Knowledge-Based Systems 53 (2013) 129–141

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

A novel belief rule base representation, generation and its inference methodology

Jun Liu [a,c,*], Luis Martinez [b], Alberto Calzada [a], Hui Wang [a]

[a] School of Computing and Mathematics, University of Ulster, Northern Ireland, UK
[b] Department of Computer Science, University of Jaen, Jaen, Spain
[c] School of Mathematics, Southwest Jiaotong University, Chengdu, Sichuan, China

ARTICLE INFO

Article history:
Received 30 May 2013
Received in revised form 13 August 2013
Accepted 16 August 2013
Available online 30 August 2013

Keywords:
Rule-based system
Belief distribution
Uncertainty
Decision-making
Inference mechanisms
Evidential reasoning approach

ABSTRACT

Advancement and application of rule-based systems have always been a key research area in computer-aided support for human decision making due to the fact that rule base is one of the most common frameworks for expressing various types of human knowledge in an intelligent system. In this paper, a novel rule-based representation scheme with a belief structure is proposed firstly along with its inference methodology. Such a rule base is designed with belief degrees embedded in the consequent terms as well as in the all antecedent terms of each rule, which is shown to be capable of capturing vagueness, incompleteness, uncertainty, and nonlinear causal relationships in an integrated way. The overall representation and inference framework offers a further improvement and great extension of the recently developed belief Rule base Inference Methodology (refer to as RIMER), although they still share a common scheme at the final step of inference, i.e., the evidential reasoning (ER) approach is applied to the rule combination. It is worth noting that this new extended belief rule base representation is a great extension of traditional rule base as well as fuzzy rule base by encompassing the uncertainty description in the rule antecedent and consequent. Subsequently, a simple but efficient and powerful method for automatically generating such extended belief rule base from numerical data is proposed involving neither time-consuming iterative learning procedure nor complicated rule generation mechanisms but keeping the relatively good performance, which thanks to the new features of the extended rule base with belief structures. Then some case studies in oil pipeline leak detection and software defect detection are provided to illustrate the proposed new rule base representation, generation, and inference procedure as well as demonstrate its high performance and efficiency by comparing with some existing approaches.

1. Introduction

Among many varieties for knowledge representation, it is widely recognized that the rule based one is one of the most common frameworks for expressing various types of knowledge in an intelligent system [22,1]. Taking advantage of the beauty of representing and manipulating human knowledge at first, in the design and implementation of advanced rule-based systems for supporting human decision making, it is desirable to endower the rule-based system with certain representation scheme and processing capabilities to handle simultaneously vagueness, incompleteness and uncertainty in conjunction with the flexibility in incorporating different types of input data formats, such as numerical, interval, uncertain value, or even subjective judgments.

During the last quarter of a century many different types of rule-based systems emerged, certainly including the fuzzy rule-based system [33,25], which, as one of the dominant and main frameworks in rule-based systems, has been widely accepted, investigated and applied in many application areas. Moreover, in recognition of the need to handle hybrid information with uncertainty in human decision making, a new methodology has been proposed recently for modeling a hybrid rule base using a belief structure and for inference in the belief rule-based system using the evidential reasoning (ER) approach [29]. The methodology is referred to as a belief Rule base Inference Methodology using the Evidential Reasoning approach – RIMER [30], where a rule base is designed with belief degrees embedded in the consequent term of a rule, called a *belief rule base* (BRB), is used to capture nonlinear causal relationships as well as uncertainty. The inference of the belief rule based system is implemented using the ER approach, this has been a distinct feature compared with the existing rule based inference methodologies. RIMER approach has been further investigated and its results and relevant extensions have proved to be highly

* Corresponding author at: School of Computing and Mathematics, University of Ulster, Northern Ireland, UK. Tel.: +44 2890368919.
E-mail address: j.liu@ulster.ac.uk (J. Liu).

0950-7051/$ - see front matter
http://dx.doi.org/10.1016/j.knosys.2013.08.019

positive in solving decision problems cross different application areas, such as, among others, safety and risk analysis, oil pipe leak detection and some other application in health care and engineering systems [31,32,24,2,17,16,15,14,13,12,19,18,6,21,10,11,34–37].

Among other issues, how to generate the rule base is a fundamental issue when designing and implementing a rule base system. Hence, various methods were proposed for automatically generating the rule base from sample data set. Most of these methods have involved iterative training algorithm or complicated generation scheme, e.g., gradient descent learning methods, genetic-algorithm-based methods, least-squares methods, a fuzzy c-means method, and a fuzzy-neuro method for learning fuzzy rule base, however they are either time consuming or using complicate rule generation strategies with the need of additional learning tool. For the RIMER approach, an optimal modeling has been proposed [31,17], with some further improved learning algorithms [34–36].

In this paper, to facilitate the more general and advanced application cases to handle simultaneously imprecision, incompleteness and uncertainty in conjunction with the flexibility in incorporating different types of input data format, as well as more flexible and simpler rule base generation scheme, the belief rule base in RIMER is extended with belief degrees embedded also in the entire antecedent terms of each rule. Most importantly, a simple but efficient and powerful method for automatically generating such extended belief rule base from numerical data is proposed, which is mainly attributed to the new features of the extended belief rule base. The main advantages of which over most of traditional learning approaches is its simplicity and efficiency because it involves neither time-consuming iterative learning procedure nor complicated rule generation mechanisms, but keep the relatively good performance. The work has been briefly outlined in [13]. This paper aims at extending, refining, completing and systematizing the results in [13].

The rest of this paper is organized as follows. Extended belief rule base and its inference framework are proposed in Section 2, including a brief overview of belief rule base in the RIMER approach. In Section 3, we propose a simple but efficient extended belief rule generation method with no time-consuming iterative procedure. Section 4 discusses and proves extended belief rule base inference system as a universal approximator, which shows the soundness of the methodology and provides the theoretical basis for successful applications of this new methodology to many different practical problems. The proposed rule representation, generation, and inference scheme as well as its performance are demonstrated in Section 5 using some case studies in oil pipe leak detection as well as in software defect prediction compared with some existing approaches. Conclusions are drawn in Section 6.

2. Extended belief rule based inference methodology

This section introduces the extended belief rule base along with its inference procedures.

2.1. Extended belief rule base

The belief rule base (BRB) introduced in RIMER approach [30] is summarized firstly in this section, which is designed with belief degrees embedded in the entire consequent terms. Then it is extended into a new belief rule base with belief degrees embedded in the consequent terms as well as in the all antecedent terms of each rule.

Suppose a BRB is given by $\boldsymbol{R} = \{R_1, R_2, \ldots, R_L\}$ with the kth rule represented as follows [30]:

$$R_k: \text{ IF } \boldsymbol{U} \text{ is } \boldsymbol{A}^k \text{ THEN } V \text{ is } \{\boldsymbol{D}, \boldsymbol{\beta}^k\}, \text{ with a rule weight } \theta_k \text{ and the attribute weight } \boldsymbol{\delta} \tag{1}$$

where $\boldsymbol{U}$ represents the antecedent attribute vector $(U_1, \ldots, U_T)$, V is the consequent attribute, $\boldsymbol{A}^k$ the packet referential values $\left\{A_1^k, \ldots, A_T^k\right\}$ of antecedents, here A_i^k $(i = 1, \ldots, T)$ is the referential value of the antecedent attribute U_i in the kth rule. T is the total number of antecedent attributes used in the rule; $(\boldsymbol{D}, \boldsymbol{\beta}^k)$ represents $\boldsymbol{D}$ with belief degrees $\boldsymbol{\beta}^k$, i.e., $\{(D_1, \beta_{1k}), \ldots, (D_N, \beta_{Nk})\}$, where $\boldsymbol{D}$ is the consequent vector $(D_1, \ldots, D_N)$, and $\boldsymbol{\beta}^k$ the vector of the belief degrees $(\beta_{1k}, \ldots, \beta_{Nk})$ for $k \in \{1, \ldots, L\}$, and β_{sk} $(s \in \{1, \ldots, N\})$ represents the belief degree to which D_s is believed to be the consequent if in the kth packet rule the input satisfies the packet antecedents $\boldsymbol{A}^k$. θ_k $(\in \Re^+$, $k = 1, \ldots, L)$ is the relative weight of the kth rule and $\boldsymbol{\delta}$ is the relative weight vector of the antecedent attributes, $\delta_i \in \Re^+$; $i = 1, \ldots, T$. L (>0) is the number of all the packet rules in the rule base. Moreover, $\sum_{s=1}^{N} \beta_{sk} \leqslant 1$. If $\sum_{s=1}^{N} \beta_{sk} = 1$, the kth packet rule is said to be complete; otherwise, it is incomplete.

In a rule base, a referential set can be a set of meaningful and distinctive evaluation standards for describing an attribute, which are commonly described by linguistic terms to reflect and model the vagueness or imprecision in the concepts. In (2) and (3) illustrated the referential values used in antecedent attributes in a belief rule relevant to safety analysis.

Take for example the following belief rule in safety analysis [17]:

R_k: IF the **failure rate** is *frequent* AND the **consequence severity** is *critical* AND the **failure consequence probability** is *unlikely*, THEN the **safety estimate** is $\{(good, 0), (average, 0), (fair, 0.7), (poor, 0.3)\}$ (2)

This is a special case of the rule in (1) while $N = 4$, $T = 3$, $\boldsymbol{U} = (U_1, U_2, U_3)$ = (failure rate, consequence severity, failure consequence probability), $\boldsymbol{A}^k = \left(A_1^k, A_2^k, A_3^k\right) = (frequent, critical, unlikely)$, $\boldsymbol{D} = (D_1, \ldots, D_N) = (good, average, fair, poor)$, and $\boldsymbol{\beta}^k = (\beta_{1k}, \beta_{2k}, \beta_{3k}, \beta_{4k}) = (0, 0, 0.7, 0.3)$. The consequent V (= safety estimate) is described as a belief distribution representation, stating that it is 70% sure that safety level is *fair* and 30% sure that safety level is *poor*. This kind of rule reflects another kind of uncertainty caused because sometimes evidence available is not sufficient or experts are not 100% certain to believe in a hypothesis but only to degrees of belief. If $\beta_{4k} = 0.2$, then $\sum_{s=1}^{4} \beta_{sk} = 0.9 \leqslant 1$, which means the correlation assessment is incomplete with 10% ignorance, which may be due to lack of knowledge, so this may reflect the incompleteness.

In this paper, to facilitate the more general application cases to handle simultaneously vagueness, incompleteness and uncertainty and more flexible and simpler rule base generation scheme, the belief rule in (1) is extended with belief degrees embedded in the entire possible antecedent terms of each rule as well, for example, the belief rule (2) can be extended as follows:

R_k: IF the **failure rate** is $\{(very\ low, 0), (low, 0), (reasonably\ low, 0), (average, 0), (reasonably\ frequent, 0), (frequent, 0), (highly\ frequent, 1)\}$ AND the **consequence severity** is $\{(negligible, 0), (marginal, 0), (moderate, 0), (critical, 0.3), (catastrophic, 0.7)\}$ AND the **failure consequence probability** is $\{(highly\ unlikely, 0.7), (unlikely, 0.2), (reasonably\ unlikely, 0.1), (likely, 0), (reasonably\ likely, 0), (highly\ likely, 0), (definite, 0)\}$ THEN the **safety estimate** is $\{(Good, 0), (Average, 0.1), (Fair, 0.3), (Poor, 0.6)\}$ (3)

This rule may look complex but is still easy to understand intuitively. This kind of extended belief rule is generic in the sense that it is able to not only capture the following several aspects in an integrated way, but also provide the flexible way to incorporate the hybrid input information (see Section 2.2 for details) and possibility to build up more efficient rule generation scheme (as illustrated in Section 3) although this extended belief rule looks complex: (1) fuzziness, e.g., representing the referential values of the antecedent and consequent attributes via linguistic terms which may be characterized by fuzzy membership as the way in fuzzy rule base; (2) uncertainty, e.g., with beliefs characterizing randomness or likelihood of some attribute in nature or reflecting the uncertain causal relationship; (3) incompleteness, e.g., handling partially known beliefs in antecedents and/or consequents due to incomplete information or knowledge; and (4) nonlinear relationships, e.g., via IF-THEN rules compared with the traditional mathematical modeling approaches between those three parameters and the safety level.

It is worth noting that the belief distributions incorporated in the antecedent of the rule shows some distinct and positive features compared with the traditional rule base as well as the belief rule base in RIMER [30]: (a) it provides a scheme to reflect the potential uncertainty involved essentially in the antecedent attribute itself, e.g., for some attributes which are random in nature (for an example of car evaluation, fuel consumption of car could change under different weather and road condition, the evaluation of it would be a probability distribution in essence); (b) it provides the flexibility in incorporating different types of input data format, such as numerical, interval, uncertain value, or even subjective judgments (see Section 2.2 for illustration); and (c) it accommodates effectively the relationship between the given input and the antecedent referential values by using the belief distribution without loss of information. This feature is especially important and has contributed to the new rule base generation scheme as proposed in Section 3, where the belief distributions in both IF part and THEN part play an important role in memorizing/storing the essential features of each input–output sample data pair.

More generally, rule (1) can be extended as follows:

$$R_k^*:\ \text{IF } \boldsymbol{U} \text{ is } \{\boldsymbol{A}, \boldsymbol{\alpha^k}\} \text{ THEN } V \text{ is } \{\boldsymbol{D}, \boldsymbol{\beta^k}\},\ \text{with a rule weight } \theta_k \text{ and the attribute weight } \boldsymbol{\delta} \tag{4}$$

where $(\boldsymbol{A}, \boldsymbol{\alpha^k})$ is the packet antecedent with belief distributions, i.e., $\left\{\left\{\left(A_{ij}, \alpha_{ij}^k\right),\ j = 1,\ldots,J_i\right\} |\ i = 1,\ldots,T\right\}$ (as illustrated in (3) representing antecedent with belief structure; here J_i is the total number of referential values of the ith antecedent attribute U_i ($i = 1,\ldots,T$), $\{A_{ij};\ j \in \{1,\ldots,J_i\}\}$ is the set of all the referential values of U_i $(i = 1,\ldots,T)$, α_{ij}^k the likelihood to which U_i is evaluated to be the referential value A_{ij} in the kth rule with $\alpha_{ij}^k \geqslant 0$ and $\sum_j^{J_i} \alpha_{ij}^k \leqslant 1$ $(i = 1,2,\ldots T)$, $k = 1,\ldots,L^*$, L^* is the total number of all the packet rules in the rule base. A rule base with rules in the form of (4) is called an *extended belief rule base* (EBRB) compared with (1). Note that all the rules in an EBRB share the same antecedent attributes and consequent attribute, but have different belief distributions for each antecedent attribute and the consequent. The use of the superscript k in (4) reflects this feature. Obviously, traditional rule base and fuzzy rule base, as well as the BRB in RIMER are the special cases of EBRB in terms of representation.

Remark 2.1. In the example of an extend belief rule (3), $T = 3$ and $\boldsymbol{U} = (U_1, U_2, U_3)$ = (failure rate, consequence severity, failure consequence probability). $\{A_{1j}\ (j \in \{1,\ldots,J_1\})\}$ = {*very low, low, reasonably low, average, reasonably frequent, frequent, highly frequent*} is the set of all referential values corresponding to the 1st antecedent attribute U_1 = failure rate, where $J_1 = 7$. Notice that $J_2 = 5$, $J_3 = 7$, respectively.

For a BRB with rules in the form of (1), the total number of rules, L, is usually determined by the following way:

$$L = \prod_{i=1}^{T} J_i. \tag{5}$$

However, the total number of rules, L^*, for an EBRB is determined by the number of given input–output sample data pairs, which is not equal to L and will be explained further in details in Section 3 about how to generate this type of extended belief rules. The following subsections will outline the steps of inference scheme based on the EBRB.

In order to illustrate the EBRB in terms of its applicability in reality, consider again the example in car evaluation. For some attributes are random in nature, e.g. fuel economy of a car can be assessed by fuel consumption that can be measured using a quantity such as how many miles a car can travel per gallon of fuel (mpg). However, fuel consumption could change under different weather and road condition, in winter could be 31 mpg in urban areas and 39 mpg in suburb; in summer it would be 35 mpg in urban areas and 43 mpg in suburb. If the car is to be used equally frequently in those conditions, then its fuel economy could be more realistically assessed during the following expectation: {(31, 0.25), (35, 0.25), (39, 0.25), (43, 0.25)}, where in (31, 0.25), 0.25 denotes the perceived frequency 25% that the car may be used in urban areas in winter. This belief distribution provides a more comprehensive and complete picture about the fuel consumption, using a single value, e.g., either 31 or 43 will not reflect the actual context. This description is useful and necessary in situation where the frequencies associated with different conditions are not equal or are not summed to one (due to lack of information). Hence, due to the randomness nature of fuel consumption, instead of using one or several referential values in the antecedent, a belief distribution could be more realistic to incorporate the randomness uncertainty into the rule. Considering the belief in the consequent, it reflects much realistic situation when IF-THEN rule being determined. In an assessment of IF-THEN relationship between two parameters, for example, a group of experts may provide the following assessment: 30% sure that the consequent is at the high level and 70% sure that it is at the medium level, which means 30% of experts voted for the high level, and 70% of the experts voted for the medium level. This kind of distribution considers all the decision makers' opinions so avoids the loss of information.

Remark 2.2. EBRB actually provides a flexibility to incorporate the context information in the knowledge-based system; the context information could be vague, uncertain, incomplete, quantitative or qualitative in nature.

2.2. Input transformation into belief distribution

In this paper we consider a single output belief rule-based system in the T-dimensional input space $\Re^T$. Let us assume an input vector is given as follows:

$$\underline{x} = (x_1, x_2, \ldots, x_T),$$

where x_i is the input for U_i ($i = 1,\ldots,T$). This section is to show the EBRB inference procedure to derive the output from $\underline{x}$ assuming that EBRB has been established.

Remark 2.3. The input x_i ($i = 1,\ldots,T$) in this methodology is not necessary to be numerical value depending on the nature of the antecedent attribute and the availability of numerical data; x_i could be a belief distribution of qualitative referential values or categorical referential terms, e.g., *high*, *medium*, *low*; *Yes* or *No*; *A*, *B*, *C* or *D*, etc.).

In general, the antecedent attributes involved in a rule could be quantitative or qualitative in nature, so that the input for each antecedent attribute may be different from each other both in types and in scales. The belief distribution embedded in the antecedent attributes facilitates different data collection and matching in a manner appropriate to a particular attribute respectively. Hence it provides the flexible way to incorporate the hybrid input information.

As the first step in the EBRB inference process, a given input x_i for the antecedent attribute U_i ($i = 1,\ldots,T$) will be firstly transformed or assessed into the belief distributed representation of the referential values of U_i as follows:

$$E(x_i) = \{(A_{ij}, \alpha_{ij});\ j = 1,\ldots,J_i\},\ \ i = 1,\ldots T \tag{6}$$

where $\{A_{ij}; j \in \{1,\ldots,J_i\}\}$ is the set of all the referential values of U_i ($i = 1,\ldots,T$), α_{ij} is the likelihood to which the input x_i for U_i belongs to the referential value A_{ij} with $\alpha_{i\,j} \geqslant 0$ and $\sum_j^{J_i} \alpha_{ij} \leqslant 1$ ($i = 1,\ldots,T$). $\alpha_{i\,j}$ in Eq. (6) could be generated using various ways depending on the nature of an antecedent attribute (i.e., type of its referential value) and also the types of the input data, which is described in the following three cases [30]:

(1) Matching function methods while the inputs are in numerical form, the antecedent attributes are quantitative in nature and their referential values are taken as linguistic terms characterized using fuzzy membership functions (FMFs). This covers certainly the case of fuzzy rule base with extended belief structure. Hence if $A_{ij}(x_i)$ is the fuzzy membership degree of a given numerical input x_i for the attribute U_i to the linguistic value A_{ij}, then the likelihood in (6) (also called *relative matching degree* in this case) to which the input x_i for U_i belongs to (or match to) A_{ij} is calculated as follows:

$$\alpha_{ij} = A_{ij}(x_i) \Big/ \sum_t^{J_i} A_{it}(x_i), \quad j = 1,\ldots,J_i; \quad i = 1,\ldots T. \tag{7}$$

The FMFs can be applied in different forms and shapes depending on the system and the application context.

(2) Rule-based or utility-based transformation methods while the input is in numerical forms, the antecedent attributes are quantitative in nature, and their referential values are represented by linguistic terms but characterized by numerical utility values using utility-based equivalence transformation techniques [26], where the equivalence rules need to be extracted from decision makers to transform a numerical value to an equivalent expectation, thereby relating a particular numerical value to each referential value. This is an alternative way when fuzzy membership function is not available (so only suits for the quantitative attributes). The input transformation is done based on the following procedure:
Suppose a utility value a_{ij} for an antecedent attribute U_i is judged to be equivalent to a referential value A_{ij} ($j = 1,\ldots,J_i$) or

$$a_{ij} \text{ means } A_{ij}\ (j = 1,\ldots,J_i). \tag{8}$$

Without loss of generality, suppose U_i is a 'profit' attribute, that is a large value $a_{i(j+1)}$ is preferred to a smaller value a_{ij}. Let a_{iJ_i} be the largest feasible value and a_{i1} the smallest. Then an input value x_{pi} for U_i can be represented using the following equivalent expectation:

$$E(x_i) = \{(A_{ij}, \alpha_{ij});\ j = 1,\ldots,J_i\}, \tag{9}$$

where

$$\alpha_{ij} = \frac{a_{i(j+1)} - x_i}{a_{i(j+1)} - a_{ij}}, \quad \alpha_{i(j+1)} = 1 - \alpha_{ij} \text{ if } a_{ij} \leqslant x_i \leqslant a_{i(j+1)} \tag{9a}$$

$$\alpha_{it} = 0 \quad \text{for } t \in \{1,\ldots,J_i\}, t \neq j, j+1 \tag{9b}$$

It has been proven that the above equivalence relations are justified if the underlying implicit utility function of the attribute U_i is assumed to be piecewise linear [26]. This type of transformation scheme covers the case of utility-based rule base with extended belief structure.

(3) Subjective assessment methods (for quantitative and qualitative attribute). In this case the referential values of each attribute are still represented by linguistic terms, but α_{ij} in (6) can be assessed based on expert judgments. This subjective assessment can be taken as an alternative solution due to the lack of information/knowledge, e.g. when neither the membership function or utility rules of each linguistic term nor numerical forms of the input are available at all, especially useful for qualitative attribute assessments, which sometimes are totally subjective, e.g., an expert may provide an assessment: 30% sure that an attribute is at the *medium* level and 70% sure at the *high* level. This type of transformation scheme covers the case of linguistic-valued rule base with extended belief structure.

Remark 2.4. from the above different input transformation schemes, we can see clearly that the EBRB covers different types of rules representations, and is generic and able to accommodate the hybrid antecedent types and inputs.

2.3. Individual matching degree

Assume that the input x_i for an antecedent attribute U_i can be assessed to a distribution representation of the referential values using belief degrees as shown in (6), considering the belief distribution of antecedent attribute in the kth rule as shown in (4), *the individual matching degree* of x_i to U_i in the kth rule, denoted as S_i^k, can be calculated based on the similarity measure between two belief distributions. Since they are actually the probability distributions (either an objective probability, e.g., relative frequency, or a subjective probability, e.g., confidence or belief), using the Euclidean distance given by

$$d^k(x_i, U_i) = d_i^k = \sqrt{\sum_{j=1}^{J_i} (\alpha_{ij} - \alpha_{ij}^k)^2}, \tag{10}$$

S_i^k can then be calculated as follows:

$$S^k(x_i, U_i) = S_i^k = 1 - d_i^k. \tag{11}$$

It follows that $S_i^k \in [0, 1]$. More options about distance measure in (10) can be found in the literatures, the choice may depend on the application context. Note that this individual matching degree here is different from the one defined in RIMER approach.

2.4. Activation weights

Suppose the "$\wedge$" connective is used for all antecedents in a rule, such as "if $A \wedge B \wedge C$". In other words, the consequent of a rule is not believed to be true unless all the antecedents of the rule are activated. In such cases, it is justified to use the simple weighted multiplicative aggregation function to calculate the combined matching degree [30]. Accordingly, for a given input x_i for the antecedent attribute U_i ($i = 1,\ldots,T$), based on the individual matching degree in (11), the activation weight w_k, which measures the degree to which the kth rule is weighted and activated, is calculated by:

$$w_k = w_k(\underline{x}) = \frac{\theta_k * \prod_{i=1}^{T} (S^k(x_i, U_i))^{\bar{\delta}_i}}{\sum_{j=1}^{L^*} \left[\theta_j * \prod_{l=1}^{T} (S^j(x_l, U_l))^{\bar{\delta}_l}\right]} = \frac{\theta_k * \prod_{i=1}^{T} \left(S_i^k\right)^{\bar{\delta}_i}}{\sum_{j=1}^{L^*} \left[\theta_j * \prod_{l=1}^{T} \left(S_l^j\right)^{\bar{\delta}_l}\right]}, \quad \text{with } \bar{\delta}_i = \frac{\delta_i}{\max_{i=1,\ldots,T}\{\delta_i\}} \tag{12}$$

where it is assumed that $\theta_k \in \Re^+$ $(k = 1,\ldots,L^*)$ and $\delta_i \in \Re^+$ $(i = 1,\ldots,T)$ θ_k and δ_i can be assigned to any value in $\Re^+$ because w_k will be eventually normalized, so that $w_k \in [0,1]$ using (12). $S_i^k \in [0,1]$ is based on (11), $i = 1,\ldots,T$. Compared with the RIMER approach [30], the formula to calculated the activation weight is different from each other mainly due to the difference in S_i^k, the rule weight θ_k (see the update of rule weights in Section 3.1), and the total number of rules $L^*(\neq L$ in (5)).

2.5. Rule inference using the ER approach

Having represented each rule by (4), similar to the RIMER approach, the evidential reasoning (ER) approach [29] is applied to combine rules and generate final conclusions. The final conclusion for an output generated by aggregating the L^* rules using the ER algorithm, which are activated by the actual input vector $\underline{x} = (x_1, x_2,\ldots,x_T)$, is also a belief distribution, can be represented as follows

$$f_D(\underline{x}) = \{(D_s, \beta_s),\ s = 1,\ldots,N\} \tag{13}$$

The ER recursive algorithm used in [29] has been equivalently transformed into the ER analytical algorithm [31]. Using it, the overall combined degree of belief β_s in D_s is generated as follows:

$$\beta_s(\underline{x}) = \frac{\mu * \left[\prod_{k=1}^{L^*}\left(w_k(\underline{x})\beta_{sk} + 1 - w_k(\underline{x})\sum_{s=1}^{N}\beta_{sk}\right) - \prod_{k=1}^{L^*}\left(1 - w_k(\underline{x})\sum_{s=1}^{N}\beta_{sk}\right)\right]}{1 - \mu * \left[\prod_{k=1}^{L^*}(1 - w_k(\underline{x}))\right]} \tag{14}$$

where $s = 1,\ldots,N$, $w_k(\underline{x})$ is given in Eq. (12) and

$$\mu = \left[\sum_{s=1}^{N}\prod_{k=1}^{L^*}\left(w_k(\underline{x})\beta_{sk} + 1 - w_k(\underline{x})\sum_{s=1}^{N}\beta_{sk}\right) - (N-1)\prod_{k=1}^{L^*}\left(1 - w_k(\underline{x})\sum_{s=1}^{N}\beta_{sk}\right)\right]^{-1}.$$

Here if $\sum_{j=1}^{N}\beta_j = 1$ or not depends on if the input or the EBRB is complete or not.

Suppose that β_D represents the remaining belief degrees unassigned to any D_j. The upper bound of the likelihood is given by $(\beta_s + \beta_D)$. It has been proved that $\sum_{j=1}^{N}\beta_j + \beta_D = 1$ [29]. The output that is shown in (13) is represented as a belief distribution, suppose the utility (or score) of an individual consequent D_s is denoted by $u(D_s)$. Then its expected average score (utility) is given by

$$f_U(\underline{x}) = \sum_{s=1}^{N} u(D_s)\beta_s(\underline{x}) \tag{15}$$

Note that $u(D_s)$ can be either given using a scale or estimated using the decision maker's preferences [26].

Remark 2.5. without loss of generality, we suppose that $u(D_{n+1}) \geqslant u(D_n)$, i.e., D_n has the lower utility so is the less preferred consequent element compared with D_{n+1}. Hence the function (15) is accordingly a bounded function, i.e., $u(D_1) \leqslant f_D(\underline{x}) \leqslant u(D_N)$.

Remark 2.6. Complementary to the distribution assessment as shown by Eq. (13), a utility interval can also be established [26] if the assessment is incomplete or imprecise, characterized by the maximum, minimum and average utilities of $f_D(\underline{x})$ defined as follows, provided that $u(D_{n+1}) \geqslant u(D_n)$:

$$u_{\max}(f_D(\underline{x})) = \sum_{j=1}^{N-1}\beta_j u(D_j) + (\beta_N + \beta_D)u(D_N) \tag{15a}$$

$$u_{\min}(f_D(\underline{x})) = (\beta_1 + \beta_D)u(D_1) + \sum_{j=2}^{N}\beta_j u(D_j) \tag{15b}$$

$$u_{avg}(f_D(\underline{x})) = \frac{u_{\max}(f_D(\underline{x})) + u_{\min}(f_D(\underline{x}))}{2}. \tag{15c}$$

These utilities are used for characterizing an assessment but not used in the aggregation process. Note that if the assessment of $f_D(\underline{x})$ is complete, then $\beta_D = 0$ and $u(f_D(\underline{x})) = u_{\max}(f_D(\underline{x})) = u_{\min}(f_D(\underline{x})) = u_{avg}(f_D(\underline{x}))$.

It should also be noted that conflicting information can be explicitly modeled using the normalized activation weight w_k and can be logically processed using the ER algorithm described earlier in this sub-section, thereby overcoming another drawback of the Dempster's original combination rule in dealing with conflicting evidence [30]. The computational complexity of reasoning using Dempster's rule based on the above specific ER decision analysis framework becomes linear rather than #P-complete [26–28] (the class #P is a functional analog of the class NP of decision problems).

As seen from Eqs. (12) and (15), the activation weights for rules and the belief degrees in each rule play an essential role in the inference procedure and determine the actual inference performance. The activation weights are also based on the actual input and the belief distribution in the antecedent, as well as the rule weights and the attribute weights. Therefore, in the following section, the EBRB generation algorithm is proposed in order to generate those beliefs embedded in the rules from the given input–output data pairs.

3. Generating EBRB from numerical data

In [23], an efficient rule generation method with no time-consuming iterative procedure was proposed and its high performance was demonstrated. In this section, based on the feature of new EBRB as shown in (4) and following the similar way used in [23], we propose a simple but efficient and powerful method for automatically generating EBRB from numerical data. The common features shared in both approaches are no need of time-consuming iterative learning procedures or complicated rule generation mechanisms. However, the procedures are very different from each other in that our belief rule base has been a great extension of traditional fuzzy if-then rule base in terms of representation power, applicability, and flexibility. In addition, the parameters to be generated and also the activation weight calculation and the inference algorithm are all different from each other.

3.1. Steps to generate EBRB

In this paper we consider a single output belief rule-based system in the T-dimensional input space $\Re^T$. Let us assume that the following M input–output pairs are given as training data for constructing an EBRB:

$$\{(\boldsymbol{x_p}; y_p)| \quad p = 1, 2, \ldots, M\}, \tag{16}$$

where $\boldsymbol{x_p} = (x_{p1}, x_{p2}, \ldots, x_{pT})$ is the input vector of the pth input–output pair and y_p is the corresponding output. The objective here is to derive a set of extended belief rules in the form of (4) from these training data pairs.

Our approach consists of the following steps.

Step 1: Determine the referential set of each antecedent and consequent attribute and their representations

To establish a rule base, one has to determine which referential set for each antecedent and consequent attribute needs to be used and how many referential values should be used. A referential value could be assigned a fuzzy membership function (the same way as it in the fuzzy rule base case). Alternatively each referential value can be assigned a numerical value using utility-based equivalence transformation techniques [26] as also detailed in Section 2.2. This provides an additional flexibility and applicability over other rule base approaches. Finally, each A_{ij} ($j \in \{1, \ldots, J_i\}$), i.e., the jth referential value of the ith antecedent attribute U_i ($i = 1, \ldots, T$), will be determined (in terms of the names, notations, and the total number J_i) and represented either in fuzzy set or utility values.

Step 2: Generate belief distribution from the given input–output data pairs

This step aims at setting up the relationship between an input (fact) and each referential value in the antecedents, as well as the relationship between the output data and each referential value in the consequent. The basic idea is to examine all the referential values of each attribute in order to determine a matching degree to which an input (output) belongs to a referential value. This is equivalent to transforming an input data (output data) into a distribution on referential values using belief degrees. This is an important step where the belief distribution plays a critical role to reflect and store the features of each input–output data. These belief rule bases which store all the sample data information will be used to predict an output once a new input is given.

Step 2-1: *Antecedent belief distribution generation from the input data*

Given the input vector $\boldsymbol{x_p} = (x_{p1}, x_{p2}, \ldots, x_{pT})$ can be assessed to a distribution representation of the referential values using belief degrees as follows:

$$E(x_{pi}) = \left\{ \left(A_{ij}, \alpha_{ij}^p \right);\ j = 1, \ldots, J_i \right\}, \quad i = 1, \ldots T; \quad p = 1, \ldots, M \tag{17}$$

where α_{ij}^p is the likelihood to which the pth input x_{pi} for U_i belongs to the referential value A_{ij} with $\alpha_{ij}^p \geqslant 0$ and $\sum_j^{J_i} \alpha_{ij}^p \leqslant 1$ ($i = 1, 2, \ldots, T$). The superscript "p" in α_{ij}^p reflects the fact that different input x_{pi} will correspond to different belief distribution. α_{ij}^p in Eq. (17) could be generated using various ways depending on the given representation schemes of the referential values as discussed in Section 2.2. An assessment $E(x_{pi})$ is complete if $\sum_j^{J_i} \alpha_{ij}^p = 1$ and incomplete if $\sum_j^{J_i} \alpha_{ij}^p < 1$.

Step 2-2: *Consequent belief distribution generation from the output data*

Consider the pth output y_p, it can be assessed to a distribution representation of the referential values using belief degrees as follows:

$$E(y_p) = \{(D_s, \beta_{sp});\ s = 1, \ldots, N\}, \quad p = 1, 2, \ldots M \tag{18}$$

where β_{sp} is the belief degree to which D_s is believed to be the consequent given the pth input vector $\boldsymbol{x}_p$, with $\beta_{sp} \geqslant 0$ and $\sum_{s=1}^N \beta_{sp} \leqslant 1$ ($s = 1, \ldots, N$; $p = 1, 2, \ldots, M$).

Remark 3.1. given the pth input vector $\boldsymbol{x_p}$, if $\sum_j^{J_i} \alpha_{ij}^p < 1$ or $\sum_{s=1}^N \beta_{sp} \leqslant 1$, that means the pth input for the ith antecedent attribute is incomplete or the pth input–output causal relationship evaluation is incomplete respectively. This leads to the fact the pth rule generated is also incomplete. Therefore, the so-called context information of the sample data or the imperfect features of the sample data are reflected in the generated EBR. Following the inference scheme in Section 2, however, one may still infer or predict the output given a new input, based on the generated imperfect EBRB. The output may not be definite, but in a certain range, as specified in (15a)–(15c). This is mostly useful and preferred in the real world application by providing the output with uncertainty range based on the imperfect known fact. This is also a unique feature of this rule generation scheme compared with the normal rule leaning algorithm.

Step 3: Create the extended belief rule base

Based on Step 2, the pth extended belief rule can be generated and represented as follows:

$$\begin{aligned} R_p^*: &\text{ IF } U_1 \text{ is } \left\{ \left(A_{1j}, \alpha_{1j}^p \right), j = 1, \ldots, J_1 \right\} \text{ and } U_2 \text{ is} \\ &\left\{ \left(A_{2j}, \alpha_{2j}^p \right), j = 1, \ldots, J_2 \right\} \text{ and} \ldots \text{and } U_T \text{ is} \\ &\left\{ \left(A_{Tj}, \alpha_{Tj}^p \right), j = 1, \ldots, J_T \right\} \text{ THEN } V \text{ is } (D_1, \beta_{1p}), \ldots, (D_1, \beta_{Np}) \end{aligned} \tag{19}$$

Remark 3.2. From Step 2 and (19), it is clear that one input–output pair will generate one extended belief rule, so p is exactly the same as the k in (4), hence $L^* = M$, i.e., the total number of generated extended belief rules is equal to the total number of the given training data pairs. Hence, the extended belief rule base consists of M extended belief rules in the form of (19). Note that each data has been transferred into a belief distribution, so it is the beliefs that store and reflect the feature of each input–output data pair. This is also a novel element added in the new and extended belief rule base compared with the belief rule base in RIMER [30], this new element makes the reflections feasible.

Step 4: Assign both rule and antecedent attribute weights

Since there are normally lots of data pairs and each one generates one rule as indicated in Step 3, it is high likely there will be some rules more important than others due to the noise, inconsistency or reliability of the data. Hence, the rule weights depend on the importance, consistency or reliability of the data pairs, along with the antecedent weights (which reflect the importance of the antecedents), will be pre-determined by the domain experts, in addition, as discussed in the Section 3.2, the consistency degree of each rule will be calculated after rules being generated and can be combined with the importance of the rules to generate the overall rule weights for inference purpose. Both rule and antecedent attribute weights will affect the performance of the system as they are involved and interact each other in determining the overall activation weight of each rule.

It follows from Step 1 to 4 that this rule generation method is simple, straightforward and quick to construct in that it is a one-pass build procedure without need of complicated or time consuming training procedure. Once the EBRB is established from the given training data, following the inference procedure detailed from Sections 2.2–2.4, the predictive output can be obtained given the new input data.

Remark 3.3. it is worth noting that (1) the size and quality of the data play a crucial role in this EBRB generation methods, actually, so did in all the learning methods generated from data; (2) one concern may be the computational complexity because the large amount of training data corresponds to large amount of rules generated. However, as justified in the end of Section 2.4 about the linear computational complexity of reasoning using the ER algorithm rather than #P-complete, and also illustrated in the case study in Section 5 that the rule generation and inference is very efficient compared with the existing optimization approach even considering the large amount of training data. The case study has shown that this learning method have the same or even better performance in term of accuracy based on the same data set compared with the time consuming optimization based training approaches, efficiency is the most shining aspect of the proposed EBRB generation method; (3) considering a set of sample data A, if a subset of A is used for generating rules, one question would be how is the subset chosen? It is worth noting that for this EBRB generation method from sample data set A, in order to test the overall performance of the EBRB generation method, the k-cross validation method under the set A is used (as illustrated in the case study in Section 5) to avoid random or ad hoc subset selection for the comparative analysis purpose. Afterwards, the EBRB generated by using all the sample data in A (called EBRB based on A, denoted as EBRB_A) will be used for prediction or classification purpose; and (4) the proposed EBRB generation method provides another simplicity and flexibility in rule updating and refining: any single new data or any new set of data generated rule or rules can be simply added into the EBRB_A, without need of re-training through the optimization algorithm; moreover, the individual rule can be easily modified or adjusted based on the expert's domain knowledge. In the generated EBRB_A, it is the beliefs that store and reflect the feature of each input–output data pair in A. This is also a novel element compared with the belief rule base in RIMER. This nature is quite similar to artificial neural network (ANN) in term of the fact the in ANN it is the weights and also the structure of network store the feature of the sample data. However, ANN is in essence a black-box modeling approach, and its internal structure does not allow the explicit representation of subjective expert knowledge.

3.2. Inconsistency of rules in EBRB

The consistency of generated rule base is actually important for a rule-based system to exhibit a reliable performance because the inconsistency generally exists in the process of knowledge representation and acquisition. The consistency of the rules is usually thought to be trivial if the rules are extracted from expert knowledge. However, if the rules are automatically generated from a set of data affected by noise, this can become serious. In this subsection, the measurement of EBRB inconsistency is provided and then incorporated into the rule weights, which can improve the performance of the EBRB system to reduce the inconsistency via adjusted the rule weights.

3.2.1. Similarity measures

Rules are regarded as inconsistent, if they have very similar premise parts, but possess rather different consequents; or they conflict with the expert knowledge or heuristics. Before we discuss the consistency, we first provide the similarity of rule antecedent (SRA) and the similarity of rule consequent (SRC). Because the rule antecedent and the consequent of a rule in the EBRB are both represented as a probability distribution, it is to determine the similarity measure for probability distributions.

Considering two probability distributions $P = (p_1, p_2, \ldots, p_n)$ and $Q = (q_1, q_2, \ldots, q_n)$, to calculate the similarity $S(P, Q)$, we use the Euclidean distance given by

$$d(P,Q) = \sqrt{1 \sum_{i=1}^{n} (p_i - q_i)^2},$$

So

$$S(P,Q) = 1 - d(P,Q). \tag{20}$$

3.2.2. Consistency measure of EBRB

Consider two rules R_i and R_k in the EBRB:

R_i: IF U_1 is $\left\{\left(A_{1j}, \alpha^i_{1j}\right),\ j = 1,\ldots,J_1\right\}$ and U_2 is $\left\{\left(A_{2j}, \alpha^i_{2j}\right),\ j = 1,\ldots,J_2\right\}$ and $\ldots$ and U_T is $\left\{\left(A_{Tj}, \alpha^i_{Tj}\right),\ j = 1,\ldots,J_T\right\}$ THEN V is $\{(D_1, \beta_{1i}), \ldots, (D_N, \beta_{Ni})\}$;
R_k: IF U_1 is $\left\{\left(A_{1j}, \alpha^k_{1j}\right),\ j = 1,\ldots,J_1\right\}$ and U_2 is $\left\{\left(A_{2j}, \alpha^k_{2j}\right),\ j = 1,\ldots,J_2\right\}$ and $\ldots$ and U_T is $\left\{\left(A_{Tj}, \alpha^k_{Tj}\right),\ j = 1,\ldots,J_T\right\}$ THEN V is $\{(D_1, \beta_{1k}), \ldots, (D_N, \beta_{Nk})\}$.

Here $\left\{\left(A_{tj}, \alpha^i_{tj}\right),\ j = 1,\ldots,J_t\right\}$ (denoted as $\boldsymbol{A}^i_t$) and $\left\{\left(A_{tj}, \alpha^k_{tj}\right),\ j = 1,\ldots,J_t\right\}$ (denoted as A^k_t) represent the belief distributions for the tth attribute in the ith rule and the kth rule respectively; $\{(D_1, \beta_{1i}), \ldots, (D_N, \beta_{Ni})\}$ (denoted as $\boldsymbol{D}^i$) and $\{(D_1, \beta_{1k}), \ldots, (D_N, \beta_{Nk})\}$ (denoted as $\boldsymbol{D}^k$) represent the belief distributions for the consequent attribute in the ith rule and the kth rule respectively.

The SRA and SRC of these two rules are defined as follows respectively:

$$\text{SRA}(i,k) = \min_{t=1}^{T} S\left(\boldsymbol{A}^i_t, \boldsymbol{A}^k_t\right) \tag{21}$$

$$\text{SRC}(i,k) = S(\boldsymbol{D}^i, \boldsymbol{D}^k) \tag{22}$$

S is given in (20). Then the consistency of two rules R_i and R_k is defined by [9]:

$$Cons(R_i, R_k) = \exp\{-(\text{SRA}(i,k)/\text{SRC}(i,k) - 1.0)^2/(1/\text{SRA}(i,k))^2\} \tag{23}$$

An inconsistency degree for the ith rule is calculated as follows:

$$\text{Incons}(i) = \sum_{\substack{1 \leqslant k \leqslant L \\ k \neq i}} [1.0 - \text{Cons}(R_i, R_k)], \quad i = 1, \ldots, L^*, \tag{24}$$

which can be incorporated in the rule weight of EBRB. So the updated rule weight can be calculated as follows:

$$\theta^*_p = \theta_p + \lambda(1 - \text{Incons}(p)/\xi_{\text{Incons}}) \tag{25}$$

Without loss of generality, here θ_p can be normalized into [0,1] ($p = 1, \ldots, L^*$), λ is a weighting constant to control the consistency level over θ_p. It will be predetermined according to the application context. ξ_{Incons} indicates the inconsistency degree of a rule base and

$$\xi_{\text{Incons}} = \sum_{i=1}^{L*} \text{Incons}(i). \tag{26}$$

If no importance weight of rules are initially specified, then θ_p in (25) is taken as zero, the consistency measure $(1 - \text{Incons}(p)/\xi_{\text{Incons}})$ will be taken as the weight for the pth rule to reflect the reliability of the pth rule in terms of consistency.

4. EBRB inference system as a universal approximator

The four step procedure of the last section generates an EBRB which can be regarded as a mapping from input space to output space. For the general T-input one output case, assume that the input are defined on a compact set $Q \subset \Re^T$, we could have the following existence theorem showing that there exists a way of defining the input–output region, a way of choosing membership function or utility function, and a way of assigning the belief distribution of the rule base, such that the resulting mapping (15) approximates an arbitrary nonlinear continuous mapping from Q to $\Re$ to any accuracy.

The following Theorem 1 is similar to the result of [23] about a fuzzy system as a universal approximator, as well as the one of [8], regarding a three-layer feed-forward neural network as a universal approximator provided that there are sufficiently large number of hidden-layer neurons. Theorem 1 provides the theoretical basis for successful applications of the proposed EBRB inference method to many different practical problems.

Theorem 1. *Assume the mapping defined by (15) l-defined, i.e., for any input $\underline{x} = (x_1, x_2, \ldots, x_T) \in Q$, (15) generate an output $f(\underline{x}) \in \Re$, and also the following assumptions are true:*

AS.1. The utility values or fuzzy regions corresponding to the referential values for the input and output spaces can be arbitrarily defined (i.e., we have total freedom in defining the utility value or dividing the domain interval into fuzzy region for the referential values).

AS.2. The membership function (or the utility functions) can be any continuous function which satisfies that the matching degree of an antecedent cannot be equal to zero if the actual input value of this antecedent falls into the required region of the rule.

AS.3. Any rule can be assigned to a EBRB, i.e., there is no consequent with zero belief distribution.

Then the mapping defined by (15) is capable of approximating any real continuous function over the compact set Q to arbitrary accuracy.

Proof of Theorem 1. Part 1: Let F be the family of functions in the form (15) on the compact set Q under AS.1, AS.2 and AS.3, to analyze the properties of the function family F, it is necessary to establish that the mapping defined by (15) is well defined, i.e., for any input $\underline{x} = (x_1, x_2, \ldots, x_T) \in Q$, (15) will generate an output $f(\underline{x}) \in \Re$.

Proof of Part 1: Note that for any input $\underline{x} \in Q$, if all the individual matching functions S_i^k $(i = 1, \ldots, T;\ k = 1, \ldots, L^*)$ in (11) are non-zero, and there is at least one rule in the EBRB, then there exists at least one rule, say the kth rule, the activation weight $w_k(\underline{x}) \neq 0$ for $\underline{x} \in Q$. In addition, there is no consequent with zero belief distribution (AS.3), so assume we have $u(D_N) > 0$, $\beta_{sk} = 1$ when $s = N$ and $\beta_{sk} = 0$ when $s \neq N$, $k = 1, \ldots, L^*$. Then the output of the EBRB system $f(\underline{x})$ in (15) will be equal to $u(D_N) \neq 0$ for any $\underline{x} \in Q$. Therefore, the mapping defined by (15) from Q to $\Re$ is well-defined.

Part 2: If we could prove that (1) F is an algebra of real continuous functions (i.e., F is closed under addition, multiplication, and scalar multiplication); (2) F separate points on Q (i.e., for any $\underline{x}, \underline{z} \in Q$ and $\underline{x} \neq \underline{z}$, there exists a $f \in F$ such that $f(\underline{x}) \neq f(\underline{z})$); and (3) F vanishes at no point of Q (i.e., for any $\underline{x} \in Q$, there exists a $f \in F$ such that $f(\underline{x}) \neq 0$), then the Stone-Weierstrass Theorem [20] guarantees the conclusion of Theorem 1, i.e., it follows from the above 3 assumptions and the Stone-Weierstrass Theorem that the uniform closure B of F consists of all real continuous functions on Q. The uniform closure B of F is the union of F and its limit points; hence, if B consists of all real continuous functions on Q, then F is capable of approximating any real continuous function on Q to arbitrary accuracy.

For notational convenience, the numerical output function (15) of EBRB system can be re-written as follows:

$$f(\underline{x}) = \frac{\sum_{s=1}^{N} u(D_s)\eta_s(\underline{x})}{\sum_{n=1}^{N} \eta_n(\underline{x})} \tag{27}$$

where $\eta_m(\underline{x}) = \prod_{k=1}^{L^*}(w_k(\underline{x})\beta_{mk} + 1 - w_k(\underline{x})) - \prod_{k=1}^{L^*}(1 - w_k(\underline{x}))$, $m = 1, \ldots, N$.

Proof of Part 2 (1): Firstly, we will prove that F is a family of real continuous function. By AS.2, the matching functions are assumed to be real continuous functions. For any input vector $\underline{x}$, $w_k(\underline{x})$ is calculated based on Eq. (12) by the normalized multiplicative individual matching function defined in Eq. (11), so $w_k(\underline{x})$ is a real continuous function. From the form of $\eta_m(\underline{x})$ in (27), it is a polynomial form of the $w_k(\underline{x})$. According to the new form of the output function $f(\underline{x})$ of the EBRB system in (27), it follows that F is a family of real continuous functions.

Now we prove that F is an algebra of real continuous functions. For any input vector $\underline{x}$, let $f_1, f_2 \in F$, so that we can write each of them as:

$$f_1(\underline{x}) = \frac{\sum_{s_1=1}^{N_1} u^1(D_{s_1})\eta_{s_1}^1(\underline{x})}{\sum_{n_1=1}^{N_1} \eta_{n_1}^1(\underline{x})} \quad \text{and} \quad f_2(\underline{x}) = \frac{\sum_{s_2=1}^{N_2} u^2(D_{s_2})\eta_{s_2}^2(\underline{x})}{\sum_{n_2=1}^{N_2} \eta_{n_2}^2(\underline{x})}.$$

Hence

$$\begin{aligned} f_1(\underline{x}) + f_2(\underline{x}) &= \frac{\sum_{s_1=1}^{N_1} u^1(D_{s_1})\eta_{s_1}^1(\underline{x})}{\sum_{n_1=1}^{N_1} \eta_{n_1}^1(\underline{x})} + \frac{\sum_{s_2=1}^{N_2} u^2(D_{s_2})\eta_{s_2}^2(\underline{x})}{\sum_{n_2=1}^{N_2} \eta_{n_2}^2(\underline{x})} \\ &= \frac{\sum_{s_1=1}^{N_1}\sum_{s_2=1}^{N_2} [u^1(D_{s_1}) + u^2(D_{s_2})]\eta_{s_1}^1(\underline{x})\eta_{s_2}^2(\underline{x})}{\sum_{n_1=1}^{N_1}\sum_{n_2=1}^{N_2} \eta_{n_1}^1(\underline{x})\eta_{n_2}^2(\underline{x})}. \end{aligned} \tag{28}$$

Now suppose $\eta_{s_1}^1(\underline{x})\eta_{s_2}^2(\underline{x})$ corresponds to a new belief output function of $\underline{x}$ (because $\beta_s(\underline{x}) = \frac{\eta_s(\underline{x})}{\sum_{n=1}^{N}\eta_n(\underline{x})}$), and suppose that $u^1(D_{s_1}) + u^2(D_{s_2})$ is the utility function of the new rule, then (28) is of the form of (27). Hence, $f_1 + f_2 \in F$. Similarly, $f_1(\underline{x})f_2(\underline{x})$ can be written as:

$$f_1(\underline{x})f_2(\underline{x}) = \frac{\sum_{s_1=1}^{N_1}\sum_{s_2=1}^{N_2} u^1(D_{s_1})u^2(D_{s_2})\eta_{s_1}^1(\underline{x})\eta_{s_2}^2(\underline{x})}{\sum_{n_1=1}^{N_1}\sum_{n_2=1}^{N_2} \eta_{n_1}^1(\underline{x})\eta_{n_2}^2(\underline{x})},$$

which is again of the form (27); hence $f_1 f_2 \in F$. Finally, for any $c \in \Re$ and $f \in F$, we have

$$cf(\underline{x}) = \frac{\sum_{s=1}^{N} c \cdot u(D_s)\eta_s(\underline{x})}{\sum_{n=1}^{N} \eta_n(\underline{x})},$$

which is also of the form of (27). Hence $cf \in F$. In summary, F is an algebra of real continuous functions.

Proof of Part 2 (2): we prove that F separates points on Q, i.e., for any $\underline{x}, \underline{z} \in Q$ and $\underline{x} \neq \underline{z}$, there exists a $f \in F$ such that $f(\underline{x}) \neq f(\underline{z})$. Because (15) or (27) is well-defined, which guarantees that there is at least one active rule for $\underline{x}$ and at least one active rule for $\underline{z}$. Since $\underline{x} \neq \underline{z}$, there must be at least one j such that $x_j \neq z_j$, hence the only active rule for $\underline{x}$ and the only active rule for $\underline{z}$ have different consequents belief distribution. Since we are free to assign any rules to a consequent with non-zero belief distribution of the EBRB (according to AS.3), we just assign two different rules to these two consequent belief distributions, and obtain the required $f \in F$ with $f(\underline{x}) \neq f(\underline{z})$. Actually, without loss of generality, considering a single-input EBRB system with two different rules as follows:

R_1: IF U is $\{\boldsymbol{A}, \boldsymbol{\alpha}^1\}$ THEN V is $\{\boldsymbol{D}, \boldsymbol{\beta}^1\}$ with $\beta_{11} = 1$;
R_2: IF U is $\{\boldsymbol{A}, \boldsymbol{\alpha}^2\}$ THEN V is $\{\boldsymbol{D}, \boldsymbol{\beta}^2\}$ with $\beta_{N2} = 1$.

Without loss of generality, assume that $\underline{x}, \underline{z} \in Q$, $\underline{x} \neq \underline{z}$. It is easy to prove that $f(\underline{x}) \neq f(\underline{z})$ based on the above EBRB.

Proof of Part 2 (3): we prove that F vanishes at no point of Q, i.e., for any $\underline{x} \in Q$, there exists a $f \in F$ such that $f(\underline{x}) \neq 0$. By AS.1 and AS.2, we can make all the $f(\underline{x}) > 0$. Since (15) is well-defined, there exists at least one i such that the individual matching degree is not equal to zero for any $\underline{x} \in Q$, which leads to non-zero weight in (12), the result $f(\underline{x})$ should be also positive for any $\underline{x} \in Q$. Actually, according to Eqs. (14) and (15), we can construct a EBRB system where $u(D_N) > 0$, $\beta_{sk} = 1$ when $s = N$ and $\beta_{sk} = 0$ when $s \neq N$, $k = 1, \ldots, L^*$. So the output of the EBRB system $f(\underline{x})$ will be equal to $u(D_N) \neq 0$ for any $\underline{x} \in Q$.

This completes the proof of Theorem. □

Theorem 1 has shown the EBRB Inference System as a Universal Approximator which shows the soundness of the methodology and provides the theoretical basis for successful applications of this new method to many different practical problems. However, this universal approximation theorem is an existence theorem to guarantees that there exists a way of defining reference values in the rule base, a way of choosing membership functions or utility functions, and a way of assigning belief structure of the rule base, such that the resulting mapping, (15), approximating an arbitrary nonlinear continuous mapping from Q to $\Re$ to any accuracy.

5. Case study

Two case studies in oil pipeline leak detection and software defect detection are provided in this section to illustrate the proposed new rule base representation, generation, and inference procedure as well as demonstrate its high performance and efficiency by comparing with some existing approaches.

5.1. Oil pipeline leak detection

5.1.1. Problem description and data set

An example for oil pipeline leak detection in [24] is used here. We use the training data to generate EBRB and then the testing data for detecting and estimating the leak sizes. A pipeline with the mass flow meters at the inlet and outlet and the pressure meters at the inlet and outlet. Data are collected from those meters every 10 s. The pipeline is mostly operated in leak free (normal) condition. However, during a leak trial period, a series of leaks were created in the pipeline. The inlet and outlet flow and pressure readings were collected in a period of time slot. The difference between inlet flow and outlet flow, called Flow Difference (FD), and the average pipeline pressure change over time, called Pressure Difference (PD), are the two very important factors in detecting whether there is a leak in the pipeline. Therefore, they are the two antecedent attributes of the rule base. The consequent attribute is the Leak Size (LS). LS values are controlled during the leak trial and therefore are more or less (though not exactly) known.

We have chosen the following terms (words in natural language) for each type of numerical attribute as the referential values.

FD partition: $\{A_{1j}; j = 1, \ldots, 8\}$ = {NL: Negative large; NM: Negative medium; NS: Negative small; NVS: Negative very small; Z: Zero; PS: Positive small; PM: Positive medium; PL: Positive large}. Based on the range of total sample data set and the expert knowledge, the following equivalence rules are extracted to transform a numerical value to an equivalent expectation, i.e., A_{1j} $(j = 1, \ldots, 8)$ as the referential values for FD are assumed to be equivalent to the numerical value −11, −6, −3, −1, 0, 1, 2, and 3, respectively.

PD partition: $\{A_{2j}; j = 1, \ldots, 7\}$ = {NL: Negative large; NM: Negative medium; NS: Negative small; Z: ZERO; PS: Positive small; PM: Positive medium; PL: Positive large}. Similarly, the following equivalence rules are extracted to transform a numerical value to an equivalent expectation, i.e., A_{2j} $(j = 1, \ldots, 7)$ referential values for PD are assumed to be equivalent to the numerical value −0.061, −0.005, −0.002, 0, 0.005, 0.01, and 0.06, respectively.

We have discretized the leak size to five levels, i.e., $D = \{D_1, \ldots, D_5\}$ = {N: none; VS: very small; M: medium; F: fair; L: large}, which are supposed to be corresponding to the leak size 0, 2, 4, 6, and 8, respectively.

Notice that in this case study the input is in numerical forms, the antecedent attributes are quantitative in nature, and their referential values can be represented by linguistic terms but characterized by numerical utility values where the equivalence rules (8) need to be extracted from decision makers to transform a numerical value to an equivalent expectation, thereby relating a particular numerical value to each referential value, as illustrated above for FD, PD and Leak Size. Therefore rule-based or utility-based transformation methods discussed in Section 2.2 is utilized to transform any input into belief distribution. The more elaborated work to compare the using of fuzzy membership function instead or further analysis of the equivalence rules are not the focus of this paper, will be discussed in the future work. The utility based approached actually work well in this application as illustrated in the following subsection.

5.1.2. Generation of EBRB from training data for leak detection

In order to compare with the some existing approaches (e.g., including the original RIMER approach) and also to demonstrate the performance of the proposed EBRB method, we consider two ways of selecting the training data and testing data: one way is to follow exactly the same way as done in [24,34,35] while using the original RIMER approach but with different rule base generation methods for the comparison purpose; the other way is to consider more but different data set for training and testing in order to illustrate the performance of the EBRB method against the method in [23].

Case 1: In the existing learning models based on the RIMER, such as [24] (model denoted as M1), [34] (model denoted as M2), and [35] (model denoted as M3), the authors used the same experimental data setting (the leak data is completely same to the one used by Xu et al.[24]: during the leak trial, 2008 samples of 25% leak data were collected at the rate of 10 s per sample, they are the data collected in about 5 and half hours during a leak trail from 7 am to 12:30 pm, with the leak period clearly marked by the large discrepancy between the inlet and outlet flow readings. In order to train the rule base, 500 data samples are selected and about half of them are collected during the leak period. They are the data collected in the three periods of 7 a.m. to 7:33 a.m., 9.46 a.m. to 10:20 a.m. and 10:50 a.m. to 11:08 a.m. [24]. For testing the trained belief rules, all the 2008 samples are used. In this work, we use the same training and test data set to generate and test the proposed EBRB model (denoted as M4), and then compare it with the performance of M1, M2, M3, as well as the fuzzy rule-based model by Wang and Mendel [23] (denoted as M5).
Case 2: during the leak trial, samples of 4%, 16% and 25% leak data were collected at the rate of 10s per sample, respectively. For the sake of comparison, in order to generate the EBRB expert system for pipeline leak detection using the proposed algorithm in Section 3, 3733 data pairs from 8236 sample data are selected as training data, which include 853 from 25% leak, 880 from 16% leak and 2000 from 4% leak data sets. For testing the generated EBRB, the remaining 4503 data pairs from 8236 sample data is taken as the first test set, and on the other hand, 1135 data pairs specially extracted from 25% data set is used as the second test set (because of more chances of leak). For this case, we only consider the comparison between M4 and M5.

Here we consider the utility-based transformation method approaches discussed in Section 2.2 and follow Steps 2.1 and 2.2 in rule generation procedure, those belief distributions are obtained via (9), (16) and (17) based on the utility value of each linguistic term given in Section Section 5.1. All the other rules are generated in the similar way based on the input–output data pairs. Considering a total of 3733 training data pairs, it will lead to 3733 rules generated in the EBRB. For example, for the kth data pair: (FD, PD; LS) = (−4.35, −0.02; 3.656255), the generated extended belief rule is shown as follows:

R_k: IF FD is {(NM, 0.675), (NS, 0.325)} AND PD is {(NL, 0.2), (NM, 0.8)} THEN LS is {(VS, 0.1917), (M, 0.8281)}.

The process of generation and testing the EBRB is implemented using a Java-based EBRB inference system [4].

5.1.3. Performance comparison

Quite a few criteria have been defined in the literature to evaluate the performance of a model, where mean absolute error (MAE), mean square error (MSE), root mean squared error (RMSE), mean absolute percentage error (MAPE) are the most widely used performance evaluation criteria and were used in this study. The model with the smallest MAE, MSE, RMSE and MAPE is the best. The most commonly reported error measurement is the MAE/RMSE. The RMSE is more sensitive to outliers in the data than the MAE. In order to minimize the effect of outliers on the classification methods, we choose MAE as the standard error model in the following analysis; others are also provided in some case study as additional information.

M1–M3 models all started from the initial rule-based provided by the expert, 500 data (for M1, M2, M3) are selected respectively as the training data for learning or updating the belief rule base, the calculated times (i.e., the training and testing time in minutes) of three algorithms have been 300 (for M1), 20 (for M2), and 2 min (for M3) respectively [35]. M1, M2 and M3 have all used the initial rule base provided by the expert, which essentially all need extra time spent on initial rule generation, but were not counted in the calculated times above though, however this initial rule base have indeed saved a lot of training or updating time compared with the randomly given or the initially empty belief rule base.

Comparatively, we used the same training and test data set, there is no any initial rule base needs to be generated, just starts from the empty rule base. The time counting for loading the 500 rules for 500 training data (i.e., the time for generating rules) is less than 1 s (0.21 s), the execution time for the testing for 2008 data pairs is less than 40 s (31.7 s) on the PC with Inter® Pentium® 4 CPU @3.40 GHz Processor with 2 GB RAM. In the study of Case 2 data set, we used 3733 training data and two sets of testing data with the number 1135 and 4503 respectively. The time counting for loading the 3733 rules for 3733 training data (i.e., the time for generating rules) is less than 1 s (0.55 s), the execution time for the testing for 1135 data pairs is less than 30 s (20.6 s) and the testing for 4503 data pairs for less than 50 s, In addition, the execution time for 5-cross validation for M4 is less than 1 and half minutes and less than 3 min for 10-cross validation. It is clear that one input–output pair will generate one extended belief rule, there is no iterative step involved, and the complexity is obvious much less than the normal optimization based learning approach. Therefore the rule generation and the execution of test are highly efficient and much suitable for on-line or real-time application. Overall, the EBRB model outperforms M1–M3 in terms of efficiency. This is mainly attributed to the new belief representation and inference mechanism and the simplicity of rule generation approach.

The following provides a performance comparison in terms of accuracy.

Case 1: The performances of the 5 models mentioned above are summarized in Table 1 regarding MAE, from which it is quite clear that the EBRB model achieves much better performance (accuracy) than the other models. It is worth noting that the inconsistency measurement and adjustment into the rule weight also takes apparent effects on the accuracy of the proposed EBRB model.

Case 2: Table 2 summarizes the performance compared between M4 and M5 regarding different evaluation criteria, from which it is quite clear that the EBRB model achieves much better performance than M5. The efficiency comparison is similar to each other.

The MAE for M4 in Tables 1 and 2 reflects the mean absolute error between the actual and predicted LS. It demonstrates that the predicted outcomes from M4 match the real ones very closely, and can detect the leak in a high accurate rate. Overall, the EBRB model (M4) outperforms M1–M3 in terms of performance and efficiency, and M4 outperforms M5 in terms of the overall performance. This is mainly attributed to the new belief representation and inference mechanism and the simplicity of rule generation approach.

To illustrate further the performance of the proposed methodology, we conducted another case study in software defection prediction where more data sets are used and more existing approaches are compared with the proposed methods to illustrate the performance in terms of the accuracy and efficiency). See Section 5.2 for more details.

5.2. Software defect prediction

5.2.1. Problem description and public domain defect data

Statistical, machine learning, and mixed techniques are widely used in the literature to predict software defects. Software defect prediction is the task of classifying software modules into fault-prone and non-fault-prone by means of metric-based classification [3,7]. Early discovery of software errors is very important for the quality of a software system and may cause significant cost savings, especially for large and complex systems. Accurate estimates of defective modules may help software developers in terms of allocating the limited resources and thus, decreasing testing times. So in this section, the proposed EBRB method will be applied on the software defects prediction which will be compared with some existing approaches based on the same data set in order to illustrate its performance.

We used the public domain defect data from the CM1, JM1, PC1, KC1 and KC2 accessed from the NASA's MDP (Metric Data Program) data repository (available online at http://mdp.ivv.nasa.gov) as shown in Table 3. There are different types of predictor software metrics (independent variables) used in the analysis. These complexity and size metrics include well known metrics, such as Halstead, McCabe, line count, operator/operand count, and branch count metrics. Halstead metrics are sensitive to program size and help in calculating the programming effort in months. The different

Table 1
Performances comparison of several models.

Training data Testing data	500 2008 Testing data	500	500	500	2008 Data (10 cross validation)
Models	M1	M2	M3	**M4**	**M4**
MAE	0.2245	0.4227	0.1732	**0.1653**	**0.2169**

Table 2
Performances comparison of M4 and M5 using different data set.

Training data	3733			
Testing data	1135		4503	
Models	M4	M5	M4	M5
MAE	0.1796	0.3616	0.2058	0.2504
MSE	0.0829	0.5292	0.091608	0.136987
RMSE	0.2879	0.7275	0.302668	0.356353
MAPE	8.014	26.46	34.14928	74.71288

Table 3
Data sets used in this study.

Project	# Modules	% With defects	Language	Notes
CM1	506	9.5	C	A NASA spacecraft instrument
JM1	10,879	19.3	C++	Real-time predictive ground system
KC1	2108	15.4	C++	Science data processing
PC1	1108	6.8	C	Flight software system

Halstead metrics include length, volume, difficulty, intelligent count, effort, error, and error estimate. McCabe metrics measure code (control flow) complexity and help in identifying vulnerable code. The different McCabe metrics include cyclometric complexity, essential complexity, design complexity and lines of code. The target metrics (dependent variables) are the "Error Count" and "Defects". "Error Count" refers to the number of module errors associated with the corresponding set of predictor software metrics, while "Defect" metric refers to whether the module is fault-prone or fault-free. In order to identify reliable classification algorithms, Challagulla et al. [5] recommended trying a set of different predictive models. Based on the results from [5], in this paper those data sets are again used to validate and compare the proposed EBRB approach with those predictive models.

5.2.2. Prediction techniques via EBRB

In [5] the WEKA machine learning tool kit (http://www.cs.waikato.ac.nz/ml/weka/) was used to conduct these experiments. It is important to note that the prediction techniques used to comparative purpose in this paper are limited only by the available techniques in WEKA. We evaluated empirically the accuracy of predicting defects using machine learning and statistical prediction systems covered in WEKA. Firstly 70% of the data as training data and 30% as the test data, this follows the same way as in [5]. For more precise comparison, we also used 10-folder cross validation. Our input attributes (input data) are treated as continuous values, while the output takes discrete or continuous values depending on the classifier used. In order to compare them in the same framework, we assume that the target metric should be of discrete type. It concluded in [5] that the better way is to analyze the modules as faulty or faultless. A module with one or more defects is considered faulty. In this way, we have only two classes of data which facilitates the application of classification prediction techniques.

Due to space constraints, we use abbreviations in the table to represent different predictive methods: Learning Method (LM), Mean Absolute Error (MAE), Linear Regression (LR), Pace Regression (PR), Support Vector Regression (SVR), Neural Network for continuous goal field (NNC), Support Vector Logistic Regression (SVLR), Neural Network for discrete goal field (NND) with 12 hidden layers, Logistic Regression (LoR), Naïve Bayes (NB), Instance Based Learning for 10 nearest neighbors (IBL), J48 Trees (JDT), and 1-Rule (1R).

The process of generation and testing the EBRB is implemented using a Java-based EBRB inference system [4]. Here we consider the utility-based transformation method approaches discussed in Section 2.2 and follow Steps 2.1 and 2.2 in rule generation procedure, those belief distributions are obtained via (9), (16) and (17) based on the utility value of each linguistic term given in Section Section 5.1. All the other rules are generated in the similar way based on the input–output data pairs.

5.2.3. Performance comparison

As mentioned earlier, quite a few criteria have been defined in the literature to evaluate the performance of a model; the WEKA software computes the MAE, RMSE, relative absolute error (RAE), and root relative squared error (RRSE). In order to minimize the effect of outliers on the classification methods, we choose MAE as the standard error model in all the following analysis.

The data sets we used are amenable to comparison because they belong to the same application domain meaning that they solve similar problems: they are developed with similar methods and implementation environments, and they are developed by teams with similar technical background and procedures. Table 4 shows the errors (regarding MAE) predicted by different predictor techniques on different data sets as faulty or faultless, from which it is quite clear that the EBRB model achieves much better performance than the other models. In addition, we notice that there is no consistency in the rank of other learning techniques in terms of prediction accuracy across different data sets. However, as shown in Table 4 EBRB shows consistently better performance than other methods.

In Table 4, the results before the last row are based on the analysis that the entire data set is split into two independent data sets, 70% as training set and 30% as test set, which is known as split-sample/partition methodology, is a commonly used approach for evaluating the classification accuracy of different predication techniques. However, there are other widely used data sampling methods, including cross-validation, jackknifing, and bootstrapping. However, as compared in [5] about the error estimates obtained on different data sets by using the next most commonly used 10-fold cross validation methodology with the split-sample methodology, the error estimates for both methodologies are nearly the same for all the data sets except for NND, NB of the CM1 data set and SVLR, NB of PC1 data set. The variations of these techniques are attributes to the smaller size of CM1 and PC1 data sets. However, as shown in the last row of Table 4 about 10-fold cross validation results from EBRB, it shows much better performance than almost all the other methods.

Moreover, the time counting for loading the rule is very short, for example, for 4000 training data (i.e., the time of generating 4000 rules) is less than 1 s, on the PC with Inter® Pentium® 4 CPU @3.40 GHz Processor with 2 GB RAM, hence the rule generation is highly efficient and much suitable for on-line or real-time application.

Table 4
Errors shown by different models on the predicting "defect" of the CM1, JM1, KC1, and PC1 data sets.

Learning Method (LM)	CM1 (MAE)	JM1 (MAE)	KC1 (MAE)	PC1 (MAE)
SVLR	0.159	0.2844	0.2088	0.111
NND	0.184	0.2648	0.1944	0.1123
LoR	0.1569	0.2834	0.2031	0.1059
NB	0.1311	0.191	0.1638	0.092
IBL	0.1567	0.269	0.1913	0.089
JDT	0.1752	0.275	0.2066	0.0895
1R	0.1184	0.2013	0.169	0.0631
EBRB (split-sample)	**0.1538**	**0.0352**	**0.169**	**0.059**
EBRB (10-fold cross)	**0.097**	**0.1928**	**0.1542**	**0.0689**

Overall, the EBRB model outperforms others in terms of performance and efficiency. This is mainly attributed to the new belief representation and inference mechanism and the simplicity of rule generation approach.

6. Conclusions

A new belief rule base representation, called EBRB, was proposed as a great extension of traditional fuzzy rule based systems and also a further improvement and extension of the belief rule base introduced in [30] in terms of flexibility, applicability and predictive performance. Specially, the major advantage of this EBRB is that it offers and facilitates a very simple and efficient rule base generation approach with high performance from the given sample data, which was shown having sound theoretical foundation as a universal approximator. It is worth noting that the distinct feature of this proposed EBRB model leads to decision attributes definitions, rule base representation and generation, inference can be designed and implemented in an integrated system, no need to link to a stand-alone training package.

The two case studies in a real world application to oil pipe leak detection and software defect detection have shown the high efficiency and consistently better performance compared with some existing approaches. As traditional rule base including fuzzy rule base as well as belief rule base in RIMER are all special cases of the EBRB, we believe that such more general, flexible, efficient and effective rule base representation, inference and generation system is more acceptable in more complex systems including online or real time system.

It is worth noting that the inconsistency measurement and adjustment also play an important role in the proposed EBRB model in term of accuracy. Inconsistency is often aggravated when dealing with highly dimensional data or data with noise, where different rules belonging to several classes are activated together for each set of inputs. Although in the proposed EBRB model, the inconsistency has been handled in certain level, but more elaborate work to fit with the dynamic situation and large-scale data set are expected to be done and reported in the future work.

Rule base updating is also an interesting issue as well to be investigated to fit with the dynamic situation, it is easy to see that the proposed EBRB actually already provided a much easier way to update the rule base when new sample data added, that is simply to add a new rule generated from this new data. It possible to add each new test data result into the generated rule base iteratively in order to obtain a better overall performance, however, this issue would be investigated and reported in the future work.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China (Grant Nos. 60875034, 61175055), the project TIN2012–31263.

References

[1] P. Angelov, R. Yager, A simple fuzzy rule-based system through vector membership and kernel-based granulation, in: Proc. of 2010 IEEE Int. Conference on Intelligent Systems, London, UK, 7–9 July, 2010, pp. 349–354.

[2] 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, International Journal of Computational Intelligence Systems 1 (4) (2008) 361–378.

[3] L.C. Briand, W.L. Melo, J. Wust, Assessing the applicability of fault-proneness models across object-oriented software projects, IEEE Transactions on Software Engineering 28 (7) (2002) 706–720.

[4] A. Calzada, J. Liu, H. Wang, L. Martinez, An intelligent decision support tool based on belief rule-based inference methodology, in: Proceeding of 2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE2011), Taipei, Taiwan, June 27–30, 2011, pp. 2638–2643.

[5] V.U.B. Challagulla, F.B. Bastani, I.-L. Yen, R.A. Paul, Empirical assessment of machine learning based software defect prediction techniques, International Journal on Artificial Intelligence Tools 17 (2) (2008) 389–400.

[6] Y.W. Chen, J.B. Yang, D.L. Xu, Z.J. Zhou, D.W. Tang, Inference analysis and adaptive training for belief rule based systems, Expert Systems with Applications 38 (10) (2011) 12845–12860.

[7] J. Demsar, Statistical comparisons of classifiers over multiple data sets, Machine Learning Research 7 (2006) 1–30.

[8] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural Network 2 (1989) 359–366.

[9] Y.C. Jin, W. von Seelen, B. Sendhoff, On generating FC^3 fuzzy rule systems from data using evolution strategies, IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics 29 (6) (1999) 829–845.

[10] B. Li, H.W. Wang, J.B. Yang, M. Guo, C. Qi, A belief-rule-based inventory control method under nonstationary and uncertain demand, Expert Systems with Applications 38 (12) (2011) 14997–15008.

[11] W.J. Li, J. Liu, H. Wang, A. Calzada, R.M. Rodriquez, L. Martínez, A qualitative decision making model based on belief linguistic rule based inference methodology, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 20 (1) (2012) 105–118.

[12] J. Liu, S.W. Chen, L. Martinez, A belief rule-based generic risk assessment framework, in: B. Vitoriano, J. Montero, D. Ruan (Eds.), Decision Aid Models for Disaster Management and Emergencies, Atlantis Computational Intelligence Systems Series, vol. 7, AP/Springer Joint Publisher, 2013, pp. 145–169.

[13] J. Liu, L. Martinez, H. Wang, A. Calzada, S.W. Chen, A new belief rule base representation scheme and its generation by learning from examples, in: The 10th International FLINS Conference on Uncertainty Modeling in Knowledge Engineering and Decision Making (FLINS2012), Istanbul, Turkey, 27–29th August, 2012.

[14] J. Liu, L. Martinez, D. Ruan, R. Rodriguez, A. Calzada, Optimization algorithm for learning consistent belief rule base from examples, Journal of Global Optimization 51 (2) (2011) 255–270.

[15] J. Liu, L. Martínez, H. Wang, R.M. Rodriquez, V. Novozhilov, Computing with words in risk assessment, International Journal of Computational Intelligence Systems 3 (4) (2010) 396–419.

[16] J. Liu, D. Ruan, H. Wang, L. Martinez, Improving IAEA safeguards evaluation through enhanced belief rule-based inference methodology, International Journal of Nuclear Knowledge Management 3 (3) (2009) 312–339.

[17] J. Liu, J.B. Yang, D. Ruan, L. Martinez, J. Wang, Self-tuning of fuzzy belief rule bases for engineering system safety analysis, Annals of Operations Research 163 (1) (2008) 143–168.

[18] G.L. Kong, D.L. Xu, R. Body, J.B. Yang, K. Mackway-Jones, S. Carley, A belief rule-based decision support system for clinical risk assessment of cardiac chest pain, European Journal of Operational Research 219 (3) (2012) 564–573.

[19] G.L. Kong, D.L. Xu, X.B. Liu, J.B. Yang, Applying a belief rule base inference methodology to a guideline-based clinical decision support system, Expert Systems 26 (5) (2009) 391–408.

[20] W. Rudin, Principles of Mathematical Analysis, McGraw-Hill, New York, 1964.

[21] X.S. Si, C.H. Hu, J.B. Yang, Z.J. Zhou, A new prediction model based on belief rule base for system's behaviour prediction, IEEE Transactions on Fuzzy Systems 19 (4) (2011) 636–651.

[22] R. Sun, Robust reasoning: integrating rule-based and similarity-based reasoning, Artificial Intelligence 75 (2) (1995) 241–295.

[23] L.X. Wang, J.M. Mendel, Generating fuzzy rules by learning from examples, IEEE Transactions on Systems, Man, and Cybernetics 22 (6) (1992) 1414–1427.

[24] D.L. Xu, J. Liu, J.B. Yang, G.P. Liu, J. Wang, I. Jenkinson, J. Ren, Inference and learning methodology of belief-rule-based expert system for pipeline leak detection, Expert Systems with Applications 32 (1) (2007) 103–113.

[25] R.R. Yager, P.F. Dimitar, Essentials of Fuzzy Modeling and Control, New York, 1994.

[26] J.B. Yang, Rule and utility based evidential reasoning approach for multi-attribute decision analysis under uncertainties, European Journal of Operational Research 131 (2001) 31–61.

[27] J.B. Yang, P. Sen, A general multi-level evaluation process for hybrid MADM with uncertainty, IEEE Transactions on Systems, Man, and Cybernetics 24 (10) (1994) 1458–1473.

[28] J.B. Yang, M.G. Singh, An evidential reasoning approach for multiple attribute decision making with uncertainty, IEEE Transactions on Systems, Man, and Cybernetics 24 (1) (1994) 1–18.

[29] J.B. Yang, D.L. Xu, On the evidential reasoning algorithm for multiple attribute decision analysis under uncertainty, IEEE Transactions on Systems, Man, and Cybernetics – Part A 32 (3) (2002) 289–304.

[30] 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 Transactions on Systems, Man, and Cybernetics – Part A 36 (2) (2006) 266–285.

[31] J.B. Yang, J. Liu, D.L. Xu, J. Wang, H.W. Wang, Optimization models for training belief-rule-based systems, IEEE Transactions on Systems, Man, and Cybernetics – Part A 37 (4) (2007) 569–585.
[32] J.B. Yang, Y.M. Wang, D.L. Xu, K.S. Chin, L. Chatton, Belief rule-based methodology for mapping consumer preferences and setting product targets, Expert Systems with Applications 39 (2012) 4749–4759.
[33] L.A. Zadeh, Fuzzy logic, Computer 21 (4) (1988) 83–93.
[34] 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 Systems with Applications 36 (4) (2009) 7700–7709.
[35] Z.J. Zhou, C.H. Hu, J.B. Yang, D.L. Xu, D.H. Zhou, Bayesian reasoning approach based recursive algorithm for online updating belief rule based expert system of pipeline leak detection, Expert Systems with Applications 38 (2011) 3937–3943.
[36] Z.J. Zhou, C.H. Hu, J.J. Yang, D.L. Xu, D.H. Zhou, Online updating belief-rule-based systems using the RIMER approach, IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans 41 (6) (2011) 1225–1243.
[37] Z.J. Zhou, J.B. Yang, C.H. Hu, D.L. Xu, Belief Rule Base Expert Systems and Complex System Modelling, Science Publisher, Beijing, 2011.