A New Dynamic Rule Activation Method for Extended Belief Rule-Based Systems

Alberto Calzada, Jun Liu, *Member, IEEE*, Hui Wang, *Member, IEEE*, and Anil Kashyap

Abstract—Data incompleteness and inconsistency are common issues in data-driven decision models. To some extend, they can be considered as two opposite circumstances, since the former occurs due to lack of information and the latter can be regarded as an excess of heterogeneous information. Although these issues often contribute to a decrease in the accuracy of the model, most modeling approaches lack of mechanisms to address them. This research focuses on an advanced belief rule-based decision model and proposes a dynamic rule activation (DRA) method to address both issues simultaneously. DRA is based on "smart" rule activation, where the actived rules are selected in a dynamic way to search for a balance between the incompleteness and inconsistency in the rule-base generated from sample data to achive a better performance. A series of case studies demonstrate how the use of DRA improves the accuracy of this advanced rule-based decision model, without compromising its efficiency, especially when dealing with multi-class classification datasets. DRA has been proved to be beneficial to select the most suitable rules or data instances instead of aggregating an entire rule-base. Beside the work performed in rule-based systems, DRA alone can be regarded as a generic dynamic similarity measurement that can be applied in different domains.

Index Terms—Rule-based processing, incompleteness, inconsistency, knowledge base verification, decision support

1 Introduction

Among the various knowledge representation schemes that can be found in the literature, rule-based systems have been widely recognised as one of the most used for decision support systems (DSSs), where elements of human reasoning are combined with some degree of intelligence provided by the computer-based system [1], [2], [3]. In many cases, the human knowledge commonly represented with IF-THEN rules needs to be complemented with other information, usually collected as a dataset, in order to model a problem in more detail. In this regard, data-driven approaches provide mechanisms to construct a decision model directly from sample data. However, due to the noise, varied quality or reliability of the data set used to generate a rule-based decision model, it is likely that the resulting model produces unreliable and inaccurate outputs. This research focuses on two particularly-common scenarios in the rule-base decision model generated from sample data: incompleteness and inconsistency. Incompleteness occurs when a data driven rule-based decision model is unable to produce a decision result due to a lack of information in the generated rule-base (caused essentially from the incompleteness of the sample data set). On the other hand, a situation of inconsistency occurs when the rule-based DSS is unable to provide a confident result because two or more rules with different consenquents (THEN part) are activated together for the same input. This can be seen as an excess of conflicting information in the rule-base.

Some studies in recent literature have attempted to simultaneously approach situations of incompleteness and inconsistency of rule-bases. The authors in [4] proposed a Petri-nets-based mechanism to represent the rule-base and to identify conflicting situations. However, this method required three processes to be performed to transform the entire rule-base into a Petri-net, which could be computationally expensive as the number of rules and complexity of the rule-base increase. A similar Petri-net-based mechanism was proposed in [5]. Another approach was presented in [6], where the notion of execution paths is proposed to identify these issues. An execution path is a chain of rules, where the consequent of a fired rule can serve as the antecedent of another rule. In this case, it is necessary to find all the possible execution paths of the rule-base, which may become a prohibitive process as the complexity of the rule-base increases. In addition, in many rule-bases directly generated from data, all the rules share the same consequent attribute and therefore it would not be possible to chain rules in order to generate execution paths. Some works such as [7] criticise the cost that simplification implies when fuzzy rule-bases are generated from linguistic and qualitative knowledge, and propose the notions of e-completeness and e-consistency to offset this negative effect. However, their approach is focused just on qualitative information and would not apply to data-driven approaches, usually generated from numerical data. The knowledge representation model selected for this research, so called extended belief rule-base (E-BRB) [8] is a generic and effective method to handle the related vague, incomplete and/or uncertain

- *A. Calzada, J. Liu, and H. Wang are with the School of Computing and Mathematics, University of Uster at Jordanstown Campus, Shore Rd, Newtownabbey, United Kingdom.*
  E-mail: {a.calzada, j.liu, h.wang}@ulster.ac.uk.
- *A. Kashyap is with the School of the Built Environment, University of Uster at Jordanstown Campus, Shore Rd, Newtownabbey, United Kingdom. E-mail: a.kashyap@ulster.ac.uk.*

Manuscript received 7 Aug. 2013; revised 8 Aug. 2014; accepted 11 Aug. 2014. Date of publication 8 Sept. 2014; date of current version 3 Mar. 2015.
Recommended for acceptance by G. Karypis.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TKDE.2014.2356460

knowledge of a problem, along with precise data. Therefore, it is able to involve qualitative and quantitative data, overcoming some of the main issues of other state-of-the-art decision models.

In this context, the aim of this research is to enhance data-driven knowledge representation schemes, such as E-BRB, complementing them with a specialised activation algorithm, named Dynamic Rule Activation (DRA), designed to find a suitable set of instances (in our case rules) for each input data sample. The main aim of DRA is to tackle the incompleteness and the inconsistency found in the decision model by automatically adjusting the size of the set of activated rules [9]. In situations of incompleteness, the size of this set would increase, capturing some other contextual, neighbourhood information [10]. On the other hand, in situations of inconsistency DRA searches for smaller sets of activated rules, trying to find more precise and homogeneous information [9]. The main emphasis of this research is to highlight the benefits, in terms of accuracy (number of instances correctly predicted) and efficiency (in terms of time complexity), of DRA as an activation method that finds the most suitable pieces of information contained in a knowledge-base, instead of aggregating the entire knowledge-base to get a result. Ultimately, this research also compares the performance of DRA as an activation method against the Euclidean Distance similarity, which is utilised by the inference methods when DRA is not employed.

The remainder of this paper is organised as follows: Section 2 briefly reviews the E-BRB representation, generation and inference procedure. Section 3 represents E-BRB as tuples that can be grouped to form hypertuples, using this idea as a starting point for E-BRB aggregation; Section 4 introduces the concept of DRA and details how the search for a suitable hypertuple is performed. Section 5 discusses the concepts described in the previous sections and explores how to integrate them to form a standalone, customisable DSS using DRA. Section 6 demonstrates and discusses the performance of the system by presenting several case studies using benchmark datasets. Finally, Section 7 concludes this paper.

2 Overview of Extended Belief Rule-Bases

To evaluate the proposed DRA, this research makes use of a recently developed data-driven rule-based decision model: E-BRBs. E-BRBs can be described as an extension of the hybrid belief rule bases (BRBs) included in the RIMER approach [11]. The RIMER approach—along with its BRBs—has been investigated and successfully applied to different areas: from safety and risk analysis, health care and engineering systems, among others [11], [12], [13], [14], [15], [16].

BRBs provide methods to handle causal, non-linear relationships as well as uncertainty by embedding belief degree distributions in the consequents of the rules. Embedding these belief distributions in the rule consequent allows BRBs representing additional information, e.g., vagueness (i.e., handling partially known beliefs in consequents due to incomplete knowledge). Moreover, BRBs provide additional parameters to represent the importance of different attributes and rules.

The E-BRBs used in this research import all the benefits from BRBs and extend them, since they include belief degree distributions not only in the consequent of the rules, but also in all the antecedents. E-BRBs have been successfully combined with the evidential reasoning (ER) approach to create the new RIMER+ decision model [8]. Thanks to the mentioned capabilities and the elegant way in which they are comprised within each rule and the data-driven nature of E-BRBs, they are considered as a suitable scheme to exemplify the DRA method presented in this study. Subsequent sub-sections briefly explore the underpinning concepts of E-BRBs [8], taking RIMER's BRBs as a starting point to describe the advantages of this method as a knowledge representation scheme.

2.1 Extended Belief Rule Bases

As aforementioned, E-BRBs import all the advantages of RIMER's BRBs [11] and expand them. This section summarizes BRBs as a starting point to introduce the E-BRBs.

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

$$\begin{aligned} R_k &: \text{IF } \boldsymbol{U} \text{ is } \boldsymbol{A_k} \text{ THEN } D \text{ with belief degrees } \boldsymbol{\beta^k}, \\ &\text{with a rule weight } \theta_\text{k} \text{ and attribute weights } \boldsymbol{\delta_\text{k}}, \end{aligned} \tag{1}$$

where $\boldsymbol{U}$ represents the antecedent attribute vector $(U_1, \ldots, U_T)$, A_k the packet referential values $\{A_1^k, \ldots, A_T^k\}$ of antecedents, here $A_i^k (i = 1, \ldots, T)$ is the referential value of the antecedent attribute U_i in the kth rule and $\delta_i (\in R^+, i = 1, \ldots, T)$ is the relative weight of the ith antecedent attribute. T is the total number of antecedent attributes used in the rule base and L (>0) is the number of rules in the rule-base. $\theta_k (\in R^+, k = 1, \ldots, L)$ is the relative weight of the kth rule. D is the consequent attribute, defined using N linguistic terms included in the vector $(D_1, \ldots, D_N)$, $\boldsymbol{\beta^k}$ is the vector of the belief degrees $(\beta_{1k}, \ldots, \beta_{Nk})$ for $k \in \{1, \ldots, L\}$. Therefore, $\beta_{sk} (s \in \{1, \ldots, N\})$ is the belief degree to which the D_s linguistic term is believed to be the consequent. Note that $\sum_{s=1}^{N} \beta_{sk} \leq 1$. If $\sum_{s=1}^{N} \beta_{sk} = 1$, the consequent of the kth rule is complete; otherwise, it is incomplete.

In a rule base, a referential set is a set of meaningful and distinctive evaluation standards for describing an attribute, which are commonly described by linguistic terms. Rules (2) and (3) below illustrate the linguistic terms (in italics) used in attributes (highlighted in bold) relevant to urban planning. Take for example the following belief rule to assess social deprivation:

$$\begin{aligned} R_k &: \text{IF } \textbf{Unemployment} \text{ is Medium-High AND } \textbf{Crime} \text{ is} \\ &\quad \text{Frequent THEN } \textbf{Social\_Deprivation} \text{ is } \{(\mathit{Alert}, 0.4), \\ &\quad (\mathit{Caution}, 0.3), (\mathit{Moderate}, 0.1), (\mathit{Good}, 0)\} \text{ with rule weight} \\ &\quad \text{of } 0.9 \text{ and attribute weights of } \{0.8, 1\}. \end{aligned} \tag{2}$$

Here $\boldsymbol{U}$ = (Unemployment, Crime), $\boldsymbol{A_k}$ = (Medium-High, Frequent), and the consequent is a belief distribution representation for the consequent, stating that it is 40 percent sure that social deprivation is alert, 30 percent sure that it is caution, 10 percent that it is moderate. This type of rule reflects another kind of uncertainty caused because sometimes an

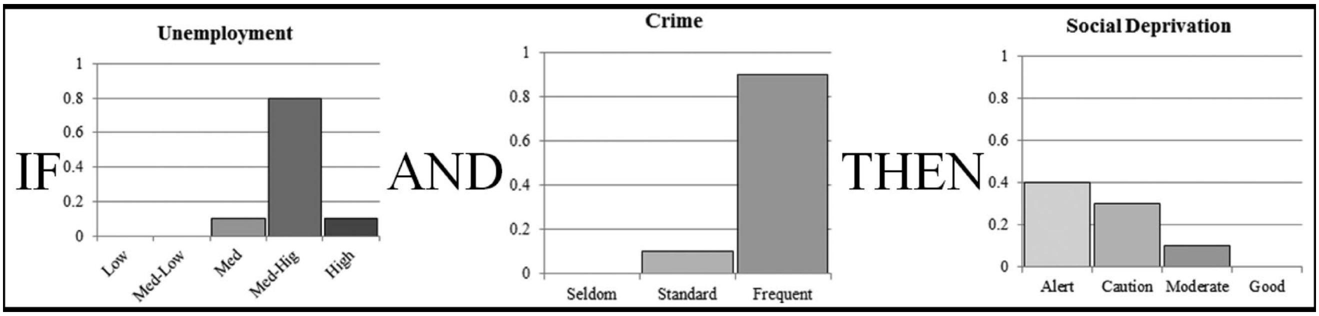

Fig. 1. An extended belief rule graphically represented. Belief degree distributions can be easily represented as bar charts in order to visually analyse the exact relationship between attributes of each rule.

expert is unable to establish a strong correlation between premise and conclusion. In addition, since $\sum_{i=1}^{N} \overline{\beta}_{ik} = 0.4 + 0.3 + 0.1 = 0.8 < 1$, the belief rule expressed in (2) has an incomplete consequent. In this case, the remaining 20 percent belief until 1 (1 – 0.8 = 0.2) is unknown, and it can be used to represent the degree of ignorance about the assessment described in the rule. Note also that the Crime attribute has a higher relative weight than the Unemployment one (1 and 0.8, respectively). The rule weight of 0.9 represents the importance of R_k over other rules.

To facilitate more general application cases and more flexible and simpler rule-base generation scheme, this belief rule is extended with belief degrees embedded in all the antecedent terms of each rule, for example, belief rule (2) can be extended as follows:

$$\begin{aligned} R_k\text{: IF } & \textbf{Unemployment} \text{ is } \{(Low, 0), (Medium\text{-}Low, 0), \\ & (Medium, 0.1), (Medium\text{-}High, 0.8), (High, 0.1)\} \text{ AND} \\ & \textbf{Crime} \text{ is } \{(Seldom, 0), (Standard, 0.1), (Frequent, 0.9)\} \\ & \text{THEN } \textbf{Social\_Deprivation} \text{ is } \{(Alert, 0.4), (Caution, 0.3), \\ & (Moderate, 0.1), (Good, 0)\} \text{with rule weight of 0.9 and} \\ & \text{attribute weights of} \{0.8, 1\}. \end{aligned} \tag{3}$$

The rule illustrated in (3) might look complex, but it is still easy to intuitively understand [8], [16]. The belief degree distributions embedded in each antecedent attribute provide more information about the status of the antecedent than the linguistic terms of BRBs. Also, E-BRBs explicitly show the corresponding input-output relationship through the IF-THEN rule mapping. Fig. 1 graphically illustrates rule (3) reflecting the added feature in terms of the interpretation and transparency of the E-BRB.

This kind of extended belief rule is able to capture vagueness (with linguistic terms), uncertainty (with beliefs), incompleteness (partially known beliefs in antecedents and/or consequents), nonlinear relationships (IF-THEN rules) between the two antecedents and the consequent, a flexible way to embed hybrid information and a more efficient rule generation scheme.

The belief distributions incorporated in the antecedent show distinct features compared with traditional rule-bases as well as the BRBs included in RIMER [11]. On the one hand, it provides a scheme to reflect the potential uncertainty involved in the antecedent attribute, e.g., some attributes are random in nature, (e.g. fuel consumption of car could change under different weather conditions or a temperature of 23 degrees Celsius could be "warm" for some people and "hot" for others), so its evaluation could be a probability distribution. On the other hand, it accommodates the relationship between the given input and the antecedent referential values by using the belief distribution without loss of information, which contributes to the rule-base generation scheme detailed in Section 2.2. More generally, rule (1) can be extended as follows:

$$\begin{aligned} R_k^* : \text{IF } \boldsymbol{U} \text{ is } (\boldsymbol{A}, \boldsymbol{\alpha^k}) \text{ THEN D with belief degrees } \boldsymbol{\beta^k}, \\ \text{with a rule weight } \theta_k \text{ and attribute weights } \boldsymbol{\delta_i}, \end{aligned} \tag{4}$$

where $(\boldsymbol{A}, \boldsymbol{\alpha^k})$ is the packet antecedents $\{\{(A_{ij}, \alpha_{ij}^k), j = 1, \ldots, J_i\} | i = 1, \ldots, T\}$ (as illustrated in rule (3)) representing each antecedent with its belief structure; here $A_{ij} (j \in \{1, \ldots, J_i\})$ is the jth linguistic term of the ith antecedent attribute $U_i (i = 1, \ldots, T)$ (note that U_i is defined using J_i linguistic terms), α_{ij}^k is the likelihood to which U_i is evaluated to be the linguistic term A_{ij} in the kth rule with $\alpha_{ij}^k \geq 0$ and $\sum_j^{J_i} \alpha_{ij}^k \leq 1 \quad (i = 1, 2, \ldots, T), k = 1, \ldots, L^*.L^*$ is the total number of rules in the rule-base. Note that the THEN part of rules (1) and (4) are equivalent. A rule base with rules in the form of (4) is called an E-BRB. Note that all the rules of an E-BRB share the same antecedent and consequent attributes, but have different belief distributions for each antecedent attribute and the consequent. The use of the superscript k in (4) reflects this feature. It is crucial to note that traditional fuzzy rule-bases and the belief rule-base in RIMER are special cases of E-BRB. For a BRB with rules in the form of (1), the total number of rules, L, is usually determined as $L = \prod_{i=1}^{T} J_i$, but the total number of rules (L^*) for an E-BRB is determined by the number of given input-output data pairs [8]. This paper does not focus on this specific property of E-BRB, which is briefly outlined in the next section, but on how the inference performed on top of E-BRB can be improved with DRA.

2.2 Generating E-BRB from Numerical Data

The rule generation methodology presented in [8] allows transforming a dataset into an E-BRB. Basically, this methodology is a linear process consisting of four basic steps (see [8] for more information). This section briefly highlights the key points of this rule generation methodology that are relevant to this research.

As aforementioned, the data instances of the original dataset are directly transformed as extended belief rules. Therefore, a dataset containing L^* elements would be converted into an E-BRB with L^* extended belief rules. In this regard, it is important to highlight that in this process, all the information included in the original dataset is transferred to the E-BRB [8]. Thanks to this, a direct mapping between the E-BRB and the original dataset is created, and each extended belief rule can be re-converted as an input-output data pair. This important characteristic is crucial for the generality of the algorithm proposed in this research, and is exploited in Section 3.

As a data-driven approach provided to generate E-BRBs for RIMER+, it indeed sacrifices some interpretability/transparency of the rule-based system to a certain degree (although still maintains the essential features of IF-THEN mapping, taking into account that normal fuzzy rule bases or belief rule bases in RIMER are special cases of E-BRBs, without loss of their interpretability/transparency). However, E-BRB enhances the overall performance in terms of applicability (suitable for online or real-time application), efficiency (no need of time consuming training or optimisation algorithm) and accuracy (as illustrated in the case studies presented in Section 6). Usually, the complexity increases opposite to the interpretability, and RIMER+ certainly is not perfect to fulfil all the criteria. Despite this fact, further research is being currently carried out to enhance the interpretability of E-BRB, reducing the number of rules and attributes and simplifying the model, following the indications from some research studies in this area, such as [17], [18].

Due to the immediate correlation between the original dataset and the rule-base, systems based on E-BRB can be categorized as instance-based learners. Therefore, the inference process performed on top of E-BRB can be categorized as “lazy”, in the sense that little effort is placed on learning from the original dataset and building the model. Instead, more work needs to be done when classifying new samples [19]. The following subsection outlines the inference steps to retrieve decision outputs from E-BRBs.

2.3 Extended Belief Rule-Base Inference

This section briefly describes the E-BRB inference methodology. For a more detailed description of this process, please see [8]. Basically, this inference methodology consists of 3 main steps. Fig. 2 illustrates them.

In the inference process, the first step required is to compare the inputs of the system against the antecedents of the extended belief rules in order to evaluate which rules should be activated. Let us assume an input vector, given as follows:

$$\boldsymbol{X} = (x_1, x_2, \ldots, x_T), \tag{5}$$

where x_i is the input for the U_i attribute $(i = 1, \ldots, T)$. In order to measure the similarity of $\boldsymbol{X}$ against the rule antecedents (see Step 1 in Fig. 2), it is required to transform each $x_i (i = 1, \ldots, T)$ as a belief degree distribution using the linguistic terms and membership functions used to define each U_i in Eq. (4). Therefore, x_i could be expressed as a belief degree distribution as follows:

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

where A_{ij} is the jth linguistic term of the U_i attribute and α_{ij} is the degree to which the input x_i is believed to match A_{ij}. Once the input is transformed, individual matching degrees

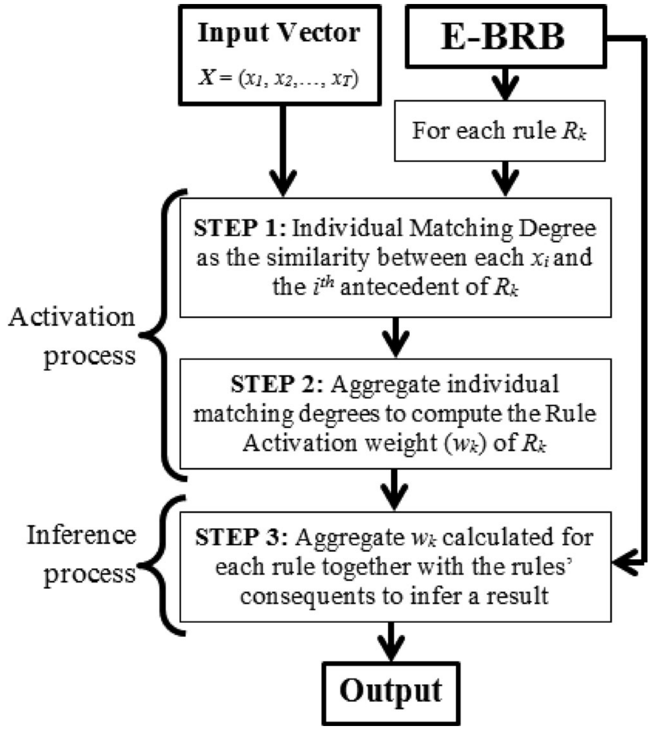
Input Vector
$X = (x_1, x_2, \ldots, x_T)$
E-BRB
For each rule R_k
Activation process
STEP 1: Individual Matching Degree as the similarity between each x_i and the i^{th} antecedent of R_k
STEP 2: Aggregate individual matching degrees to compute the Rule Activation weight (w_k) of R_k
Inference process
STEP 3: Aggregate w_k calculated for each rule together with the rules' consequents to infer a result
Output

Fig. 2. E-BRB inference methodology.

(S_i^k) can be computed to measure the similarity between each input (x_i) and its corresponding antecedent attribute (U_i) in the kth rule. A popular distance function that can be used to meet this purpose is the Euclidean distance:

$$d^k(x_i, U_i) = d_i^k = \sqrt{\sum_{j=1}^{J_i} \left(\alpha_{ij} - \alpha_{ij}^k\right)^2}. \tag{7}$$

Then, S_i^kcan be calculated as follows:

$$S^k(x_i, U_i) = S_i^k = 1 - d_i^k. \tag{8}$$

To obtain a rule activation weight (w_k) as illustrated in Step 2 in Fig. 2, the individual matching degrees calculated for the kth rule are aggregated as follows:

$$w_k = \frac{\theta_k * \prod_{i=1}^{T} (S_i^k)^{\bar{\delta}_i}}{\sum_{j=1}^{L^*} \left[\theta_j * \prod_{l=1}^{T} (S_l^j)^{\bar{\delta}_l}\right]}, \text{ with } \bar{\delta}_i = \frac{\delta_i}{\max_{i=1,\ldots,T}\{\delta_i\}}, \tag{9}$$

where θ_i and δ_i are given in rule (4). Finally, the rule activation weights of all the rules in the E-BRB can be aggregated using the ER algorithm as inference engine (Step 3 in Fig. 2) to get an output:

$$\beta_s = \frac{\mu * \left[\prod_{k=1}^{L^*} \left(w_k \beta_{s,k} + 1 - w_k \sum_{s=1}^{N} \beta_{s,k}\right) - \prod_{k=1}^{L^*} \left(1 - w_k \sum_{s=1}^{N} \beta_{s,k}\right)\right]}{1 - \mu * \left[\prod_{k=1}^{L^*} (1 - w_k)\right]}, \tag{10}$$

where $s = 1, \ldots, N$, w_k is given in Eq. (9) and

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

Note that the final output of RIMER+ is provided as a belief degree distribution in the form $\{(D_s, \beta_s), s = 1, \ldots, N\}$.

TABLE 1
A Dataset of *M* Input-Output Data Pairs (or Tuples) Represented as an E-BRB

	Tuples in the dataset (Input-Output Pairs)			
p	x_{p1} (input for $\boldsymbol{U_1}$)	x_{p2} (input for $\boldsymbol{U_2}$)	y_p (output for V)	Possible Extended Belief Rules generated from process briefly outlined in Section 2.3 (See [8] for more details)
$p = 1$	Up	Zero	White	IF $\boldsymbol{U_1}$ is {(Up, 1), (Down, 0)} AND $\boldsymbol{U_2}$ is {(Zero, 1), (One, 0), (Two, 0)} THEN $\boldsymbol{V}$ is {(White, 1), (Black, 0)}
$p = 2$	Up	One	White	IF $\boldsymbol{U_1}$ is {(Up, 1), (Down, 0)} AND $\boldsymbol{U_2}$ is {(Zero, 0), (One, 1), (Two, 0)} THEN $\boldsymbol{V}$ is {(White, 1), (Black, 0)}
$p = 3$	Down	One	White	IF $\boldsymbol{U_1}$ is {(Up, 0), (Down, 1)} AND $\boldsymbol{U_2}$ is {(Zero, 0), (One, 1), (Two, 0)} THEN $\boldsymbol{V}$ is {(White, 1), (Black, 0)}
$p = 4 = M$	Down	Two	Black	IF $\boldsymbol{U_1}$ is {(Up, 0), (Down, 1)} AND $\boldsymbol{U_2}$ is {(Zero, 0), (One, 0), (Two, 1)} THEN $\boldsymbol{V}$ is {(White, 0), (Black, 1)}

For simplicity purposes, the original dataset is a two-dimensional dataset consisting of qualitative values (Adapted from [20]).

3 Mapping Extended Belief Rules as Hypertuples

In order to provide a consistent theoretical base for the proposed method, it is worth considering extended belief rules (or data samples/instances for other decision models) as input-output data pairs (known as tuples) that can be grouped, creating hypertuples [20]. Situating the concept of E-BRB in this framework is a straightforward procedure, since all the information of the original dataset is transmitted to the E-BRB during rule generation process (see Section 2.2). Hence, a direct mapping between the original dataset, the tuples space and the E-BRB can be provided. That is, each extended belief rule can be directly represented as a data sample and/or as a tuple. Table 1 in the next page provides an illustrative example of a dataset in a two-dimensional space.

Each one of these tuples can be mapped by using its values in a way in which tuples with similar values are closely located and can be grouped, creating hypertuples. To exemplify this, Fig. 3 shows a dataset with three possible classes, depicted by crosses, circles and triangles. For simplicity issues, the dataset shown has just two attributes (U_1 and U_2). Some hypertuples are also illustrated in Fig. 3 using rectangles. All of them are consistent hypertuples because they just contain tuples of the same class—that is, all the rules within each hypertuple have the same consequent part: either cross, circle or triangle.

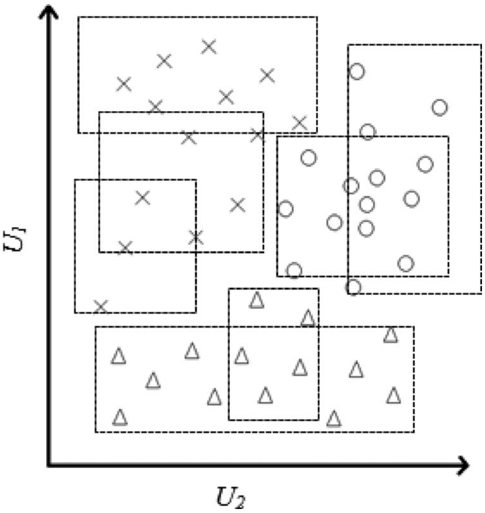

Fig. 3. A two-attribute (U_1 and U_2) dataset. Consistent hypertuples are depicted using dashed rectangles and each cross, circle or triangle represents a simple tuple. Adapted from [20].

In ideal conditions, it would be possible to assume that any input vector in the form of Eq. (5) would activate (see Eq. (9)) a set of rules that would correspond to a consistent hypertuple of the dataset. However, due to the noise, inconsistency or reliability of the data used to generate the rule base this is not usually the case. Note that in the case of E-BRBs, belief distributions are utilised to define the rule consequents. Therefore it is highly unlikely that for an E-BRB generated from data many rules share identical belief distributions in their consequents.

Assuming these issues, in this research the concept of hypertuple is relaxed and a hypertuple can be considered as a set of extended belief rules with *similar* consequent part (that is, a set of tuples with *similar* output part). A simple way to define the criterion of *similarity* between rules for a given input vector would be using the activation weight already calculated in Eq. (9). Therefore, in this research a hypertuple Δ is defined as the set of extended belief rules whose activation weight is greater than zero:

$$\Delta = \{R_i; w_i > 0; (i = 1, \ldots, |\Delta|)\}, \tag{11}$$

where R_i are in the form of (4) and w_i is given by Eq. (9). In this context, the process of assessing the inputs and rule activation in an E-BRB can be described as a method to adjust the Δ hypertuple obtained in Eq. (11) in order to approximate it as much as possible to a consistent hypertuple (i.e. a consistent set of *similar* extended belief rules); to be aggregated in further inference steps.

However, data-driven rule-based systems usually rely on static rule activation procedures, where the hypertuple retrieved (whether it is empty, consistent or not) after inputs and rules are compared using a similarity metric is directly processed to get a decision support outcome. However, for

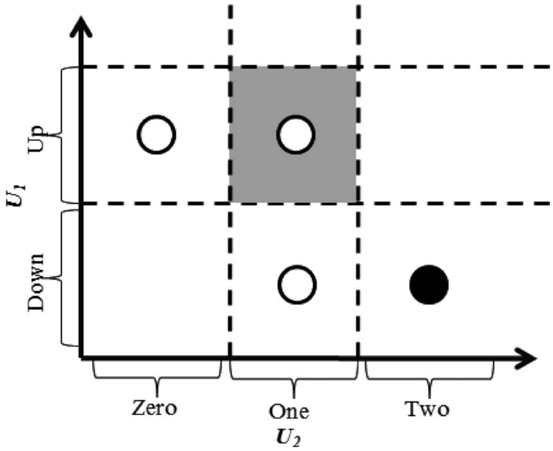

Fig. 4. A two-attribute (U_1 and U_2) dataset with four tuples (represented by white and black circles). For the input (*Up*, *One*), the Δ hypertuple would be the grey rectangle delimited by the activation intervals.

an input it may occur that Δ is empty (situation of incompleteness) or Δ is not consistent and contains rules with different consequents (inconsistency).

4 Dynamic Rule Activation Algorithm to Adjust the Set of Activated Rules

The activation process is one of the most important steps in rule-based systems and other data-driven methodologies. Varying the formulation in this part of the system can significantly affect the outcomes produced. In rule-based systems, the similarity measure used to evaluate the matching degree between inputs and rule antecedents is fundamental, since it affects the performance of the entire system. Research has highlighted the importance of similarity measures [21]; and many researches are focused on finding new similarity metrics to improve system performances. The proposed method adds a new level in this field of research, since it evaluates the values produced by similarity measures in a "smart" way, automatically assessing each particular situation found.

Note that in this context, most rule-based decision models (including RIMER and RIMER+) are based on static rule activation, relying on the entire set of activated rules retrieved after the similarity metric is applied. It is called static in the sense that the set of activated rules is fixed once the similarity measure has been applied. That is, there is no adjustment or control based on different situations that commonly affect the performance of the decision model, such as incompleteness or inconsistency of the rule-base.

4.1 Extended Belief Rule-Base Inference

This section describes how the concepts of incompleteness and inconsistency of the rule-base are expressed in terms of hypertuples. To achieve so, the notion of activation interval of each attribute for an input is introduced. These concepts would help representing and understanding the proposed DRA method.

4.1.1 Activation Intervals

By definition, the Δ hypertuple (Eq. (11)) contains only rules whose activation weight is greater than zero. Following the representation scheme proposed in Fig. 3, the activation intervals of each antecedent attribute U_i are the boundaries which contain all the possible values of U_i that would be activated for the input x_i. In this context, the hypertuple Δ can be described as the area where all the possible activated values for each antecedent attribute intersect. Therefore, having the input (Up, One) (as specified in Eq. (5)), the example dataset presented in Table 1 could be represented as illustrated in Fig. 4.

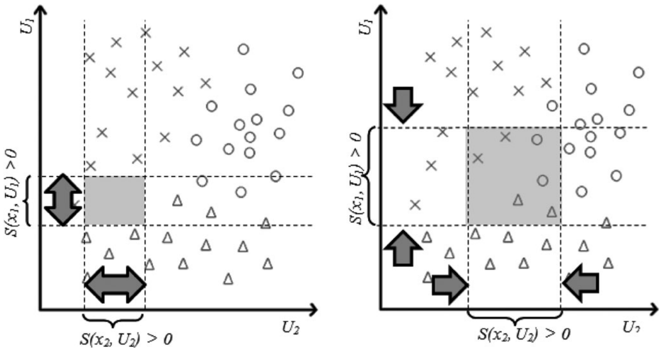

Fig. 5. Situation of incompleteness (left) and inconsistency (right) of the E-BRB for the input vector {*x1*, *x2*}.

In Fig. 4, only one rule was activated (there is only one white circle contained in the Δ hypertuple) for the (Up, One) input. It is worth noting that in this study it is supposed that all the antecedent attributes of each rule of the E-BRB are connected with an "AND" relationship, usually aggregated with the product operation (as in Eq. (9)). Therefore, the Δ hypertuple is defined as the intersection of the activation intervals, depicted in Fig. 2 using a grey rectangle. For other relationships (like "OR"), Δ would have a more complex shape. In this study the "AND" relationship is chosen as the default one.

4.1.2 Rule-Base Incompleteness and Inconsistency Situations

The ideal situation represented in Fig. 4 is not usually the case. Instead, the activation intervals often define a Δ hypertuple that is either empty or includes distinct rules to be aggregated. Fig. 5 below depicts these two situations using a 2-dimensional dataset with three possible classes: cross, circle or triangle.

In Fig. 5, the left chart depicts a situation of rule-base incompleteness, where the similarity function (S) applied to the input x_1 and its attribute U_1 and to the input x_2 and its attribute U_2 define activation intervals (dashed lines) that delimit an empty Δ hypertuple (depicted as a grey rectangle). The right chart illustrates a situation of inconsistency, in which the Δ hypertuple, also defined through the similarity function S, contains tuples from different classes. Bold arrows represent the possibilities available to improve these two situations. Basically, the key actions proposed in this research are to automatically open or close the activation intervals of each attribute searching for a suitable Δ hypertuple. When searching for such hypertuple, an important aspect to take into account is that incompleteness and inconsistency of the rule-base might be seen as opposite situations. While the former occurs due to a lack of activated rules, the latter appears when there is an excess of activated rules for a given input. That is, incompleteness occurs when

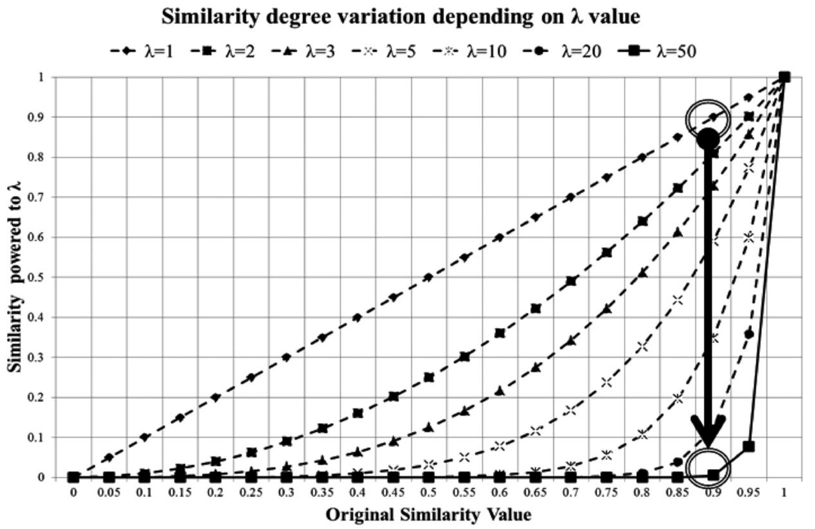

Fig. 6. Penalty applied to similarity values by powering them to some representative λ values. The case of $\lambda = 50$ previously mentioned in the text is highlighted with a bold arrowillustrating the penalty applied.

$|\Delta| = 0$ (the Δ hypertuple is empty because none of the rules were activated, $w_i = 0, \forall R_i, i = 1, \ldots, M$) and inconsistency happens when one input activates two or more rules with different consequents ($\exists i, j : w_i, w_j > 0; \beta_{si} \neq \beta_{sj}$).

Therefore, in this research, the right balance between incompleteness and inconsistency of the rule-base is considered as the most suitable situation to aggregate Δ. In this regard, the proposed DRA method searches for that balance by opening or closing the activation intervals depending on the situation found after calculating the activation weights using Eq. (9).

4.2 Automatic Adjustment of λ to Control the Size of the Δ Hypertuple: Dynamic Rule Activation

Based on the assessment performed to Δ, situations of incompleteness or inconsistency in the rule-base can be detected and fixed by augmenting or reducing the size of Δ, respectively. As mentioned, the size of Δ is determined by its activation intervals (see Fig. 5). DRA controls the opening and closing of the activation intervals by adjusting one single parameter, named λ. This is performed by powering the similarity function to this λ:

$$S_\lambda(x_i, U_i) = (S(x_i, U_i))^\lambda, \tag{12}$$

where S is the similarity function applied to the input x_i and its corresponding antecedent attribute U_i. Since the size of Δ changes for each value of λ while DRA is adjusting it, we can notate the current state of Δ by using λ as an index. That is, Δ_λ is the hypertuple obtained after calculating the rule activation weights for a λ value.

In the decision model described in Section 2.3, DRA is applied by powering the individual matching degree (Eq. (8)) to λ, as described in Eq. (12). Therefore, Eq. (12) is used to tune the individual matching degree obtained in Eq. (8) and ultimately selects which rules have an activation weight greater than zero (Eq. (9)).

It is worth noting that any similarity function could be used as S in Eq. (12), as long as it meets the following requirements: (i) it can be normalised within a [0, 1] interval; and (ii) for the result, the values of 1 and 0 would mean perfect matching and total dissimilarity, respectively. In this research, Minkowski's (Euclidean) distance was chosen as a function that meets these requirements, although any other function could be selected depending on the nature of the decision model or application context.

Powering the similarity metric S to λ may affect the outcome of S_λ in two different ways: (i) higher λ values penalise rules with low activation weights (decreasing their activation weight to almost zero) and (ii) lower λ values reward them (increase their value). The former helps tackling inconsistency situations by searching for smaller, more specific and consistent Δ hypertuples. Meanwhile, the latter helps avoiding incompleteness situations by opening the activation intervals, obtaining larger Δ hypertuples.

4.2.1 *Reward and Penalty Applied by Tuning λ*

The λ value controls the range of the activation intervals for an input vector, penalising or rewarding rules with low activation weights, eventually resolving the incompleteness or inconsistency of the rule-base directly generated from data. In this regard, the higher the λ value is; the smaller and more restrictive hypertuple Δ is retrieved. For example, in the case of $\lambda = 50$, the similarity values less than 0.9 will be powered via λ down to almost zero (see Fig. 6). Accordingly, all the rules initially activated, but with similarity values of less than 0.9 are discarded from the hypertuple Δ_{50}. In other words, Δ_{50} contains only the rules whose individual matching degree is higher than 0.90. Fig. 6 shows how activation weights are penalised for some representative λ values.

Considering Φ as the hypertuple containing all the rules included in the E-BRB, some definitions can be stated for the DRA method:

1) If $\lambda = 0$, $\Delta_\lambda = \Phi$
2) If $\lambda = \infty$, $|\Delta_\lambda| = H$, where H is the number of rules that perfectly match the given input $\boldsymbol{X}$. That is, rules where Eq. (8) retrieves 1 for all its antecedent attributes U_i
3) $\lambda^* > \lambda$, $|\Delta_\lambda{}^*| \leq |\Delta_\lambda|$, and moreover, $\Delta_\lambda{}^* \in \Delta_\lambda$.

Finally, if $\lambda = 1$, there is no penalty or reward, just the basic similarity measure is applied. Therefore, $\lambda = 1$ is considered as the starting point of the DRA algorithm. From this point, the method chooses to either raise the value of λ or decrease it, controlling in this way the size of Δ. The decision of either increasing or decreasing λ depends on three basic policies:

1) If there is a situation of incompleteness (i.e. $|\Delta_\lambda| = 0$), decrease λ in order to open the activation intervals and therefore increase the size of Δ_λ (Section 4.4.1).
2) If there is a situation of inconsistency (i.e. there is no agreement in the Δ_λ hypertuple because its rules have diverse consequents belonging to different classes), increase λ to penalise rules with low activation weights, close the activation intervals and therefore decrease the size of Δ_λ, searching for higher levels of agreement within the Δ_λ hypertuple (Section 4.4.2 for further details).
3) It is desirable to have a large amount of information to be aggregated in the Δ_λ hypertuple. It is considered that the larger the amount of relevant information used to generate the output, the more reliable it will be. To put this situation in another way, consider the worst case scenario, having a hypertuple of size $|\Delta_\lambda| = 1$. In this case, almost every classifier would behave similarly to the Nearest Neighbour (1NN) method, since it would only have one rule to aggregate and therefore getting a decision result based just on one training sample, which is not a very reliable situation in most cases (for example, if the only instance available is highly inconsistent, providing the wrong information to be aggregated). To avoid this DRA searches for as large Δ_λ hypertuples as possible.

4.3 DRA Algorithm to Balance the Incompleteness and Inconsistency of the Rule-Base

Algorithm 1 details the implementation of DRA in pseudocode, using the variable Λ to store the best performing λ. In order to quantify the desired balance between the three principles mentioned in the previous section, DRA utilises the function C(Δ) to measure the performance of λ in terms of consistency or agreement in the Δ hypertuple. In the case of this research, C(Δ) was defined as:

$$C(\Delta) = \frac{\max_{s=1,.,N}\{C_s\}}{|\Delta|}, \tag{13}$$

where C_s is given by:

$$C_s = \left| D_s; s = \arg\left(\max_{i=1,\ldots,N} \{\beta_{ik}\} \right) \right|.$$

For example, in a Δ hypertuple where three belief-rules have the linguistic term "*Low*" with the maximum belief in their consequent; four rules have "*Medium*" and 6 rules have "*High*". Then, $C(\Delta)$ would be the maximum of rules with the same maximum value in the consequent divided by the total of rules (in this case, the six rules that agree with "*High*" are the maximum number, and the total number of rules in Δ is $3 + 4 + 6 = 13$). Therefore, $C(\Delta) = 6/13 = 0.46$.

Any other function to measure the agreement or consistency of C(Δ) could have been used depending on the particular features of the data-driven model and the nature of the decision problem.

Algorithm 1. DRA—an algorithm to tune the value of λ searching for an appropriate hypertuple of activated rules

```
Input: The E-BRB ($\Phi$) containing all the rules ($R_k$) available
for the problem domain and a given input $\boldsymbol{X}$, where the
input $x_j$ corresponds to the attribute $U_j$ of the rule-base
Output: A set of activated rules to be aggregated in further
steps ($\Delta$).
begin
$\lambda = 1$;
$\Lambda = 1$;
while $\lambda > 0$ and ($|\Delta_\lambda| \leq 1$ or $C(\Delta_\lambda) \leq C(\Delta_\Lambda)$) do
    for each $R_k$ of $\Phi$ do
        for each $U_i$ in $R_k$ do
            Calculate $S_i^k(x_i,U_i)$ (Eq. (3.22))
            Apply $\lambda$: $(S_i^k(x_i,U_i))^\lambda$ (Eq. (4.2))
        end for
        Calculate $w_k$ using each $(S_i^k(x_i,U_i))^\lambda$ (Eq. (3.12))
        if $w_k > 0$ then
            $\Delta_\lambda = \Delta_\lambda \cup R_k$
        end if
    end for
    if $|\Delta_\lambda| = 0$ then
        decrease $\lambda$ (e.g. $\lambda = \lambda$-1)
    else if $C(\Delta_\lambda) > C(\Delta_\Lambda)$ then
        $\Lambda = \lambda$
    end else if
    increase $\lambda$ (e.g. $\lambda = \lambda$+1)
    $\Delta_\lambda = \varnothing$
end while
for each $R_k$ of $\Phi$ do
    for each $U_i$ in $R_k$ do
        Calculate $S_i^k(x_i,U_i)$ (Eq. (3.22))
        Apply $\Lambda$ (best $\lambda$ found): $(S_i^k(x_i,U_i))^\Lambda$ (Eq. (4.2))
    end for
    Calculate $w_k$ using each $(S_i^k(x_i,U_i))^\Lambda$ (Eq. (3.12))
    if $w_k > 0$ then
        $\Delta_\Lambda = \Delta_\Lambda \cup R_k$
    end if
end for
detect outliers in $\Delta_\Lambda$
end
```

The increase and decrease intervals for λ are not necessarily fixed to 1. Depending on the specific focus of each particular study, different intervals can be set to further tune the performance and efficiency of the system.

Detecting the outliers in Δ_Λ is an optional method that can be used to ease subsequent inference procedures. In the case of our particular research, this method calculates the average of the consequents included in Δ_Λ. Then, it measures the distance of each rule consequent to the average to identify possible outliers. An outlier would have normalised distances of 1 (far away from the average of Δ_Λ), meaning that its consequent belief distribution does not have any referential value in common with the rest of the rules in Δ_Λ. In the case studies presented in this paper, these outliers are considered as inconsistent pieces of information and DRA discards them. However, the information provided by this method may be valuable. For example, once the decision output is inferred, the information about outliers can be used in order to further tune the result, represent uncertainty, or to

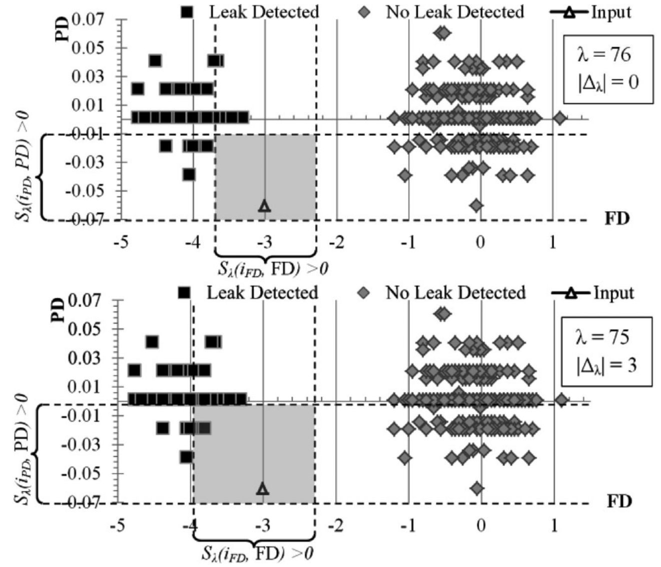

Fig. 7. Decreasing λ, searching for contextual information. In each chart, activation intervals are depicted using dashed lines and $\Delta\lambda$ is represented with a grey rectangle.

provide specific feedback about the most conflicting inconsistencies found in the rule activation process.

Finally, it is worth noting that DRA cannot be considered as an optimisation process, since none of the parameters or attributes of the knowledge-base are tuned to enhance the performance of the system. DRA is better categorised as a supportive method, which could be run before the inference process of the data-driven DSS. Therefore, it has been adjusted to achieve high levels of efficiency, compared the time that most common optimisation methods would delay when tuning the amount of parameters of an entire data-driven decision model (in this case E-BRB). Nevertheless, optimisation models may still apply without affecting the performance of DRA.

4.4 Sensitivity Analysis of Adjusting λ

The following subsections provide step-by-step graphical examples describing how DRA works in order to control the size of each hypertuple Δ_λ, searching for a balance between incompleteness and inconsistency and trying to meet the three policies aforementioned.

4.4.1 Decreasing λ Searching for More Contextual Information to Resolve the Incompleteness

The mechanism provided by DRA to compensate or resolve incompleteness is simply to reduce the λ value and re-calculate the activation weights of the rules. Accordingly, the size of the hypertuple Δ_λ grows, trying to capture more contextual information associated with the given input vector. This mechanism is activated when Δ_λ is empty. For illustrative purposes, a real world example about leak detection is utilized here. The dataset has two attributes, pressure difference (PD) and flow difference (FD), which are plotted in a two-dimensional scatter chart. Based on these two attributes, the aim is to predict whether there is a leak or not in an oil pipeline. The application background and data set has been detailed in [8]. Fig. 7 below details the distribution of the leak detection dataset in relation to its two attributes and how the size of Δ varies corresponding to the given input vector: $\{(\text{iPD} = -0.06), (\text{iFD} = -3)\}$, where iPD is the input for the PD attribute and iFD is the input for the FD attribute.

Decreasing the λ value allows recovering certain information that was discarded when the λ value was larger. In the example depicted in Fig. 7, with $\lambda = 76$ there was a situation of incompleteness, since none of the rules was activated and therefore the Δ_{76} hypertuple was empty. Decreasing λ to 75 allowed rewarding rules with residual activation weights in order to capture more information associated to the given input vector.

In the worst case scenario of decreasing λ to 0, all the rules in the E-BRB would be activated with the same (maximum) activation weight and the system output provided, would most likely be some compromise value towards the average output of the whole sample dataset used to generate the E-BRB. However, this compromise solution would contain a large amount of uncertainty, indicating to the user the poor reliability of the output.

In any case, having a compromise solution still provides some information to the user, compared to what happens when none of the rules in the rule-base is activated, returning a failed execution with no result at all. The compromise solution approach is especially remarkable in complex rule-based systems, where attributes are organised as a hierarchy which consists of several sub-rule-bases. Retrieving a failed execution in any of the leaf rule-bases would propagate the failed execution error to the root of the hierarchy. However, having a compromise solution in some nodes would still allow inferring a result in the remaining rule-bases.

4.4.2 Increasing λ to Resolve Inconsistency

Situations of inconsistency are common during the rule activation process. Fig. 8 below shows an example of a hypertuple containing rules from diverse nature. In this case, the rule aggregation process would be complex, since some amount of uncertainty is being induced from the inconsistent rules that should not be activated for an input (in the example below: $\{(\text{iPD} = -0.01), (\text{iFD} = -1.4)\}$). Increasing the value of λ reduces the size of Δ, discarding the less relevant rules as an attempt to get more homogeneous information.

In Fig. 8, the λ value was increased from 1 to 3. As λ increased, Δ_λ was reduced; discarding the rules with lower activation weights until an appropriate λ value is reached, having a suitable Δ_λ containing just the most relevant information. In the example illustrated in Fig. 8, the 33 less relevant rules were discarded during the DRA process, keeping the 18 most highly activated ones. Activating fewer rules simplifies subsequent aggregation procedures by providing just the most useful and relevant information.

5 Embedding DRA in Decision Models

To demonstrate the performance of DRA, it was embedded into two different inference schemes used to combine/aggregate Δ and generate final conclusions:

A) *Inference scheme I*. The RIMER+ model, reviewed in Section 2.3, using the ER algorithm [11] (Eq. (10)).

B) *Inference scheme II*. A simple, yet efficient, weighted average (WA) algorithm, performed by aggregating

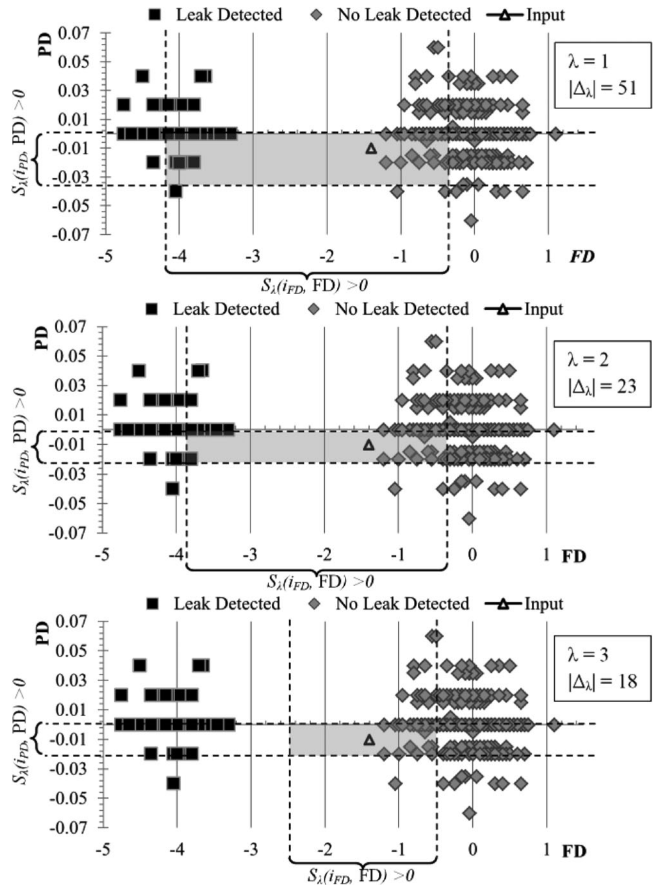

Fig. 8. Increasing λ, searching for a homogeneous hypertuple $\Delta\lambda$. In each chart, activation intervals are depicted using dashed lines and $\Delta\lambda$is represented with a grey rectangle.

the consequents of the rules in Δ and using their activation weight (Eq. (9)) as weighting parameter:

$$\beta_s = \frac{1}{|\Delta|}\sum_{k=1}^{|\Delta|} w_k \beta_{sk}. \tag{14}$$

This E-BRB based decision model is called EBRB-WA.

Therefore, the two-step decision model proposed uses DRA as an activation method and either the EBRB-WA or the ER algorithm as inference engine.

Note that, thanks to the flexibility provided by DRA, any other rule-based knowledge base, similarity function or inference engine to aggregate Δ could be used to test DRA. In this research, E-BRBs were chosen because of their easy rule generation method, flexibility, applicability and simplicity when representing vagueness and uncertainty and the EBRB-WA and ER were selected because of their simplicity and efficiency when aggregating the information from Δ. However, any other decision model meeting these requirements could be potentially upgraded using DRA as a rule activation method. Fig. 9 summarizes the architecture of the proposed decision model and how DRA can fit within other decision models as a data activation step.

The next section demonstrates the relevance of DRA, by comparing the performance of EBRB-WA and ER, used with and without DRA.

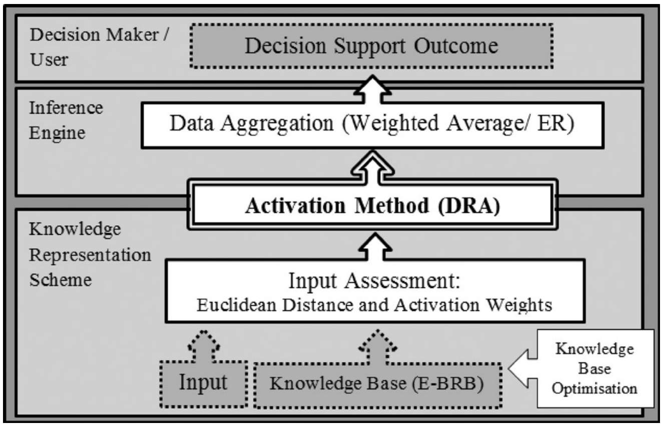
Decision Maker / User
Decision Support Outcome
Inference Engine
Data Aggregation (Weighted Average/ ER)
Activation Method (DRA)
Knowledge Representation Scheme
Input Assessment:
Euclidean Distance and Activation Weights
Input
Knowledge Base (E-BRB)
Knowledge Base Optimisation

Fig. 9. Architecture of a decision model upgraded with DRA.

6 Case Studies

This section illustrates how DRA can help improving the performance of E-BRB based decision models. The improvement becomes especially prominent when dealing with datasets with multiple classes, since DRA helps discarding similar, yet inconsistent rules from different classes. In this regard, the more classes the dataset has, the more inconsistent rules can be discarded during the activation stage, becoming DRA a more relevant process in the decision making procedure. To demonstrate this, two series of sensitivity and comparative tests were implemented. The first tests illustrate how the overall accuracy in datasets with numerous classes improves when DRA is applied. Meanwhile, the second tests were divided into three main groups depending on the number of classes in order to perform a comparative analysis to illustrate how DRA can improve the results in terms of accuracy, without compromising the efficiency of the DSS.

6.1 Case Study 1: How DRA Improves the Performance of E-BRB Based Decision Models

Three unbalanced, multi-class datasets named *Glass Identification*, *Ecoli* and *Yeast* were selected from the UCI repository [22] to perform sensitivity analyses for DRA.

Taking into account that $\lambda = 1$ is the base case where no penalty or reward is applied, this case study analyses the sensitivity of DRA by varying the range of acceptable values for λ. For example, when setting a threshold for a maximum λ of 10, DRA has less relevance than if that threshold is set to 20, since the upper threshold indicates the maximum degree of penalty applied to activated rules. In this context, by setting a maximum λ threshold it is possible to control the amount of inconsistent rules to be discarded, and therefore adjust the relevance of this essential aspect of DRA to evaluate its sensitivity.

For each range of λ values, each dataset was randomised by permuting its samples in order to perform ten series of 10-fold Cross Validation tests. For comparative purposes, all the tests performed in this case study use the ER algorithm included in the RIMER+ approach as inference engine. Table 2 illustrates how the accuracy (the percentage of data instances correctly predicted against the total number of instances) of the method augments as DRA becomes more relevant and able to use higher λ values. It also presents the minimum, mean and maximum results obtained from each

TABLE 2
Results Summary for the Sensitivity Tests Performed to Measure the Relevance of the λ Value, Controlled by DRA

		Range of λ values that DRA is able to select				
Dataset		[0, 3]	[0, 5]	[0, 10]	[0, 20]	[0, 50]
Glass Acc. (%)	Max	66.82	69.62	70.56	71.49	71.96
	Mean	**63.71**	**68.08**	**68.45**	**69.62**	**69.62**
	Min	60.28	65.42	65.88	67.75	67.75
	Time	14	14.1	16.3	18.6	18.7
Ecoli Acc. (%)	Max	81.84	83.63	84.22	84.52	85.11
	Mean	**80.86**	**82.61**	**83.27**	**83.39**	**83.95**
	Min	79.46	81.54	81.84	82.44	83.03
	Time	11.9	11.9	14.8	16.4	17.8
Yeast Acc. (%)	Max	39.48	52.56	54.58	55.32	55.32
	Mean	**38.67**	**51.87**	**53.61**	**54.48**	**54.52**
	Min	37.87	51.28	52.89	53.30	53.36
	Time	263	283	304	322	393

series of ten tests run for each interval of λ values and the average time per test, in milliseconds.

As Table 2 illustrates, the accuracy increases as DRA has more flexibility when tuning λ. This is because for the DRA method, being able to adjust λ from 0 to 50 is much less restrictive than selecting just among a few possible values between the [0, 3] interval, for example.

Note that Table 2 also includes the average time response (in milliseconds) that the system needs to derive an output from an input. As the computational cost might be a crucial aspect in several real-world decision problems, a compromise interval for λ might be chosen for a specific problem to obtain the desired balance between accuracy and time complexity. In this regard, it is important to highlight that the increase of accuracy is usually greater when dealing with small λ values. Table 2 illustrates that the increase of performance is more noticeable between the [0, 3] and [0, 10] columns than between the [0, 20] and the [0, 50] ones. Therefore, for some specific problems where computational cost is crucial and some precision errors can be assumed, it might be desirable to select a compromise interval for λ.

6.2 Comparative Analysis with Some Existing Approaches on More Datasets

Twenty-two benchmark datasets were collected from the UC Irvine Machine Learning Repository [22]. The EBRB-WA (performing the Weighted Average—WA) and RIMER + (performing the Evidential Reasoning—ER) decision models were tested with and without applying DRA as an activation algorithm (see Section 5). The 22 datasets were organised by their number of classes, creating three main groups of datasets: two-class datasets (Table 3), three-class datasets (Table 5) and multi-class datasets (Table 6). Each dataset was randomised by permuting its samples in order to perform a series of twenty 10-fold Cross Validation tests for each method and dataset. The average results of the 20 tests performed for each dataset and method were measured with: (1) accuracy (percentage of classes correctly classified from the total); (2) MAPE (Mean Absolute Percentage Error); (3) failed executions (number of tests where the method could not retrieve any result due to lack of relevant rules activated—that is, number of situations of incompleteness in the E-BRB); and (4) Average time response of the system in milliseconds for each input processed.

TABLE 3
Comparison of the Average Results for Two-Class Datasets

Dataset		DRA + WA	WA	DRA + ER	ER
Breast Cancer	Acc. (%)	94.52	**94.57**	**94.61**	94.59
	MAPE	0.037	0.050	0.036	0.048
	Failed	0	8.6	0	8.8
	Time	8.6	7.1	10	9.3
PIMA Diabetes	Acc. (%)	71.34	**73.59**	71.44	**73.39**
	MAPE	0.232	0.274	0.231	0.266
	Failed	0	0	0	0
	Time	11.7	9.1	14.3	12.3
Mammographic	Acc. (%)	**78.67**	77.61	**78.39**	77.64
	MAPE	0.162	0.228	0.164	0.211
	Failed	0	0.05	0	0.1
	Time	29.3	25.0	28.0	30.2
ILPD	Acc. (%)	70.42	**71.76**	70.24	**72.28**
	MAPE	0.213	0.265	0.214	0.250
	Failed	0	0	0	0
	Time	25.7	15.4	25.7	20.6
Banknote	Acc. (%)	**99.70**	98.60	**99.70**	98.53
	MAPE	0.002	0.197	0.002	0.171
	Failed	0	0	0	0
	Time	42.7	24.1	73.1	71.0
Blood Trans.	Acc. (%)	76.55	**77.33**	76.57	**77.37**
	MAPE	0.145	0.237	0.145	0.214
	Failed	0	0	0	0
	Time	24.6	13.0	34.6	20.1
BUPA Liver	Acc. (%)	65.72	**67.81**	64.9	**67.65**
	MAPE	0.278	0.336	0.280	0.330
	Failed	0	0	0	0
	Time	4.3	2.9	5.8	4.4
AVERAGE	Acc. (%)	79.56	**80.18**	79.40	**80.20**
	MAPE	**0.153**	0.227	**0.153**	0.213
	Failed	**0**	1.23	**0**	1.27
	Time	20.98	**13.8**	27.35	**23.98**

As Table 3 illustrates, the average accuracy for two-class datasets like Breast Cancer Wisconsin, BUPA Liver Disorders or PIMA does not vary greatly (being even slightly worse when using DRA in some cases). Despite this fact, it is still possible to see a considerable decrease in the overall MAPE percentage error. This is so because DRA discards the inconsistent rules during the activation step and therefore eliminates much of the output uncertainty represented in the output belief degree distributions of RIMER+ and EBRB-WA methods, trying in this way to retrieve a more confident result. In any case, the accuracies retrieved in the tests completed for this research are comparable to some of the best performances of classifiers found in literature. In [23], the results of some state-of-the-art classifiers for the Breast Cancer Wisconsin and BUPA Liver Disorders datasets were compiled. For comparative purposes, the tests performed in [23] were emulated, randomising the original datasets by permuting their samples and running 20 series of tests using 75 percent of the samples for training and the remaining 25 percent for testing. Table 4 summarises the results.

TABLE 4
Average Accuracies Comparison of Two-Class Datasets with Some of the Most Popular Classifiers

Method	Dataset	
	Breast Cancer Wisconsin	BUPA Liver Disorders
DRA + WA	0.940	0.678
WA	0.947	0.684
DRA + ER	0.941	0.656
ER (RIMER +)	0.942	0.660
SVM	0.932	0.693
KNN	0.898	0.611
Naive Bayes	0.876	0.592
FDT	0.862	0.601

Source: [23].

SVM stands for support vector machine, KNN refers to the K-Nearest Neighbours classifier and FDT represents the Fuzzy Decision Tree algorithm (see [23]). Even if the results of DRA for the two-class datasets are comparable to some of the most advanced methods, DRA capabilities start being noticeable from the case of three-class datasets, as illustrated in Table 5. In such scenario, DRA needs to search more exhaustively for specific hypertuples to provide a confident result, becoming a more important step than in the two-classes case. Although for the Wine and Iris datasets the average accuracy improves just slightly, in the remaining cases the results improved considerably, getting around 6 percent higher accuracy on average when using DRA in both decision models. Moreover, DRA significantly enhances the precision of the system, measured in terms of decrease of the MAPE error, having a reduction of over 25 percent on average. However, as the number of classes of the tested datasets increase, the performance of DRA becomes more relevant. Table 6 highlights the accuracies obtained for the datasets with higher number of classes: for the Wine Quality (Red) dataset, the accuracy improves around 8 percent and the Yeast dataset case shows the greatest increase, of around 23 percent higher accuracy.

TABLE 5
Comparison of the Average Results for Three-Class Datasets

Dataset		DRA+ WA	WA	DRA+ ER	ER
Iris	Acc. (%)	**95.63**	95.1	**95.5**	95.2
	MAPE	0.030	0.128	0.031	0.145
	Failed	0	0	0	0
	Time	>3.3	>3.3	>3.3	>3.3
Blogger	Acc. (%)	**70.00**	65.40	**70.20**	63.90
	MAPE	0.159	0.226	0.165	0.236
	Failed	0	7.8	0	8.8
	Time	>5	>5	>5	>5
Balance Scale	Acc. (%)	**89.36**	71.60	**89.04**	76.68
	MAPE	0.071	0.207	0.073	0.179
	Failed	0	0	0	0
	Time	16.8	11.4	16.3	16.7
Wine	Acc. (%)	**96.4**	96.29	**96.46**	96.32
	MAPE	0.017	0.038	0.017	0.031
	Failed	0	0	0	0
	Time	>5.6	>5.6	11.2	5.6
Seeds	Acc. (%)	**92.14**	84.83	**92.02**	87.04
	MAPE	0.099	0.172	0.101	0.153
	Failed	0	0	0	0
	Time	9.5	4.7	9.5	9.5
Contraceptive Method Choice	Acc. (%)	**35.69**	30.47	**36.41**	32.17
	MAPE	0.542	0.580	0.542	0.582
	Failed	0	231.05	0	229.7
	Time	73.3	50.9	91.6	85.5
Thyroid	Acc. (%)	**97.18**	95.52	**97.19**	95.60
	MAPE	0.019	0.038	0.020	0.037
	Failed	0	42.4	0	43.75
	Time	425.5	290.3	509	954.9
Vertebral Column	Acc. (%)	**83.66**	74.6	**84.29**	75.56
	MAPE	0.129	0.266	0.123	0.247
	Failed	0	1	0	1
	Time	22.6	16.1	25.8	16.1
AVERAGE	Acc. (%)	**82.50**	76.72	**82.63**	77.80
	MAPE	**0.152**	0.207	**0.134**	0.201
	Failed	**0**	35.28	0	35.40
	Time	70.2	**48.41**	**83.96**	137.1

TABLE 6
Comparison of the Average Results for Multi-Class Datasets

Dataset		DRA+WA	WA	DRA+ER	ER
Glass	Acc. (%)	**70.26**	47.85	**69.65**	51.43
	MAPE	0.238	0.392	0.238	0.375
	Failed	0	3.55	0	3.35
	Time	>9.3	>9.3	18.7	9.3
Car Evaluation	Acc. (%)	**94.69**	90.43	**94.51**	92.43
	MAPE	0.037	0.119	0.039	0.100
	Failed	0	0	0	0
	Time	52.7	62.1	54.1	136.3
Wine Quality (Red)	Acc. (%)	**67.38**	60.36	**67.28**	61.37
	MAPE	0.068	0.098	0.068	0.096
	Failed	0	0	0	0
	Time	193.2	113.8	327.7	258.3
Ecoli	Acc. (%)	**83.75**	19.53	**83.76**	33.72
	MAPE	0.160	0.568	0.158	0.466
	Failed	0	1	0	1
	Time	17.8	8.9 ms	17.8	11.9
Knowledge	Acc. (%)	**80.67**	73.77	**80.71**	75.07
	MAPE	0.102	0.166	0.101	0.161
	Failed	0	0	0	0
	Time	11.6	7.8 ms	11.6	7.8
Yeast	Acc. (%)	**54.15**	31.2	**54.13**	30.91
	MAPE	0.327	0.547	0.326	0.516
	Failed	0	0	0	0
	Time	334.9	203.5	393.5	251.3
Vowel Context	Acc. (%)	**98.74**	46.09	**98.69**	53.28
	MAPE	0.005	0.193	0.005	0.158
	Failed	0	0	0	0
	Time	49.5	48.5	38.4	85.9
AVERAGE	Acc. (%)	**78.52**	52.31	**78.39**	56.88
	MAPE	**0.134**	0.298	**0.134**	0.267
	Failed	**0**	0.65	0	0.62
	Time	95.57	**64.84**	123.1	**108.7**

The results obtained for the multi-class datasets are among the best ones found in literature. Table 7 compares the results of the multi-class datasets with those retrieved by [24], which also presents methodologies to specifically address datasets with numerous classes using an ANN (Artificial Neural Network) structure as a base. The authors in [25] present a series of results obtained from diverse classifiers based on linear discriminant analysis (LDA) and

TABLE 7
Percent Error Comparison for Multi-Class Datasets

	Dataset			
Method	Glass	Yeast	Ecoli	Car*
DRA + WA	29.74	45.85	16.25	5.31
DRA + ER	30.35	45.86	16.24	5.49
OAAE	28.64	47.51	14.50	9.83
OAA	29.57	47.72	14.80	11.39
OAO	31.00	47.31	14.51	9.25
A&O	30.04	47.11	14.21	9.42
ECOC	31.86	47.44	14.80	10.18

Source: [24].
**Car refers to the Car Evaluation dataset.*

SVM for a number of datasets. The same tests were emulated, using half of the dataset for training and the other half for testing purposes. Table 8 compiles some of the accuracies and their standard deviations (in parenthesis) from [25] with those obtained from this research.

As Tables 5 and 6 illustrate, adding DRA as an activation step for rule-base classifiers boosts the output accuracy of the system and enhances the accuracy of multi-class datasets to competitive levels, compared to other state-of-the-art methodologies. The ER and weighted average algorithms were selected for these studies as examples. However, thanks to the flexibility of the DRA method, there are chances to further improve the accuracies obtained in this study by applying it to other decision models and inference algorithms. It is worth noting that number of classes and increase of accuracy using DRA are not perfectly correlated. Although the increase of accuracy becomes more significant as the number of classes augment, the relationship between these two factors is not linear. For example, the increase of accuracy for the Vertebral Column dataset is greater that the Wine Quality (Red) one, even if the number of classes of the latter is double (3 and 6 classes, respectively). However, even without having an exact relationship between these two factors, the case studies presented in this research demonstrate the significant increase of accuracy when dealing with multi-class data. This enhancement is further underpinned when checking the MAPE values obtained from the case studies for all datasets, showing how DRA improves the precision of the two rule-based classifiers.

One last issue to note is the increase of time when using DRA. Even if DRA has been designed to work as efficiently as possible, adding an activation algorithm between the rule activation and inference processes adds some computational effort. However, this increase is relatively small, compared to the prohibitive cost that would be needed to pre-process, in order to optimise, all the parameters of an entire E-BRB. For example, the E-BRB of the Iris Dataset (one of the smallest datasets used in this study) contains no less than 2,200 parameters to be optimised for each fold of the cross validation test. As DRA is based on tuning a single λ parameter, it is far simpler and still significantly effective. In addition, discarding irrelevant and inconsistent pieces of information using DRA simplifies subsequent aggregation methods and reduces their computational cost. This can be seen in some large datasets, such as Thyroid or Car Evaluation, where thanks to the rule-discarding process performed by DRA, the overall time complexity was substantially reduced.

Another important feature demonstrated by DRA is its ability to avoid incompleteness situations by learning from contextual data. In this regard, failed executions occurring in some datasets due to lack of information could be approached. For example, failed executions found in Breast Cancer Wisconsin (with an average of 8.7 failed executions), Thyroid (43), Glass (3.45) and Contraceptive Method Choice (230) were eliminated and results were inferred for every case.

7 CONCLUSION

A new DRA method was proposed to select the most relevant information to be aggregated within a data-driven decision model. In the case of data with numerous classes, this research has demonstrated how focusing just on the most relevant information not only simplifies subsequent aggregation methods, but also increases the overall

TABLE 8
Accuracy Rates Compared to the Results Presented in [25]

	2-class		3-class		Multi-class	
Method	**PIMA**	**BUPA**	**Wine**	**Iris**	**Thyroid**	**Glass**
DRA + WA	69.40 (2.17)	65.08 (2.34)	**97.30 (1.82)**	95.55 (1.25)	**96.87 (0.29)**	**70.95 (1.66)**
WA	73.59 (2.23)	67.86 (2.70)	93.48 (3.09)	94.93 (2.96)	94.21 (0.48)	50.66 (4.12)
DRA + ER	72.52 (2.41)	65.60 (2.77)	95.39 (1.77)	95.60 (1.58)	**96.97 (0.23)**	68.75 (2.24)
RIMER+	69.86 (2.72)	65.54 (4.24)	94.83 (1.95)	94.26 (1.69)	94.81 (0.44)	54.57 (5.24)
SVM (LK)	**77.76 (1.85)**	67.51 (3.16)	96.40 (1.21)	96.40 (1.21)	96.19 (1.57)	61.21 (3.08)
LDA (LK)	68.26 (1.41)	54.97 (1.48)	71.24 (3.72)	71.24 (3.72)	88.33 (2.48)	52.90 (3.47)
LapSVM (LK)	68.52 (3.86)	68.53 (6.92)	96.85 (1.48)	**96.85 (1.48)**	96.11 (2.30)	63.08 (4.32)
LapRLS (LK)	66.78 (1.05)	**71.08 (13.4)**	96.52 (0.98)	96.52 (0.98)	85.56 (2.48)	58.88 (3.33)
DRLSC (LK)	**77.60 (2.27)**	68.73 (3.66)	**97.75 (0.75)**	**97.75 (0.75)**	85.19 (2.76)	58.04 (3.90)
SSDR (LK)	**77.94 (2.00)**	68.67 (3.73)	**98.09 (1.19)**	**98.09 (1.19)**	93.70 (2.68)	58.97 (3.78)
SVM (RBFK)	77.21 (1.73)	**70.58 (5.64)**	85.17 (2.89)	85.17 (2.89)	96.39 (1.66)	68.79 (3.85)
LapSVM (RBFK)	77.01 (2.78)	64.01 (5.26)	88.76 (2.54)	88.76 (2.54)	94.81 (2.00)	68.41 (5.06)
LapRLS (RBFK)	77.18 (2.50)	67.37 (4.00)	93.60 (1.50)	93.60 (1.50)	96.48 (2.08)	**71.50 (3.00)**
DRLSC (RBFK)	67.89 (3.25)	61.21 (2.89)	87.42 (4.13)	87.42 (4.13)	96.30 (1.38)	69.07 (3.00)
SSDR (RBFK)	75.96 (1.45)	**71.10 (1.57)**	94.27 (1.45)	94.27 (1.45)	**96.67 (1.86)**	**71.96 (1.39)**

The three best results obtained for each dataset are highlighted in bold.

accuracy of the system. At the same time, DRA provides an elegant, yet effective way of tackling both incompleteness and inconsistency situations at once, using a simple, flexible, efficient and effective methodology background.

Further research is being currently carried out to enhance the interpretability of E-BRBs, reducing the number of rules and attributes and simplifying the model. In this regard, many possibilities are available to approach this problem, such as including clustering algorithms to group similar extended belief rules or data instances.

Although RIMER+ has been chosen as a representative framework to exemplify the DRA approach; any other instance-based (or data-driven) model could have been used, just by adapting Eq. (13) to the context of the model. In this regard, at the moment DRA is being embedded in some of the most popular classifiers found in literature. This may lead to higher accuracies and lower percentage error values in the near future. Moreover, there are chances of further improving the accuracies presented in this research by using any other similarity and agreement functions within the DRA framework. The combined use of optimisation techniques along with DRA may also retrieve positive resuts. Especially for the case of multi-class data, DRA can be considered as a supportive method, whether the rule-base is optimised or not.

Note that DRA not only opens the possibility of achieving higher overall accuracies, but also provides mechanisms to add more functionality to the decision model. One option could be measuring the quality of the decision result, based on both the consistency and size of the hypertuple of activated rules. Techniques of quality and risk assessment could be developed and included as an integral part of the DRA procedure to enhance the feedback information to be retrieved.

ACKNOWLEDGMENTS

The authors would like to express their gratitude and appreciation to the reviewers for their constructive and helpful comments that have been a great support to improve the quality of the paper. This work was supported by the Computer Science Research Institute at the University of Ulster.

REFERENCES

[1] P. Angelov and R. Yager, "A simple fuzzy rule-based system through vector membership and kernel-based granulation," in *Proc. IEEE Int. Conf. Intell. Syst.*, 2010, pp. 349–354.

[2] J. Huysmans, K. Dejaeger, C. Mues, J. Vanthienen, and B. Baesens, "An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models," *Decision Support Syst.*, vol. 51, no. 1, pp. 141–154, 2011.

[3] A. P. Sinha and H. Zhao, "Incorporating domain knowledge into data mining classifiers: An application in indirect lending," *Decision Support Syst.*, vol. 46, no. 1, pp. 287–299, 2008.

[4] S. J. H. Yang, "Fuzzy rule base systems verification using high-level Petri nets," *IEEE Trans. Knowl. Data Eng.*, vol. 15, no. 2, pp. 457–473, Feb. 2003.

[5] X. He, W. C. Chu, H. Yang, and S. J. H. Yang, "A new approach to verify rule-based systems using petri nets," in *Proc. IEEE 23th Annu. Int. Conf. Comput. Softw. Appl.*, pp. 462–467, 1999.

[6] S. Kim, "Checking a rule base with certainty factor for incompleteness and inconsistency," in *Proc. 2nd Int. Conf. Inform. Process. Manage. Uncertainty Knowl.-Based Syst. Uncertainty Intell. Syst.*, 1988, vol. 313, pp. 201–208.

[7] N. Fantana and J. Weisbrod, "Detecting local inconsistency and incompleteness in fuzzy rule bases," in *Proc. 4th Eur. Congr. Fuzzy Intell. Tech.*, 1996, vol. 1, pp. 656–660.

[8] J. Liu, L. Martínez, A. Calzada, and H. Wang, "An extended belief rule based inference methodology and its rule-base generation by learning from examples," *Knowl. Based Syst.*, vol. 53, pp. 129–141, 2013.

[9] A. Calzada, J. Liu, H. Wang and A. Kashyap, "Dynamic rule activation for extended belief rule bases," in *Proc. Int. Conf. Mach. Learning Cybern.*, 2013, pp. 1836–1841.

[10] H. Wang, "Contextual probability and neighbourhood counting," *Intl J. Softw. Inform.*, vol. 6, no. 3, pp. 435–452, 2012.

[11] 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 Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 36, no. 2, pp. 266–285, Feb. 2006.

[12] J. Liu, L. Martinez, H. Wang, A. Calzada, and S. W. Chen, "A new belief rule base representation scheme and its generation by learning from examples," in *Proc. 10th Int. FLINS Conf. Uncertainty Model. Knowl. Eng. Decision Making*, 2012, pp. 1030–1035.

[13] X. S. Si, C. H. Hu, J. B. Yang, and Z. J. Zhou, "A new prediction model based on belief rule base for system's behaviour prediction," *IEEE Trans Fuzzy Syst.*, vol. 19, no. 4, pp. 636–651, Aug. 2011.

[14] W. J. Li, J. Liu, H. Wang, A. Calzada, R. M. Rodriquez, and L. Martínez, "A qualitative decision making model based on belief linguistic rule based inference methodology," *Int. J. Uncertainty Fuzziness Knowl. Based Syst.*, vol. 20, no. 1, pp. 105–118, 2012.

[15] Z. J. Zhou, C. H. Hu, J. J. Yang, D. L. Xu, and D. H. Zhou, "Online updating belief-rule-based systems using the RIMER approach," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 41, no. 6, pp. 1225–1243, 2011.

[16] A. Calzada, J. Liu, H. Wang, and A. Kashyap, "A novel spatial belief rule-based intelligent decision support system," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, 2013, pp. 639–644.

[17] H. Ishibuchi and Y. Nojima, "Discussions on interpretability of fuzzy systems using simple examples", in *Proc. Int. IFSA/EUSFLAT Conf.*, 2009, pp. 1649–1654.

[18] M. J. Gacto, R. Alcalá, and F. Herrera, "Interpretability of linguistic fuzzy rule-based systems: An overview of interpretability measures," *Inform. Sci.*, vol. 181, pp. 4340–4360, 2011.

[19] B. Martin, "Instance-based learning: Nearest neighbour with generalisation," Thesis submitted at the Dept. Comput. Sci., Univ. Waikato, Hamilton, New Zealand, 1995.

[20] H. Wang, I. Düntsch, G. Gediga, and A. Skowron, "Hyperrelations in version space," *Int. J. Approx. Reason.*, vol. 36, no. 3, pp. 223–241, 2004.

[21] S. H. Cha, "Comprehensive survey on distance/similarity measures between probability density functions," *Int. J. Math. Models Methods Appl. Sci.*, vol. 1, no. 4, pp. 300–307, 2007.

[22] A. Frank and A. Asuncion, (2010). UCI machine learning repository. School Comput. Sci., Univ.California, Los Angeles, CA, USA. [Online]. Available: http://archive.ics.uci.edu/ml

[23] C. Y. Fan, P. C. Chang, J. J. Lin, and J. C. Hsieh, "A hybrid model combining case-based reasoning and fuzzy decision tree for medical data classification," *Appl. Soft Comput.*, vol. 11, no. 1, pp. 632–644, 2011.

[24] T. H. Oong and N. A. M. Isa, "One-against-all ensemble for multiclass pattern classification," *Appl. Soft Comput.*, vol. 12, no. 4, pp. 1303–1308, 2012.

[25] F. Wu, W. H. Wang, Y. Yang, Y. T. Zhuang, and F. P. Nie, "Classification by semi-supervised discriminative regularization," *Neurocomputing*, vol. 73, no. 10–12, pp. 1641–1651, 2010.

Alberto Calzada received the BSc degree in Computer Sciences and the MSc degree in 3D computer graphics from the University of Jaen, Jaen, Spain, in 2008 and 2010, respectively. He received the PhD degree in spatial decision support systems from the University of Ulster, UK, in 2014. His current research interests include spatial analysis, DSS, GIS and health-care applications. He has 17 peer-reviewed publications in international journals and conference proceedings.

Jun Liu received the BSc and MSc degrees in applied mathematics, and the PhD degree in information engineering from Southwest Jiaotong University, Chengdu, China, in 1993, 1996, and 1999, respectively. He is currently a senior lecturer in Computer Science at University of Ulster, Newtownabbey, United Kingdom. He has been working in the field of AI for many years. His current research interests include nonclassical logic and reasoning methods for intelligent systems; intelligent DSSs and information management, with applications in management, engineering, and industry field, etc. (e.g., safety and risk analysis; situation awareness and emergency systems; video scenario recognition); information fusion and data combinations; data mining and KBS; applied computational intelligence for uncertainty analysis and optimisation. He is a member of the IEEE.

Hui Wang received the BSc and MSc degrees in AI in Jilin University, Changchun, China, and the PhD degree in AI from the University of Ulster, Newtownabbey, United Kingdom. His current research interests include learning and reasoning (contextual probability, similarity kernels; lattice machine), and applications in text/video/time series/spectral data understanding. He has over 170 research publications in international journals and conference proceedings. He is an Associate Editor of the *IEEE Transactions SMC Cybernetics* since 2009, an Associate Editor of the *Springer International Journal of Machine Learning and Cybernetics* since 2010 and a member of Editorial Board, *International Journal of Computational Intelligence Systems* since 2009 He is a PC member of over 20 international conferences. He is a member of the IEEE.

Anil Kashyap received the PhD degree from the School of the Built Environment at University of Ulster, Newtownabbey, United Kingdom. He is an urban planning professional with 20 years of industry and academic experience in India and abroad. His current research interests include financing urban infrastructure, urban regeneration, appraisal and finance, market analysis, value creation and sustainability in real estate. He has published his research widely and is currently member of editorial board of *American Journal of Energy Research*. He is a chartered member of RICS, chartered member of Royal Town Planning Institute London, fellow of Higher Education Academy, United Kingdom, and a fellow of the Institute of Town Planners, India.