J Glob Optim (2011) 51:255–270
DOI 10.1007/s10898-010-9605-x

Optimization algorithm for learning consistent belief rule-base from examples

Jun Liu · Luis Martinez · Da Ruan · Rosa Rodriguez · Alberto Calzada

Received: 1 April 2010 / Accepted: 26 August 2010 / Published online: 22 September 2010

Abstract A belief rule-based inference approach and its corresponding optimization algorithm deal with a rule-base with a belief structure called a belief rule base (BRB) that forms a basis in the inference mechanism. In this paper, a new learning method is proposed based on the given sample data for optimally generating a consistent BRB. The focus is given on the consistency of BRB knowing that the consistency conditions are often violated if the system is generated from real world data. The measurement of BRB inconsistency is incorporated in the objective function of the optimization algorithm. This process is formulated as a non-linear constraint optimization problem and solved using the optimization tool provided in MATLAB. A numerical example is demonstrated the effectiveness of the proposed algorithm.

Keywords Belief rule base · Optimization · Consistency · Learning

1 Introduction

To extend the rule-base framework for covering credibility uncertainty, a generic Rule-base Inference Methodology using the Evidential Reasoning approach (RIMER) was proposed in [15] by combining decision science, fuzzy logic [19] and Dempster-Shafer (D-S) theory of evidence [3,9]. In the RIMER framework, a rule base designed on the basis of a belief structure, called a Belief Rule Base (BRB), is used to capture uncertainty and non-linear

J. Liu (✉) · A. Calzada
School of Computing and Mathematics, University of Ulster, Newtownabbey, Co. Antrim BT37 0QB, Northern Ireland, UK
e-mail: j.liu@ulster.ac.uk

L. Martinez · R. Rodriguez · A. Calzada
Department of Computer Science, University of Jaén, 23071 Jaén, Spain

D. Ruan
Belgian Nuclear Research Centre (SCK.CEN), Boeretang 200, 2400 Mol, Belgium

D. Ruan
Department of Applied Mathematics & Computer Science, Ghent University, 9000 Gent, Belgium

relationships between the parameters, and the inference is implemented using the evidential reasoning algorithm in [16] and [17].

Belief rule base forms a basis in the inference mechanism of RIMER, which is a framework for representing expert knowledge. In a BRB, input data, belief structures, attribute weights, rule weights, and parameters of referential values for each attribute are combined to generate activation weights for rules, and all activated belief rules are then combined to generate appropriate conclusions using the evidential reasoning approach, such a combination process is finally formulated as non-linear objective functions with some constraints, which provides the flexibility to be handled by many optimization algorithms [8]. Those parameters involved in RIMER are the important factors for the system performance.

However, it is difficult to determine those weights and parameters elements entirely subjectively, in particular for a large scale rule base with hundreds of rules. Accordingly, optimal models for training the elements of general belief rule bases in RIMER have been proposed in [18], which has also been revised and applied into engineering system safety analysis and leak detection [7,13]. The optimization process is formulated as non-linear objective functions to minimize the differences between observed outputs and the outputs of a belief rule base whilst parameter specific limits and partial expert judgments can be formulated as constraints. The optimization problems can be solved using existing tools such as the optimization tool box provided in *MATLAB* [2].

One of the important issues in designing the rule-base, i.e., the consistency of generated rule-base is however not addressed in the above work. This is actually important for a rule-based system to exhibit a reliable performance because the inconsistency generally exists in the knowledge itself provided by the domain experts and in the process of knowledge representation and acquisition as well.

Hence, the main focus of this paper is to extend the optimal algorithm in [18] to generate the consistent BRB from sample data. The measurement of BRB inconsistency is provided and incorporated into the objective function of the optimization algorithm, which can prevent the algorithm from generating rules that seriously contradict with each other or with the heuristic knowledge.

The paper is organized as follows. Section 2 briefly reviews the RIMER approach. Section 3 provides the detailed optimization algorithm including the way on how to measure the inconsistency of BRB. Section 4 provides a numerical example to demonstrate the effectiveness of the proposed algorithm. Section 5 concludes the paper.

2 Outline of RIMER

This section reviews the RIMER framework in [15]. A belief rule-base is given by $\boldsymbol{R} = \{R_1, R_2, \ldots, R_L\}$, where the kth rule can be represented as follows:

$$R_k\text{: } \textit{IF } \boldsymbol{U} \text{ is } \boldsymbol{A^k} \textit{ THEN } \boldsymbol{D} \text{ with belief degree } \boldsymbol{\beta^k}\text{, with a rule weight } \theta_k \text{ and attribute weights } \delta_{K1}, \delta_{K2}, \ldots, \delta_{KT}. \quad (1)$$

This is the vector form of *a belief rule*. Here $\boldsymbol{U}$ represents the antecedent attribute vector $(U_1, \ldots, U_T)$, $\boldsymbol{A^k}$ the packet antecedents $\{A_1^k, \ldots, A_T^k\}$ ($A_i^k (j = 1, \ldots, T)$ is the referential value of the jth antecedent attribute in the kth rule), T the number of antecedent attributes used in the rule, $\boldsymbol{D}$ the consequent vector $(D_1, \ldots, D_N)$, and $\boldsymbol{\beta^k}$ the vector of the belief degrees $(\beta_{1k}, \ldots, \beta_{Nk})$, for $k \in \{1, \ldots, L\}$ with $\sum_{i=1}^{N} \beta_{ik} \leq 1$. β_{ik}measures the degree to which D_i is the consequent if the input activates the antecedent A^k in the kth rule for

Table 1 A belief rule expression matrix

Output	Input					
	$A^1(w_1)$	$A^2(w_2)$	…	$A^k(w_k)$	…	$A^L(w_L)$
D_1	β_{11}	β_{12}	…	β_{1k}	…	β_{1L}
⋮	⋮	⋮	…	⋮	…	⋮
D_N	β_{N1}	β_{N2}	…	β_{Nk}	…	β_{NL}

$i = 1, \ldots, N, k = 1, \ldots, L$. L is the number of rules and N is the number of possible consequents. If $\sum_{i=1}^{N} \beta_{ik} = 1$, the output assessment or the kth rule is complete; otherwise, it is incomplete. The belief rule-base in the form (1) is referred to as *a fuzzy belief rule-base* where either $\boldsymbol{U}$or $\boldsymbol{D}$ or both could be described as linguistic terms modeled typically by fuzzy membership functions. The rule-base can be summarized using a belief rule expression matrix shown in Table 1.

Take, for example, the following belief rule in safety analysis:

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)}

where {(good, 0), (average, 0), (fair, 0.7), (poor, 0.3)} is a belief distribution representation for safety consequent, stating that it is 70% sure that safety level is fair and 30% sure that safety level is poor. In this belief rule, the total belief degree is $0.3 + 0.7 = 1$, so that the assessment is complete. This kind of rule reflects another kind of uncertainty caused because sometime an expert is unable to establish a strong correlation between premise and conclusion. In other words, evidence available is not sufficient or experts are not 100% certain to believe in a hypothesis but only to degrees of belief. The linguistic value set for failure rate is given by {very low, low, reasonably low, average, reasonably frequent, frequent, and highly frequent}, which can be represented by fuzzy membership functions respectively.

In the matrix, w_k is the activation weight of A^k, which measures the degree to which the kth rule is weighted and activated. w_k is calculated as follows:

$$w_k = \left(\theta_k * \prod_{j=1}^{T} \left(\alpha_j^k\right)^{\bar{\delta}_j}\right) \Bigg/ \left(\sum_{t=1}^{L} \left[\theta_t * \prod_{l=1}^{T} \left(\alpha_l^t\right)^{\bar{\delta}_l}\right]\right), \quad \text{here } \bar{\delta}_j = \delta_j \Bigg/ \left(\max_{j=1,\ldots,T} \{\delta_j\}\right), \tag{2}$$

Here θ_k and δ_i can be assigned to any value in $\mathbf{R}^+$ depending on a specific application because w_k will be normalized so that $w_k \in [0, 1]$ by (2). Without loss of generality, however, we assume that $\theta_k \in [0, 1]$, $(k = 1, \ldots, L)$ and $\delta_j \in [0, 1] (j = 1, \ldots, T)$. Note that $0 \leq w_k \leq 1 (k = 1, \ldots, L)$ and $\sum_{i=1}^{L} w_i = 1$. In addition, $\alpha_j^k \in \{\alpha_{js}; s = 1, \ldots, S_j\} (j = 1, \ldots, T)$ is the degree of belief to which the input for U_j belongs to A_j^k of the jth individual antecedent in the kth rule, called individual matching degree, where $A_j^k \in \left\{A_{js}; S = 1, \ldots, S_j\right\} \in$ (S_j is the total number of referential values for the attribute U_j).

In RIMER, two major ways of representing the referential values are utilized, i.e., the way using linguistic terms which can be represented by fuzzy membership function, and the way of using linguistic terms which can be represented by linear utility function. Accordingly, the ways to calculate the individual matching degree are different.

2.1 The ways to calculate the individual matching degree

This is given a bit in more detail because it is necessary to determine the inconsistency of rule-base.

2.1.1 *For the fuzzy membership function case*

Suppose $\alpha_j^k = A_j^k(x_j)$ is the fuzzy membership degree of a given real input x_j for the attribute U_j to the linguistic term A_j^k. The fuzzy membership function can be applied in different forms depending on the system. In [6], the straight-line membership functions are used due to its advantage of simplicity, such as the triangular membership function and trapezoidal membership function. In this paper continuous and differentiable Gaussian function is used, i.e.,

$$A_j^k(x_j) = \exp\left(-(1/2)\left((x_j - c_j^k)\Big/(\sigma_j^k)\right)^2\right), \tag{3}$$

where c_j^k is the central value of the fuzzy membership function and σ_j^k is the variance at the central value. The reason to choose the Gaussian on is that a generalized formula in analytical form for the similarity measure of two Gaussian fuzzy sets with different widths is available. Although triangular/trapezoidal ones can be also applied, however, for any two fuzzy sets described with triangular/trapezoidal membership functions, there are nine total possible overlapping cases, that means no single analytical form can cover all the cases, which will cause some computation issues as well.

2.1.2 *For the utility case*

If the referential values for each antecedent attribute are given by numerical values using utility-based equivalence transformation techniques [14], where the equivalence rules need to be extracted from decision makers to transform a value to an equivalent expectation, thereby relating a particular value to each referential value, then the individual matching degree can be calculated based on the following procedure.

Suppose $\boldsymbol{U} = \{U_j;\ j = 1, \ldots, T\}$ is the set of antecedent attributes, $A_j = \left\{A_{js}, s = 1, \ldots, S_j\right\}$ $(j = 1, \ldots, T)$ is the referential set of the antecedent attribute U_j, and a value a_{js} for an antecedent attribute U_j is judged to be equivalent to a referential value $A_{js}(s = 1, \ldots, S_j)$, or

$$a_{js} \ \text{ means } \ A_{js}(s = 1, \ldots, S_j). \tag{4}$$

Without loss of generality, suppose U_i is a 'profit' attribute, that is a large value $a_{j(s+1)}$ is preferred to a smaller value a_{js}. Let a_{jS_j} be the largest feasible value and a_{j1} the smallest. Then an input value x_j^* for U_j may be represented using the following equivalent expectation:

$$S(x_j^*) = \left\{\left(a_{js}, \gamma_{js}\right); s = 1, \ldots, S_j\right\}, \tag{5}$$

where

$$\gamma_{js} = \frac{a_{j(s+1)} - x_j^*}{a_{j(s+1)} - a_{js}}, \quad \gamma_{j(s+1)} = 1 - \gamma_{js} \quad \text{if} \quad a_{js} \leq A_j^* \leq a_{j(s+1)} \tag{5a}$$

$$\gamma_{jt} = 0 \quad \text{for} \quad t = 1, \ldots, S_j, t \neq s, s+1 \tag{5b}$$

$\alpha_j^k \in \{\gamma_{js}; s = 1, \ldots, S_j\}(j = 1, \ldots, T)$ is the belief degree to which the input for U_j belongs to A_j^k $\left(\in \{A_{js}, s = s = 1, \ldots, S_j\}\right)$ of the jth individual antecedent in the kth rule.

2.2 Reasoning based on the evidential reasoning algorithm

Based on the above belief rule expression matrix, we apply the analytical evidential reasoning (ER) algorithm in [18] (which is equivalent to the recursive ER algorithm in [16] and [17]) to combine rules and generate final conclusions. The combined degree of belief β_j in D_j is generated as follows:

$$\beta_i = \frac{\mu * \left[\prod_{k=1}^{L}\left(w_k\beta_{i,k} + 1 - w_k\sum_{i=1}^{N}\beta_{i,k}\right) - \prod_{k=1}^{L}\left(1 - w_k\sum_{i=1}^{N}\beta_{i,k}\right)\right]}{1 - \mu * \left[\prod_{k=1}^{L}(1 - w_k)\right]}, \quad i = 1, \ldots, N \tag{6}$$

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

Because the inference process in the proposed method is a kind of non-linear algorithm using the ER algorithm, the assessment of the parameters of the inference procedure is not easy. That is indeed the main objective of the proposed work trying to use self-tuning to help efficiently generate near-optimal parameters.

The logic behind the approach is that if the consequent in the kth rule includes D_i with $\beta_{ik} > 0$ and the kth rule is activated then the overall output must be D_i to a certain degree. The degree is measured by both the degree to which the kth rule is important to the overall output and the degree to which the antecedents of the kth rule are activated by the actual input $\boldsymbol{x}^*$.

3 Extended optimization algorithm for generating consistent BRB

The performance of inference can be improved if the following parameters in (6) are adjusted by autonomous learning and if they are not given a priori or only known partially or imprecisely: (1) rule weights $\theta_k(k = 1, \ldots, L)$ and attribute weights $\delta_j(j = 1, \ldots, T)$; (2) the degrees of belief $\beta_{ik}(i = 1, \ldots, N; k = 1, \ldots, L)$. For constraint conditions on each parameter, we refer to the ref. in [18]

Figure 1 shows the process of training a belief rule base [18], where $\hat{x}_m$ is a given input, $\hat{y}_m$ the corresponding observed output, either measured using instruments or assessed by experts, y_m the simulated output generated by the belief rule based system, and $\xi(P)$ the difference between $\hat{y}_m$ and y_m, as defined later.

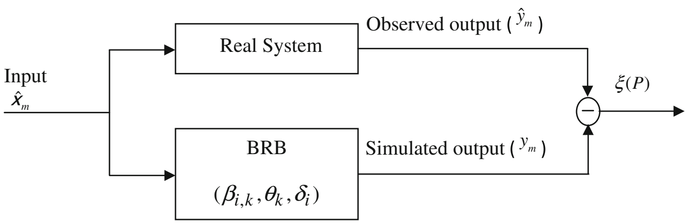

Fig. 1 Illustration of the optimal learning process

It is desirable that $\xi(P)$ is as small as possible where P is the vector of training parameters including $\beta_{i,k}$, θ_k, and δ_i. This objective is difficult to achieve if a BRB is constructed using expert judgments only. An optimal learning method is used to adjust the parameters to minimize the difference between the observed output ($\hat{y}_m$) and the simulated output (y_m), or $\xi(P)$.

3.1 Initial optimization algorithm

The objective function is to minimize the mean square error (MSE) criterion defined as follows:

$$\min_{\boldsymbol{P}} \{\xi(\boldsymbol{P})\} \tag{7}$$

where, $\xi(\boldsymbol{P}) = \frac{1}{M}\sum_{m=1}^{M}(y_m - \hat{y}_m)^2$, $y_m = \sum_{j=1}^{N} u(D_j)\beta_j(m)$ is the expected utility (or score) of the output corresponding to the given input x_m^* for the mth input vector in a training set ($m = 1, \ldots, M$), $\beta_j(m)$ is given by (6) for the mth input in the training set ($m = 1, \ldots, M$). M is the number of points in the training set, $\hat{y}_m$ is the measured confidence score or the expected output. ($y_m - \hat{y}_m$) is the residual at the mth point. The constraint conditions for a belief rule base are given as follows [18]:

(a) A belief degree (subjective probability) must not be less than zero or more than one

$$0 \le \beta_{jk} \le 1,\ j = 1, \ldots, N;\ k = 1, \ldots L. \tag{7a}$$

(b) If the kth belief rule is complete, its total belief degree in the consequent is equal to one, i.e.,

$$\sum_{j=1}^{N} \beta_{jk} = 1. \tag{7b}$$

(c) An attribute weight is normalized so that it is between zero and one

$$0 \le \delta_i \le 1, \qquad i = 1, \ldots, T. \tag{7c}$$

(d) A rule weight is normalized so that it is between zero and one

$$0 \le \theta_k \le 1, \qquad k = 1, \ldots, L. \tag{7d}$$

Therefore, all generations of the optimization algorithm are used to obtain the minimal mean square error. Equation (7) is a multi-variable constrained non-linear single-objective optimization problem and can be solved using existing non-linear optimization software packages. In this paper, the Matlab Optimization Toolbox is used [2]. Since the function to be minimized and the constraints are all continuous, this optimization problem can be solved using the optimization function FMINCON provided in MATLAB.

3.2 Extended optimization algorithm to minimize the inconsistency

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. So we focus only on the consistency among the rules in the generated rule base.

Fuzzy 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

[5]. Before we discuss the definition of the consistency, we first provide the definition of the similarity of rule antecedent (SRA) and the similarity of rule consequent (SRC) again with the help of fuzzy similarity measures, similarity measure of utility, and similarity measure of discrete probability distribution.

3.2.1 Similarity measures

A. Similarity measures of rule antecedents

This similarity depends on the ways of representing the referential values in the rule antecedent attributeand can be determined in the following two ways: for fuzzy sets or for utility values.

(a) Similarity measure for fuzzy sets

Similarity measure for fuzzy sets indicates the degree to which two fuzzy sets are equal. For any two fuzzy sets A and B, the set-theoretic similarity measure usually used in interpretability analysis is in the following form [4]:

$$S_F(A,B) = (|A \cap B|)/(|A| + |B| - |A \cap B|), \tag{8}$$

where $|\cdot|$ denotes the cardinality of the set. For the Gaussian fuzzy sets A and B defined in (3), the cardinality calculation becomes an integration as follows [4]:

$$|A| = \int_{-\infty}^{\infty} A(x)dx = \int_{-\infty}^{\infty} \exp\left[-\left((x - c_A)/\sigma_A\right)^2\right] dx = \sqrt{\pi}\sigma_A$$

$$|B| = \int_{-\infty}^{\infty} B(x)dx = \int_{-\infty}^{\infty} \exp\left[-\left(\frac{x - c_B}{\sigma_B}\right)^2\right] dx = \sqrt{\pi}\sigma_B,$$

$$|A \cap B| = \frac{\sqrt{\pi}}{2}[2\sigma_{\min} + \Omega],$$

$$\Omega = (\sigma_{\max} - \sigma_{\min})\operatorname{erf}\left(\frac{c_{\max} - c_{\min}}{\sigma_{\max} - \sigma_{\min}}\right) - (\sigma_{\max} + \sigma_{\min})\operatorname{erf}\left(\frac{c_{\max} - c_{\min}}{\sigma_{\max} + \sigma_{\min}}\right)$$

$$\sigma_{\max} = \max(\sigma_A, \sigma_B) \quad \text{(others are defined similarly)},$$

$$\operatorname{erf}(x) = \left(2/\sqrt{\pi}\right) \int_0^x e^{-t^2} dt.$$

More options about similarity measure for fuzzy sets can be found in [11] and [12].

(b) Similarity measure for utility values

For the comparison functions $d(a,b)$ for two utilities a and b, we use the Minkowski's (or Euclidean) distance given by

$$d(a,b) = |a - b|,$$

so the similarity between a and b is calculated as

$$S_u(a,b) = 1 - d(a,b). \tag{9}$$

B. Similarity measures of rule consequents

Because the rule consequent is represented as a probability distribution, it is to determine the similarity measure for probability distributions.

For the comparison functions $m(P, Q)$ for two probability distributions P and Q, we use the Minkowski's (or Euclidean) distance given by

$$d(P, Q) = \sum_{y \in Y} |p(y) - q(y)|^2, \text{ So } S_P(P, Q) = 1 - d(P, Q). \tag{10}$$

More options about m can be found in [1] and [20].

3.2.2 Consistency measure of BRB

Consider two rules in the rule base:

R_i: IF $\boldsymbol{U}_1$ is A_1^i and ... and $\boldsymbol{U}_T$ is A_T^i THEN $\boldsymbol{D}$ is $\{(D_1, \beta_{1i}), \ldots, (D_N, \beta_{Ni})\}$,

R_k: IF $\boldsymbol{U}_1$ is A_1^k and... and $\boldsymbol{U}_T$ is A_T^k THEN $\boldsymbol{D}$ is $\{(D_1, \beta_{1k}), \ldots, (D_N, \beta_{Nk})\}$.

The SRP of these two rules is defined as follows:

$\text{SRA}(i, k) = \min_{j=1}^{T} S_f(A_j^i, A_j^k)$ if A_j^i and A_j^k are represented by fuzzy membership functions; where $S_f(A, B)$ is defined in (8).

$\text{SRA}(i, k) = \min_{j=1}^{T} S_u(A_j^i, A_j^k)$ if A_j^i and A_j^k are represented by utility values; where $S_u(A, B)$ is defined in (9).

The SRC of these two rules is defined as follows:

$$\text{SRC}(i, k) = S_p[\{(D_1, \beta_{1i}), \ldots, (D_N, \beta_{Ni})\}, \{(D_1, \beta_{1k}), \ldots, (D_N, \beta_{Nk})\}],$$

where S is defined in (10).

Then the consistency of rule R_i and R_k is defined by [5]:

$$\text{Cons}(R_i, R_k) = \exp\left\{-\left(\text{SRA}(i, k)/\text{SRC}(i, k) - 1.0\right)^2 / \left(1/\text{SRA}(i, k)\right)^2\right\}. \tag{11}$$

An inconsistency degree of a rule base is suggested based on the consistency index provided in (11). At first, an inconsistency degree for the i-th rule is calculated as follows:

$$\text{Incons}(i) = \sum_{\substack{1 \le k \le L \\ k \ne i}} [1.0 - \text{Cons}(R_i, R_k)], \quad i = 1, \ldots, L, \tag{12}$$

The inconsistency degree of each rule is then summed up to indicate the inconsistency degree of a rule base:

$$\xi_{\text{Incons}} = \sum_{i=1}^{L} \text{Incons}(i), \tag{13}$$

which can be incorporated in the objective function of the algorithm.

3.2.3 Extended optimization formulation

Combining the inconsistency indices, the quality of a generated FBRB is evaluated with the following objective function to minimize the mean square error criterion and also minimize the inconsistency level:

$$\text{MIN}\{\xi + \lambda \xi_{\text{Incons}}\}, \tag{14}$$

where ξ is the same as in (7), ξ_{Incons} is provided in (13). λ is a weighting constant to control the consistency level. It will be predetermined according to the application context, not a parameter to be optimized.

4 Case study

4.1 Problem description

An example for oil pipeline leak detection in [13] is used here. We use the data to train and validate a BRB system 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. Each leak lasted up to a few hours and the size of the leaks was controlled through a valve. Figure 2 shows the inlet and outlet flow and pressure readings (f_0, f_1, p_0 and p_9, respectively) collected in about five and half hours during a leak trail, with the leak period clearly marked by the large discrepancy between the inlet and outlet flow readings.

4.2 Data and belief rule base

The difference between the inlet flow and the outlet flow, denoted by FlowDiff, and the average pipeline pressure change over time, denoted by PressureDiff, are the two important factors in detecting a leak in the pipeline. Under normal operations, when the inlet flow is larger (or less) than the outlet flow, the pressure in the pipeline will build up (or decrease) because the total content in the pipeline is increasing (or decreasing, respectively). However, if the pattern is violated, for example, when the inlet flow is larger than the outlet flow, yet the pressure in the line still decreases, then it is highly likely that there is a leak in the pipeline.

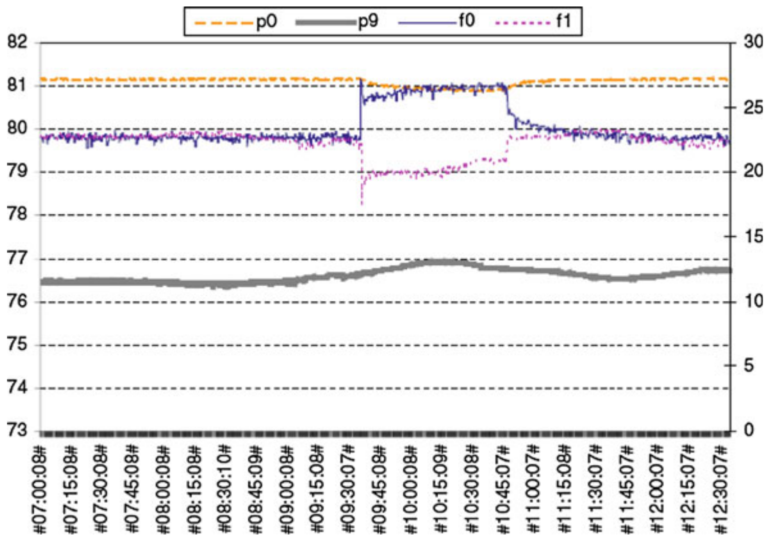

Fig. 2 Inlet and outlet flow and pressure readings

Table 2 Calculated flow and pressure difference

	FlowDiff $(f1 - f0)$	PressureDiff $(P_{\text{average}}(t) - P_{\text{average}}(t-1))$	Leak size t(h)
#07:00:18#	0.05	0	0
…	…	…	…
#10:04:08#	−6.15	−0.007	6.35
#10:04:18#	−6.15	0	6.38
#10:04:28#	−6.15	0	6.40
#10:04:39#	−6.1	0.0085	6.39
#10:04:49#	−6.1	0	6.35
…	…	…	…
#12:34:27#	−0.1	0	0

FlowDiff and PressureDiff are the two-antecedent attributes of the rule base and are calculated as follows:

$$U_1 = \mathit{FlowDiff}(t) = f_1(t) - f_0(t)$$

$$U_2 = \mathit{PressureDiff}(t) = [p_0(t) + \cdots + p_9(t)]/10 - [p_0(t-1) + \cdots p_9(t-1)]/10$$

where $f_1(t)$, $f_0(t)$, $p_0(t), \ldots,$ and $p_9(t)$ are collected instrument readings at time t. The consequent attribute is the leak rate, denoted by LeakSize. LeakSize values are controlled during the leak trial and therefore are more or less (though not exactly) known. Table 2 lists a few antecedent attribute values and the corresponding consequent attribute values.

We have chosen the following terms (words in natural language) for each type of numerical attribute in the table:

Flow difference partition:

NL: Negative large
NM: Negative medium
NS: Negative small
NVS: Negative very small
Zero
PS: Positive small
PM: Positive medium
PL: Positive large

The terms NL, NM, NS, NVS, Zero, PS, PM, and PL are assumed to be equivalent to the numerical value −7, −5, −3, −1, 0, 0.5, 1, and 2, respectively.

Pressure difference partition:

NL: Negative large
NM: Negative medium
NS: Negative small
Z: ZERO
PS: Positive small
PM: Positive medium
PL: Positive large

The terms NL, NM, NS, Z, PS, PM, and PL are assumed to be equivalent to the numerical value −0.061, −0.005, −0.002, 0, 0.002, 0.01, and 0.06, respectively.

We have discretized the leak size to five levels, i.e., none, very small, medium, fair, and large, which are suppose to be corresponding to the leak size 0, 0.5, 2, 4, 6, and 8, respectively.

Based on the above definitions, the similarity measures (9) and (10) will be used for an inconsistency measure of the belief rule base.

Because the flow difference is divided into eight terms and the pressure difference is divided into seven terms, there are a total of 56 combinations of the two antecedents leading to 56 rules in the rule-base. For example, a sample rule could be:

IF FlowDiff is Negative Medium AND PressureDiff is Negative Large THEN LeakSize is {(*none*, 0), (*very small*, 0), (*medium*, 0), (*fair*, 0.2), (*large*, 0.8)}.

4.3 Learning of the belief-rule-base for leak detection

In the rule-based leak detection system based on RIMER, the belief rule base plays an important role, where an initial belief distribution may be extracted from experts. In this case, the initial BRB can be built from experts' partial knowledge, which may not be optimal. It can be then refined by an autonomous learning process. The belief rule-base is initially roughly given by experts. The training followed will use training data to adjust and refine the rule-base parameters.

(1) Training based on the training data without checking the inconsistency

Six hundred samples, including inconsistent data, are used for training. Compared with the initial belief-rule-base, there exist some inconsistent rules, for example, the rule 2 is inconsistent with the rules 3 and 4; the rules 5 and 6, the rules 10 and 11 etc. They are illustrated in the following Table 3.

Table 3 Inconsistent rules illustration

RN	RW	FlowDiff and PreDiff	No 0	Very low 0.5	Medium 4	Fair 6	High 8	LS
1	1	*Negative large AND Negative large*	0.0005	0.0006	0.0001	0.0017	0.997	7.9869
2	1	*Negative large AND Negative medium*	0.1011	0.0907	0.0062	0.2087	0.5933	6.06875
3	1	*Negative large AND Negative small*	0.0649	0.0578	0.0042	0.3892	0.4838	6.2513
4	1	*Negative large AND ZERO*	0.0352	0.0307	0.0004	0.6814	0.2523	6.12375
5	1	*Negative large AND Positive small*	0.0086	0.029	0.0413	0.6856	0.2356	6.1781
6	1	*Negative large AND Positive medium*	0.0381	0.0761	0.0599	0.203	0.623	6.47965
7	1	*Negative large AND Positive large*	0.7888	0.1905	0.0013	0.0053	0.0141	0.24505
8	1	*Negative medium AND Negative large*	0.0006	0.0007	0	0.2006	0.7981	7.58875
9	1	*Negative medium AND Negative medium*	0.0524	0.0501	0.0079	0.6773	0.2122	5.81805
10	1	*Negative medium AND Negative small*	0.023	0.0207	0.0909	0.7977	0.0677	5.70175
11	1	*Negative medium AND ZERO*	0.0075	0.0064	0.1462	0.7008	0.1391	5.9056
12	1	*Negative medium AND Positive small*	0.0177	0.0724	0.3263	0.1481	0.4356	5.7148

Here RN means the rule number, RW means the rule weight, the 2nd row of the table (i.e., the 2nd rule) is read as follows:

IF FlowDiff is *Negative large* AND PressDiss is *Negative medium* THEN Leak size is {(*none*, 0.1011), (*very small*, 0.0907), (*medium*, 0.0062), (*fair*, 0.2087), (*large*, 0.5933)} with the rule weight 1

The corresponding leak size (LS, in the last column) corresponding to the belief distribution is calculated as: $0.1011 * 0 + 0.0907 * 0.5 + 0.0062 * 4 + 0.2087 * 6 + 0.5933 * 8 = 6.06875$

Table 4 Summary of effect of increasing leak on system variables

FlowDiff	PressDiff	Leak size
=(Negative) (More input than output)	↑(Negative) (i.e., more negative)	↑
=(Negative)	↑(Positive) More positive	↓
=Zero	↑(Negative) More negative	↑
=Zero	↑(Positive)	↓
↑(Negative)	=(Negative)	↑
↑(Positive)	=(Negative)	Uncertain
=(Positive)	↑(Negative) or ↓(Negative)	Normal operation (no leak)
=(Positive)	↑(Positive) or ↓ (Positive)	Normal operation (no leak)

Here, =Zero: constant zero; =(Negative): Negative constant; ↑(Negative): Negatively increasing; ↓(Positive): Positively decreasing; ↑: increasing; ↓: decreasing

As we can see, the inconsistency degrees among those rules are quite high. Intuitively, while FlowDiff is negative (i.e., $f_0 > f_1$), the negative increase of pressure difference will increase the leak size, but the positive increase of pressure will decrease the leak size. Table 4 shows the trend of leak size with the variation of two parameters:

Based on this trained rule-base, the system output for 600 training data is shown in Fig. 3a.

Comparing with the system output using initial rule-base, the system output after training is much better. However, there are some points far away from the assumed output, which is caused partially by the inconsistency among the trained rules, which are mainly caused by some noise training data. The consistency of the system is violated due to the noise data, which may be caused by turbulence and dynamic changes in the pipeline, or possible instrument and data communication errors.

After checking the training data, some training data are inconsistent according to common sense. For example, while the flow difference is the same, larger pressure drops indicate larger leaks. The data pairs (32, 78) and (74, 76) seem inconsistent with this rule (due to noises in data and dynamics in pipeline operations).

(2) Training based on the training data minimizing/excluding the inconsistent set

To make the trained rule consistent, the proposed optimization algorithm for learning consistent BRB is used so that the inconsistency caused due to those 21 inconsistent training data can be minimized. Based on the new algorithm, the training is implemented again. We have checked that there is minor inconsistency in the rule-base (the overall inconsistency degree of rule-base is less than 0.02). Moreover, the comparative analysis between the system output based on the initial rule-base and the trained rule base using the new proposed optimization algorithm is shown in Fig. 3b. Clearly, the latter system output is better than the former inconsistent one, especially on some discrete points in between the top and bottom data.

From Fig. 3a, b, we notice that the performance of trained rule-base based on the proposed optimization algorithm is better than that based on training data without considering the inconsistent rule. The a priori knowledge from experts about the problem to be solved is given in an initial rule-base form. Although this initial BRB may lead to symptoms or abnormal conditions being overlooked (as shown in Fig. 3b, green points), it does have the advantage of not relying on a large amount of experimental data. It actually can be refined by the proposed learning process not only based on the data available but also maintaining the minimal consistency of available data.

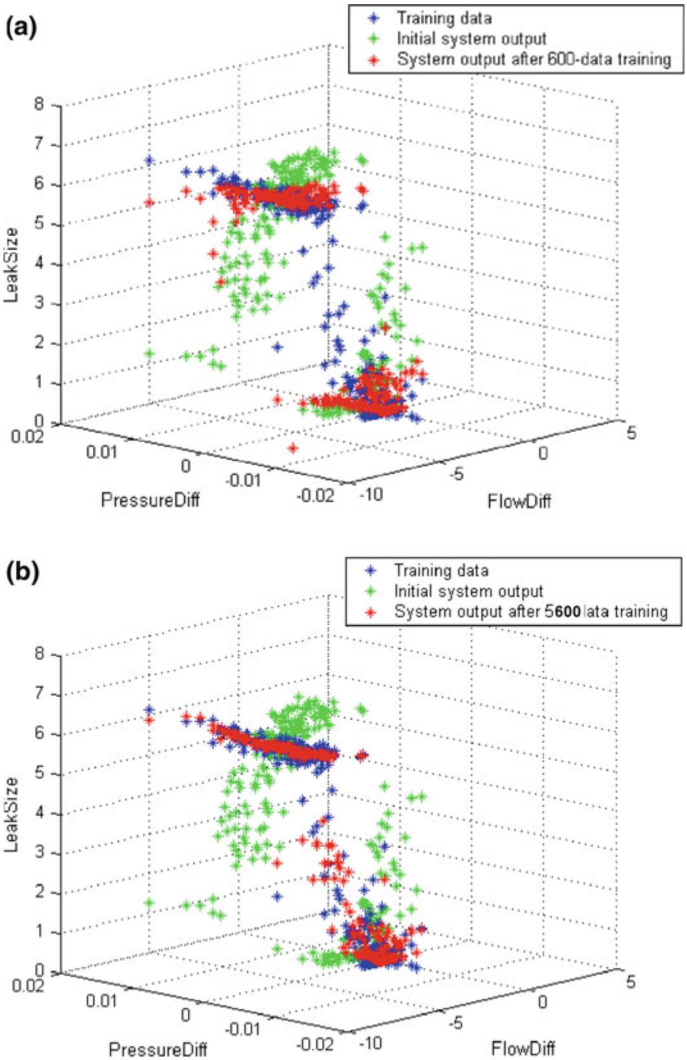

Fig. 3 **a** Comparison between system output based on initial rule-base and that based on the trained rule-base using 600 raw training data. **b** Comparison between the system output based on initial rule-base and that based on the trained rule-base for 600 raw training data (while minimize the inconsistency of rule-base)

The performance of the rule-base system has been greatly improved after the autonomous learning. Indeed, when the system was tested on the training data, all of the instances were classified approximately correctly (the mean square error is less than 0.03). Actually, this gives only an indication of the maximum success rate. When not enough data is available to create a test data set, a set of rules that infer selected training cases 100% correctly may be used with high confidence in its prediction; cases that are not covered by this set of rules require another set of rules of low accuracy.

Moreover, we may notice that the change of the rule base after the training and the performance of the trained rule-base are dependent on the quality and range of the training data. Not only we just mentioned the inconsistency of the training data, but also the range of training data is also important. For example, some rules seem no changes after the training compared with the initial rule base (i.e., the belief structures are kept the same). After checking the range of the training data, actually there is no data or very few data closed to the antecedent term of those rules, so the activation weights of those rule are equal to zero or approaching to zero. That means they play no or very little role in the final system output, so the training does not influence their belief structure. One reason that there is no or very few data in the range of training data is that actually these rules seem impossible state for the leak in the real pipeline operation state. They can be regarded as the system in a normal operation condition (no leak state). But if consider additional condition, like temperature, these rules may influence the leak status.

4.4 Test result comparisons

For test purposes, the data are selected in a similar way as the training data, the total numbers of 457 data from the whole set of data set are selected for the test purpose, which are different from the training data set.

Here we consider the trained rule-base based on 600 observations, as shown in Fig. 3b, the trained rule-base was tested on the training data, all of the instances were classified approximately correctly (the mean square error is less than 0.03). The trained rule-base is now tested on the 457 test data, which is shown in Fig. 4, and looks good.

Note a point (with 0 pressureDiff) at the bottom seems a very big error. Actually, this is a wrongly selected odd point, the test point here is FlowDiff $= -7.05$, PressDiff $= 0$. Here -7.05 is out of range of the flow difference we assumed. Hence the activation weight is zero, so the output becomes zero. This point is called *point out of range*. It can be certainly deleted or revised without influence on the whole performance of the trained rule-base.

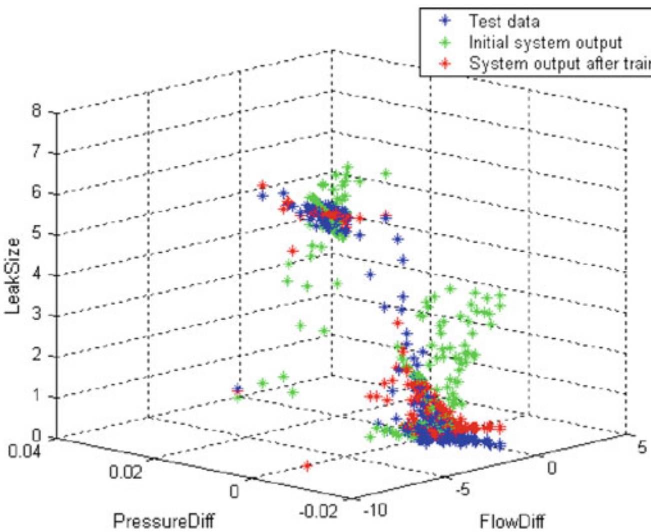

Fig. 4 Comparison between the system output based on initial rule-base and that based on the trained rule-base for 457 raw test data from table (25% controlled leak size)

The tests show that the trained rule-base seems reasonable, the success rate is good, together with the consistency, the training process and results seem more or less successful.

The above training and test experimental results demonstrate that the estimated outcomes match the observed ones very closely. The accuracy of the test output has been effectively improved thanks to minimize the inconsistency in an optimal way. The consistency of the system is violated due to the noise data, which may be caused by turbulence and dynamic changes in the pipeline, and possible instrument and data communication errors. Such noise is intrinsic to almost all pipeline operation data and poses significant challenges to developing pipeline leak detection systems. To minimize the inconsistency can be an effective way to minimize such noises.

However, the computational complexity is much higher and slower in the training process than the training without minimizing the inconsistency. So the proposed method is only suitable for off-line applications, not yet for real-time applications. Some new methods and algorithms can be investigated and applied, e.g., among others, approaches in [10], in order to increase the efficiency and effectiveness of the proposed methods.

In addition, the similarity measures defined in Sect. 3 are only for illustration purpose. There have been many types of similarity measurements being introduced. The utilization of them depends on the type of antecedent and consequent attributes, which should be chosen accordingly based on the application context.

5 Conclusions

In the presented work, an optimization method for generating consistent rule base with the belief structure was proposed. The main focus was given on the consistency of the system. The measurement of inconsistency of BRB was provided and finally incorporated in the objective function of the optimization algorithm which is formulated as a non-linear constraint optimization problem and solved using the optimization tool provided in MATLAB. The case study for application in oil pipeline leak detection is provided to illustrate the effectiveness of the proposed approach. The proposed optimization algorithm provided a practical and reliable support for the proposed RIMER approach.

Acknowledgments This work is partially supported by the research project TIN2009-08286 and P08-TIC-3548

References

1. Appice, A., D'Amato, C., Esposito, F., Malerba, D.: Classification of symbolic objects: a lazy learning approach. Intell. Data Anal. **10**, 301–324 (2006)
2. Coleman, T., Branch, M.A., Grace, A.: Optimization Toolbox—for Use with Matlab. The Mathworks Inc., Natick, MA (1999)
3. Dempster, A.P.: A generalization of Bayesian inference. J. Royal Stat. Soc. Ser. B **30**, 205–247 (1968)
4. Hefny, H.A.: Comments on "Distinguishability quantification of fuzzy sets". Inf. Sci. **177**, 4832–4839 (2007)
5. Jin, Y.C., von Seelen, W., Sendhoff, B.: On generating FC^3 fuzzy rule systems from data using evolution strategies. IEEE Trans. Syst. Man Cybern. Part B: Cybern. **29**(6), 829–845 (1999)
6. Liu, J., Yang, J.B., Wang, J., Sii, H.S., Wang, Y.M.: Fuzzy rule-based evidential reasoning approach for safety analysis. Int. J. Gen. Syst. **33**(2–3), 183–204 (2004)
7. Liu, J., Yang, J.B., Ruan, D., Martinez, L., Wang, J.: Self-tuning of fuzzy belief rule bases for engineering system safety analysis. Ann. Oper. Res. **163**(1), 143–168 (2008)

8. Pardalos, M., Hansen, P.: Data Mining and Mathematical Programming, CRM, vol. 45. American Mathematical Society, Providence, RI (2008)
9. Shafer, G.: A Mathematical Theory of Evidence. Princeton University Press, Princeton, NJ (1976)
10. Triantaphyllou, E.: Data Mining and Knowledge Discovery via Logic-Based Methods Theory, Algorithms, and Applications, Series: Springer Optimization and Its Applications, vol. 43 (2010)
11. Triantaphyllou, E., Mann, S.H., Pardalos, P.M.: A minimization approach to membership evaluation in fuzzy sets and error analysis. J. Optim. Theor. Appl. **66**, 275–287 (1990)
12. Triantaphyllou, E., Lootsma, F., Mann, S.H., Pardalos, P.M.: On the evaluation and application of different scales for quantifying pairwise comparisons in fuzzy sets. J. Multi-Criteria Decis. Anal. **3**, 133–155 (1994)
13. Xu, D.L., Yang, J.B., Liu, J., Liu, G.P., Wang, J., Jenkinson, I., Ren, J.: Inference and learning methodology of belief-rule-based expert system for pipeline leak detection. Expert Syst. Appl. **32**(1), 103–113 (2007)
14. Yang, J.B.: Rule and utility based evidential reasoning approach for multi-attribute decision analysis under uncertainties. Eur. J. Oper. Res. **131**, 31–61 (2001)
15. Yang, J.B., Liu, J., Wang, J., Sii, H.S.: A generic rule-base inference methodology using the evidential reasoning approach—*RIMER*. IEEE Trans. Syst. Man Cybern. **36**(2), 266–285 (2006)
16. Yang, J.B., Xu, D.L.: On the evidential reasoning algorithm for multiple attribute decision analysis under uncertainty. IEEE Trans. Syst. Man Cybern.-Part A **32**(3), 289–304 (2002)
17. Yang, J.B., Xu, D.L.: Nonlinear information aggregation via evidential reasoning in multi-attribute decision analysis under uncertainty. IEEE Trans. Syst. Man Cybern.—Part A: Syst. Hum. **32**(3), 376–393 (2002)
18. Yang, J.B., Liu, J., Xu, D.L., Wang, J., Wang, H.W.: Optimization models for training belief rule based systems. IEEE Trans. Syst. Man Cybern. Part A: Syst. Hum. **37**(4), 569–585 (2007)
19. Zadeh, L.A.: Fuzzy sets. Inf. Control **8**, 338–353 (1965)
20. Zimmerman, H.J.: Fuzzy Set Theory and Its Application. Kluwer, Norwell, MA (1991)