are converted to a matrix of single attribute values that is expressed in Table 3.

During the process of constructing a KDT-based MaSF, the variances of the two antecedent attributes are calculated to be 0.133 and 0.142, in the first iteration of the KDTSF algorithm. Thus, the splitting attribute is U_2 and the splitting value is 0.7. As a result, the root node is R_1, the set Φ^{left} is $\{R_2, R_5\}$ and the set Φ^{right} is $\{R_3, R_4\}$. After more iterations, a complete KDT-based MaSF can be shown as Fig. 4.

3.3. Large-scale attributes search framework based on Burkhard-Keller tree

The Burkhard–Keller tree (BKT) is a metric tree suggested by Burkhard and Keller [8] specifically adapted to metric spaces, and is defined in the following way. An arbitrary datapoint is selected as the root node, whose sub-trees are identified by metric distance. The sub-trees are recursively built from the other datapoints. This process can be repeated until there is only one datapoint. Thus, the BKT is an n-ary tree and its construction depends on the metric distance between two datapoints rather than the splitting value of one of the splitting attributes. Therefore, a BKT is suitable for

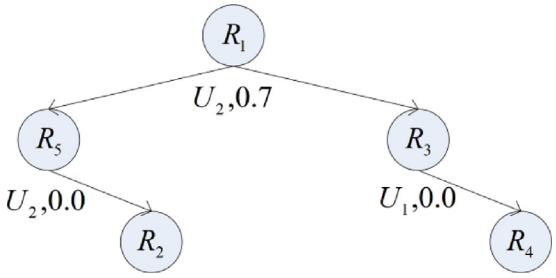

Fig. 4. Structure of KDT-based MaSF.

efficiently finding the desired datapoints in high-dimensional spaces without being influenced by increasing dimensionality. However, when the metric distance belongs to the set of real number, the branch of a BKT is far greater than its height. The average number of comparisons is then required to determine the desired datapoints tend toward $S/2$ [5], whereas KDT only needs $\log S$ in low-dimensional spaces [8]. Thus, when the number of rules L fails to satisfy $L >> 2^T$, a BKT is a better choice for optimizing an EBRB. In addition, it is important that the metric distance of BKT should satisfy the following axioms.

Axiom 1.1 (Positive definite). For two arbitrary datapoints x and y, the metric distance of the datapoints is no less than zero; if and only if x is equal to y, then the metric distance is zero.

Axiom 1.2 (Symmetry). For two arbitrary datapoints x and y, the metric distance from x to y is equal to the distance from y to x.

Axiom 1.3 (Triangle inequality). For three arbitrary datapoints x, y and z, the metric distance from x to y is smaller than the metric distance from x to y via z.

The above three axioms guarantee the efficiency and validity of BKT, which are the main method for reducing the number of comparisons as well. For constructed EBRB search framework using BKT (BKT-based MaSF, for short), the definitions of distance are given as follows.

Definition 3.1 (Distance of extended belief rule)**.** Suppose $\{y_1^k, ..., y_T^k\}$ and $\{y_1^l, ..., y_T^l\}$ are the single attribute value set of rule R_k and R_l, respectively. The distance of the extended belief rule is then defined as follow.

$$d(R_k, R_l) = \sqrt{\sum_{i=1}^{T} (y_i^k - y_i^l)^2}. \tag{18}$$

Definition 3.2 (Distance between rule and query)**.** Suppose $\{y_1^k, ..., y_T^k\}$ is the single attribute value set of rule R_k and $y_q = (y_1^q, y_2^q, ..., y_T^q)$ is the non-dimensional query. The distance between rule and query is then defined as follow.

$$d(R_k, y_q) = \sqrt{\sum_{i=1}^{T} (y_i^k - y_i^q)^2}. \tag{19}$$

The above distance formulas are actually the Euclidean distance, which is also used to calculate activated weights in the EBRB system. The more distance between the rule and the query, the smaller rule's activated weight in inferring the final result of the query. Therefore, the goal of the BKT-based MaSF is to find the rule that has the biggest activated weight. Given the BKT and the distance of extended belief rule, the detailed algorithm for constructing a BKT-based MaSF is introduced as follows.

BKTSF algorithm: an algorithm to construct a BKT-based MaSF using the entire EBRB.

Input: *node* is an index-unit of BKT-based MaSF, which contain a set of sub-index-unit (*node.cns*) and an index of extended belief rule (*node.r*); R_k is an extended belief rule of EBRB, which is used to construct the BKT-based MaSF.

Output: A node set of BKT-based MaSF can be used to index extended belief rule in the EBRB.

```
function BKTSF (node, R_k)
    sign = false.
    for each sub-index-unit cnode in node.cns do
        if d(R_node.r, R_cnode.r) == d(R_node.r, R_k) then
            BKTSF (cnode, R_k).
            sign = true.
        end if
    end for
    if sign==false then
        initialize a new index-unit cnode and cnode.r = k.
        node.cns = node.cns ∪ {cnode}.
    end if
end
```

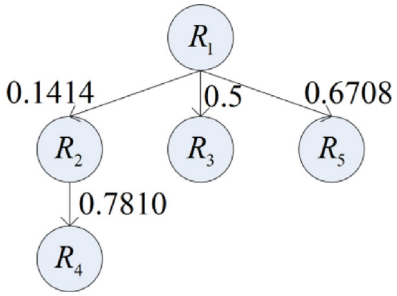

Fig. 5. Structure of BKT-based MaSF.

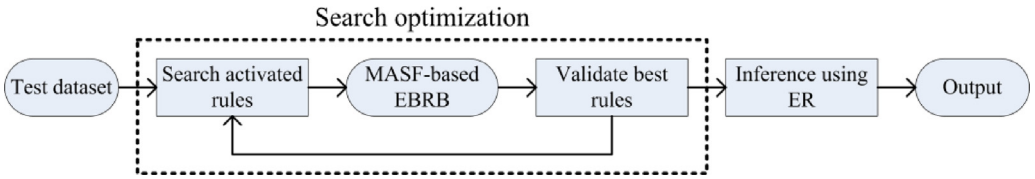

Fig. 6. Search optimization in MaSF-based EBRB.

According to the BKTSF algorithm, the core of constructing a BKT-based MaSF is to distinguish the rules using the distance of extended belief rules. Nodes with the same distance should be divided into the same sub-index-unit set in the BKT-based MaSF. The set of sub-index-units is listed by distance. An example for constructing BKT-based MaSF is introduced as follows.

Example 3 (Construction of BKT-based MaSF)**.** The dataset is the same as in Example 2. For simplicity, the EBRB is expressed in Table 3.

To construct a BKT-based MaSF, the rule R_1 is selected as the root node. Then, the distance between R_1 and other rules are $d(R_1, R_2) = 0.1414$, $d(R_1, R_3) = 0.5$, $d(R_1, R_4) = 0.1414$, and $d(R_1, R_5) = 0.6708$. Thus, rules R_2, R_3 and R_5 should be used as child nodes of rule R_1, and rule R_4 as child node of rule R_2. The resulting BKT-based MaSF is shown as Fig. 5.

4. Search optimization for EBRB systems

With the BKT-based MaSF and KDT-based MaSF for EBRB systems described in Section 3, the correlative algorithms of search optimization are introduced in this Section, as shown in Fig. 6.

4.1. Preconditions for search optimization

According to [31], the mechanism of rule activation requires visiting and combining the entire EBRB. However, the inconsistency of rules commonly impacts the performance of EBRB systems. A smart strategy is to activate the consistent rules and not to visit the inconsistent rules. To illustrate the smart strategy, a practical case about leak detection [31,49,65] is analyzed here. The antecedent attributes, including pressure difference and flow difference, are used to construct the leak detection EBRB system. Based on Definition 2.1, all rules can be expressed by two single-attribute values and divided into two classes, leak detected and no leak detected, which are plotted in a two-dimensional scatter chart shown as Fig. 7.

Suppose a non-dimensional query is {(Flow difference $= -5$), (Pressure difference $= -0.01$)}, an intuitive way to predict whether a leak is occurring is to activate several rules close to the query. As shown in Fig. 7, the rules lying within the circle should be used to infer the result of the query and it is unnecessary to visit the rules outside it.

In this case, the smart strategy of MaSF-based EBRBs aims to activate the consistent rules by not visiting the rules that are distant from the query. Although some additional operations should be inevitably considered to construct a MaSF, this smart strategy can improve the efficiency of EBRB systems. In particular, when the rule scale or query frequency is larger, the more superior MaSF-based EBRBs are. To conceptually explain the smart strategy of MaSF-based EBRBs, two lemmas about KDT-based MaSF and BKT-based MaSF are given as follows.

Lemma 1 (Smart strategy of KDT-based MaSF)**.** *Suppose node is an index-unit in a KDT-based MaSF, ξ is the maximum distance, R_t is indexed by node ($node.r = t$), $\{R_1, R_2, \ldots, R_K\}$ are indexed by node.lsn or node.rsn, and the splitting value of R_t* is y_j^t

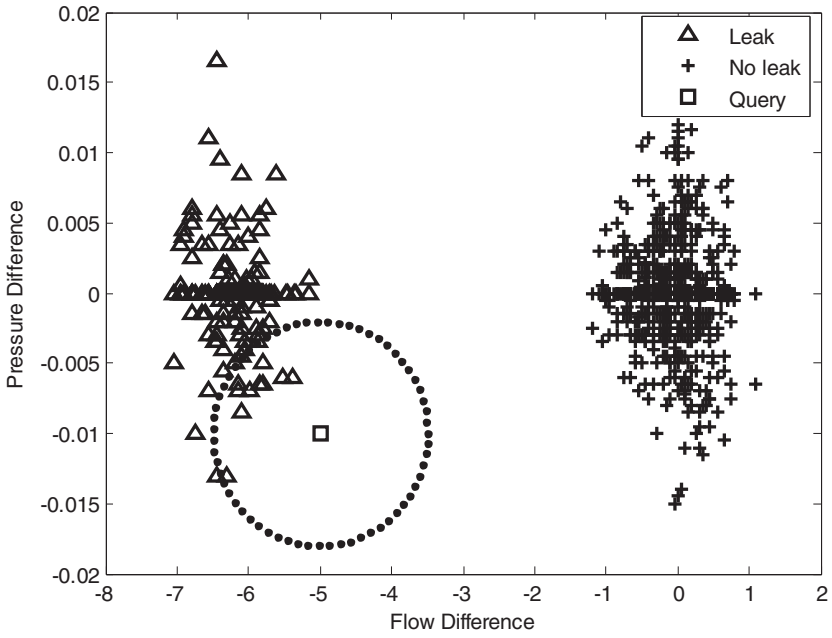

Fig. 7. Case of leak detection in oil pipeline.

($node.sv = y_j^t$). *For a non-dimensional query* y_q, *the following condition ensures that it is unnecessary to visit* $\{R_1, R_2, ..., R_K\}$.

$$y_j^q \le y_j^t \le y_j^k \text{ and } |y_j^t - y_j^q| \ge \xi, \text{ while } R_k \in \{R_1, R_2, ..., R_K\}, \tag{20}$$

Proof. See Appendix A.

Lemma 2 (Smart strategy of BKT-based MaSF)**.** *Suppose node is an index-unit in BKT-based MaSF,* ξ *is the maximum distance,* R_t *is indexed by node* ($node.r = t$), *and* $\{R_1, R_2, ..., R_K\}$ *are indexed by node.cns. For a non-dimensional query* y_q, *the following condition ensures that it is unnecessary to visit* $\{R_1, R_2, ..., R_K\}$.

$$|d(y_q, R_t) - d(R_t, R_k)| > \xi, \text{ while } R_k \in \{R_1, R_2, ..., R_K\}, \tag{21}$$

Proof. See Appendix B.

Based on Lemma 1 and 2, $\{R_1, R_2, ..., R_K\}$ is the set of rules that are distant from the query in the MaSF-based EBRBs. Neither BKT-based MaSF nor KDT-based MaSF should take $\{R_1, R_2, ..., R_K\}$ into consideration while activating the consistent rules in accordance with the smart strategy of MaSF-based EBRBs. Detailed search algorithms using BKT-based MaSF and KDT-based MaSF are introduced in Section 4.2.

4.2. K-neighbor searches in MaSF-based EBRB

The k-neighbor search [2] finds the nearest k neighbors for a query from a given dataset. This problem appears in many scientific and engineering applications, including pattern recognition, object recognition, data clustering, function approximation, vector quantization, and pattern classification [13,30]. The intuitive method of finding the nearest k neighbors from a dataset is to compare the distance between the query and all datapoints. To reduce the number of comparisons during k-neighbor search process, two search optimization algorithms are proposed in Section 4.2.1 and 4.2.2.

4.2.1. Search optimization using KDT-based MaSF

A k-neighbor search algorithm using KDT-based MaSF is proposed to optimize the process of searching for activated rules. The detailed algorithm is introduced as follows.

KSKDT algorithm: a k-neighbor search algorithm to search for the top k nearest activated rules.

Input: *node* is an index-unit of KDT-based MaSF; y_q is a given query; η is the number rules need to be activated.

Output: A set of activated rules (Q) to be aggregated in further steps.

```
function KSKDT (node, y_q, η)
    if |Q| < η then
        Q = Q ∪ {R_node.r}.
    else if max_{R_k∈Q}(d(R_k, y_q)) > d(node.r, y_q) then
        Q = (Q − R_k) ∪ {R_node.r}.
    end if
    if y^q_node.sa ≤ node.sv then
        KSKDT (node.lcn, y_q, max_{R_k∈Q}(d(R_k, y_q))).
        if |y^q_node.sa − node.sv| ≤ max_{R_k∈Q}(d(R_k, y_q)) then
            KSKDT (node.rcn, y_q, max_{R_k∈Q}(d(R_k, y_q))).
        end if
    else if y^q_node.sa > node.sv then
        KSKDT (node.rcn, y_q, max_{R_k∈Q}(d(R_k, y_q))).
        if |y^q_node.sa − node.sv| ≤ max_{R_k∈Q}(d(R_k, y_q)) then
            KSKDT (node.lcn, y_q, max_{R_k∈Q}(d(R_k, y_q))).
        end if
    end if
end
```

According to the KSKDT algorithm, the set of activated rules Q would be put into η extended belief rules and the maximum distance between query y_q and R_k within Q is $\xi(\xi = \max_{R_k \in Q}(d(R_k, y_q)))$. Based on Lemma 1, either the rules indexed by *node.lcn* or *node.rcn* are visited. Note that the value of ξ should be updated as the set Q changes. An example of the KSKDT algorithm is introduced as follows.

Example 4 (K-neighbor searches in KDT-based MaSF)**.** A KDT-based MaSF is shown as Fig. 4. The non-dimensional input for each antecedent attribute is $y_q = \{(y_1^q, 0.5), (y_2^q, 0.5)\}$ and $\eta = 1$.

To search for the activated rule set, four steps are expressed as follows.

Step 1: Visit rule R_1 and determine which rule should be visited in the next step. (1) According to $d(R_1, y_q) = 0.2236$ and $|Q| = 0 < \eta$, rule R_1 should put into the set Q and $\max_{R_k \in Q}(d(R_k, y_q)) = 0.2236$; (2) According to $y_2^q = 0.5 \le node.sv = 0.7$, rule R_5 needs to be visited in the next step.

Step 2: Visit rule R_5 and determine which rule should be visited in the next step. (1) According to $d(R_5, y_q) = 0.7071 > \max_{R_k \in Q}(d(R_k, y_q))$, rule R_5 should not put into the set Q; (2) According to $y_1^q = 0.5 > node.sv = 0$, rule R_2 should be visited in the next step.

Step 3: Visit rule R_2. According to $d(R_2, y_q) = 0.1 < \max_{R_k \in Q}(d(R_k, y_q))$, rule R_1 in the set Q should be replaced by rule R_2 and $\max_{R_k \in Q}(d(R_k, y_q)) = 0.1$.

Step 4: Return to rule R_1 and terminate the algorithm. (1) According to $|y_2^q - node.sv| = 0.2 > \max_{R_k \in Q}(d(R_k, y_q))$, the right branch of R_1 is unnecessary; (2) None of rules need to be visited in the next step, and the algorithm is concluded.

4.2.2. Search optimization using BKT-based MaSF

A k-neighbor search algorithm using BKT-based MaSF is proposed to optimize the process of searching for activated rules. The detailed algorithm is introduced as follows.

KSBKT algorithm: a k-neighbor search algorithm to search for the top k nearest activated rules.

Input: *node* is a node of BKT-based MaSF; y_q is a given query; η is the required number of activated rules.

Output: A set of activated rules (Q) to be aggregated in further steps.

```
function KSBKT (node, y_q, η)
    if |Q| < η then
        Q = Q ∪ {R_node.r}.
    else if max_{R_k∈Q}(d(R_k, y_q)) > d(R_node.r, y_q) then
        Q = (Q − R_k) ∪ {R_node.r}.
    end if
    for each child node cnode in node set node.cns do
        d_1 = d(R_node.r, R_cnode.r)
        d_2 = d(R_node.r, y_q)
        if d_1 ≥ d_2 − max_{R_k∈Q}(d(R_k, y_q)) and d_1 ≤ d_2 + max_{R_k∈Q}(d(R_k, y_q)) then
            KSBKT (cnode,y_q, max_{R_k∈Q}(d(R_k, y_q)))
        end if
    end for
end
```

According to the KSBKT algorithm, the set of activated rules Q would be put into η extended belief rules, the maximum distance between query y_q and R_k within Q is $\xi(\xi = \max_{R_k \in Q}(d(R_k, y_q)))$, and the rules indexed by *node.cns* are visited

based on Lemma 2. Note that the value of ξ should be updated while set Q changes. An example of the KSBKT algorithm is introduced as follows.

Example 5 (K-neighbor searches in BKT-based MaSF)**.** A BKT-based MaSF is shown as Fig. 5, the non-dimensional input for each antecedent attribute is $y_q = \{(y_1^q, 0.5), (y_2^q, 0.5)\}$ and $\eta = 1$.

To search for the activated rule set, three steps are expressed as follows.

Step 1: Visit rule R_1 and determine which rule should be visited in the next step. (1) According to $d(R_1, y_q) = 0.2236$ and $|Q| = 0 < \eta$, rule R_1 should put into the set Q and $\max_{R_k \in Q}(d(R_k, y_q)) = 0.2236$; (2) According to $d_1 = d(R_1, R_2) = 0.1414$and $d_2 = d(R_1, y_q) = 0.2236$, rule R_2 must be visited because $d_1 \geq d_2 - \max_{R_k \in Q}(d(R_k, y_q))$ and $d_1 \leq d_2 + \max_{R_k \in Q}(d(R_k, y_q))$.

Step 2: Visit R_2 and determine which rule should be visited in the next step. (1) According to $d(R_2, y_q) = 0.1 < \max_{R_k \in Q}(d(R_k, y_q))$, rule R_1 in the set Q should be replaced by rule R_2 and $\max_{R_k \in Q}(d(R_k, y_q)) = 0.1$; (2) According to $d_1 = d(R_2, R_4) = 0.781$ and $d_2 = d(R_2, y_q) = 0.1$, it is unnecessary to visit the rule R_4 because $d_1 > d_2 + \max_{R_k \in Q}(d(R_k, y_q))$.

Step 3: Return to visit rule R_1 and terminate the algorithm: (1) According to $d_1 = d(R_1, R_3) = 0.5$and $d_2 = d(R_1, y_q) = 0.2236$, it is unnecessary to visit the other rules because $d_1 > d_2 + \max_{R_k \in Q}(d(R_k, y_q))$; (2) None of rules needs to be visited in the next step and the algorithm is concluded.

4.3. Search optimization for the best activated rule set

The Section 4.2 has introduced k-neighbor searches in MaSF-based EBRBs. When a query is given, it is unnecessary to visit the entire MaSF-based EBRB, and the search process is thus highly efficient. However, it is difficult to determine a threshold η in k-neighbor searches, which is vital for improving the performance of EBRB systems. In this section, we propose a solution to take search optimization and the best activated rule set into consideration. Our definition of the best rule set draws inspiration from the DRA method [9], which only takes the number of rules in the activated rule set into account. In this situation, we append another two elements, namely each activated rule's belief values and the distance between activated rules and the query, to assess the best rule set. Thus, a new definition is written as follows.

Definition 4 (Assessment of the activated rule set)**.** Let $Q = \{R_1, ..., R_L\}$ be the activated rule set, $\{\beta_{1,k}, ..., \beta_{N,k}\}$ be the consequent values of the $k(k = 1, ..., L)$th activated rule, and y_q be the non-dimensional query. The assessment of the activated rule set is then defined as follows.

$$C(Q) = \max_{n=1,...,N} \left\{ \frac{\sum_{k=1}^{L} \tau(n,k)\beta_{n,k}(\sqrt{T} - d(R_k, y_q))}{L\sqrt{T}} \right\}, \tag{22a}$$

where

$$\tau(n,k) = \begin{cases} 1, & if\ n = \arg(\max_{i=1,...,N}\{\beta_{i,k}\}) \\ 0, & otherwise. \end{cases}. \tag{22b}$$

Example 6 (Assessment of the activated rule set)**.** A MaSF-based EBRB is shown in Table 2 and 3; the non-dimensional input for each antecedent attribute is $y_q = \{(y_1^q, 0.5), (y_2^q, 0.5)\}$.

In this example, the distance between the input and all rules are $d(R_1, y_q) = 0.2236$, $d(R_2, y_q) = 0.1$, $d(R_3, y_q) = 0.7071$, $d(R_4, y_q) = 0.3606$, and $d(R_5, y_q) = 0.7071$. R_3 belongs to stage 0, R_1 and R_2 belong to stage 1, R_1 and R_4 belong to stage 2, and R_5 belongs to stage 3. Thus, the values of each class are calculated as $\frac{1*(\sqrt{2}-0.7071)}{5\sqrt{2}} = 0.1$, $\frac{0.5*(\sqrt{2}-0.2236)+1*(\sqrt{2}-0.1)}{5\sqrt{2}} = 0.27$, $\frac{0.5*(\sqrt{2}-0.2236)+1*(\sqrt{2}-0.3606)}{5\sqrt{2}} = 0.2332$, $\frac{1*(\sqrt{2}-0.7071)}{5\sqrt{2}} = 0.1$, And then $C(Q) = \max\{0.1,\ 0.27,\ 0.2332,\ 0.1\} = 0.27$.

Based on Definition 4, the detailed algorithm for assessing the best activated rule set in MaSF-based EBRB is given as follows.

SBR algorithm: an algorithm to tune the value of p searching for the best activated rule set.

Input: *node* is an index-unit in a MaSF-based EBRB; y_q is a given query;

Output: A set of the best activated rules (Q_{p_b}) to be aggregated in further steps.

```
function SBR (node, y_q)
    p = 1.
    while p > 0 ∧ p ≤ 1 do
        get a rule set Q_p using parameters: node, y_q and p. (such as KSKDT and KSBKT)
```

```
        for each rule R_k in rule set Q_p do
            if ω_k == 0 then
                Q_p = Q_p − {R_k}.
            end if
        end for
        if C(Q_p) > C(Q_b) then
            Q_b = Q_p.
        end if
        update p.
    end while
end
```

The value of p is described in terms of a percentage, which should transform into η. For example, $\eta = \lfloor p * L \rfloor$ when there are L rules in the EBRB. When updating p, the interval for varying p depends on the specific focus of each particular study.

5. Case studies

This section illustrates how MaSFs can improve the performance of EBRB systems, what differences exist between the two kinds of MaSFs for small- and large-scale attributes, and which aspects reveal the advantage of the k-neighbor search and the best activated rules algorithms using the MaSF-based EBRB. To demonstrate these, various classification datasets with different numbers of rules and antecedent attributes are taken into consideration.

5.1. Datasets and experiment conditions

The performance of the proposed MaSF-based EBRBs is empirically assessed through several different experiments with 19 classification datasets from the well-known UCI Repository of Machine Learning Databases [20]. The main characteristics of the 19 datasets are summarized in Table 4. Note that for the Mammographic, Cancer and Diabetes datasets, samples with missing attribute values have been excluded.

To carry out the different experiments, k-fold cross-validation (K-CV) [38] is applied to test each dataset. Each dataset is divided into k blocks, with k-1 blocks as a training dataset and the remaining block as a test dataset. Therefore, each datapoint can be guaranteed to be used exactly once in the K-CV. In comparative analyses with MaSF-based EBRBs, we use 5-CV to analyze the difference between KDT-based MaSF and BKT-based MaSF for classification datasets with small- and large- scale attributes. In the comparative analysis with some existing approaches, we utilize 4-CV and 10-CV to maintain consistency with experimental conditions from the literatures. In addition, the proposed MaSF-based EBRBs are implemented in Microsoft Visual C++ 6.0 on Inter(R) Core(TM) i5-4570 CPU @ 3.20 GHz with Windows 7 system.

5.2. Comparative analysis with two kinds of MaSF-based EBRBs

Four datasets with different antecedent attributes, Banknote, Pageblocks, Forest, and Satimage, are selected from the UCI repository to compare KDT-based MaSF and BKT-based MaSF. Note that Banknote and Pageblocks are dataset with a small number of attributes, and Forest and Satimage are datasets with a large number of attributes. Table 5 shows the results

Table 4
Basic information of classification datasets.

Number	Names	No. of samples	No. of attributes	No. of classes
1	Banknote	1372	4	2
2	Transfusion	748	4	2
3	Iris	150	4	3
4	Mammographic	830	5	2
5	Thyroid	215	5	3
6	Knowledge	403	5	4
7	Vertebral	310	6	3
8	Seeds	210	7	3
9	Ecoli	336	7	8
10	Diabetes	393	8	2
11	Yeast	1484	8	10
12	Glass	214	9	6
13	Pageblocks	5473	10	4
14	Wine (Red)	1599	11	6
15	Wine	178	13	3
16	Forest	523	27	4
17	Cancer	569	30	2
18	Ionosphere	351	34	2
19	Satimage	6435	36	6

Table 5
Comparison of different datasets.

		5-CV						
EBRB	Attributes	Time(s)	VR	VRR (%)	CRR (%)	Accuracy (%)	Failed	Error
Banknote	4	5.20	1,505,906	100	99.99	96.65	0	46
Pageblocks	10	125.01	22,966,516	100	99.96	93.06	0	380
Forest	27	2.84	218,822	100	99.79	86.23	0	72
Satimage	36	554.21	33,127,356	100	98.92	89.37	0	684

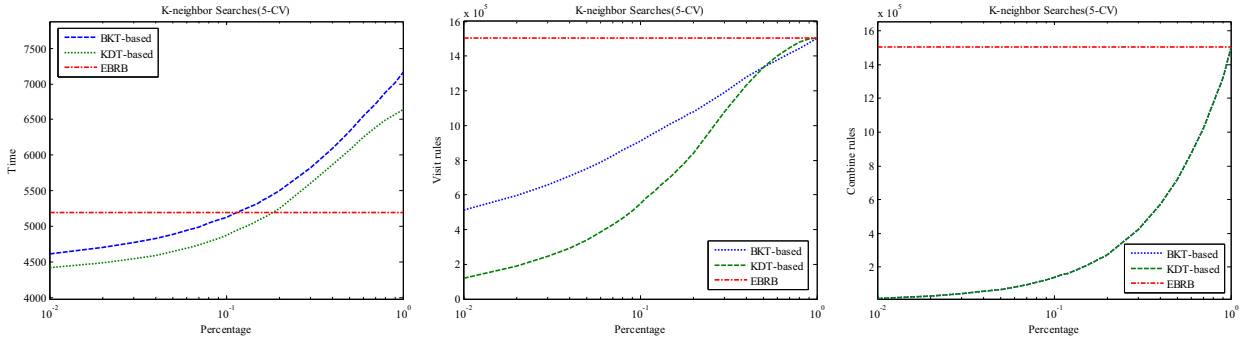

Fig. 8. Comparison of banknote about efficiency.

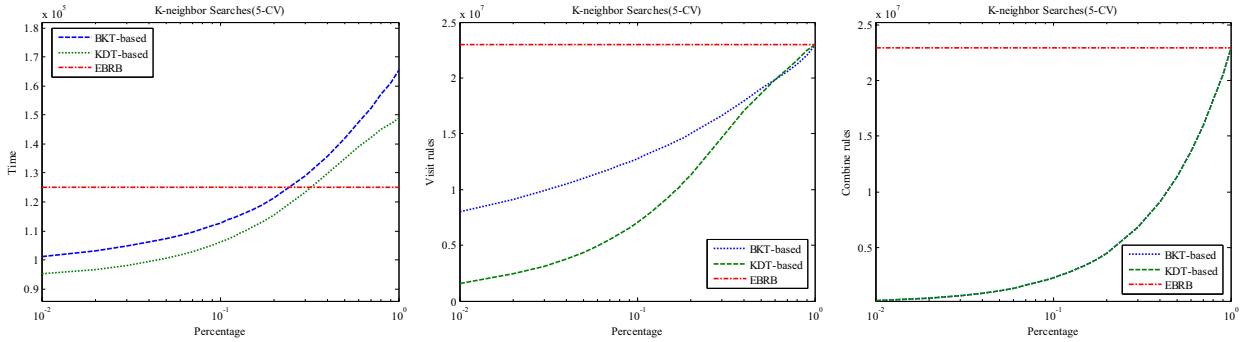

Fig. 9. Comparison of pageblock about efficiency.

from the four datasets using 5-CV in the EBRB system. Algorithms are measured in terms of: (1) time: the total time of algorithm, including the time needed to construct the MaSF in the comparative analysis; (2) visitation rule (VR): the number of rules that the algorithms need to visit; (3) visitation rule rate (VRR): the percentage of rules visited of the total rules; (4) combination rule rate (CRR): the percentage of combined rules of the total rule; (5) accuracy: the percentage of classes correctly classified from the total test datapoints; (6) failed: the number of test datapoints where the algorithm could not activate any rule; and (7) error: the number of test datapoints that the algorithm could not correctly classify. For the four datasets, the entire EBRB would have to be visited due to its unorganized structure. Hence, the VRR is 100%, which may be time-consuming. However, the CRR is smaller than 100% because of the distance between rules and the query, and these rules cannot be activated.

Based on the KDTSF and BKTSF algorithms, KDT-based MaSF and BKT-based MaSF can be constructed using all rules from benchmarks. Taking into account that η may be any value in its domain, this case study analyzes the sensitivity of k-neighbor search by varying the range of acceptable values for η. Therefore, the interval value is 0.01 in the range (0, 0.2) and others are 0.1. Figs 8–15 illustrate how time, VRR, CRR, failed, and error of the method increase as k-neighbor searches become more relevant.

For the classification problems in small-scale attribute datasets, Banknote (4 attributes) and Pageblocks (10 attributes) are used to validate the effectiveness of KDT-based MaSF and BKT-based MaSF. Figs. 8 and 9 show the efficiency of Banknote and Pageblocks using k-neighbor search, and Figs. 10 and 11 show the accuracy of Banknote and Pageblocks using k-neighbor search.

In view of its efficiency in small-scale attribute datasets, KDT-based MaSF are faster than BKT-based MaSF using k-neighbor search. This is also revealed by the number of visiting rules and combined rules. Meanwhile, Figs. 10 and 11 show

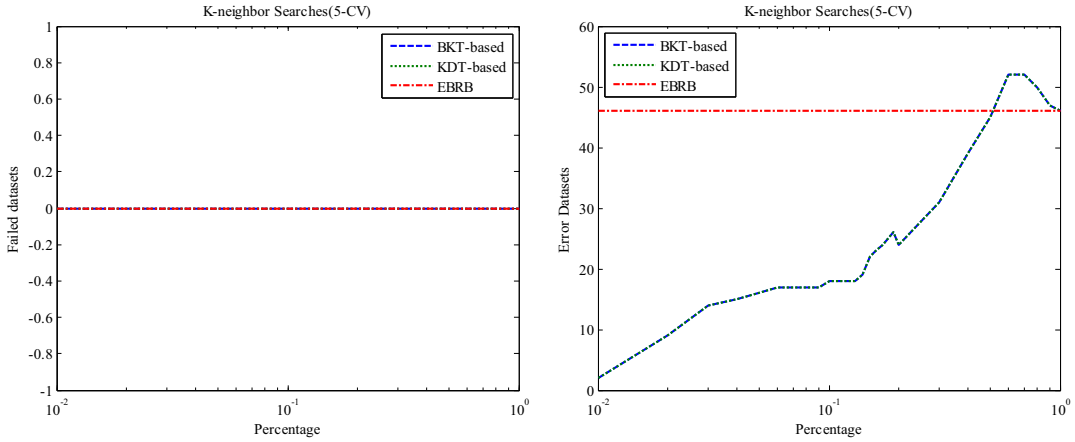

Fig. 10. Comparison of banknote about accuracy.

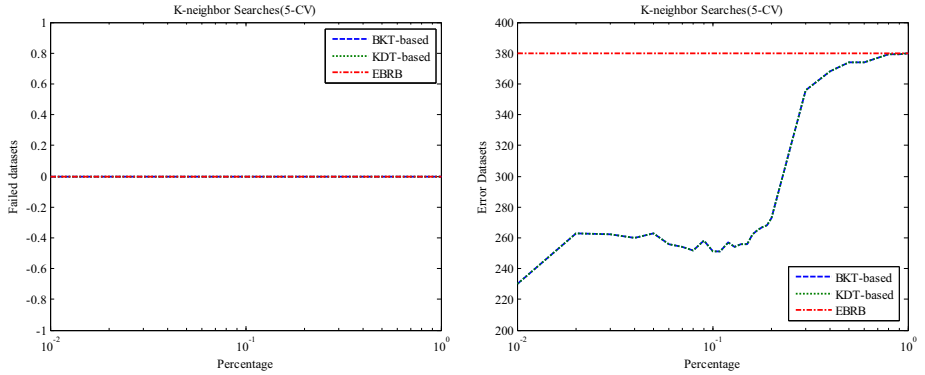

Fig. 11. Comparison of pageblocks about accuracy.

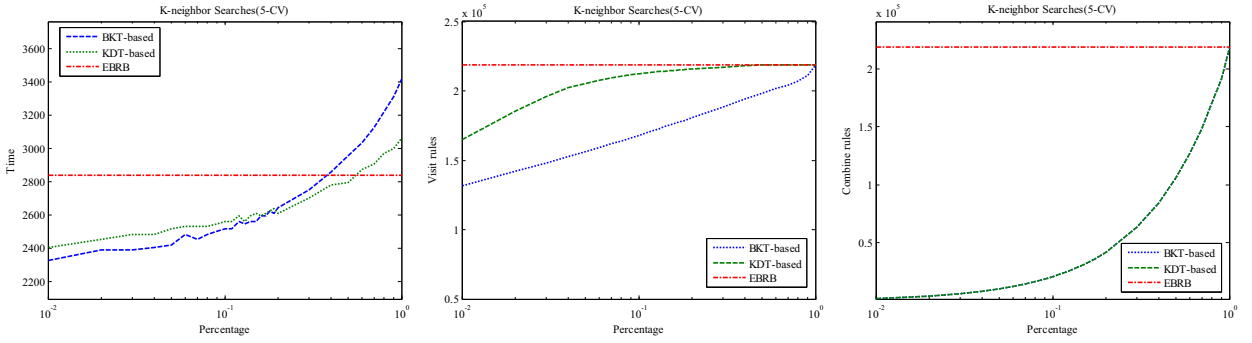

Fig. 12. Comparison of forest about efficiency.

that even though the time needed to construct the MaSFs has been taken into account, the proposed MaSFs still enhance the performance of EBRB systems.

In view of the accuracy in small-scale attribute dataset, both the BKT-based MaSF and the KDT-based MaSF perform more accurately than the initial EBRB. However, the process of search optimization can effectively eliminate failed data due to the nearest k neighbors for a query from the classification dataset being put into the activated rule set initially.

For the classification problems in large-scale attribute datasets, Forest (18 attributes) and Satimage (36 attributes) are used to validate the effectiveness of KDT-based MaSF and BKT-based MaSF. Fig. 12 and 13 show the efficiency of Banknote and Satimage using k-neighbor search, and Figs. 14 and 15 show the accuracy of Banknote and Satimage using k-neighbor search.

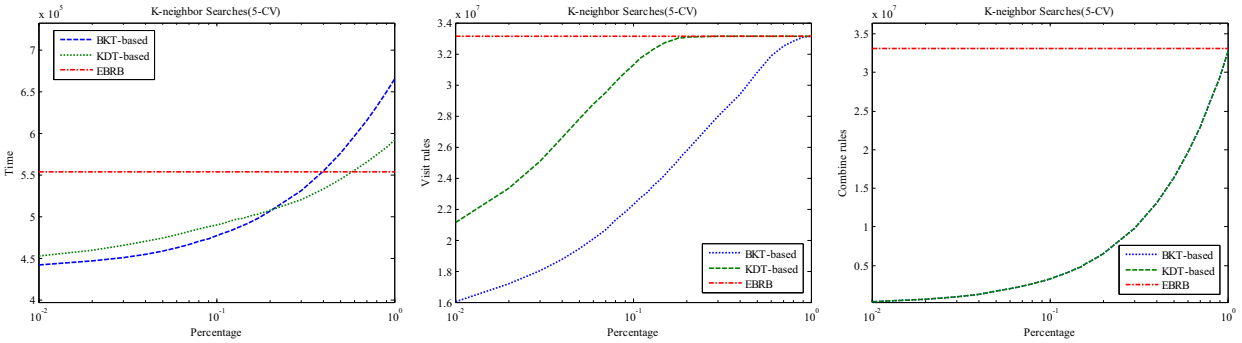

Fig. 13. Comparison of satimage about efficiency.

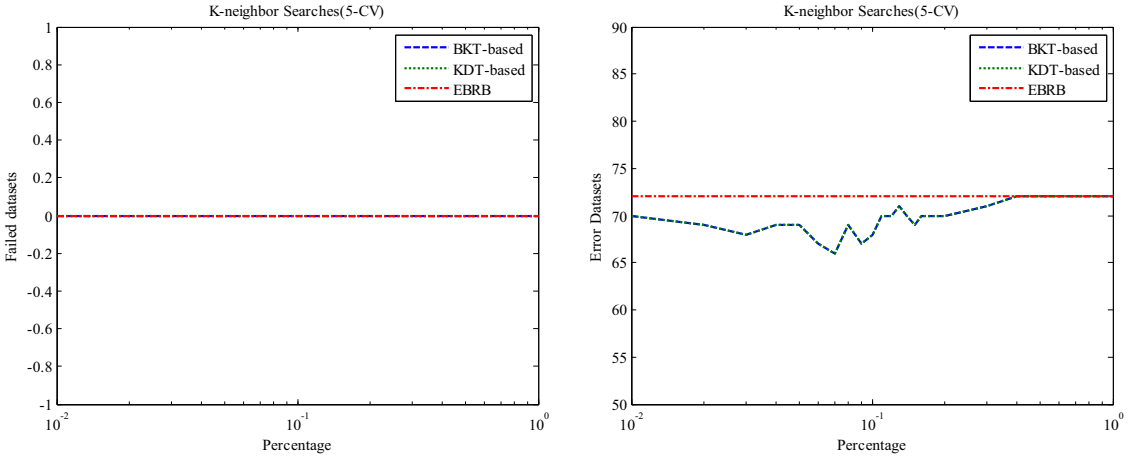

Fig. 14. Comparison of forest about accuracy.

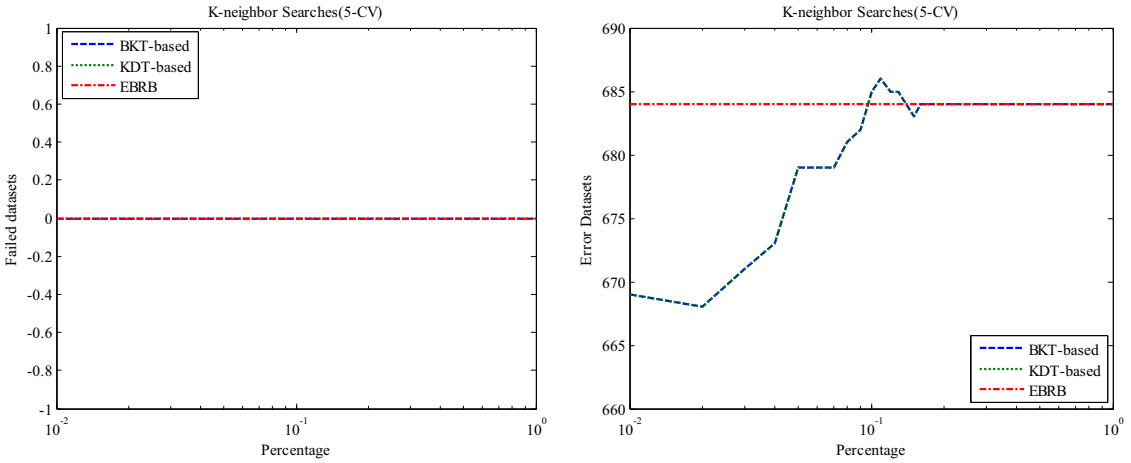

Fig. 15. Comparison of satimage about accuracy.

In terms of their efficiency in handling large-scale attribute datasets, BKT-based MaSF outperform KDT-based MaSF using *k*-neighbor searches, and the MaSFs can reduce the time for searching rules even though the time needed to construct the MaSFs has been considered. The number of visiting rules and combined rules supports this conclusion.

In terms of their accuracy in handling large-scale attribute datasets, both the BKT-based MaSF and the KDT-based MaSF perform more accurately than the initial EBRB. The process of search optimization can also eliminate failed data using the MaSF-based EBRBs.

Table 6
Comparison of the expanded datasets.

EBRB	Attributes	5-CV						
		Time (s)	VR	VRR (%)	CRR (%)	Accuracy (%)	Failed	Error
Banknote(2x)	8	7.504	1,505,906	100	99.99	98.54	0	20
Banknote(3x)	12	10.298	1,505,906	100	99.99	98.83	0	16
Banknote(4x)	16	12.245	1,505,906	100	99.99	99.13	0	12
Pageblocks(2x)	20	219.58	22,966,516	100	99.96	95.54	0	244
Pageblocks(3x)	30	316.08	22,966,516	100	99.96	96.31	0	202
Pageblocks(4x)	40	411.36	22,966,516	100	99.96	96.57	0	188

Note: kx represent k times more antecedent attributes than initial numbers

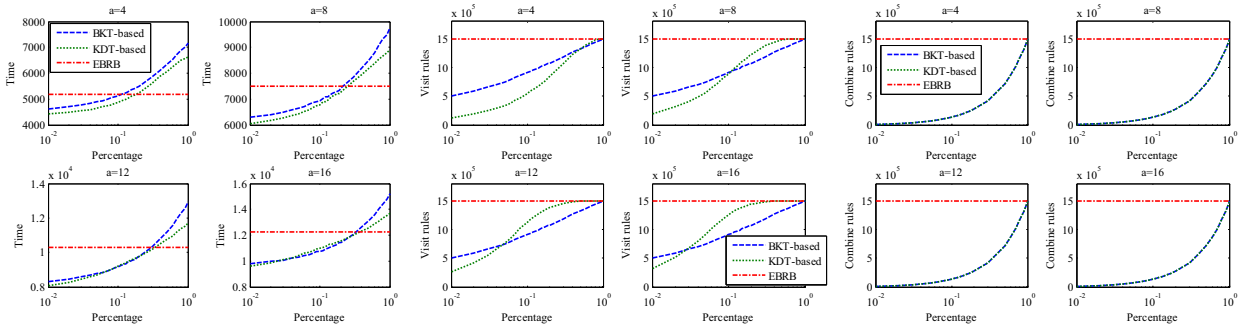

Fig. 16. Comparison of banknote about efficiency using *k*-neighbor search.

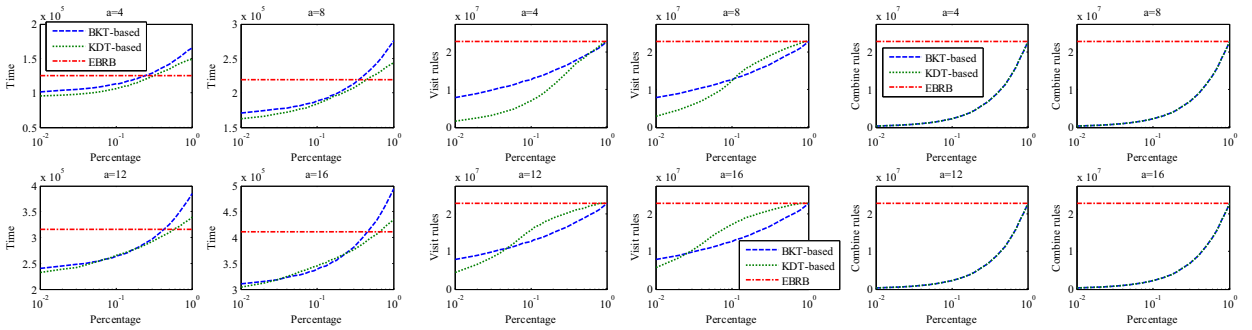

Fig. 17. Comparison of Pageblocks about efficiency using *k*-neighbor search.

To demonstrate the influence of the number of antecedent attributes, the small-scale attribute classification datasets are expanded in their number of antecedent attributes. For example, Banknote has 4 attributes, which can be expanded by coping the initial attribute and numerical values. The expanded dataset and its result are shown in Table 6.

Based on the differing numbers of antecedent attributes, the time results, VR, and combined rules using *k*-neighbor search are shown in Fig. 16 and 17. Note that the superiority of the KDT-based MaSF decreases as the number of antecedent attributes increases. Taking Banknote for example, when there are 4 antecedent attributes in the dataset, the number of visiting rules for KDT-based MaSF is better than for BKT-based MaSF using *k*-neighbor search. When the dataset has 16 attributes, the result of comparison is that BKT-based MaSF are almost better than KDT-based MaSF using *k*-neighbor search.

Preliminary conclusions are drawn based on these comparison results:

(1) The MaSF-based EBRBs outperform the initial EBRB. This can be seen in the comparison between the BKT-based MaSF, KDT-based MaSF and initial EBRB using the four benchmarks. When the percentage p is in [0, 0.2], the MaSF-based EBRBs have nearly better accuracy and efficiency than the initial EBRB. However, the advantage of MaSF-based EBRBs decreases by as p increase. Due to additional operations that are used to construct MaSFs, the efficiency of MaSF-based EBRBs is worse than that of the initial EBRB during search for all MaSF-based EBRBs.

(2) The KDT-based MaSF is suitable for small-scale attributes decision-making problems. For the Banknote and Pageblocks benchmarks, the efficiency of the KDT-based MaSF is greater than that of the BKT-based MaSF. Meanwhile, the more attributes the benchmark has, the worse the performance of the KDT-based MaSF is. This means that the number of attributes has an influence on the KDT-based MaSF.

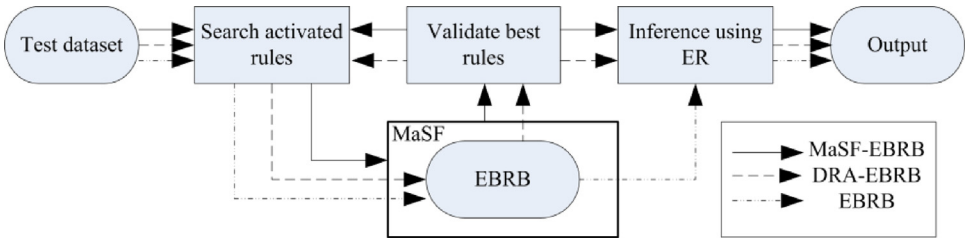

Fig. 18. Difference between three EBRB.

Table 7
Comparison with DRA method in small-scale attributes.

Dataset	Attributes	Aspects	DRA+WA	DRA+ER	EBRB	KSBKT+SBR	KSKDT+SBR
Banknote	4	Accuracy (%)	99.70(1)	99.70(1)	96.74(5)	99.58(3)	99.58(3)
		VRR (%)	100	100	100	22.81 ± 2.00	**8.23 ± 0.05**
Transfusion	4	Accuracy (%)	76.55(4)	76.57(3)	76.14(5)	78.65(1)	78.65(1)
		VRR (%)	100	100	100	17.47 ± 1.16	**14.14 ± 0.24**
Iris	4	Accuracy (%)	95.63(1)	95.50(2)	95.33(3)	95.20(5)	95.20(5)
		VRR (%)	100	100	100	**16.96 ± 1.38**	19.70 ± 0.22
Mammographic	5	Accuracy (%)	78.67(4)	78.39(5)	79.70(3)	80.61(1)	80.61(1)
		VRR (%)	100	100	100	**16.75 ± 2.42**	18.42 ± 0.33
Thyroid	5	Accuracy (%)	97.18(2)	97.19(1)	81.35(5)	93.02(3)	92.70(4)
		VRR (%)	100	100	100	35.70 ± 2.40	**27.33 ± 0.36**
Knowledge	5	Accuracy (%)	80.67(4)	80.71(3)	80.47(5)	87.59(1)	87.59(1)
		VRR (%)	100	100	100	56.72 ± 2.82	**31.65 ± 0.37**
Vertebral	6	Accuracy (%)	83.66(2)	84.29(1)	72.94(5)	76.39(3)	76.39(3)
		VRR (%)	100	100	100	40.14 ± 5.41	**31.45 ± 0.40**
Seeds	7	Accuracy (%)	92.14(3)	92.02(4)	91.33(5)	93.33(2)	93.71(1)
		VRR (%)	100	100	100	37.73 ± 2.70	**30.39 ± 0.27**
Ecoli	7	Accuracy (%)	83.75(4)	83.76(3)	81.16(5)	86.93(1)	86.93(1)
		VRR (%)	100	100	100	42.01 ± 6.13	**34.01 ± 0.42**
Diabetes	8	Accuracy (%)	71.34(5)	71.44(4)	72.88(3)	75.04(1)	75.04(1)
		VRR (%)	100	100	100	64.43 ± 6.38	**63.29 ± 0.18**
Yeast	8	Accuracy (%)	54.15(3)	54.13(4)	45.61(5)	59.77(1)	59.77(1)
		VRR (%)	100	100	100	57.94 ± 3.36	**46.85 ± 0.24**
Glass	9	Accuracy (%)	70.26(1)	69.65(4)	67.85(5)	70.19(2)	69.86(3)
		VRR (%)	100	100	100	**35.70 ± 3.39**	39.05 ± 0.37
Average rank			2.83(3)	2.92 (4)	**4. 50(5)**	**2.00 (1)**	**2.08(2)**

(3) The BKT-based MaSF is suitable for large-scale attributes decision-making problems. For the Forest, Satimage, expanded Banknote, and expanded Pageblocks benchmarks, the efficiency of the BKT-based MaSF is greater than that of the KDT-based MaSF using *k*-neighbor search. The accuracy of KDT-based MaSFs is nearly equal to the BKT-based MaSF, which result from the nearest rules being searched first.

5.3. Comparative analysis with some existing approaches

Nineteen benchmarks were collected from the UCI Machine Learning Repository. The classification algorithms from [9,34,48] are introduced for comparison with the proposed MaSFs.

In the first experiment, we aim to compare the performance of the MaSF-based EBRBs with the dynamic rule activation (DRA) method of EBRB [9], which is an optimization algorithm to improve the accuracy of the EBRB system. Diagrams of the EBRB system, DRA-based EBRB system and MaSF-based EBRB system are displayed in Fig. 18. The DRA-based EBRB system and the MaSF-based EBRB system each have an extra step to validate the best rules before rule inference using the ER algorithm, but the MaSF-based EBRB system visits the EBRB through MaSF when searching for activated rules.

For distinguishing attribute scale, fifteen classification datasets from [9] were divided into two groups. The first group consisted of datasets with less than 10 attributes, and the other, those with more than 10 attributes. The average results of 10 tests performed for each dataset using 10-CV are expressed as Tables 7 and 8, which show the classification accuracy and VRR of the proposed MaSF-based EBRBs compared with other EBRB algorithms. It is worth noting that VRR, rather than time, is used to analyze the proposed MaSF-based EBRBs. Both the DRA-based EBRB system and the MaSF-based EBRB system have to visit the EBRB repeatedly, so the running time of these algorithms can be measured by the VRR. Moreover, the running time of the algorithm is most affected by experimental equipment and programs, so VRR is more precise than time to compare EBRB systems. The number in brackets represents the rank of each method and the best VRR is marked by bold font, where the reported values of VRR is organized as: Mean value±Standard deviation.

Table 8
Comparison with DRA method in large-scale attributes.

Dataset	Attributes	Aspects	DRA + WA	DRA + ER	EBRB	KSBKT + SBR	KSKDT + SBR
Wine Red	11	Accuracy (%)	67.38(1)	67.28(2)	62.30(5)	66.30(3)	66.30(3)
		VRR (%)	100	100	100	**58.38 ± 1.57**	63.03 ± 0.16
Wine	13	Accuracy (%)	96.40(3)	96.46(2)	96.24(5)	96.40(3)	96.52(1)
		VRR (%)	100	100	100	**72.39 ± 10.83**	80.48 ± 0.49
Cancer	30	Accuracy (%)	94.52(5)	94.61(4)	96.52(3)	96.63(1)	96.63(1)
		VRR (%)	100	100	100	**65.85 ± 8.67**	89.31 ± 0.14
Average rank			3.00(4)	2.66 (3)	**4.33(5)**	**2.33(2)**	**1.67(1)**

Table 9
Comparison with traditional methods from [34].

Method	Iris (4 attributes)	Glass (9 attributes)	Wine (13 attributes)	Cancer (30 attributes)	Average rank
KSBKT + SBR	95.67 ± 0.68(5)	70.09 ± 1.00(1)	96.63 ± 0.71(4)	96.70 ± 0.31(3)	**3.25(3)**
KSKDT + SBR	95.73 ± 0.68(4)	69.72 ± 1.40(2)	96.52 ± 0.79(5)	96.70 ± 0.31(3)	**3.50(4)**
EBRB	95.13 ± 0.31(6)	67.90 ± 1.18(4)	96.46 ± 0.56(6)	96.38 ± 0.39(5)	**5.25(6)**
Naïve Bays	96.00 ± 0.30(3)	42.90 ± 1.70(7)	96.75 ± 2.32(3)	95.90 ± 0.20(6)	4.75(5)
C4.5	95.13 ± 0.20(6)	67.90 ± 0.50(4)	91.14 ± 5.12(7)	94.71 ± 0.09(7)	6.00 (7)
SMO	96.69 ± 2.58(2)	58.85 ± 6.58(6)	97.87 ± 2.11(2)	97.51 ± 0.97(2)	3.00(2)
FGM	96.88 ± 2.40(1)	69.14 ± 4.69(3)	98.36 ± 1.26(1)	98.14 ± 0.90(1)	1. 50(1)

Table 10
Comparison with traditional improvement methods from [48].

Method	Thyroid (5 attributes)	Glass (9 attributes)	Wine (13 attributes)	Ionosphere (34 attributes)	Average rank
KSBKT + SBR	92.84 ± 0.84(9)	70.09 ± 1.00(3)	96.63 ± 0.71(4)	86.47 ± 0.60(8)	**6.00(4)**
KSKDT + SBR	92.79 ± 0.67(10)	69.72 ± 1.40(4)	96.52 ± 0.79(5)	86.47 ± 0.60(8)	**6.75(6)**
EBRB	81.72 ± 0.42(14)	67.90 ± 1.18(8)	96.46 ± 0.56(7)	85.87 ± 0.60(11)	**10.00(12)**
SVM(LK)	96.19 ± 1.57(5)	61.21 ± 3.08(10)	96.40 ± 1.21(8)	87.44 ± 1.38(6)	7.25(9)
LDA(LK)	88.33 ± 2.48(11)	52.90 ± 3.47(14)	71.24 ± 3.72(14)	71.76 ± 5.53(14)	13.25(14)
LapSVM(LK)	96.11 ± 2.30(6)	63.08 ± 4.32(9)	96.85 ± 1.48(3)	84.59 ± 2.24(12)	7.50(10)
LapRLS(LK)	85.56 ± 2.48(12)	58.88 ± 3.33(12)	96.52 ± 0.98(5)	81.95 ± 2.65(13)	10.50(13)
DRLSC(LK)	85.19 ± 2.76(13)	58.04 ± 3.90(13)	97.75 ± 0.75(2)	85.91 ± 1.28(10)	9.50(11)
SSDR(LK)	93.70 ± 2.68(8)	58.97 ± 3.78(11)	98.09 ± 1.19(1)	87.50 ± 1.72(5)	6.25(5)
SVM(RBFK)	96.39 ± 1.66(3)	68.79 ± 3.85(6)	85.17 ± 2.89(13)	93.30 ± 1.91(1)	5.75(3)
LapSVM(RBFK)	94.81 ± 2.00(7)	68.41 ± 5.06(7)	88.76 ± 2.54(11)	88.69 ± 2.33(3)	7.00(7)
LapRLS(RBFK)	96.48 ± 2.08(2)	71.50 ± 3.00(2)	93.60 ± 1.50(10)	88.68 ± 1.96(4)	4.50(2)
DRLSC(RBFK)	96.30 ± 1.38(4)	69.07 ± 3.00(5)	87.42 ± 4.13(12)	87.22 ± 3.21(7)	7.00(7)
SSDR(RBRF)	96.67 ± 1.86(1)	71.96 ± 1.39(1)	94.27 ± 1.45(9)	88.75 ± 1.65(2)	3.25(1)

As Table 7 illustrates, the average accuracy for 4- through 9- attribute-datasets like Seeds, Ecoli and Glass are almost unacceptable when only using the EBRB system. Although the DRA method of [9] can improve the accuracy of the EBRB systems, the MaSF-based EBRBs are better optimized, based on their average ranks. The VRR of MaSF-based EBRBs is smaller than those the other EBRBs due to their special search framework, which also reveals that the time needs of the DRA method are greater than those of the MaSF-based EBRBs. Meanwhile, Table 7 also reveals that the KDT-based MaSF outperforms the BKT-based MaSF in terms of VRR.

As shown in Table 8, the average accuracy for datasets with more than 10 attributes also indicates that MaSF-based EBRBs are better than the DRA-based EBRB, and that the initial EBRB has worst accuracy. Comparing VRR, when the number of attributes increases, the KDT-based MaSF's rate is almost higher than that of the BKT-based MaSF, as shown as Table 8.

In the second experiment, state-of-the-art classifiers for 6 common classification datasets were compiled from [34,48]. For comparative purposes, the tests performed in [34,48] were emulated: randomizing the original datasets by permuting their samples and running 10 series of 4-CV (75 percent of the samples for training and the remaining 25 percent for test). Table 9 summarizes the results.

Table 9 presents the comparison of MaSF-based EBRBs with EBRB, the naïve Bays method, the C4.5 method, sequential minimal optimization (SMO), and fuzzy gain measure (FGM). Table 10 shows the comparison between the various classifiers based on linear discriminate analysis (LDA) and SVM. The reported values are organized as: Mean value±Standard deviation.

From the average rank, the accuracy of the EBRB system is worse than traditional methods or traditional improved methods. For example, the average rank of the EBRB system is 12 within 14 methods. On the other hand, the highest rank of the MaSF-based EBRB system is 4 when the EBRB is optimized with the BKT-based MaSF. These comparisons demonstrate that MaSFs can improve the accuracy of EBRB systems.

Insights are derived based on the comparison results:

(1) The applications of MaSFs are influenced by the number of attributes. For classification datasets with small number of attributes, the number of visiting rules in KDT-based MaSF is smaller than in BKT-based MaSF. Otherwise, the performance of KDT-based MaSF is worse than that of BKT-based MaSF as the number of attributes increases. Note that the performance of MaSF-based EBRBs outperforms the initial EBRB in both accuracy and VRR.
(2) The assessment of the best activated rule set is useful. As shown in Tables 7 and 8, the average accuracy rank of MaSF-based EBRBs is higher than those of the DRA-based EBRB and the initial EBRB, which owes to their taking the support of each class according to the activated rule set into consideration. The assessment of the best activated rule set guarantees that the crucial parameter η can determine its optimal value automatically.
(3) The accuracy of the EBRB system is improved. Compared with some existing methods, the accuracy of EBRB system is the worst. The MaSFs make the EBRB system feasible for use by improving its accuracy with a lower VRR. This is direct proof that validates the efficiency of MaSFs. Meanwhile, the MaSFs are a general optimization framework, so other rule-based systems also can improve their accuracy using MaSFs to optimize their rule base.

6. Conclusions

In this study, a novel multi-attribute search framework (MaSF) is proposed to improve the performance of rule-based systems. When constructing the MaSF using the entire EBRB (MaSF-based EBRB), detailed k-neighbor search algorithms are proposed as an efficient strategy to search for the desired rules without visiting the entire MaSF-based EBRB. To ensure the accuracy of rule-based system, a search algorithm for finding the best activated rule set is also proposed. In experiment with small- and large-scale attributes classification datasets, this study has demonstrated that the MaSF plays an important role in optimizing EBRB systems. The main conclusions of this paper can be summarized as follows:

(1) The two kinds of MaSFs are constructed based on the KDT and BKT. A series of case studies, according to k-neighbor search, are carried out in KDT-based MaSF and BKT-based MaSF, demonstrating that the former is good for small-scale attributes decision-making problems and the latter for those with large-scale attributes.
(2) Although these MaSFs are appropriate for different decision-making problems, the efficiency of MaSF-based EBRBs, including time cost and VRR, is superior to the initial EBRB system and the DRA-based EBRB system. Note that the construction of MaSF-based EBRBs cost extras time, though the process of searching for the best activated rule is more efficient in return, especially when the number of rules is large or the frequency of decision-making processes increases.
(3) In comparison with various classification datasets, the accuracy of MaSF-based EBRB systems is higher than that of the DRA-based EBRB system, and the initial EBRB system has correspondingly poor performance. On the other hand, the MaSFs can improve the accuracy of the EBRB system in comparison with traditional approaches.

For future research, the failed rule problem should be attended to further, as it has a great influence on the robustness and interpretability of EBRB systems. Therefore, a rule-adding mechanism must be designed to deal with that problem. On the contrary, reducing the redundancy of rules is another problem that also requires work.

Acknowledgements

This research was supported by the GRF of RGC-HK (No. CityU112111), the National Natural Science Foundation of China (Nos. 71231007, 71371053, and 71501047), the Research Fund for the Doctoral Program of Higher Education of China (No. 20123514110012), the Natural Science Foundation of Fujian Province, China (No. 2015J01248), and by the Science and Technology Project of Fujian Education Department under Grant (No. JA13036).

Appendix A. Proof of Lemma 1

Proof. Based on the given information from Lemma 1, we can deduce the distance formula as follows:

$$d(R_k, y_q) = \sqrt{\sum_{i=1}^{T} (y_i^k - y_i^q)^2} \geq \sqrt{(y_j^k - y_i^q)^2} = \left| y_j^k - y_j^q \right|, \tag{A.1}$$

According to the known condition $y_j^q \leq y_j^t \leq y_j^k$, we can get two inequalities

$$\begin{cases} y_j^k - y_j^t \geq 0 \\ y_j^t - y_j^q \geq 0 \end{cases}. \tag{A.2}$$

And then

$$\left| y_j^k - y_j^q \right| = \left| y_j^k - y_j^t + y_j^t - y_j^q \right| = \left| y_j^k - y_j^t \right| + \left| y_j^t - y_j^q \right| \geq \left| y_j^t - y_j^q \right|. \tag{A.3}$$

Equation (A.3) represents that the distance between y_q and R_k is greater than $|y_j^t - y_j^q|$. Therefore, while the maximum distance ξ is greater than $|y_j^t - y_j^q|$, none of rules in $\{R_1, R_2, ..., R_K\}$ would be selected.

Appendix B. Proof of Lemma 2

Proof. Based on the given information from Lemma 2, we can analyze its properties as follows:

$$d(R_t, R_k) = \sqrt{\sum_{i=1}^{T} (y_i^t - y_i^k)^2} \geq 0, iff\ t = k, d(R_k, R_k) = \sqrt{\sum_{i=1}^{T} (y_i^k - y_i^k)^2} = 0, \tag{B.1}$$

$$d(R_t, R_k) = \sqrt{\sum_{i=1}^{T} (y_i^t - y_i^k)^2} = \sqrt{\sum_{i=1}^{T} (y_i^k - y_i^t)^2} = d(R_k, R_t). \tag{B.2}$$

It is clear from Equation (B.1) and (B.2) that the distance between rules satisfies Axiom 1.1 and 1.2. Thereafter, we can deduce the distance formula using Cauchy-Buniakowsky-Schwarz inequality [29]:

$$\begin{aligned} d(R_t, R_s) + d(R_s, R_k) &= \sqrt{\left(\sqrt{\sum_{i=1}^{T} (y_i^t - y_i^s)^2} + \sqrt{\sum_{i=1}^{T} (y_i^s - y_i^k)^2}\right)^2} \\ &= \sqrt{\sum_{i=1}^{T} (y_i^t - y_i^s)^2 + \sum_{i=1}^{T} (y_i^s - y_i^k)^2 + 2\sqrt{\sum_{i=1}^{T} (y_i^t - y_i^s)^2}\sqrt{\sum_{i=1}^{T} (y_i^s - y_i^k)^2}}. \\ &\geq \sqrt{\sum_{i=1}^{T} (y_i^t - y_i^s)^2 + \sum_{i=1}^{T} (y_i^s - y_i^k)^2 + 2\sum_{i=1}^{T} ((y_i^t - y_i^s)(y_i^s - y_i^k))} \\ &= \sqrt{\sum_{i=1}^{T} (y_i^t - y_i^s + y_i^s - y_i^k)^2} = d(R_k, R_t) \end{aligned} \tag{B.3}$$

Hence, the distance has been proved to be a metric distance. Next, we can get two inequalities using Axiom 1.3:

$$\begin{cases} d(y_q, R_k) + d(R_t, R_k) >= d(y_q, R_t) \\ d(y_q, R_k) + d(y_q, R_t) >= d(R_t, R_k) \end{cases}. \tag{B.4}$$

And then

$$\begin{cases} d(y_q, R_k) >= d(y_q, R_t) - d(R_t, R_k) \\ d(y_q, R_k) >= d(R_t, R_k) - d(y_q, R_t) \end{cases}. \tag{B.5}$$

Finally, we can combine two inequalities and get another inequality

$$d(y_q, R_k) >= |d(y_q, R_t) - d(R_t, R_k)|. \tag{B.6}$$

Equation (B.6) represents that the distance between y_q and R_k is greater than $|d(y_q, R_t) - d(R_t, R_k)|$. Therefore, while the maximum distance ξ is greater than $|d(y_q, R_t) - d(R_t, R_k)|$, none of rules in $\{R_1, R_2, ..., R_K\}$ would be selected. For simplification, if the distance $d(y_q, R_t)$ is smaller than $|d(y_q, R_t) - d(R_t, R_k)|$, then the rule R_t is a better choice than R_k, where R_k belongs to $\{R_1, R_2, ..., R_K\}$. Therefore, it is unnecessary to search for any rules in $\{R_1, R_2, ..., R_K\}$.

References

[1] K. AbuDahab, D.L. Xu, Y.W. Chen, A new belief rule base knowledge representation scheme and inference methodology using the evidential reasoning rule for evidence combination, Expert Syst. Appl. 51 (2016) 218–230.
[2] N.S. Altman, An introduction to kernel and nearest-neighbor nonparametric regression, Am. Stat. 46 (3) (1992) 175–185.
[3] P. Angelov, R. Yager, A simple fuzzy rule-based system through vector membership and kernel-based granulation, in: Process of 2010 IEEE International Conference on Intelligent Systems, London, UK, 7-9 July, 2010, pp. 349–354.
[4] J. Augusto, J. Liu, P. McCullagh, H. Wang, J.B. Yang, Management of uncertainty and spatio-temporal aspects for monitoring and diagnosis in a smart home, Int. J. Comput. Intel. Syst. 1 (4) (2008) 361–378.
[5] R. Baeza-Yates, G. Navarro, Fast approximate string matching in a dictionary, in: String Processing and Information Retrieval: A South American Symposium, Proceedings IEEE, 1998, pp. 14–22.
[6] J.M. Bernardo, F M S. Adrian, Bayesian Theory, John Wiley and Sons, 2009.
[7] L.C. Briand, W.L. Melo, J. Wust, Assessing the applicability of fault-proneness models across object-oriented software projects, IEEE Trans. Softw. Eng. 28 (7) (2002) 706–720.
[8] W.A. Burkhard, R.M. Keller, Some approaches to best-match file searching, Commun. ACM 16 (4) (1973) 230–236.
[9] A. Calzada, J. Liu, H. Wang, K. Anil, A new dynamic rule activation method for extended belief rule-based systems, IEEE Trans. Knowl. Data Eng. 27 (4) (2015) 880–894.
[10] L.L. Chang, J.B. Sun, J. Jiang, M.J. Li, Parameter learning for the belief rule based system in the residual life probability prediction of metalized film capacitor, Knowl.-Based Syst. 73 (2015) 69–80.
[11] L.L. Chang, Y. Zhou, J. Jiang, M.J. Li, X. Zhang, Structure learning for belief rule base expert system: a comparative study, Knowl.-Based Syst. 39 (2) (2013) 159–172.
[12] L.L. Chang, Z.J. Zhou, Y. You, L.H. Yang, Z.G. Zhou, Belief rule based expert system for classification problems with new rule activation and weight calculation procedures, Inf. Sci. 336 (2016) 75–91.
[13] C.H. Chen, Feature selection for clustering using instance-based learning by exploring the nearest and farthest neighbors, Inf. Sci. 318 (2015) 14–27.

[14] Y.W. Chen, J.B. Yang, C.C. Pan, D.L. Xu, Z.J. Zhou, Identification of uncertain nonlinear systems: constructing belief rule-based models, Knowl.-Based Syst. 73 (2015) 124–133.
[15] Y.W. Chen, J.B. Yang, D.L. Xu, S.L. Yang, On the inference and approximation properties of belief rule based systems, Inf. Sci. 234 (11) (2013) 121–135.
[16] Y.W. Chen, J.B. Yang, D.L. Xu, Z.J. Zhou, D.W. Tang, Inference analysis and adaptive training for belief rule based system, Expert Syst. Appl. 38 (10) (2011) 12845–12860.
[17] D.A. Clark, Numerical and symbolic approaches to uncertainty management in AI, Artif. Intel. Rev. 4 (2) (1990) 109–146.
[18] A.P. Dempster, A generalization of Bayesian inference, J. Royal Stat. Soc., Series B (Methodological) 30 (2) (1968) 205–247.
[19] T. Denoeux, A k-nearest neighbor classification rule based on Dempter-Shafer theory, IEEE Trans. Syst. Man Cybern. 25 (5) (1995) 804–813.
[20] A. Frank, A. Asuncion, UCI machine learning repository, School Computer Science, University California, Los Angeles, 2010 [Online]. Available:http://archive.ics.uci.edu/ml.
[21] J.E. Goodman, J.O. Rourke, P. Indyk, Nearest neighbors in high-dimensional spaces, Handbook Of Discrete And Computational Geometry, CRC Press, 2004.
[22] S.J. Henkind, M.C. Harrison, An analysis of four uncertainty calculi„ IEEE Trans. Syst., Man Cybern. 18 (5) (1988) 700–714.
[23] G.R. Hjaltason, H. Samet, Distance browsing in spatial databases, ACM Trans. Database Syst. 24 (2) (1999) 265–318.
[24] J. Hodges, S. Bridges, C. Sparrow, B. Wooley, B. Tang, C. Jun, The development of an expert system for the characterization of containers of contaminated waste, Expert Syst. Appl. 17 (3) (1999) 167–181.
[25] L.M. Jiao, Q. Pan, T. Denoeux, Y. Liang, X. Feng, Belief rule-based classification system: extension of FRBCS in belief functions framework, Inf. Sci. 309 (2015) 26–49.
[26] L.Q. Jin, J. Liu, Y. Xu, X. Fang, A novel rule base representation and its inference method using the evidential reasoning approach, Knowl.-Based Syst. 87 (2015) 80–91.
[27] L.B. Jon, Multidimensional binary search trees used for associative searching, Commun. ACM 18 (9) (1975) 509–517.
[28] N. Ken, I. Hisao, T. Hideo, Adaptive fuzzy rule-based classification systems, IEEE Trans. Fuzzy Syst. 4 (3) (1996) 238–250.
[29] S. Kurepa, On the Buniakowsky-Cauchy-Schwarz inequality, Glasnik Matematicki 27 (2) (1966) 147–158.
[30] Z. Li, G.F. Ding, R. Li, Qin S.F, A new extracting algorithm of k nearest neighbors searching for point clouds, Pattern Recognit. Lett. 49 (2014) 162–170.
[31] J. Liu, L. Martinez, A. Calzada, H. Wang, A novel belief rule base representation, generation and its inference methodology, Knowl.-Based Syst. 53 (2013) 129–141.
[32] J. Liu, J.B. Yang, J. Wang, H.S. Sii, Review of uncertainty reasoning approaches as guidance for maritime and offshore safety-based assessment, J. UK Safety Reliability Soc. 23 (1) (2003) 63–80.
[33] S.A. Mcchrystal, Joint Capabilities Integration And Development System, Chairman of the Joint Chiefs of Staff Instruction, Washington DC, 2009.
[34] F. Mehdi, H.M. Mohammad, Z. Salman, A hybrid higher order neural classifier for handling classification problems, Expert Syst. Appl. 38 (2011) 386–393.
[35] S. Parsons, Current approaches to handling imperfect information in data and knowledge bases„ IEEE Transactions on, Know. Data Eng. 8 (3) (1996) 353–372.
[36] J. Pearl, Probabilistic reasoning in intelligent systems: networks of plausible inference, Morgan Kaufmann (1988).
[37] F.P. Preparata, M.I. Shamos, Computational Geometry: An Introduction, Springer, 1988.
[38] S. Raudys, A. Jain, Small sample effects in statistical pattern recognition: recommendations for practitioners, IEEE Trans., Patt. Anal., Mach, Intell. 13 (1991) 252–264.
[39] H. Samet, Foundations Of Multidimensional And Metric Data Structures, Morgan Kaufmann, 2006.
[40] G. Shafer, A Mathematical Theory of Evidence, Princeton University Press, Princeton, 1976.
[41] X.S. Si, C.H. Hu, J.B. Yang, Z.J. Zhou, A new prediction model based on belief rule base for system's behavior prediction, IEEE Trans. Fuzzy Syst. 19 (4) (2011) 636–651.
[42] A.P. Sinha, H. Zhao, Incorporating domain knowledge into data mining classifier: an application in indirect lending, Decis. Support Syst. 46 (1) (2008) 297–299.
[43] R. Sun, Robust reasoning: integrating rule-based and similarity-based reasoning, Artif. Intell. 75 (2) (1995) 241–295.
[44] Y.M. Wang, J.B. Yang, D.L. Xu, Environmental impact assessment using the evidential reasoning approach, Eur. J. Oper. Res. 174 (2006) 1885–1913.
[45] Y.M. Wang, J.B. Yang, D.L. Xu, K.S. Chin, Consumer preference prediction by using a hybrid evidential reasoning and belief rule-based methodology, Expert Syst. Appl. 36 (4) (2009) 8421–8430.
[46] Y.M. Wang, L.H. Yang, L.L. Chang, Y.G. Fu, Rough set method for rule reduction in belief rule base, Cont. Decis. 29 (11) (2014) 1943–1950 (In Chinese).
[47] Y.M. Wang, L.H. Yang, Y.G. Fu, L.L. Chang, K.S. Chin, Dynamic rule adjustment approach for optimizing belief rule-based expert system, Knowl.-Based Syst. 96 (2016) 40–60.
[48] F. Wu, W.H. Wang, Y. Yang, Y.T. Zhuang, F.P. Nie, Classification by semi-supervised discriminative regularization, Neurocomputing 73 (10-12) (2010) 1641–1651.
[49] D.L. Xu, J. Liu, J.B. Yang, G.P. Liu, J. Wang, I. Jenkinson, R. Jun, Inference and learning methodology of belief-rule-based expert system for pipeline leak detection, Expert Syst. Appl. 32 (1) (2007) 103–113.
[50] J.B. Yang, Rule and utility based evidential reasoning approach for multi-attribute decision analysis under uncertainties, Eur. J. Oper. Res. 131 (1) (2001) 31–61.
[51] J.B. Yang, J. Liu, J. Wang, H.S. Sii, H.W. Wang, Belief rule-base inference methodology using the evidential reasoning approach-RIMER, IEEE Transaction on Syst., Man, Cybernet.-Part A 37 (4) (2006) 569–585.
[52] J.B. Yang, M.G. Singh, An evidential reasoning approach for multiple attribute decision making with uncertainty, IEEE Trans. Syst. Man Cybernet. 24 (1) (1994) 1–18.
[53] J.B. Yang, D.L. Xu, On the evidential reasoning algorithm for multiple attribute decision analysis under uncertainty, IEEE Trans. Syst., Man, Cybernet.-Part A 32 (3) (2002) 289–304.
[54] Y. Yang, C. Fu, Y.W. Chen, D.L. Xu, S.L. Yang, A belief rule based expert system for predicting consumer preference in new product development, Knowl.-Based Syst. 94 (2016) 105–113.
[55] R.Y. Yu, L.H. Yang, Y.G. Fu, Data driven construction and inference methodology of belief rule-base, J. Comput. Appl. 34 (8) (2014) 2155–2160 (In Chinese).
[56] L.A. Zadeh, Fuzzy logic, neural networks, and soft Computing, Commun. ACM 37 (3) (1994) 77–84.
[57] L.Z. Zadeh, Fuzzy sets, Inf. Cont. 8 (3) (1965) 338–353.
[58] L.A. Zadeh, On the validity of Dempster rule of combination, in: Memo m 79/24, University of California, Berkeley, 1979, pp. 3–28.
[59] J.F. Zhang, X.P. Yan, D. Zhang, S. Haugen, X. Yang, Safety management performance assessment for maritime safety administration (MSA) by using generalized belief rule based methodology, Saf. Sci. 63 (2014) 157–167.
[60] Z.G. Zhou, F. Liu, L.C. Jiao, Z.J. Zhou, J.B. Yang, M.G. Gong, X.P. Zhang, A bi-level belief rule based decision support system for diagnosis of lymph node metastasis in gastric cancer, Knowl.-Based Syst. 54 (2013) 128–136.
[61] Z.G. Zhou, F. Liu, L.L. Li, L.C. Jiao, Z.J. Zhou, J.B. Yang, Z.L. Wang, A cooperative belief rule based decision support system for lymph node metastasis diagnosis in gastric cancer, Knowl.-Based Syst. 85 (2015) 62–70.
[62] Z.J. Zhou, C.H. Hu, G.Y. Hu, X.X. Han, B.C. Zhang, Y.W. Chen, Hidden behavior prediction of complex systems under testing influence based on semi-quantitative information and belief rule base, IEEE Trans. Fuzzy Syst. 23 (6) (2015) 2371–2386.
[63] Z.J. Zhou, C.H. Hu, D.L. Xu, J.B. Yang, D.H. Zhou, New model for system behavior prediction based on belief rule based systems, Inf. Sci. 180 (2010) 4834–4864.

[64] Z.J. Zhou, C.H. Hu, J.B. Yang, D.L. Xu, A sequential learning algorithm for online constructing belief rule based systems, Expert Syst. Appl. 37 (2) (2010) 1790–1799.
[65] Z.J. Zhou, C.H. Hu, J.B. Yang, D.L. Xu, D.H. Zhou, Online updating belief rule based system for pipeline leak detection under expert intervention, Expert Syst. Appl. 36 (2009) 7700–7709.
[66] Z.J. Zhou, C.H. Hu, J.B. Yang, D.L. Xu, D.H. Zhou, Online updating belief-rule-base using the RIMER approach, IEEE Trans. Syst., Man, Cybernetics-Part A 41 (6) (2011) 1225–1243.
[67] H.J. Zimmermann, An application-oriented view of modeling uncertainty, Eur. J. Oper. Res. 122 (2) (2000) 190–193.