Knowledge-Based Systems 171 (2019) 93–105

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Online updating belief-rule-base using Bayesian estimation

Xilang Tang *, Mingqing Xiao, Yajun Liang, Haizhen Zhu, Jianfeng Li

ATS Lab, Air Force Engineering University, Xi'an, 710038, China

ARTICLE INFO

Article history:
Received 27 June 2018
Received in revised form 18 January 2019
Accepted 6 February 2019
Available online 19 February 2019

Keywords:
Belief rule base
Knowledge-based system
Evidential reasoning
Bayesian estimation

ABSTRACT

The common point of traditional BRB parameters training methods is taking the training process as an optimization problem, which may result in overfitting problem when training data is insufficient or contains strong noises. To solve the problems, we propose a novel method based on Bayesian estimation to update parameters of BRB online. While the optimization methods consider BRB parameters as unknown but determinate values, the Bayesian estimation regards BRB parameters as random variables. Instead of finding single optimal values of parameters, the proposed method is to estimate the posterior distribution of BRB parameters and produce prediction outputs by considering all possible parameters. Since the posterior distribution of BRB parameters cannot be calculated by analytical methods due to the nonlinearity of BRB models, the Sequential Monte Carlo (SMC) sampling technique is adopted to on-line approximate the posterior distribution of BRB parameters. A numerical function and a practical case on pipeline leak detection are studied to verify the performance of proposed algorithm.

1. Introduction

For dynamical nonlinear systems or complex decision problems, it is difficult to collect complete historical data set to construct a reliable model [1]. In this context, Yang [2] extends the conventional rule-based system, and proposes a new belief rule base(BRB) inference methodology using the evidential reasoning (RIMER) approach. BRB can be used to capture nonlinear causal relationships as well as continuity, incompleteness, and vagueness. It has been widely used in many areas, such as safety assessment [3], fault diagnosis [4], risk analysis [5,6], health estimation of engineering system [7], failure prognosis [8–10], system behavior prediction [11,12], network security prediction [13–15], medicine and medical assessment [16,17], identification of uncertain nonlinear systems [18].

Because the initial parameters of a BRB are determined subjectively by experts, it is impossible to ensure that all parameters are accurate for a large-scale rule base. Additionally, a change in parameters will influence performance of a BRB. To obtain a more accurate BRB model, there are a large number of works studying on the optimization of BRB system [19–27]. However, most of these works focus on improving the accuracy of BRB, and there is no literature devoted to the overfitting problem of BRB. Although the problem of overfitting for BRB system has been mentioned in some literatures [8,25,27], it is only regarded as a motivation of structural training. However, in practical application, overfitting is a problem that cannot be ignored. The overfitting problem arises from the structure of parameters learning task: the parameters are trained on a set of training dataset, but they are applied to make predictions on new data points [28,29]. Indeed, the over-seeking of the best fit to the training data will lead to a risk of getting a very poor prediction. The Fig. 1 is an illustration of overfitting problems [30,31]. As the degree of fitting increases, the training error of the model on the training data set will gradually decrease, but when the degree of fitting reaches a certain level, the error of the model on the verification set will increase. At this point, an overfitting occurs, that is, the model has a high degree of fit with the training data, but the model does not work on a data set other than the training set.

There are two main reasons for the overfitting problem [32]:

(1) From the perspective of the model. Due to the over-complex -ity of the model, the model fits the training data well, but the model is too sensitive to training data. Therefore, overly complex models are prone to overfitting problems due to noise or information not related to the target present in the training data.

(2) From the perspective of the training data. Overfitting caused by data noise: noise is somewhat random and deceptive. If noise is taken as effective information, it will lead to overfitting. Overfitting caused by the lack of representative samples: The training data set is small and cannot reflect the overall distribution well, but the over-refinement of the model will lead to over-fitting.

From the perspective of the model, downsizing the BRB models can overcome overfitting problem definitely to some extend, which has been mentioned in some literature, but overfitting caused by data noise or small training set is overlooked. However, in many applications, the reason why we choose BRB model instead of pure data-driven method is that we are incapable of collecting

* Corresponding author.
E-mail address: tangxilang@sina.com (X. Tang).

https://doi.org/10.1016/j.knosys.2019.02.007
0950-7051/

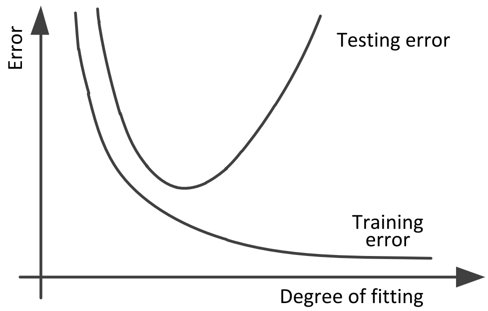

Fig. 1. Illustration of overfitting problems.

enough data. For example, for the fault diagnosis of aerospace equipment, they are often very expensive, and it is not feasible to do a lot of destructive tests to obtain enough fault data [4]. In such case, we will reasonably use the BRB model rather than the pure data-driven method. In fact, through our experiments (comparing neural network with BRB), we found that when the training data is enough, BRB model has no advantage on prediction accuracy over data-driven method, and the only advantage of BRB model is that it is an explainable system. Only when the training data set is relatively small, the BRB model can present its advantage compared with pure data-driven method, that is, make full use of the expert's prior knowledge to make up for the lack of training data. However, if the over-fitting problem occurs in BRB training due to a small training data set, that is, the training error is very small, the prediction error is very large, even larger than the initial BRB given by experts, the training will be meaningless. What is more, in order to achieve the best fitting, some BRB parameters may be changed dramatically by the training, and these parameters become inexplicable.

In this paper, a new online updating method is proposed to update BRB parameters online using Bayesian estimation to protect BRB models against overfitting [33], and to allow BRB models to update parameters adaptively [34]. Parameter training based on Bayesian estimation is totally different from current optimization methods. The common point of current methods is taking the parameters training of BRB as an optimization problem, such as the minimum mean squared error (MMSE), or maximum likelihood (ML). And the solution to the optimization problem is finding optimal model parameters, which are single and certain values [22, 35,36]. However, parameter training based on Bayesian estimation does not treat parameters as unknown determinate values, but random variables [35,34,37]. It is worth noting that "treating parameters as random variables" does not mean the initial parameters are generated randomly, conversely, the prior probability distribution of BRB parameters is decided by experts. Therefore, its objective is to compute the posterior distribution of parameters, and predict the outputs by considering all possible parameters [34, 37,38]. Consequently, Bayesian training for updating BRB can avoid overfitting problem [39].

The theory of Bayesian estimation is estimating posterior distribution of parameters with prior distribution of parameters and training dataset [40]. For a linear system, the posterior distribution can be calculated analytically [35]. But BRB is characterized as a strong nonlinear system, so analytical calculation is invalid for BRB updating. To solve this problem, the Mont Carlo sampling technique is adopted to approximately calculate the posterior distribution [41–43]. There are various Mont Carlo sampling techniques, including rejection sampling, importance sampling, Markov chain Monte Carlo (MCMC), Sequential Monte Carlo (SMC), etc. [44]. The SMC methods allow us to carry out on-line approximation of probability distributions using samples [45], so they are very appropriate when the systems have a real-time requirement [46], as discussed by zhou [21]. Therefore, the SMC method is chosen to estimate the posterior distribution of BRB parameters online in this paper.

This paper is organized as follows: Literature review is given in Section 2. BRB system is briefly introduced in Section 3; Section 4 introduces the Bayesian estimation method based on Monte Carlo sampling technique for updating the parameters of BRB, and presents two methods to produce the final predicting outputs of BRB model trained by Bayesian estimation. In Section 5, to validate the advantage of the proposed approach, two cases are studied, one of which is a simple numerical function study, and the other one is a practical case on pipeline leak detection. Section 6 comes to the conclusion.

2. Literature review

BRB is different from the traditional IF-THEN expert system in that its parameters and structures can be trained with historical data, so BRB can output more accurate results. Therefore, BRB actually combines the advantages of expert knowledge and data drive: it cannot only fully utilize the experience of experts, but also overcome the inaccuracy of expert knowledge. Accordingly, training is an important step to obtain a BRB system with excellent performance. This section reviews the literature on BRB training.

Yang [19] firstly proposes a generic learning framework to train BRB parameters, and abstracts the learning process as a nonlinear optimization problem, whose optimal target is to minimize the difference between estimated outputs of BRB system and observed outputs. Following Yang's works, there are a large number of works researching on the optimization of BRB system, and the optimization of BRB mainly includes two aspects: parameter optimization and structure optimization. Yang's approach is adopted by Xu [47] to train the BRB parameters for the practical case of oil pipeline leak detection, and the results proved that the approach improved the performance of BRB system dramatically. After that, Chen [20] analyzed the ER inference patterns on both belief degrees and activation weights, and draw a conclusion that referential values of antecedent attributes are crucial for identifying the critical points of a BRB system. So an adaptive training method which adds the referential values of antecedent attributes as one of parameter types into Yang's local optimization model is proposed, and the accuracy of the results is improved by the method. In addition, the common point of the above methods is that the optimization problem is solved by Matlab toolbox. And some works use other optimization techniques, such as evolutionary algorithms [48], to optimize the BRB parameters. These methods are offline, and Zhou [21,22] argues that the offline methods are very expensive when training data set is very large and they are not applicable to high real-time situations. To solve this problem, they proposed a new optimal training method based on recursive-expectation–maximization algorithm to update the parameters of BRB recursively. However, whether it is offline training or online training, these works regard BRB parameter training as the optimization problem. The objective is to improve the output accuracy of BRB, and not concern the problem of overfitting.

In addition to parameter optimization, a great deal of research effort has been devoted to reducing the complexity of BRB. Zhou [8] firstly put forward such idea and proposed a sequential learning algorithm to construct more compact BRB systems online. Following Zhou's work, a great deal of research effort has been devoted to reducing the complexity of BRB parameter learning. Because the size of BRB is decided by the dimensionalities of antecedent attributes and number of values for each attribute, these methods are designed to decrease them. Chang [23] used a lot of methods including Gray Target (GT), Multidimensional Scaling (MDS), Isomap and Principle Component Analysis (PCA), to reduce the number of the attributes. Wang [24] proposed a dynamic rule adjustment approach, where the density and error analysis are used to construct the structure of a BRB with appropriate number of rules. Yang [25] proposed the concept of generalization error as a new criterion for optimization, based on which, the number of referential values of each antecedent attribute are reduced efficiently by adopting a heuristic strategy. Sun [26] proposed a new approach to reduce complexity of BRB systems via Extended Causal Strength Logic. Chang [27] proposed an Akaike Information Criterion (AIC)-based objective, representing both the modeling accuracy and the modeling complexity, and developed a bi-level optimization model and a corresponding bi-level optimization algorithm. Overly complex BRB system is one of the reasons which result in the overfitting problem, so downsizing the BRB structure can avoid the overfitting problem to some extent. This has been mentioned in the above literatures, especially in Chang's work. Different from previous work, Chang's work considers not only the accuracy of BRB, but also the complexity of BRB in training objective by using AIC, so it balances the accuracy of BRB and the complexity of BRB during training [27].

These works is helpful to avoid overfitting problem, but it can be caused by other reasons, such as insufficient training data, which is overlooked by previous research. Previous works has not tried to train BRB with a small set of data, but this paper will focus on the overfitting problem caused by small set of training data. One of common ways to solve overfitting is "early stopping" [30]. The objective of "early stopping" is not to make the training error too small, because when the training error is small to a certain extent, the test error will increase depicted as Fig. 1 The method of "early stopping" is as follow: Existing data is divided into training set and verification set, and after each iteration, the respective errors are calculated, and the training is stopped before the error on verification starts to increase. Although "early stopping" can overcome the overfitting, it is necessary to check the working condition of the model on the verification set after each iteration, which greatly complicates the training process of the BRB. The common point of aforementioned methods is taking the parameters training of BRB as an optimization problem, and "early stopping" is also designed for the parameters training using optimization method. Therefore, a different learning method framework for BRB system based on Bayesian estimation from optimization method is proposed.

3. An introduction of BRB system

3.1. The basic BRB model

A belief rule is extended by IF-THEN rule with all possible results associated with belief degree. A BRB consists of a set of belief rules, which can be represented as follows [2],

$$\begin{aligned}
&R_k: \\
&\text{IF}x_1 \text{ is } A_1^k \wedge x_2 \text{ is } A_2^k \wedge \cdots \wedge x_{T_k} \text{ is } A_{T_k}^k \\
&\text{THEN} \quad \{(D_1, \beta_{1,k}), (D_2, \beta_{2,k}), \ldots, (D_N, \beta_{N,k})\}, \left(\sum_{i=1}^{N} \beta_{i,k} \le 1\right), \\
&\text{with rule weight}\theta_k\text{and attribute weights}\delta_{1k}, \delta_{2k}, \ldots, \delta_{T_k k}, \\
&k \in \{1, \ldots, L\}
\end{aligned} \tag{1}$$

where $A_i^k \quad (i = 1, 2, \ldots, T_k)$ refers to the reference value of x_i, which denotes to the ith antecedent attribute in the kth rule; $\beta_{j,k}(j = 1, 2, \ldots, N)$ is the belief degree assessed to D_j which denotes the jth consequent in the kth rule. If $\sum_{i=1}^{N} \beta_{i,k} = 1$, the kth rule is complete; Or else, it is incomplete. θ_k is the relative weight of kth rule. $\delta_{ik}(i = 1, 2, \ldots, T_k)$ is the relative weight of the ith antecedent attribute in the kth rule. L is the number of all belief rules that are used in the BRB. And T is the total number of antecedent attributes that are used in the rule base. In addition, "$\wedge$" is a logical connective to represent the "AND" operator. Therefore, a parameter model for the basic BRB can be formally represented as follows:

$$\boldsymbol{Q} = \langle \boldsymbol{\theta}, \boldsymbol{\delta}, \boldsymbol{\beta} \rangle \tag{2}$$

where $\boldsymbol{Q}$ represents the parameter vector, $\boldsymbol{\theta}$ is the vector of rule weights, $\boldsymbol{\delta}$ is the vector of attributes weights, and $\boldsymbol{\beta}$ is the vector of given belief degrees.

3.2. BRB Inference using ER approach

The BRB inference is based on ER analytical algorithms, and the final conclusion or the output is calculated by aggregating all rules that are activated [2]. The specific calculation process includes two steps: activation weight calculation and belief degree calculation. The activation weight refers to the degree of activation of the kth rule, represented by ω_k, which can be calculated by,

$$\omega_k = \frac{\theta_k \prod_{i=1}^{T_k} (\alpha_{i,j}^k)^{\bar{\delta}_i}}{\sum_{l=1}^{L} \left[\theta_l \prod_{i=1}^{T_l} (\alpha_{i,j}^l)^{\bar{\delta}_i}\right]} \text{ and } \bar{\delta}_i = \frac{\delta_i}{\max\limits_{i=1,2,\ldots T_k} \{\delta_i\}} \tag{3}$$

where $\alpha_{i,j}^k \ (i = 1, 2, \ldots, T_k)$, called individual matching degree, represents the degree of belief which the input for the ith antecedent attribute belongs to its jth referential value $A_{i,j}^k$ in the kth rule. Therefore, the input of BRB system is not a set of single values for each antecedent attribute, but the individual matching degree. The belief degree β_j can be calculated by,

$$\beta_j = \frac{\mu * \left[\prod_{k=1}^{L} \left(\omega_k \beta_{j,k} + 1 - \omega_k \sum_{i=1}^{N} \beta_{i,k}\right) - \prod_{k=1}^{L} \left(1 - \omega_k \sum_{i=1}^{N} \beta_{i,k}\right)\right]}{1 - \mu * \left[\prod_{k=1}^{L} (1 - \omega_k)\right]}, \quad j = 1, 2, \ldots, N \tag{4}$$

where,

$$\mu = \left[\sum_{j=1}^{N} \prod_{k=1}^{L} \left(\omega_k \beta_{j,k} + 1 - \omega_k \sum_{i=1}^{N} \beta_{i,k}\right) - (N-1) \prod_{k=1}^{L} \left(1 - \omega_k \sum_{i=1}^{N} \beta_{i,k}\right)\right]^{-1} \tag{5}$$

3.3. Transformation techniques for quantitative data

There is an important technique, proposed by Yang [49], which allows us transform the quantitative input data into the form of belief degree, described as follows.

Suppose that $\boldsymbol{x}= \{x_1, \ldots, x_n\}$ is a set of quantitative input, and $A_i = \left\{A_{ij}; j = 1, \ldots, J_i = \left|A_{ij}\right|\right\} (i = 1, \ldots, n)$ is a set of referential values for antecedent attribute. Without loss of generality, $A_{i(j+1)}$ is assumed to be larger than A_{ij}. Then an input variable x_i can be represented as,

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

where

$$\alpha_{ij} = \frac{A_{i(j+1)} - x_i}{A_{i(j+1)} - A_{ij}} \tag{7}$$

$\alpha_{i(j+1)} = 1 - \alpha_{ij}$ if $A_{ij} < x_i < A_{i(j+1)}$, and $\alpha_{ik} = 0$ for $k = 1, \ldots, J_i, k \neq j, j+1$ And if the output is numerical, the output of belief degree can be transformed as [2],

$$y = \sum_{j=1}^{N} u(D_j)\beta_j \tag{8}$$

here $u(D_j)$ is the utility of an individual consequent D_j. Under such circumstance, the parameter model for BRB can be extended by,

$$\boldsymbol{Q} = \langle \boldsymbol{\theta}, \delta, \boldsymbol{\beta}, \boldsymbol{A}, \boldsymbol{U} \rangle \tag{9}$$

where $\boldsymbol{A}$ is the vector of referential values for antecedent attributes, and $\boldsymbol{U}$ is the vector of utility of consequents.

4. BRB parameter updating with Bayesian estimation

4.1. Basic assumptions

In this paper, the parameter estimation for BRB systems is an on-line context. Instead of estimating single values of parameters, we try to estimate the posterior distribution of parameters, symbolized as $p(\boldsymbol{Q}_t|\boldsymbol{S}_{1:t})$, where $S_{1:t} = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_t, y_t)\}$ are observed data pairs. The Sequential Monte Carlo (SMC) sampling method is proposed to approximate the posterior distribution, and its basic idea is treating BRB parameters as random-walk variables, and using a transition probability $p(\boldsymbol{Q}_t|\boldsymbol{Q}^i_{t-1})$ to propagate samples at $t-1$ to t, and using a observation probability to update the weight of each sample. For applying this technique, two reasonable basic assumptions are needed:

(1) the adjustment of BRB parameters over time is a Markov process, which means that the distribution of BRB parameters at time t only depends on the parameters at time $t-1$. The transition of that is a random walk model, which can be described as follows,

$$\boldsymbol{Q}_{t+1} = \boldsymbol{Q}_t + \boldsymbol{\eta}_t, \quad \boldsymbol{\eta}_t \sim N(0, \boldsymbol{\nu}^2) \tag{10}$$

In this paper, we assume the random walk of parameters follows Gaussian distribution, whose mean value is 0, and variance is $\boldsymbol{\nu}^2$. Because Gaussian distribution is symmetrical, so the samples at time t will be symmetric about the sample at time $t-1$, so the probability of adjusting the parameters in any direction is equal. However, if we have prior knowledge about which direction the parameters should be adjusted, we can also assume the transition probability follows other distribution.

(2) Given an input vector, the current output is only dependent on current parameter vector, which can be represented as,

$$p(\boldsymbol{S}_t|\boldsymbol{Q}_1 \ldots, \boldsymbol{Q}_{t-1}, \boldsymbol{Q}_t) = p(\boldsymbol{S}_t|\boldsymbol{Q}_t) \tag{11}$$

This is called observation assumption, and $p(\boldsymbol{S}_t|\boldsymbol{Q}_t)$ is called observation probability. When we calculate an output of BRB at time t, we only consider the estimated parameters at time t rather than all estimated parameters before t.

4.2. Sequential Monte Carlo sampling

As discussed before, because BRB is a nonlinear system, the analytical method to calculate posterior distribution of parameters is invalid. Consequently, SMC technique is used to approximate the posterior distribution. Because the posterior distribution of parameters is unknown, it is impossible to sample from the posterior distribution directly. So, a simple distribution $q(\boldsymbol{Q}_t|\boldsymbol{S}_{1:t})$, called importance distribution, is commonly introduced and we sample from this distribution [50]. When we draw a sample from it, we need to calculate a weight corresponding this sample, and the weight reflects the importance of this sample for approximating the posterior distribution. By drawing samples $\boldsymbol{Q}^i_t$ from $q(\boldsymbol{Q}_t|\boldsymbol{S}_{1:t})$, a weighted approximation to the true posterior distribution
$p(\boldsymbol{Q}_t|\boldsymbol{S}_{1:t})$ is given by,

$$p(\boldsymbol{Q}_t|\boldsymbol{S}_{1:t}) \approx \sum_{i=1}^{N_s} w^i_t \cdot \delta(\boldsymbol{Q}_t - \boldsymbol{Q}^i_t) \tag{12}$$

where,

$$w^i_t \propto \frac{p(\boldsymbol{Q}_t|\boldsymbol{S}_{1:t})}{q(\boldsymbol{Q}_t|\boldsymbol{S}_{1:t})} \tag{13}$$

where w^i_t is called importance weight. The weights are normalized such that $\sum_{i=1}^{N_s} w^i_t = 1$. It is supposed that the importance distribution can be factorized as,

$$q(\boldsymbol{Q}_t|\boldsymbol{S}_{1:t}) = q(\boldsymbol{Q}_t|\boldsymbol{Q}_{t-1}, \boldsymbol{S}_{1:t})q(\boldsymbol{Q}_{t-1}|\boldsymbol{S}_{1:t-1}) \tag{14}$$

The posterior distribution $p(\boldsymbol{Q}_t|\boldsymbol{S}_{1:t})$ can be factorized as,

$$\begin{aligned} p(\boldsymbol{Q}_t|\boldsymbol{S}_{1:t}) &= \frac{p(\boldsymbol{S}_t|\boldsymbol{Q}_t, \boldsymbol{S}_{1:t-1})p(\boldsymbol{Q}_t|\boldsymbol{S}_{1:t-1})}{p(\boldsymbol{S}_t|\boldsymbol{S}_{1:t-1})} \\ &= \frac{p(\boldsymbol{S}_t|\boldsymbol{Q}_t, \boldsymbol{S}_{1:t-1})p(\boldsymbol{Q}_t|\boldsymbol{Q}_{t-1}, \boldsymbol{S}_{1:t-1})}{p(S_t|S_{1:t-1})} p(\boldsymbol{Q}_{t-1}|\boldsymbol{S}_{1:t-1}) \\ &= \frac{p(S_t|\boldsymbol{Q}_t)P(\boldsymbol{Q}_t|\boldsymbol{Q}_{t-1})}{p(\boldsymbol{S}_t|\boldsymbol{S}_{1:t-1})} p(\boldsymbol{Q}_{t-1}|\boldsymbol{S}_{1:t-1}) \\ &\propto p(\boldsymbol{S}_t|\boldsymbol{Q}_t)P(\boldsymbol{Q}_t|\boldsymbol{Q}_{t-1})p(\boldsymbol{Q}_{t-1}|\boldsymbol{S}_{1:t-1}) \end{aligned} \tag{15}$$

In Eq. (15), the Bayesian theory and the basic assumptions described in Section 3 are used. By substituting (14) and (15) into (13),

$$\begin{aligned} w^i_t &\propto \frac{p(\boldsymbol{S}_t|\boldsymbol{Q}^i_t)p(\boldsymbol{Q}^i_t|\boldsymbol{Q}^i_{t-1})p(\boldsymbol{Q}^i_{t-1}|\boldsymbol{S}_{1:t-1})}{q(\boldsymbol{Q}^i_t|\boldsymbol{Q}^i_{t-1}, \boldsymbol{S}_{1:t})q(\boldsymbol{Q}^i_{t-1}|\boldsymbol{S}_{1:t-1})} \\ &= w^i_{t-1} \frac{p(\boldsymbol{S}_t|\boldsymbol{Q}^i_t)p(\boldsymbol{Q}^i_t|\boldsymbol{Q}^i_{t-1})}{q(\boldsymbol{Q}^i_t|\boldsymbol{Q}^i_{t-1}, \boldsymbol{S}_{1:t})} \end{aligned} \tag{16}$$

To simplify the sampling process, the prior distribution $p(\boldsymbol{Q}^i_t|\boldsymbol{Q}^i_{t-1})$ is often chosen to be the importance distribution,

$$q(\boldsymbol{Q}_t|\boldsymbol{Q}_{t-1}, \boldsymbol{S}_{1:t}) = p(\boldsymbol{Q}_t|\boldsymbol{Q}_{t-1}) \tag{17}$$

So, the importance weight updating process can be simplified as,

$$w^i_t \propto w^i_{t-1} p(\boldsymbol{S}_t|\boldsymbol{Q}^i_t) \tag{18}$$

4.3. Degeneracy problem and resampling

A common problem of proposed method is the degeneracy phenomenon, which means that after a few iterations, the weight of some samples degrades to almost zero. This implies that the contribution of these samples to the approximation is negligible, and a resampling technique can eliminate these bad samples and allow good (important) samples to amplify themselves [46]. A resampling scheme duplicates each selected important sample to generate N^i children, such that $\sum N^i = N$, where N is the total number of samples. And the scheme should satisfy,

$$\mathbb{E}(N^i) = N * \widetilde{\omega}_t^{(i)} \tag{19}$$

where $\widetilde{\omega}_t^{(i)}$ is the weight of selected sample. To achieve the objective, a random variable J is introduced, which takes values in the set $\{1, 2, \ldots, N\}$. The variable J is called auxiliary variable, and a simple approach draws J = j with probability proportional to w^j, which can satisfy Eq. (19). After the resampling step, we get N samples distributed approximately corresponding to the distribution $p(\boldsymbol{Q}_t|\boldsymbol{S}_{1:t})$. However, a simple selection and duplication will lead to a distortion of the importance distribution. That is, many samples have no children, but some samples have a large number of children. To avoid this problem, after the resampling step, a MCMC step is introduced [51]. The basic idea is that, by applying a Markov transition kernel $K(\boldsymbol{Q}_t|\widetilde{\boldsymbol{Q}}_t)$,with invariant distribution $p(\widetilde{\boldsymbol{Q}}_t|\boldsymbol{S}_{1:t})$, such that, $\int K(\boldsymbol{Q}_t|\widetilde{\boldsymbol{Q}}_t)p(\widetilde{\boldsymbol{Q}}_t|\boldsymbol{S}_{1:t}) = p(\boldsymbol{Q}_t|\boldsymbol{S}_{1:t})$,the total variation of the current distribution with respect to the invariant distribution can only decrease. The detailed algorithm will be introduced in Section 4.5.

4.4. Constraints and rejecting sampling

For a BRB model, the parameters vector $\boldsymbol{Q}$ must satisfy some constraints [19], as follows,

$$
\begin{aligned}
&0 \le \beta_{j,k} \le 1, \quad j = 1,\ldots,N; \quad k = 1,\ldots,L \\
&\sum_{j=1}^{N} \beta_{j,k} = 1 \\
&0 \le \theta_k \le 1, \quad k = 1,\ldots,L \\
&0 \le \delta_i \le 1, \quad i = 1,\ldots,T \\
&A_{i,j} < A_{i,j+1}, \quad i = 1,\ldots,T; \quad j = 1,\ldots,J_i \\
&u(D_i) < u(D_j) \quad \text{if } i < j, \quad i,j = 1,\ldots,N
\end{aligned}
\tag{20}
$$

But the transition of parameters vector is assumed to follow Gaussian distribution, so drawing samples from the transition distribution may not satisfy the constraints listed in Eq. (20). A simple method to handle this problem is to reject the sample which deviates the constraints, and to resample until the constraint is satisfied.

4.5. Updating algorithm

By summarizing the proposed training method, the updating algorithm can be described as Algorithm 1.

4.6. Predicting the output of BRB

By updating parameter samples recursively, the posterior of parameters at current time can be obtained, based on which, there are two methods to produce the final output prediction. One method is to use an average model, $\widehat{\boldsymbol{Q}} = E_{p(\boldsymbol{Q}|\boldsymbol{S})}[\boldsymbol{Q}]$, so that the output prediction is $\widehat{f}(\boldsymbol{x}^*, \boldsymbol{Q}) = f(\boldsymbol{x}^*, \widehat{\boldsymbol{Q}})$. Another method is to calculate average output directly, represented as $\widehat{f}(\boldsymbol{x}^*, \boldsymbol{Q}) = E_{p(\boldsymbol{Q}|\boldsymbol{S})}[f(\boldsymbol{x}^*, \boldsymbol{Q})]$. Accordingly, if we use an average model, the average parameters can be approximated by

$$
\begin{aligned}
E_{p(\boldsymbol{Q}|\boldsymbol{S})}[\boldsymbol{Q}] &= \int \boldsymbol{Q}_t \cdot P(\boldsymbol{Q}_t|\boldsymbol{S}_{1:t}) \\
&\approx \sum_{i=1}^{N_s} \boldsymbol{Q}_t \cdot w_t^i \cdot \delta(\boldsymbol{Q}_t - \boldsymbol{Q}_t^i) \\
&= \sum_{i=1}^{N_s} w_t^i \boldsymbol{Q}_t^i \equiv \widehat{\boldsymbol{Q}}
\end{aligned}
\tag{21}
$$

Based on the average parameter model, the output can be calculated as Eq. (22).

$$
\widehat{f}(\boldsymbol{x}^*, \boldsymbol{Q}_t) \equiv f(\boldsymbol{x}^*, \widehat{\boldsymbol{Q}}) \tag{22}
$$

where $\boldsymbol{x}^*$ is the given vector of input. By the same approximation, we can also estimate the average output directly.

$$
E_{p(\boldsymbol{Q}|S)}[f(\boldsymbol{x}^*, \boldsymbol{Q}_n)] \approx \sum_{i=1}^{N_s} w_t^i \cdot f(\boldsymbol{x}^*, \boldsymbol{Q}_t^i) \equiv \widehat{f}(\boldsymbol{x}^*, \boldsymbol{Q}_t) \tag{23}
$$

Algorithm 1 Updating algorithm

$[\{\boldsymbol{Q}_t^i, w_t^i\}_{i=1}^{N_s}] = \text{update}[\{\boldsymbol{Q}_{t-1}^i, w_{t-1}^i\}_{i=1}^{N_s}, \boldsymbol{S}_k]$
Importance sampling step:
for $i = 1$ to N_s **do**
  while 1 **do**
    Draw $\boldsymbol{Q}_t^i \sim p(\boldsymbol{Q}_t|\boldsymbol{Q}_{t-1}^i)$
    if $\boldsymbol{Q}_t^i$ satisfy constraints listed in Eq. (20) **then**
      break
    end if
  end while
  Calculate weights using Eq. (18)
end for
Calculate total weight: $T = \text{SUM}\left[\{w_t^i\}_{i=1}^{N_s}\right]$
for $i = 1$ to N_s **do**
  Normalize: $w_t^i = T^{-1} w_k^i$
end for
Resampling and MCMC step:
for $i = 1$ to N_s **do**
  Draw J = j with probability proportional to $w^{(j)}$
  Given J = j, draw $\boldsymbol{Q}_t^{*(j)} \sim p(\boldsymbol{Q}_t|\boldsymbol{Q}_{t-1}^{(j)})$
  if $\upsilon \sim U_{[0,1]} \le \min\{1, \frac{p(S_t|\boldsymbol{Q}_t^{*(j)})}{p(S_t|\widetilde{\boldsymbol{Q}}_t^{(j)})}\}$ **then**
    accept move: $\boldsymbol{Q}_t^{(i)} = \boldsymbol{Q}_t^{*(j)}$
  else
    reject move: $\boldsymbol{Q}_t^{(i)} = \widetilde{\boldsymbol{Q}}_t^{(j)}$
  end if
end for

From, Eq. (23), all possible outputs of BRB instead of a single value are produced, as a result, this method has the capability to capture uncertainties in nature. In other words, the outputs of BRB trained by Bayesian estimation are a large number of weighted samples, from which we can estimate their distribution. The method of histogram estimation can be used to approximately calculate the probability density of BRB outputs trained by Bayesian estimation. The entire range of output values is divided into a certain enough number of intervals, and then calculate the sum of weights whose corresponding output value falls into each interval, which can be represented as,

$$
pdf(a_m) \approx \sum w^i \text{ , iff } \quad a_m \quad < y^i < a_{m+1} \tag{24}
$$

where $\{a_1, a_2, \ldots, a_K\}$ are intervals of range of output$y$, y^i is the output of ith sample, and w^i denotes to the weight of ith sample.

5. Case study

In this section, to verify the performance of Bayesian estimation approach for updating BRB parameters, we take two cases, one of which is a simple numerical function study, and the other one is a practical case on pipeline leak detection. The procedure of case study is shown in Fig. 2, including:

- Step1: Initializing BRB parameters. The initial BRB parameters generally come from experts' subjective judgments, so it cannot be accurate, and needs to be trained.
- Step2: Setting transition distribution and observation distribution. To implement the parameter updating process with Bayesian estimation, the prior distribution of parameters transition $p(\boldsymbol{Q}_t|\boldsymbol{Q}_{t-1}^i)$ and the observation distribution $P(\boldsymbol{S}_t|\boldsymbol{Q}_t^i)$ need to be given firstly, also determined subjectively by expert experience in general. Besides, initial samples

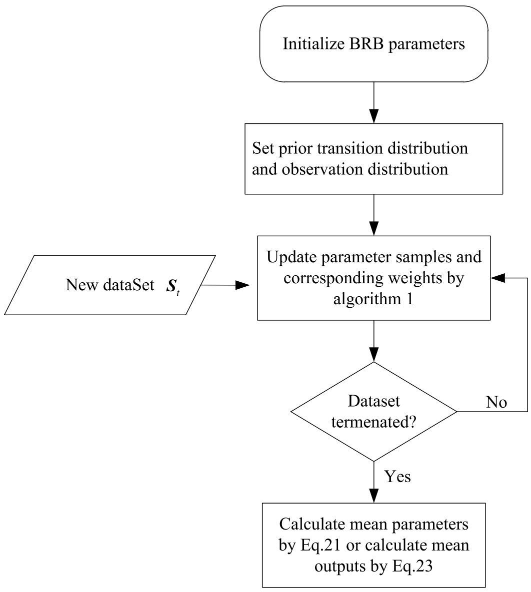

Fig. 2. The procedure of case study.

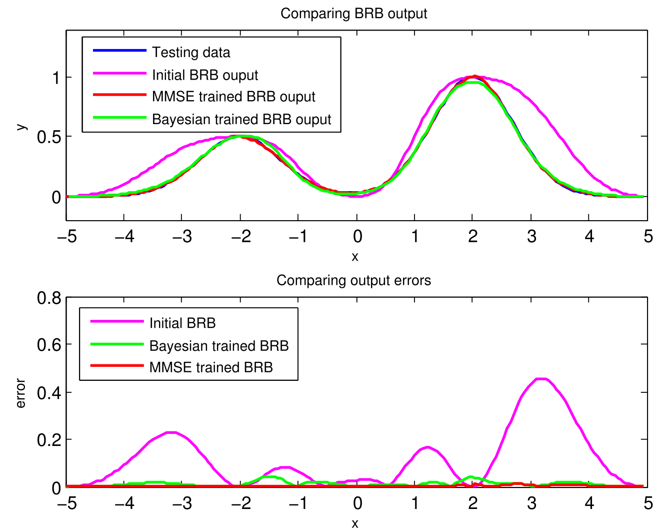

Fig. 3. Comparison of outputs of trained BRB (100 training points).. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

should be generated and their initial weights need to be assigned.

- Step3: Updating parameter samples and corresponding weights by Algorithm 1. After the preparation, the BRB parameter samples and their corresponding weights can be updated recursively by Algorithm 1 until training data set is terminated.
- Step4: Calculating mean parameters by Eq. (21) or calculating mean outputs by Eq. (23). If there is no special explanation, the subsequent results are calculated using Eqs. (21) and (22).

5.1. A numerical function study

In this case, a simple nonlinear function designed by Chen [52] is used to generate training dataset and testing dataset, which is represented as follows,

$$y = e^{-(x-2)^2} + 0.5e^{-(x+2)^2}, \quad x \in (-5, 5) \tag{25}$$

The initial BRB parameters for this experiment is set as Table 1. Since we assume the adjustment of BRB parameters over time as a Markov process, which is an online context, the training data set should be time series. Therefore, each data pair in the training data set needs to be labeled by time tag, such that, (x_t, y_t) represents the data pair at time t. The method generating the training data is as follows: at time t, an input x_t is generated randomly from a uniform distribution of its value range, formed by, $x_t \sim U([-5, 5])$. Then calculate the output y_t with Eq. (23), thus a new data pair (x_t, y_t) is obtained, and the BRB parameters is updated from $\boldsymbol{Q}_{t-1}$ to $\boldsymbol{Q}_t$ only using the data pair (x_t, y_t) rather than the complete historical data before time t.

5.1.1. Training BRB when training data is sufficient

To verify our method, we generate 100 data points as training data which can be considered sufficient, and the generation method is mentioned before. The prior distribution of parameters transition $p(\boldsymbol{Q}_t|\boldsymbol{Q}_{t-1}^i)$ and the observation distribution $P(\boldsymbol{S}_t|\boldsymbol{Q}_t^i)$ are set as follows,

$\boldsymbol{\theta}_t^i \sim N(\boldsymbol{\theta}_{t-1}^i, 0.03)$, $\boldsymbol{\beta}_t^i \sim N(\boldsymbol{\beta}_{t-1}^i, 0.006)$, $\boldsymbol{A}_t^i \sim N(\boldsymbol{A}_{t-1}^i, 0.06)$, $\boldsymbol{U}_t^i \sim N(\boldsymbol{U}_{t-1}^i, 0.006)$, $\boldsymbol{\delta}_t^i \sim N(\boldsymbol{\delta}_{t-1}^i, 0.006)$, $P(\boldsymbol{S}_t|\boldsymbol{Q}_t^i) \sim N(\boldsymbol{S}_t^*, 0.03)$.

where $\boldsymbol{S}_t^*$ is the training data set at time t. And the number of samples $N_s = 1000$, the initial weights $w_0^i = 1/N_s$.

After being trained by our method and the MMSE method, the estimated parameters are presented in Tables 2 and 3 respectively ,and the results of estimated output and the estimation errors are shown in Fig. 3. From the Figure, we can see that the output of initial BRB deviates the testing data a lot, and its MSE is 0.0277. But after trained by the MMSE method and the Bayesian estimation, the outputs of BRB is much more similar to the true output, and the errors drop to a very low level. The MSE of MMSE trained BRB is $4.90 \times 10^{(-5)}$, and the MSE of Bayesian trained BRB is $2.56 \times 10^{(-4)}$, which shows the Bayesian method has a lower accuracy than optimization method. But it does not reach to an unacceptable degree, because it can be seen from Fig. 3 that the Bayesian trained BRB output (the green curve) is very close to the testing data compared with initial BRB.

Furthermore, it is worth noting that the trained parameters by Bayesian method are only slightly adjusted instead of a thorough change compared with the initial parameters. However, the parameters trained by optimization method change drastically. For example, the second referential utility 0 becomes equal to the first referential utility −0.5, and the 4th referential utility becomes equal to the 5th referential utility 1.5. So, a conclusion can be drawn: the optimization method changes the parameters of BRB in a more radical way in order to achieve the optimization goal, but the Bayesian method adjusts the parameters of BRB in a more conservative way. A radical way changing the BRB parameters may make them inexplicable, however, compared with data-driven method, knowledge-based method is a cognitive system instead of a black box, so the value of the parameters should be interpretable. The Bayesian estimation allows experts play a major role in deciding the BRB parameters, which is more reasonable for a knowledge-based system than deciding by training process completely.

5.1.2. Training BRB when training data is insufficient and noisy

To illustrate the advantage of our method, we generate another 12 training data with Eq. (26), where $\varepsilon \sim N(0, 0.1)$, which is used to simulate noises due to measurement errors. Therefore, the training data is insufficient and noisy.

$$y = e^{-(x-2)^2} + 0.5e^{-(x+2)^2} + \varepsilon, \quad x \in (-5, 5) \tag{26}$$

Table 1
Initial belief rule base for experiment 1.

R_k	θ_k	$x(A_j)$	Consequents $U(D) = \{-0.5, 0.0, 0.5, 1.0, 1.5\}$, $\delta_1 = 1$
1	1.0	−5	$\{(D_1, 0.0), (D_2, 1.0), (D_3, 0.0), (D_4, 0.0), (D_5, 0.0)\}$
2	1.0	−2	$\{(D_1, 0.0), (D_2, 0.0), (D_3, 1.0), (D_4, 0.0), (D_5, 0.0)\}$
3	1.0	0	$\{(D_1, 0.0), (D_2, 1.0), (D_3, 0.0), (D_4, 0.0), (D_5, 0.0)\}$
4	1.0	2	$\{(D_1, 0.0), (D_2, 0.0), (D_3, 0.0), (D_4, 1.0), (D_5, 0.0)\}$
5	1.0	5	$\{(D_1, 0.0), (D_2, 1.0), (D_3, 0.0), (D_4, 0.0), (D_5, 0.0)\}$

Table 2
Mean parameters trained by Bayesian estimation (100 training points)

R_k	θ_k	$x(A_j)$	Consequents $U(D) = \{-0.5, 0.016, 0.512, 0.966, 1.5\}$, $\delta_1 = 0.8162$
1	0.8560	−5	$\{(D_1, 0.0129), (D_2, 0.9683), (D_3, 0.0045), (D_4, 0.0077), (D_5, 0.0066)\}$
2	0.3681	−1.922	$\{(D_1, 0.0061), (D_2, 0.0163), (D_3, 0.9648), (D_4, 0.0063), (D_5, 0.0065)\}$
3	0.4848	−0.151	$\{(D_1, 0.0046), (D_2, 0.9604), (D_3, 0.0061), (D_4, 0.0151), (D_5, 0.0137)\}$
4	0.3277	2.020	$\{(D_1, 0.0043), (D_2, 0.0055), (D_3, 0.0049), (D_4, 0.9714), (D_5, 0.0139)\}$
5	0.8550	5	$\{(D_1, 0.0121), (D_2, 0.9665), (D_3, 0.0116), (D_4, 0.0052), (D_5, 0.0047)\}$

Table 3
Parameters trained by MMSE (100 training points)

R_k	θ_k	$x(A_j)$	Consequents $U(D) = \{-0.5, -0.5, 0.717, 1.5, 1.5\}$, $\delta_1 = 1$
1	0.7599	−5	$\{(D_1, 0), (D_2, 0.7496), (D_3, 0), (D_4, 0.1064), (D_5, 0.1440)\}$
2	0.3223	−2.068	$\{(D_1, 0), (D_2, 0.1904), (D_3, 0.7872), (D_4, 0.0224), (D_5, 0)\}$
3	0.4186	−0.097	$\{(D_1, 0.1684), (D_2, 0.4817), (D_3, 0.2234), (D_4, 0.1265), (D_5, 0)\}$
4	0.2916	2.054	$\{(D_1, 0), (D_2, 0.2446), (D_3, 0), (D_4, 0.7554), (D_5, 0)\}$
5	0.7224	5	$\{(D_1, 0.0824), (D_2, 0.6299), (D_3, 0.0987), (D_4, 0), (D_5, 0.1889)\}$

Table 4
Mean parameters trained by Bayesian estimation (12 training points with noises).

R_k	θ_k	$x(A_j)$	Consequents $U(D) = \{-0.5, 0.018, 0.494, 1, 1.5\}$, $\delta_1 = 0.8953$
1	0.8530	−5	$\{(D_1, 0.0033), (D_2, 0.9869), (D_3, 0.0035), (D_4, 0.0027), (D_5, 0.0036)\}$
2	0.4463	−1.8467	$\{(D_1, 0.0034), (D_2, 0.0054), (D_3, 0.9850), (D_4, 0.0032), (D_5, 0.0030)\}$
3	0.8552	−0.0305	$\{(D_1, 0.0030), (D_2, 0.9846), (D_3, 0.0038), (D_4, 0.0032), (D_5, 0.0052)\}$
4	0.5218	1.8141	$\{(D_1, 0.0050), (D_2, 0.0036), (D_3, 0.0040), (D_4, 0.9842), (D_5, 0.0032)\}$
5	0.9044	5	$\{(D_1, 0.0035), (D_2, 0.9867), (D_3, 0.0027), (D_4, 0.0039), (D_5, 0.0031)\}$

Table 5
Parameters trained by MSE (12 training points with noises).

R_k	θ_k	$x(A_j)$	Consequents $U(D) = \{-0.5, -0.147, 0.659, 1.386, 1.5\}$, $\delta_1 = 1$
1	1	−5	$\{(D_1, 0.0279), (D_2, 0.9671), (D_3, 0.005), (D_4, 0), (D_5, 0)\}$
2	0.0253	−2.4041	$\{(D_1, 0), (D_2, 0.0725), (D_3, 0.1363), (D_4, 0.2290), (D_5, 0.5623)\}$
3	0.0693	−0.3066	$\{(D_1, 0.1995), (D_2, 0.6388), (D_3, 0.0088), (D_4, 0.0022), (D_5, 0.1506)\}$
4	0.0445	2.3725	$\{(D_1, 0.0361), (D_2, 0), (D_3, 0.1336), (D_4, 0.8303), (D_5, 0)\}$
5	0.5253	5	$\{(D_1, 0.1988), (D_2, 0.6368), (D_3, 0), (D_4, 0.1629), (D_5, 0)\}$

The initial BRB parameters are set exactly as before. The prior distribution of parameters transition $p(\boldsymbol{Q}_t|\boldsymbol{Q}_{t-1}^i)$ and the observation distribution $P(\boldsymbol{S}_t|\boldsymbol{Q}_t^i)$ are set as follows,

$\boldsymbol{\theta}_t^i \sim N(\boldsymbol{\theta}_{t-1}^i, 0.06)$, $\boldsymbol{\beta}_t^i \sim N(\boldsymbol{\beta}_{t-1}^i, 0.03)$, $\boldsymbol{A}_t^i \sim N(\boldsymbol{A}_{t-1}^i, 0.06)$, $\boldsymbol{U}_t^i \sim N(\boldsymbol{U}_{t-1}^i, 0.006)$, $\boldsymbol{\delta}_t^i \sim N(\boldsymbol{\delta}_{t-1}^i, 0.006)$, $P(\boldsymbol{S}_t|\boldsymbol{Q}_t^i) \sim N(\boldsymbol{S}_t^*, 0.03)$. where $\boldsymbol{S}_t^*$ is the training data set at time t. And the number of samples $N_s = 1000$, the initial weights $w_0^i = 1/N_s$.

After being trained by our method and the MMSE method, the estimated parameters are presented in Tables 4 and 5 respectively ,and the results of estimated output and the estimation errors are presented in Fig. 4. The figure shows that the estimated output curve of BRB trained by optimization method is distorted, and some estimates are very close to the true points, but some estimates are severely deviating from the real point. However, the estimated output curve of BRB trained by Bayesian method has no such problem, even though the training accuracy is not perfect because of lack of training data. The MSE of Bayesian trained BRB is 0.0023, and the MSE of MMSE trained BRB is 0.0422, worse than the MMSE of initial trained BRB, 0.0277. Therefore, When the training data is insufficient, even includes noises due to measurement error, training BRB by the optimization method may have overfitting problem, which may cause an incorrect output. Even though the accuracy of BRB trained by Bayesian estimation is lower than that of BRB trained by traditional optimization method when training data is sufficient, the Bayesian method is a more appropriate choice when training data is insufficient. The point of our work is that we sacrifice, within acceptable extent, the accuracy to make sure that the method can avoid overfitting problem when the data is insufficient. When there is no risk of overfitting because training data is enough, the higher the training accuracy, the better. But when training data is insufficient or has strong noises, it is very important to find a balance between training error and overfitting. Our method is to compute the posterior distribution of BRB parameters rather than single optimal values of BRB parameters, and predict the outputs by considering all possible parameters. It is like a proverb, "don't put your eggs in one basket", so the overfitting risk can be decreased, but the accuracy is not the best.

To further illustrate the principle of Bayesian estimation for updating BRB parameters, and its ability to deal with the training data with noises, a figure is drawn as Fig. 5. The gray background consists of output trajectories of 1000 samples calculated by Beyesian trained BRB, and each trajectory has its own weight, and the final output of Beyesian trained BRB which is represented by the red curve is the weighted average of these output trajectories.

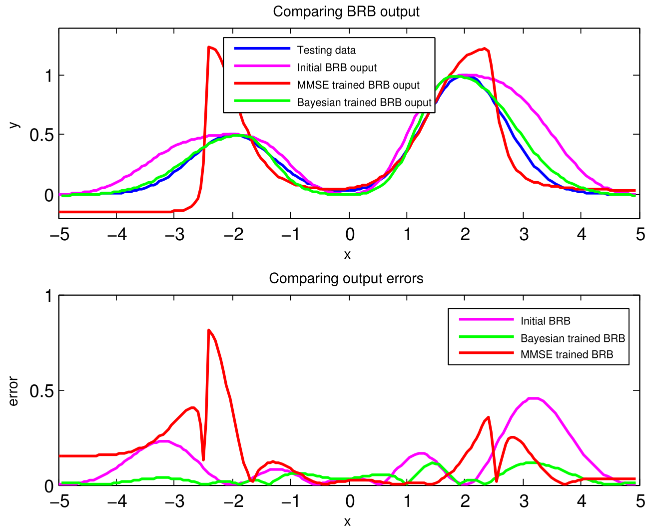

Fig. 4. Comparison of outputs of trained BRB (12 training points with noises).

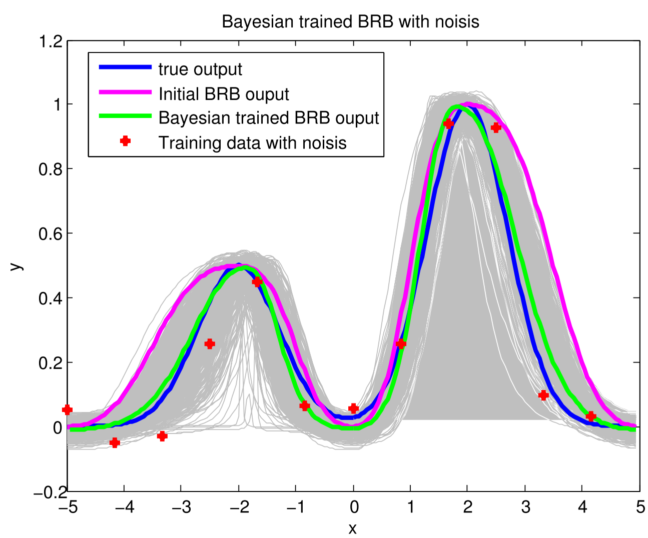

Fig. 5. Trajectories of output samples of Beyesian trained BRB. . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

As can be seen from the figure, the training data with noises are enclosed in the output trajectories, but these training data are not all in the middle of the output trajectories. This means that the Bayesian estimation assumes that an output of a BRB is equal to a certain training value only in a certain probability, which may sometimes be a very small value because other training points can change the weight. Furthermore, the Bayesian training method allows us to estimate the distribution of the BRB output by histogram estimation mentioned in Section 4.6. We divide the range of output into 100 intervals, and the density estimated by Eq. (24) is shown in a three-dimensional figure in Fig. 6. It can be seen from the figure that the shape of the distribution is similar to Gaussian distribution.

5.2. A practical case study

To better illustrate and validate our approach, a practical application for oil pipeline leak detection provided by Xu [47] is studied in this sub-section.

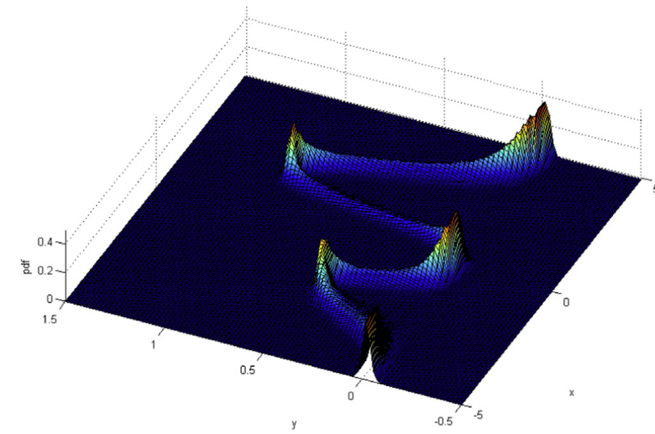

Fig. 6. Estimated density of outputs of Beyesian trained BRB.

5.2.1. Background formulation

A leak in a pipeline can change the flow and pressure in the pipeline, which has been analyzed by Xu, and the difference between inlet flow and outlet flow, and the average pipeline pressure change over time, denoted by FlowDiff, PressureDiff, respectively, are considered as the two factors in detecting whether there is a leak in the pipeline [47]. Therefore, for BRB system, FlowDiff and PressureDiff are selected as antecedent attributes, and leak size denoted LeakSize is the consequent attribute. During the leak trial, 2008 samples were collected at the rate of 10 s per sample. Fig. 7 shows the FlowDiff and PressureDiff and LeakSize. As discussed before, decreasing complexity is one of solution to overfitting problem. Many previous works have researched on methods to reduce antecedent attributes, number of referential values for each antecedent attribute, or number of rules. In Xu's work, 8 referential points were selected for FlowDiff, and 7 referential points were selected for PressureDiff, so there are $7 \times 8 = 56$ rules in total, which was considered as a over-complete structure [24]. And In Wang's work, a conclusion has been draw that the BRB with six rules is the complete structure and exhibits a better estimation performance with least mean average error [24]. Therefore, we choose the BRB structure with six rules for the practical case of pipeline leak detection. Furthermore, three referential values, i.e. negative large (NL), zeros (Z) and positive large (PL) are selected for the antecedent attribute of PressureDiff, and two referential values, i.e. negative large (NL), and positive large (PL) are chosen for the antecedent attribute of PressureDiff. Their utility values are shown as follows:

$$A_1^k \in \{(NL, -10), (Z, 0), (PL, 3)\}$$
$$A_2^k \in \{(NL, -0.02), (PL, 0.02)\} \tag{27}$$

It is worth noting that the range of the PressureDiff is $[-0.02, 0.04]$, however, the range $[-0.01, 0.01]$ for the PressureDiff is used in many works [47,20,24], including the original provider [47] of the data. Therefore, we select the range $[-0.01, 0.01]$ for the PressureDiff and treat out-of-range data as anomalous data.

For the consequent attribute of LeakSize, 5 referential points are selected: zero (Z), very small (VS), medium (M), high (H) and very high (VH).

$$D = \{D_1, D_2, D_3, D_4, D_5\} = \{(Z, 0), (VS, 2), (M, 4), (H, 6), (VH, 8)\} \tag{28}$$

5.2.2. Training data and initial BRB parameter

Although a relatively simple BRB structure is chosen, the problem of over-fitting may still occur due to the lack of training data. In

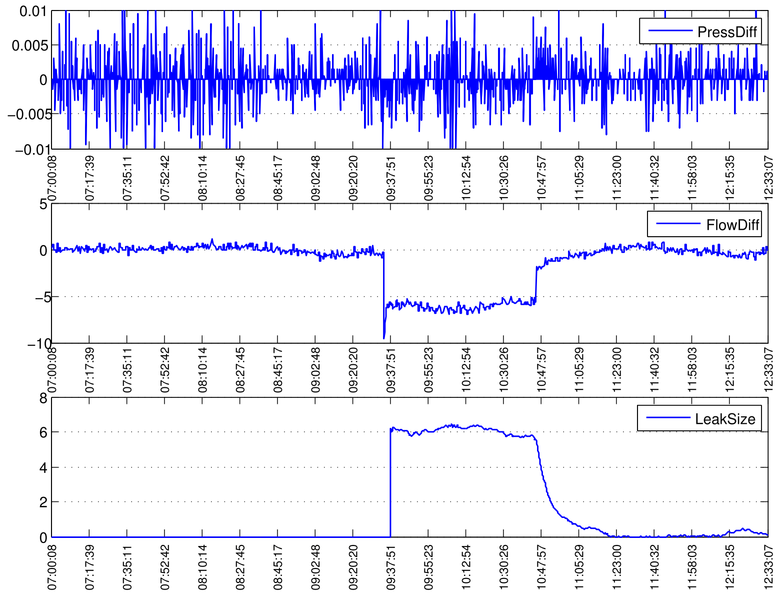

Fig. 7. Samples of 25% leak data sets.

order to compare the performance of our methods and traditional optimization methods under different size of data sets, the training data selection method is as follow:

When N data points are selected as the training set, the total 2008 points will be divided into N data sets equally, and the remainder is ignored. Then one data point is selected randomly from each data set, so that a total of N data points is obtained. For example, if 20 points are selected as the training set, the 2008 data points will be divided into 20 sets evenly, and the last 8 points are ignored. Therefore, each set includes 100 points, and then one data point is randomly extracted from the 100 points, so that a set of 20 data points is obtained. The total 2008 data points will be used as the testing set. 20 runs of experiments will be conducted for each size of training data set, the MSE described later in the results refers to the average of the MSEs obtained from these 20 runs of experiments.

An initial belief rule base is constructed as Table 6 shows.

The prior distribution of parameters transition $p(\boldsymbol{Q}_t|\boldsymbol{Q}_{t-1}^i)$ and the observation distribution $P(\boldsymbol{S}_t|\boldsymbol{Q}_t^i)$ are set as follows,

$\boldsymbol{\theta}_t^i \sim N(\boldsymbol{\theta}_{t-1}^i, 0.4)$, $\boldsymbol{\beta}_t^i \sim N(\boldsymbol{\beta}_{t-1}^i, 0.3)$, $\boldsymbol{A}_t^i \sim N(\boldsymbol{A}_{t-1}^i, 0.3)$, $\boldsymbol{U}_t^i \sim N(\boldsymbol{U}_{t-1}^i, 0.3)$, $\boldsymbol{\delta}_t^i \sim N(\boldsymbol{\delta}_{t-1}^i, 0.5)$, $P(\boldsymbol{S}_t|\boldsymbol{Q}_t^i) \sim N(\boldsymbol{S}_t^*, 0.3)$. And the number of samples $N_s = 10000$, the initial weights $w_0^i = 1/N_s$.

5.2.3. Validation of overfitting problem for BRB trained by traditional optimization method

In order to illustrate that the traditional optimization method may have over-fitting problems, a small set of points (20 points exactly) to simulate the case of insufficient training data, and use these 20 points to train the parameters of the BRB. The training data is generated by the above method. To verify whether the traditional optimization method will have the overfitting problem described in Fig. 1, the mean training error and testing error for different iteration steps are recorded as Fig. 8. It can be seen that when the iteration reaches 15–20 steps, the testing error is at the lowest level. As the number of iteration steps continues to increase, the training error gradually decreases, but the testing error starts to increase, thus the overfitting problem occurs. Besides, if the number of iteration steps is between 15 and 20, the optimization method for BRB parameter training can achieve a relative high testing accuracy, and the minimum MSE is decreased to 0.4562 in fact. This means that the 20 data points contain a lot of information about the relationship between the input and output. Even the optimization method can use these 20 points to obtain a more accurate BRB model if the training is terminated early in time. However, "early stop" will greatly complicate the training process of BRB as discussed in the introduction section.

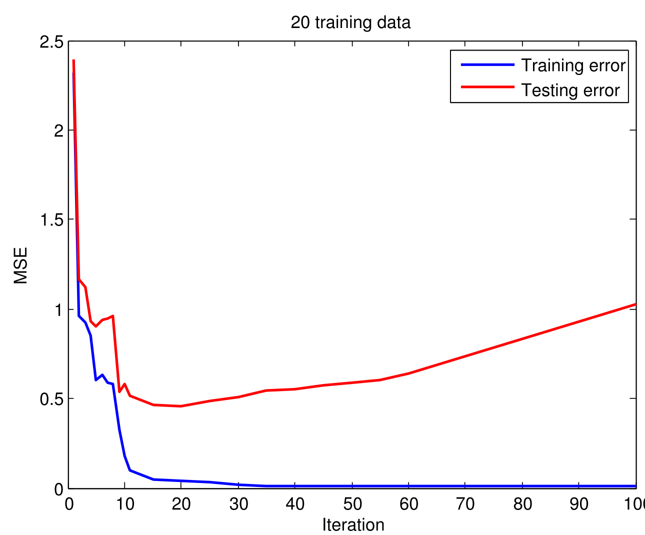

Fig. 8. Overfitting problem for BRB trained by traditional optimization method.

5.2.4. The impact of different training data sets on training results

This sub-section will compare the training results of BRB system trained by traditional optimization method, that trained by Bayesian estimation, and the BP neural network method under different data sets, but the size of data set is the same. The number of iteration steps for optimization method is 100. The parameter of BP neural network is set as follows: Two hidden layers, the first hidden layer has 6 nodes, using linear transfer function, and the second hidden layer has 1 node, using "logsig" transfer function; the maximum number of training times is 100; the learning rate is 0.05.

In this experiment, 20 data points are still used, and the method of training data generation is as mentioned above. 20 runs of

Table 6
Initial BRB for leak size detection.

R_k	θ_k	$x(A_1)$	$x(A_2)$	Consequents $U(D) = \{0, 2, 4, 6, 8\}$, $\delta_1 = 1$, $\delta_2 = 1$
1	1	-10	-0.01	$\{(D_1, 0), (D_2, 0), (D_3, 0), (D_4, 0.2), (D_5, 0.8)\}$
2	1	-10	0.01	$\{(D_1, 0), (D_2, 0), (D_3, 0), (D_4, 0.2), (D_5, 0.8)\}$
3	1	-1	-0.01	$\{(D_1, 0), (D_2, 0.8), (D_3, 0.2), (D_4, 0), (D_5, 0)\}$
4	1	-1	0.01	$\{(D_1, 0), (D_2, 0), (D_3, 0.8), (D_4, 0.2), (D_5, 0)\}$
5	1	3	-0.01	$\{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)\}$
6	1	3	0.01	$\{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)\}$

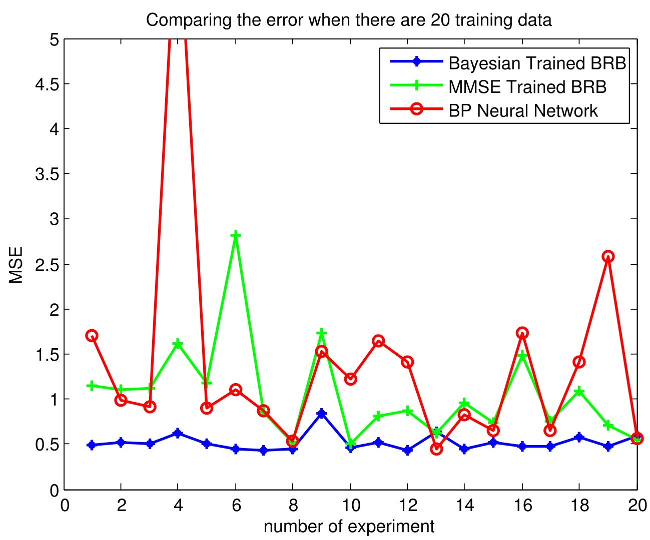

Fig. 9. The impact of different training data sets on training results.

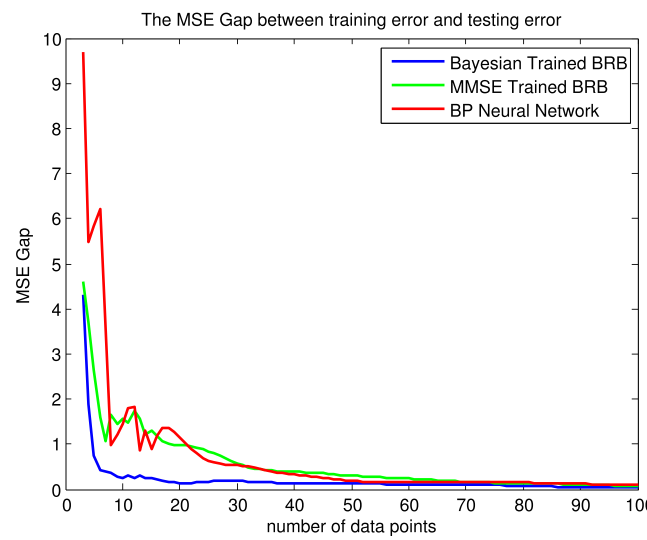

Fig. 10. The impact of different training data size on training results.

experiments are conducted, and the test errors of each run of experiments are recorded as shown in Fig. 9.

As can be seen from Fig. 9, the fluctuation of testing error of BRB trained by Bayesian estimation is obviously smaller than that trained by traditional optimization methods and the BP neural networks, and the overall testing error of BRB trained by Bayesian estimation is relative smaller. The average testing error, maximum testing error and minimum testing error of the BRB trained by Bayesian estimation are 0.5209, 0.8349, 0.4271, respectively. And these several error values are 1.0595, 2.3958, 0.4562 respectively for BRB trained by traditional optimization method, and 1.4104, 6.5115, 0.4502, respectively for BP neural networks. Therefore, for the sensitivity to the training set, it is lower for BRB trained by Bayesian estimation than that trained by traditional optimization method, but it is lower for BRB trained by traditional optimization method than BP neural networks. This indicates that the BRB trained by Bayesian estimation has a stronger adaptability to different training sets, which reflects its ability to overcome overfitting.

5.2.5. The impact of different training data size on training results

This sub-section will compare the training results of BRB system trained by traditional optimization method, that trained by Bayesian estimation, and the BP neural network method under different size of data sets. The number of iteration steps for optimization method is 100, and the parameter for BP neural network is set as mentioned previously.

The amount of overfitting can be as measured by the gap between training and testing accuracy [53], which can represented by,

$$Gap = |E_{test} - E_{train}| \tag{29}$$

where, E_{test} refers to the training error, and E_{train} refers to the testing error. Through a large number of experiments, the gap between training and testing error under different size of training data are plotted as Fig. 10.

From Fig. 10, it can be seen that when the size of training data set is smaller than 50 data points, the error gap of BRB trained by Bayesian estimation is obviously smaller than that of the other two model. This indicates that BRB trained by Bayesian estimation can overcome the overfitting problem which may occur for BRB trained by optimization method or BP neural network when training data size is relative small. When the size of training data set is smaller than 10 data points, the error gap of BP neural network is much larger than that of two kinds of BRB model. When the size of training data set is smaller than 10 data points, the error gap of BP neural network is much larger than that of the two BRB model. This means that when training data set is small to some extent, the pure data-driven method has a higher risk of overfitting than expert system.

To further validate the performance of BRB trained by Bayesian estimation, the testing errors under some different size of training data set are listed in Table 7. The results show that when the number of data points is less than 100, the performance of BRB trained by Bayesian estimation is better than that of BRB trained by MMSE, and the performance of BRB trained by MMSE is better than that of neural network. When the size of training data set is around 100, the performance of the three is almost the same. As the training data continues to increase, the performance of BRB trained by Bayesian estimation is not as good as the latter two, but it does not reach an unacceptable level. In addition, it can be seen that when there are enough training data points (more than 100 points), BRB's prediction accuracy has no advantage over BP neural network. Therefore, when the training data is sufficient, BRB has

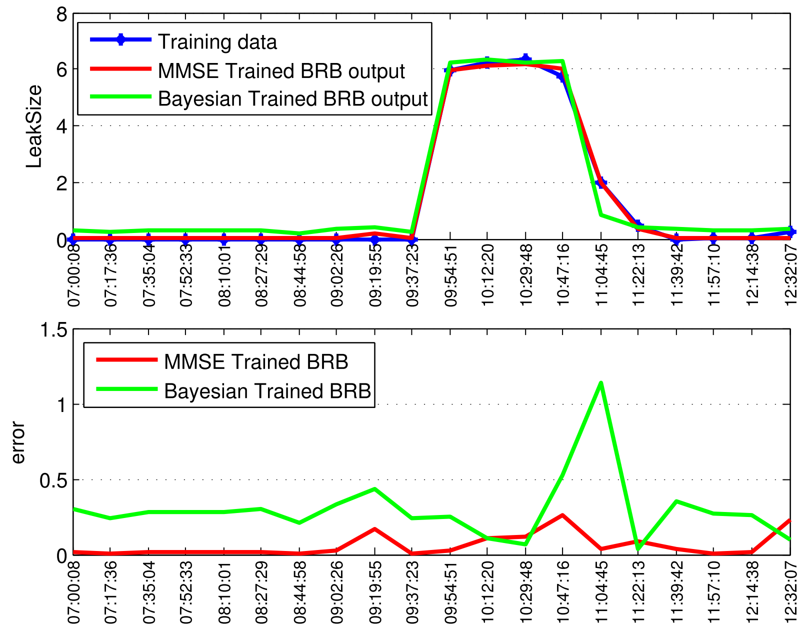

Fig. 11. Comparison of trained BRB outputs with training data.

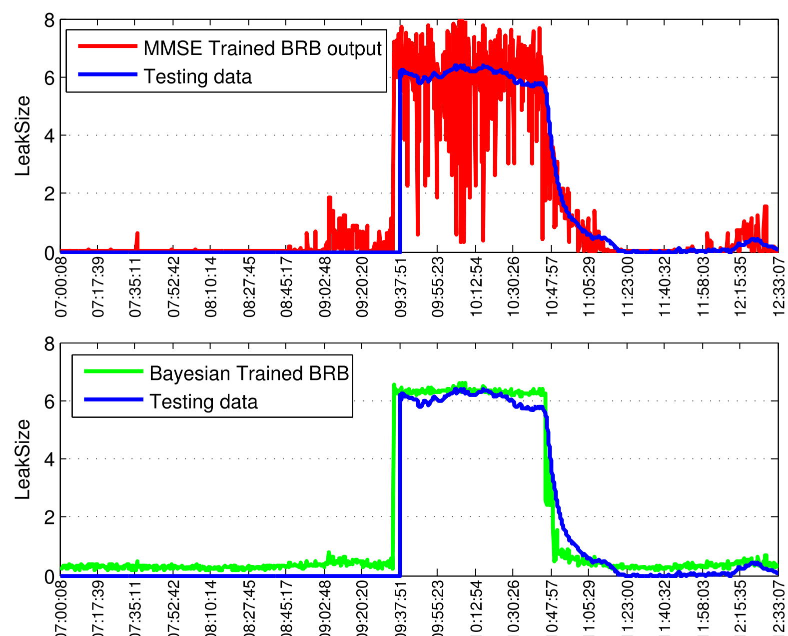

Fig. 12. Comparison of trained BRB outputs with testing data.

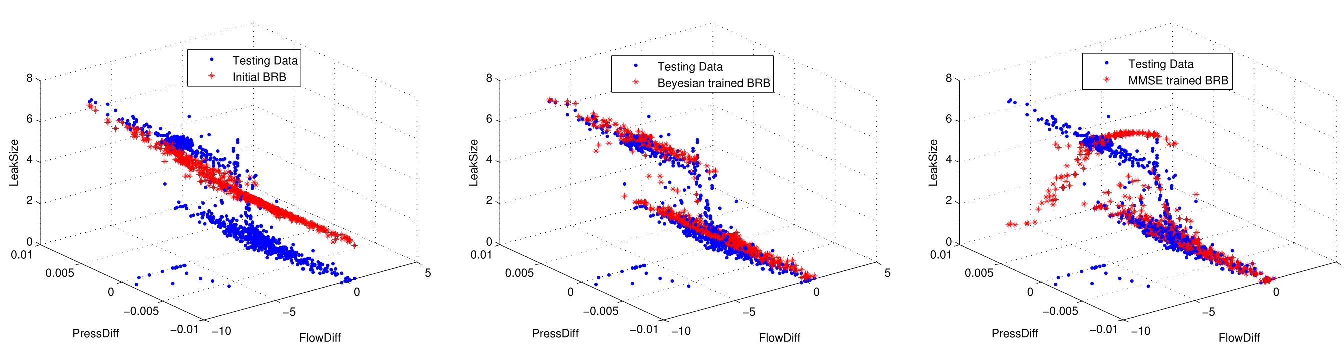

Fig. 13. Testing data and the outputs of initial BRB, MMSE trained BRB and Beyesian trained BRB.

Table 7
Testing errors under some different size of training data set.

Number of data points	Bayesian trained BRB	MMSE trained BRB	BP neural networks
5	0.8757	2.6350	5.8210
10	0.5832	1.5940	1.9584
20	0.5209	1.0595	1.4104
50	0.4789	0.5189	0.5565
100	0.4567	0.4647	0.4367
500	0.4347	0.3951	0.3737
800	0.4356	0.3900	0.3708

Table 8
Bayesian trained BRB for leak size detection.

R_k	θ_k	$x(A_1)$	$x(A_2)$	Consequents $U(D) = \{0, 1.5320, 4.0928, 5.5230, 8\}$, $\delta_1 = 0.4722$, $\delta_2 = 0.6491$
1	0.4731	−10	−0.01	$\{(D_1, 0.0047), (D_2, 0.0060), (D_3, 0.0075), (D_4, 0.4937), (D_5, 0.4881)\}$
2	0.5789	−10	0.01	$\{(D_1, 0024), (D_2, 0.0047), (D_3, 0.0051), (D_4, 0.5947), (D_5, 0.3932)\}$
3	0.0839	−1.9123	−0.01	$\{(D_1, 0.0021), (D_2, 0.4891), (D_3, 0.2015), (D_4, 0.0042), (D_5, 0.0030)\}$
4	0.0747	−1.9123	0.01	$\{(D_1, 0.0018), (D_2, 0.4602), (D_3, 0.5127), (D_4, 0.0146), (D_5, 0.0107)\}$
5	0.6514	3	−0.01	$\{(D_1, 0.9752), (D_2, 0.0061), (D_3, 0.0040), (D_4, 0.0115), (D_5, 0.0033)\}$
5	0.5387	3	0.01	$\{(D_1, 0.9805), (D_2, 0.0053), (D_3, 0.0054), (D_4, 0.0039), (D_5, 0.0050)\}$

Table 9
MMSE trained BRB for leak size detection.

R_k	θ_k	$x(A_1)$	$x(A_2)$	Consequents $U(D) = \{0, 0.5075, 3.6906, 6.1162, 8\}$, $\delta_1 = 0.7857$, $\delta_2 = 0.8207$
1	0.2035	−10	−0.01	$\{(D_1, 0), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 1)\}$
2	0.1041	−10	0.01	$\{(D_1, 0.5906), (D_2, 0.3915), (D_3, 0), (D_4, 0), (D_5, 0.0179)\}$
3	0.0331	−0.6383	−0.01	$\{(D_1, 0.6622), (D_2, 0.2473), (D_3, 0.0089), (D_4, 0.0816), (D_5, 0)\}$
4	0	−0.6383	0.01	$\{(D_1, 0.3574), (D_2, 0.0518), (D_3, 0.3671), (D_4, 0.0439), (D_5, 0.1797)\}$
5	1	3	−0.01	$\{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)\}$
6	1	3	0.01	$\{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)\}$

no advantage in accuracy over pure data-driven method, but the advantage of BRB is that the internal mechanism is transparent. If the accuracy advantage of BRB over pure data-driven method is emphasized, it can only be shown when the training data is insufficient. However, if the traditional optimization method is used to train BRB, the problem of over-fitting may occur because of insufficient data. Under such situation, the Bayesian estimation is a better choice to train the BRB.

5.2.6. Comparison of training results of Bayesian trained BRB and MMSE trained BRB

In this experiment, 20 data points are used for training data set, however, the training data is not generated randomly, but generated evenly from the complete leak data as our training data set, i.e. the 1st, 101th, 201th, …, 2001th data points.

After being trained by our method and the MMSE method, the estimated parameters are presented in Tables 8 and 9 respectively. Table 9 shows that the parameters of second rule is inexplainable. For example, $\beta_1 = 0.5906$, $\beta_2 = 0.3915$, which means there is high probability that the LeakSize is zero or very small. However, when FlowDiff is -10 and PressDiff is 0.01, there should be a high chance of a large LeakSize according to the human experience. On the contrary, from Table 8, it can be seen that the parameters of BRB, trained by Bayesian method, is much more acceptable to human, because it only slightly adjusts the parameters provided by experts.

We firstly compared the training data with the outputs of MMSE trained BRB and Bayesian trained BRB, as is shown in Fig. 11. It can be seen that the outputs of MMSE trained BRB are perfect, and their training errors are much smaller than that of Bayesian trained BRB. However, when we use the complete leak size data as the testing data, the results are very different, as is presented in Fig. 12. The outputs of MMSE trained BRB are very bad, and conversely, the Bayesian trained BRB performs much better although not perfect. This is because the problem of overfitting occurs when training BRB by using the MMSE method due to lack of training data, but it is overcome by the Bayesian estimation method. Fig. 13 shows the three dimensional figure of testing data and the outputs of initial BRB, MMSE trained BRB and Beyesian trained BRB, and it can be said that some outputs of MMSE trained BRB are completely wrong. Therefore, the capability of avoiding overfitting problem is very important when training data is insufficient, because a wrong result can lead to very serious consequences for crucial systems.

6. Conclusion

This paper focuses on updating BRB parameters online by using Bayesian estimation. The proposed updating algorithm provides a novel way to estimate the posterior distribution of BRB parameters, instead of taking the BRB training as an optimization problem to find single optimal parameter values. Because of nonlinear characteristics of BRB system, the posterior distribution of BRB parameters cannot be calculated by analytical method, therefore, the Monte Carlo sampling technique is proposed to approximate it. Two case study are implemented to verify the performance of the proposed Bayesian estimation method, which shows several features:

Firstly, the parameter updating of BRB by our method is a recursive process, when the new observed data becomes available, the proposed method can update the BRB quickly without reviewing the full historical data. Secondly, since the proposed algorithm is computing the posterior distribution of BRB parameters, and produce prediction outputs by considering all possible parameters, it can avoid overfitting problem when there is not enough training data or the training data contains strong noises. The capability of avoiding overfitting problem is very important, since a wrong result can lead to very serious consequences for crucial systems. Finally, compared with the optimization method, the Bayesian method adjusts the parameters of BRB in a more conservative way, which allows the knowledge of experts plays a major role in deciding the BRB parameters, so it can prevent parameters from becoming inexplicable. The results indicates that the accuracy of BRB trained by the proposed method is lower than that of BRB trained by traditional optimization method when training data is sufficient, however, when the training data set is small, the BRB trained by our method preforms much better than the BRB trained by traditional optimization method.

However, since we assume the adjustment of parameters as a random walk process, a large number of samples is needed to insure the performance of proposed method, and the repeat sampling may arise due to constraints. With the complexity of BRB increasing, the computational cost will increase severely. So, more efficient sampling methods need to be further researched to reduce the randomness of sampling, such as sampling along the descending gradient of ML, or certain sampling pattern determined by experts.

Acknowledgments

The authors sincerely appreciate the helpful comments and suggestions offered by the anonymous referees and chief editor for improving the quality of this paper.

References

[1] H. Kay, B. Rinner, B. Kuipers, Semi-quantitative system identification, Artificial Intelligence 119 (1) (2000) 103–140.
[2] 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 36 (2) (2006) 266–285.
[3] G. Li, Z. Zhou, C. Hu, L. Chang, Z. Zhou, F. Zhao, A new safety assessment model for complex system based on the conditional generalized minimum variance and the belief rule base, Saf. Sci. 93 (2017) 108–120.
[4] Z. Feng, Z. Zhou, C. Hu, X. Yin, G. Hu, F. Zhao, Fault diagnosis based on belief rule base with considering attribute correlation, IEEE Access PP (99) (2017) 2055–2067.
[5] G. Kongabcbdd, A belief rule-based decision support system for clinical risk assessment of cardiac chest pain, European J. Oper. Res. 219 (3) (2012) 564–573.
[6] S. Qiu, M. Sallak, W. Schön, H.X.G. Ming, A valuation-based system approach for risk assessment of belief rule-based expert systems, Inform. Sci. (2018).
[7] X. Yin, B. Zhang, Z. Zhou, X. Han, Z. Wang, G. Hu, A new health estimation model for CNC machine tool based on infinite irrelevance and belief rule base, Microelectron. Reliab. 84 (2018) 187–196.
[8] Z.J. Zhou, C.H. Hu, D.L. Xu, M.Y. Chen, D.H. Zhou, A model for real-time failure prognosis based on hidden Markov model and belief rule base, European J. Oper. Res. 207 (1) (2010) 269–283.
[9] X. Yin, Z. Wang, B. Zhang, Z. Zhou, Z. Feng, G. Hu, H. Wei, A double layer BRB model for health prognostics in complex electromechanical system, IEEE Access PP (99) (2017) 23833–23847.
[10] L. Chang, J. Sun, J. Jiang, M. Li, Parameter learning for the belief rule base system in the residual life probability prediction of metalized film capacitor, Knowl.-Based Syst. 73 (1) (2015) 69–80.
[11] Z.J. Zhou, C.H. Hu, B.C. Zhang, D.L. Xu, Y.W. Chen, Hidden behavior prediction of complex systems based on hybrid information, IEEE Trans. Cybern. 43 (2) (2013) 402–411.
[12] C.H. Hu, C.H. Hu, D.L. Xu, J.B. Yang, D.H. Zhou, New model for system behavior prediction based on belief rule based systems, Inform. Sci. 180 (24) (2010) 4834–4864.
[13] G.Y. Hu, Z.J. Zhou, B.C. Zhang, X.J. Yin, Z. Gao, Z.G. Zhou, A method for predicting the network security situation based on hidden BRB model and revised CMA-ES algorithm, Appl. Soft Comput. 48 (C) (2016) 404–418.
[14] G.Y. Hu, P.L. Qiao, Cloud belief rule base model for network security situation prediction, IEEE Commun. Lett. 20 (5) (2016) 914–917.
[15] B.C. Zhang, G.Y. Hu, Z.J. Zhou, Y.M. Zhang, P.L. Qiao, L.L. Chang, B.C. Zhang, G.Y. Hu, Z.J. Zhou, Y.M. Zhang, Network intrusion detection based on directed acyclic graph and belief rule base, ETRI J. 39 (4) (2017) 592–604.
[16] M.S. Hossain, S. Rahaman, R. Mustafa, K. Andersson, A belief rule-based expert system to assess suspicion of acute coronary syndrome (ACS) under uncertainty, Soft Comput. (6) (2017) 1–16.
[17] M.S. Hossain, F. Ahmed, Fatema-Tuj-Johora, K. Andersson, A belief rule based expert system to assess tuberculosis under uncertainty, J. Med. Syst. 41 (3) (2017) 43.
[18] Y.W. Chen, J.B. Yang, C.C. Pan, D.L. Xu, Z.J. Zhou, Identification of uncertain nonlinear systems: Constructing belief rule-based models, Knowl.-Based Syst. 73 (2015) 124–133.
[19] J.B. Yang, J. Liu, D.L. Xu, J. Wang, H. Wang, Optimization models for training belief-rule-based systems, IEEE Trans. Syst. Man Cybern. A 37 (4) (2007) 569–585.
[20] Y.W. Chen, J.B. Yang, D.L. Xu, Z.J. Zhou, D.W. Tang, Inference analysis and adaptive training for belief rule based systems, Expert Syst. Appl. 38 (10) (2011) 12845–12860.
[21] Z.J. Zhou, C.H. Hu, J.B. Yang, D.L. Xu, D.H. Zhou, Online updating belief rule based system for pipeline leak detection under expert intervention, Expert Syst. Appl. 36 (4) (2009) 7700–7709.
[22] Z.J. Zhou, C.H. Hu, J.B. Yang, D.L. Xu, D.H. Zhou, Online updating belief-rule-base using the RIMER approach, IEEE Trans. Syst. Man Cybern. A 41 (6) (2011) 1225–1243.
[23] L. Chang, Y. Zhou, J. Jiang, M. Li, X. Zhang, Structure learning for belief rule base expert system: A comparative study, Knowl.-Based Syst. 39 (2) (2013) 159–172.
[24] Y.M. Wang, L.H. Yang, Y.G. Fu, L.L. Chang, K.S. Chin, Dynamic rule adjustment approach for optimizing belief rule-base expert system, Knowl.-Based Syst. 96 (C) (2016) 40–60.
[25] L.H. Yang, Y.M. Wang, J. Liu, L. Martinez, A joint optimization method on parameter and structure for belief-rule-based systems, Knowl.-Based Syst. 142 (2011) 220–240.
[26] J.B. Sun, J. Huang, L.L. Chang, J. Jiang, Y.J. Tan, BRBcast: A new approach to belief rule-based system parameter learning via extended causal strength logic, Inform. Sci. 444 (2018).
[27] L. Chang, Z. Zhou, Y. Chen, X. Xu, J. Sun, T. Liao, X. Tan, Akaike information criterion-based conjunctive belief rule base learning for complex system modeling, Knowl.-Based Syst. 161 (2018) 47–64.
[28] T. Dietterich, Overfitting and undercomputing in machine learning, ACM Comput. Surv. 27 (3) (1995) 326–327.
[29] D.M. Hawkins, The problem of overfitting, Cheminform 35 (19) (2004) 1.
[30] L. Prechelt, Automatic early stopping using cross validation: quantifying the criteria, Neural Netw. 11 (4) (1998) 761–767.
[31] L. Prechelt, Early stopping — but when? Springer Berlin Heidelberg, 1998, pp. 55–69.
[32] C.X. Ling, Overfitting and generalization in learning discrete patterns, Neurocomputing 8 (3) (1995) 341–347.
[33] M. Welling, Exploiting the statistics of learning and inference, Comput. Sci. (2014).
[34] J. Zhu, J. Chen, W. Hu, B. Zhang, T. Lab, Big learning with Bayesian methods, Natl. Sci. Rev. (4) (2017) 627–651.
[35] C.M. Bishop, N.M. Nasrabadi, Pattern Recognition and Machine Learning, Academic Press, 2006, pp. 461–462.
[36] W. Bialek, I. Nemenman, N. Tishby, Predictability, complexity, and learning, Neural Comput. 13 (11) (2001) 2409–2463.
[37] Spragins, A note on the iterative application of Bayes' rule, IEEE Trans. Inform. Theory 11 (4) (1965) 544–549.
[38] Y.C. Ho, R.C.K. Lee, A Bayesian approach to problems in stochastic estimation and control, IEEE Trans. Automat. Control 9 (4) (1964) 333–339.
[39] Z.H.E. Chen, Bayesian filtering: From Kalman filters to particle filters, and beyond, Statistics 182 (1) (2003) 1–69.
[40] L. Lang, W. Chen, B.R. Bakshi, P.K. Goel, S. Ungarala, Bayesian Estimation via Sequential Monte Carlo Sampling-Constrained Dynamic Systems, 2007.
[41] C.P. Robert, G. Casella, Monte Carlo Statistical Methods, (Springer Texts in Statistics), 2006, p. 110.
[42] B.M. Turner, T. Van Zandt, A tutorial on approximate Bayesian computation, J. Math. Psych. 56 (2) (2012) 69–85.
[43] T.P. Minka, A Family of Algorithms for Approximate Bayesian Inference (Thesis), Massachusetts Institute of Technology, 2001.
[44] C. Andrieu, N.D. Freitas, A. Doucet, M.I. Jordan, An introduction to MCMC for machine learning, Mach. Learn. 50 (1–2) (2003) 5–43.
[45] S.N. Maceachern, M. Clyde, J.S. Liu, Sequential importance sampling for non-parametric Bayes models: The next generation, Canad. J. Statist. 27 (2) (1999) 251–267.
[46] J. Liu, RongChen, Sequential Monte Carlo methods for dynamic systems, Publ. Am. Stat. Assoc. 93 (443) (1998) 1032–1044.
[47] D.L. Xu, J. Liu, J.B. Yang, G.P. Liu, J. Wang, I. Jenkinson, J. Ren, Inference and learning methodology of belief-rule-based expert system for pipeline leak detection, Expert Syst. Appl. 32 (1) (2007) 103–113.
[48] Z.J. Zhou, G.Y. Hu, B.C. Zhang, C.H. Hu, Z.G. Zhou, P.L. Qiao, A model for hidden behavior prediction of complex systems based on belief rule base and power set, IEEE Trans. Syst. Man Cybern. A PP (99) (2017) 1–7.
[49] J.B. Yang, Rule and utility based evidential reasoning approach for multiattribute decision analysis under uncertainties, European J. Oper. Res. 131 (1) (2001) 31–61.
[50] A. Doucet, N.D. Freitas, N. Gordon, Sequential Monte Carlo Methods in Practice, Springer, 2001.
[51] C. Andrieu, N.D. Freitas, A. Doucet, Sequential MCMC for Bayesian model selection, in: Higher-Order Statistics, 1999, Proceedings of the IEEE Signal Processing Workshop on, 2002, pp. 130–134.
[52] Y.W. Chen, J.B. Yang, D.L. Xu, S.L. Yang, On the inference and approximation properties of belief rule based systems, Inform. Sci. 234 (11) (2013) 121–135.
[53] M. Cogswell, F. Ahmed, R. Girshick, L. Zitnick, D. Batra, Reducing overfitting in deep networks by decorrelating representations, 2015, arXiv preprint arXiv:1511.06068.