Applied Intelligence (2022) 52:1059–1075
https://doi.org/10.1007/s10489-021-02448-6

Research on a dynamic full Bayesian classifier for time-series data with insufficient information

Shuangcheng Wang[1,2] · Siwen Zhang[3] · Tao Wu[4] · Yongrui Duan[3] · Liang Zhou[5]

Accepted: 20 April 2021 / Published online: 16 May 2021

Abstract
Small sample time-series data with insufficient information are ubiquitous. It is challenging to improve the classification reliability of small sample time-series data. At present, the dynamic classifications for small sample time-series data still lack a tailored method. To address this, we first setup the architecture of dynamic Bayesian derivative classifiers, and then establish a dynamic full Bayesian classifier for small sample time-series data. The joint density of attributes is estimated by using multivariate Gaussian kernel function with smoothing parameters. The dynamic full Bayesian classifier is optimized by splitting the smooth parameters into intervals, optimizing the parameters by constructing a smoothing parameter configuration tree (or forest), then selecting and averaging the classifiers. The dynamic full Bayesian classifier is applied to forecast turning points. Experimental results show that the resultant classifier developed in this paper is more accurate when compared with other nine commonly used classifiers.

Keywords Dynamic full Bayesian classifier · Multivariate Gaussian kernel function · Smoothing parameter · Classification accuracy · Small sample time-series data

1 Introduction

Many classifiers have been developed and widely applicated in the past decades, such as the Support Vector Machine [1], BP Neural Network [2], Decision Tree [3]. However, these classifiers are mainly oriented to non-time-series data with an underlying assumption that the data records are independent and identically distributed. Therefore, when the data records are not independent and identically distributed, the classifiers referred to are difficult to use for processing time-series data. In addition, models applicable to econometric time-series forecasting such as ARIMA [4] and GARCH [5] are suitable for regression, but not suitable for classification. The dynamic Bayesian classifier [6] is a temporal extension to the Bayesian classifier [7], it can be used to solve the classification problems over large sample time-series data. However, there exist many forecasts that need to be made based on small sample time-series data in practice. For example, long-term global economy growth and monetary policy rules are usually predicted based on annual data such as GDP and CPI [7, 8]. There is little tailored research on finding a dynamic Bayesian classifier for processing small sample data at present. Therefore, in this paper, we propose a dynamic Bayesian derivative classifier which is predicated on the need to cater for classification problems related specifically to small sample time-series data.

The last few decades have seen many Bayesian classifiers [9] designed and proposed that can be divided principally into two classes: some with discrete attributes and others with continuous attributes. In relation to the classifiers with discrete attributes or the discretization of continuous attributes, Chow and Liu (1968) [10] proposed the Dependency Tree classifier. Friedman et al. (1997) [11] proposed the TAN (Tree augmented naïve Bayes) classifier. Domingos and Pazzani (1997) [12] optimized the simple Bayesian classifier under 0–1 loss. Campos et al. (2016) [13] proposed an extended version of the TAN

Shuangcheng Wang, Siwen Zhang and Tao Wu contributed equally to this work.

✉ Yongrui Duan
yrduan@tongji.edu.cn

[1] School of Information Management, Shanghai Lixin University of Accounting and Finance, Shanghai, China

[2] Institute of Data Science and Interdisciplinary Studies, Shanghai Lixin University of Accounting and Finance, Shanghai, China

[3] School of Economics and Management, Tongji University, Shanghai 200092, China

[4] Shanghai Jiao Tong University School of Medicine, Shanghai, China

[5] Center for Medicine Intelligent and Development, China Hospital Development Institute, Shanghai Jiao Tong University, Shanghai, China

classifier by relaxing the independence assumption. Cheng and Greiner (1999) [14] designed a Bayesian network classifier based on dependency analysis to determine the network structure. Petitjean et al. (2018) [15] presented a hierarchical Dirichlet process to estimate accurate parameters for a Bayesian network classifier. Yager (2006) [16] provided an algorithm to obtain the weights associated with the extended naïve Bayesian classifier. Webb et al. (2005) [17] presented a classifier with an aggregating one-dependence estimator. Flores et al. (2014) [18] proposed the semi-naïve Bayesian network classifier. Daniel and Aryeh (2015) [19] examined the consistency of the optimal naïve Bayesian classifier for weighted expert majority votes under the small set of samples. Wang et al. (2013) [20], Sathyaraj and Prabu (2018) [21], Yang and Ding (2019) [22] and Kuang et al. (2019) [23] proposed Bayesian network classifiers based on search and scoring.

In respect of classifiers with continuous attributes, two aspects need to receive particular attention: one is setting up the classifier structure; the other is estimating the attribute density function. The structure of the Bayesian classifier with continuous attributes is similar to that with discrete ones. Currently, Gaussian function [24], Gaussian kernel function [25] and Copula function [26] are the main functions adopted to estimate the density of attributes [27]. John and Langley (1995) [28] established two naïve Bayesian classifiers by using Gaussian function and Gaussian kernel function to estimate the marginal density of attributes. The work is widely perceived as establishing the extended naïve Bayesian classifiers based on continuous attribute density estimation. Pérez et al. (2006, 2009) [24, 25] improved the estimation of Gaussian kernel function by introducing and optimizing the smoothing parameter. He et al. (2014) [29], Luis et al. (2014) [30] and Zhang et al. (2018) [31] developed the naïve Bayesian and the full Bayesian classifiers by using Gaussian function or Gaussian kernel function to estimate attribute density and applied them to fault diagnosis and spectroscopy analysis. Xiang et al. (2016) [32] used Gaussian kernel function to estimate attribute marginal density to set up an attribute weighted naïve Bayesian classifier. Wang et al. (2016) [33] presented a Bayesian network classifier by using Gaussian kernel function to estimate attribute conditional density.

The above-mentioned Bayesian classifiers are not suitable for time-series data classification (especially for small sample time-series data).

The research that we present here on dynamic Bayesian classifiers is mainly focused on classification with discrete attributes and need large sample time-series data for learning. Dang et al. (2016) [34] proposed a new method for the early detection of emergent topics, based on Dynamic Bayesian Networks in micro-blogging networks. Xiao et al. (2017) [35] proposed a novel time-series prediction model using a dynamic Bayesian network based on a combination of the Kalman filtering model (KFM) and echo state neural networks (ESN). Premebida et al. (2017) [36] addressed a time-based Dynamic Bayesian Mixture Model (DBMM) and applied it to solve the problem of semantic place categorization in mobile robotics. Extensive experiments were performed to highlight the influence of the number of time-slices and the effect of additive smoothing on the classification performance, and the results showed the effectiveness and competitive performance of the DBMM under different scenarios and conditions. Rishu et al. (2019) [37] developed a smartphone-based context-aware driver behavior classification using a Dynamic Bayesian Network (DBN) system, which demonstrated competitive performance when considering cost-effectiveness. Song et al. (2020) [38] proposed a Dynamic Hesitant Fuzzy Bayesian Network (DHFBN) to solve the optimal port investment decision-making problem of the "21st Century Maritime Silk Road".

Almost all the dynamic Bayesian network classifiers developed at present need large sample discrete time-series data for learning, which is not available for small sample time-series data problems. In this paper, a dynamic full Bayesian ensemble classifier with continuous attributes fitting for small sample data classification is proposed. The structure of the Bayesian classifier has been developed, and the density of attributes has been estimated. In addition, the classifier is applied in forecasting the turning points for indexes. The experimental results indicate that the classifier proposed in this paper has good classification accuracy in dealing with small sample time-series problems.

The main contributions of this paper are as follows:

(1) Dislocated transformation between variables and classes is utilized in developing the temporal asynchronous (non-synchronous) dynamic Bayesian derivative classifiers with continuous attributes.
(2) The multivariate Gaussian kernel function with smoothing parameters is used to estimate the joint density of attributes. Based on that, we develop a synchronous dynamic full Bayesian ensemble classifier to solve the multivariate small sample time-series data classification problems.
(3) We propose the architecture of the dynamic Bayesian derivate classifiers by extending the dependency of variables and dislocated transforming of variables based on dynamic Bayesian classifiers, dynamic full Bayesian classifiers and dynamic Bayesian network classifiers, which provides support for further research on dynamic Bayesian derivate classifiers.

This paper is organized as follows: Section 1 reviews and analyses on the research of Bayesian classifiers and dynamic Bayesian classifiers; Section 2 presents the definition and representation of Bayesian classifiers and dynamic Bayesian classifiers, as well as the structure of dynamic Bayesian derivative classifiers; Section 3 presents the definition and representation of dynamic full Bayesian classifier, the estimation method for attribute joint density, classification accuracy criterion for the time-series progressiveness, an algorithm for constructing a

smoothing parameter configuration tree and the ensemble of dynamic full Bayesian classifiers; Section 4 conducts the experiments and analysis on small sample time-series dataset problems; Section 5 concludes this work with further directions.

2 Architecture of dynamic Bayesian derivative classifier

Definitions for Bayesian classifiers and dynamic Bayesian classifiers (both synchronous and asynchronous) are given firstly in this section, and on this basis, the architecture of dynamic Bayesian derivative classifiers is established.

2.1 Dynamic Bayesian classifier

Suppose that the attribute and class variables of a non-time-series dataset are denoted as $X_1, \ldots, X_n$ and C, and $x_1, \ldots, x_n$, c are their specific values. Let D be a non-time-series dataset with N instances.

Definition 1 We call the classifier with the structure shown in Fig. 1 the Bayesian classifier (BC) [9].

According to the Bayesian network theory [39], a BC can be represented as:

$$\underset{c(x_1,\ldots,x_n)}{argmax}\ \{p(c|x_1,\ldots,x_n)\} \tag{1}$$

We use $X_1[t], X_2[t], \ldots, X_n[t]$ and $C[t]$ for attribute and class variables of a time-series dataset, and $x_1[t], x_2[t], \ldots, x_n[t]$, $c[t]$ to denote specific values taken by those variables. Dataset sequences in cumulative time-duration are denoted by $D[1]$, $D[2]$, …, $D[t]$, where $D[1] \subset D[2] \subset \ldots \subset D[t]$, the number of instances in the corresponding time-series dataset is denoted by $N[1]$, $N[2]$, …, $N[t]$, where $1 \le t \le T$. A dynamic Bayesian classifier is an extension of the Bayesian classifier for dealing with time-series problems. It can be defined in many forms, and we give the following definition:

Definition 2 The classifier with the structure given in Fig. 2 is labelled the dynamic Bayesian classifier (DBC) [6].

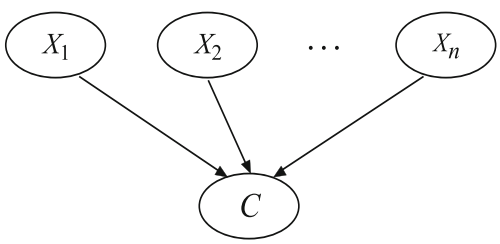


Fig. 1 The structure of the Bayesian classifier

The above dynamic Bayesian classifier may also be called the synchronous dynamic Bayesian classifier (SDBC), in which the attributes and class change synchronously in time. According to Bayesian network theory [39] and the conditional independencies contained in Fig. 2, we can get:

$$\begin{aligned} &p(c[t]|c[1],\ldots,c[t-1],x_1[1],\ldots,x_n[1],\ldots,x_1[t],\ldots,x_n[t]) \\ &\quad = p(c[t]|c[t-1],x_1[t],\ldots,x_n[t]) \end{aligned}$$

The DBC (or SDBC) can be expressed as:

$$\underset{c[t](c[t-1],x_1[t],\ldots,x_n[t])}{argmax}\ \left\{p\left(c[t]|c[t-1],x_1[t],\ldots,x_n[t]\right)\right\} \tag{2}$$

Based on the synchronous dynamic Bayesian classifier, the asynchronous (non-synchronous) dynamic Bayesian classifier can be constructed by dislocated transformation of variables (between attributes and class) in time series.

Definition 3 The classifier with the structure given in Fig. 3 is called the asynchronous dynamic Bayesian classifier (ADBC), where the classifier order $\varphi > 0$.

With the same method, the ADBC can be expressed as:

$$\underset{c[t+\varphi](c[t+\varphi-1],x_1[t],\ldots,x_n[t])}{argmax}\ \left\{p\left(c[t+\varphi]|c[t+\varphi-1],x_1[t],\ldots,x_n[t]\right)\right\} \tag{3}$$

2.2 Dynamic Bayesian derivative classifier

We label classifiers derived from a dynamic Bayesian classifier as the dynamic Bayesian derivative classifiers (DBDC). The DBDCs can be divided into two parts: the synchronous classifiers and the asynchronous classifiers. The synchronous classifier can be transformed into the corresponding asynchronous classifier by the dislocated transformation between attributes and classes over time series. According to the definition of the dynamic Bayesian derivative classifier, the structural changes in a time point (or a time slice) can be absorbed into the naïve structure, the full structure, and the Bayesian network structure (the other structures). By increasing the dependency between attributes and class over time points (or time slices), we can obtain the extended structure of temporal dependency; all these structures are synchronous classifiers. In the same way, the asynchronous classifier structures can be obtained by the dislocated transformation between attributes and class over time series. The specific structure is shown in Fig. 4.

Figure 4 shows the internal relationships between different dynamic Bayesian derivative classifiers. Future systematic and in-depth study of these dynamic Bayesian derivative classifiers can be performed based on the structure of the DBDCs.

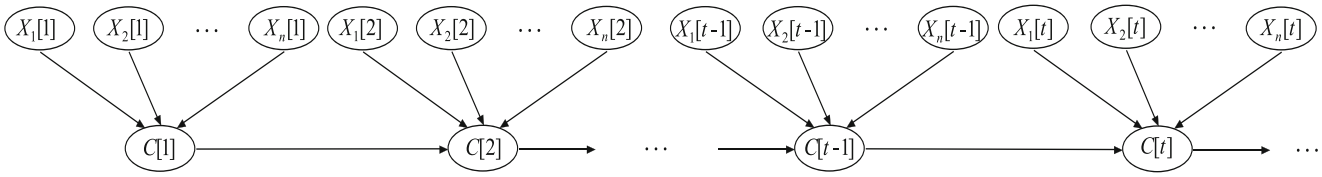


Fig. 2 The structure of the dynamic Bayesian classifier

3 Dynamic full Bayesian ensemble classifier

The information contained in small sample time-series data is very limited, and it is hard to effectively perform the classifier structure learning and parameter estimation of the parameter model. Therefore, we promote a synchronous dynamic full Bayesian classifier with continuous attributes to deal with this kind of classification problem, i.e., when operating with insufficient information. Instead of "by structure learning", the classifier can make full use of each record of information within datasets by estimating the joint density of attributes based on multivariate Gaussian kernel function. The classifier optimization is further improved by adjusting the smoothing parameters. The synchronous dynamic full Bayesian classifier is a kind of dynamic Bayesian derivative classifier that can make full use the information provided by attributes to improve the classification accuracy.

3.1 The definition and expression of the dynamic full Bayesian classifier

Definition 3 The classifier with the structure given in Fig. 5 is called as the dynamic full Bayesian classifier (DFBC).

According to the probability formula, the Bayesian network theory [39] and the conditional independencies shown in Fig. 5, we can show that:

$$\begin{aligned} &p(c[t]|c[1],\ldots,c[t-1],x_1[1],\ldots,x_n[1],\ldots,x_1[t],\ldots,x_n[t]) \\ &\quad = p(c[t]|c[t-1],x_1[t],\ldots,x_n[t]) \\ &\quad = \frac{p(c[t],c[t-1],x_1[t],\ldots,x_n[t])}{p(c[t-1],x_1[t],\ldots,x_n[t])} \\ &\quad = \alpha p(c[t]|c[t-1]) f(x_1[t],\ldots,x_n[t]|c[t]) \end{aligned} \tag{4}$$

where α is a normalization coefficient, which is independent of $C[t]$; $p(c[t]|\, c[t-1])$ denotes the transition probability of class and $f(\cdot)$ denotes the attribute density function.

The DFBC can be expressed as:

$$\underset{c[t](c[t-1],x_1[t],\ldots,x_n[t])}{argmax} \left\{ p\left(c[t]|c[t-1]\right) f\left(x_1[t],\ldots,x_n[t]|c[t]\right) \right\} \tag{5}$$

From the definition and expression of the DFBC, we can determine that the core is to estimate the joint density of attributes $f(x_1[t], \ldots, x_n[t]|\, c[t])$.

3.2 Estimation of joint attributes density function

In this section, we will use the multivariate kernel function with a diagonal smoothing parameter matrix to estimate the attribute density. This method performs well in local fitting to small sample time-series data, and it also has good anti-noise performance for dealing with time-series data classification.

For the dataset D with N records, the multivariate kernel function with smoothing parameter is denoted as [27]:

$$\phi(x_1,\cdots,x_n|D) = \frac{1}{N\rho_1\cdots\rho_n} \sum_{m=1}^{N} \prod_{i=1}^{n} K_i\left(\frac{x_i - x_{im}}{\rho_i}\right) \tag{6}$$

where $K_i(\cdot)$ is the kernel function of X_i, $K_i\left(\frac{x_i - x_{im}}{\rho_i}\right) = \frac{1}{\sqrt{2\pi}\rho_i} exp\left[-\frac{(x_i - x_{im})^2}{2\rho_i^2}\right]$, $\rho_1, \ldots, \rho_n$ are the smoothing parameters (or the bandwidth), and x_{im} is the value of the *mth* record of X_i in dataset D, $1 \le i \le n$, $1 \le m \le N$.

In this section, we will use the kernel function with diagonal smoothing parameter matrix to estimate the attribute density.

Let $\widehat{f}(x_1[t],\ldots,x_n[t]|c[t],D[t])$ denote the estimation of $f(x_1[t], \ldots, x_n[t]|\, c[t])$, which is an attribute conditional probability density function in temporal extension to $\phi(x_1,\cdots,x_n|\, D)$ under the classification, then

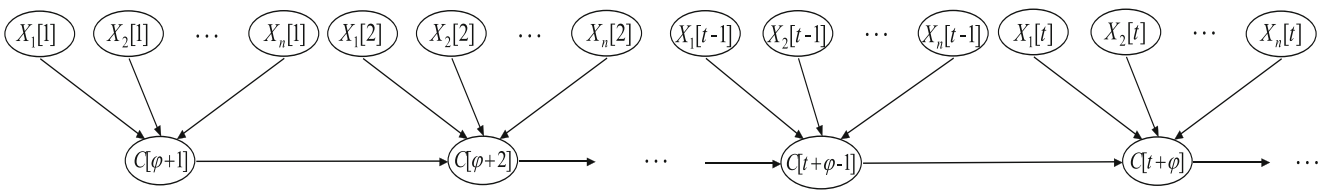


Fig. 3 The structure of the asynchronous dynamic Bayesian classifier

$$\widehat{f}(x_1[t],\ldots,x_n[t]|c[t],D[t]) = \frac{1}{(2\pi)^{n/2}N(c[t])\rho_1^2\cdots\rho_n^2}\sum_{v=1}^{t}\left\{signa(c[v])\prod_{i=1}^{n}exp\left[-\frac{(x_i[t]-x_i[v])^2}{2\rho_i^2}\right]\right\} \tag{7}$$

where $N(c[t])$ is the number of the instances when $C[t] = c[t]$ in $D[t]$, and $signa(c[v]) = \begin{cases} 1, c[v] = c[t] \\ 0, c[v] \neq c[t] \end{cases}$.

The DFBC can be denoted as:

$$\underset{c[t](c[t-1],x_1[t],\ldots,x_n[t])}{argmax}\left\{\frac{N(c[t],c[t-1])}{(2\pi)^{n/2}N(c[t])N(c[t-1])\rho_1^2\cdots\rho_n^2}\sum_{v=1}^{t}\left\{signa(c[v])\prod_{i=1}^{n}exp\left[-\frac{(x_i[t]-x_i[v])^2}{2\rho_i^2}\right]\right\}\right\} \tag{8}$$

where $N(c[t-1])$ are the numbers of the instances when $C[t-1] = c[t-1]$ in $D[t]$, and $N(c[t], c[t-1])$ are the numbers of the instances when both $C[t] = c[t]$ and $C[t-1] = c[t-1]$ in $D[t]$, respectively.

The smoothing parameters that shape the curve (or surface) of a Gaussian function will have great influence on the performances of a DFBC. To balance the training and generalization of the DFBC, we construct a smoothing parameter configuration tree (or forest) to optimize the DFBC, where the scoring and search method is used under the time-series progressiveness classification accuracy criterion.

3.3 Information analysis of attributes providing for class

In dynamic Bayesian derivative classifiers, attributes can provide three kinds of dependency information for class, namely transitive dependency information, directly induced dependency information and indirectly induced dependency information [20, 39]. The attributes of a dynamic naïve Bayesian classifier can only provide transitive dependency information, but the attributes of a dynamic full Bayesian classifier (DFBC) can provide all three kinds of dependency information, hence the DFBCs have better performance in relation to classification accuracy. Figure 6 shows the way in which the different dependency information that attribute variables providing for class variable among dynamic Bayesian derivative classifiers with different network structures.

In the dynamic naïve Bayesian classifier that is shown in Fig. 6(a), the attributes only provide transitive dependency information for class, although this kind of information is the primary one for classification, the other two kinds of information cannot be ignored. In the dynamic tree Bayesian classifier shown in Fig. 6(b), apart from transitive dependency information, $X_1[t]$ and $X_2[t]$ also provide direct induced dependency information for $C[t]$. In the dynamic full Bayesian

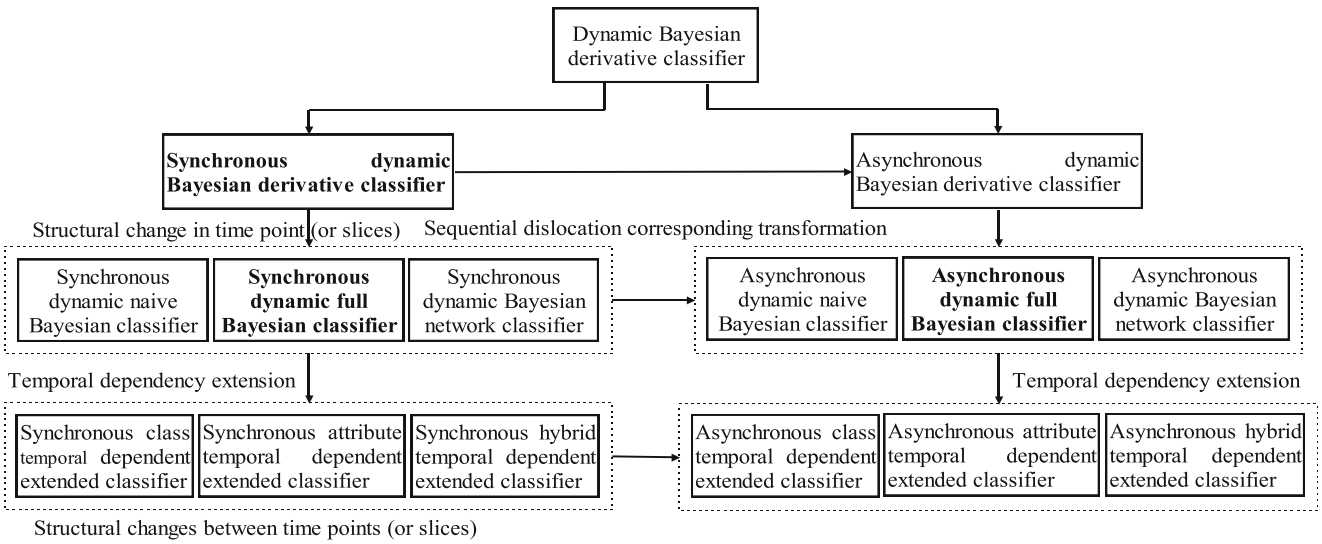


Fig. 4 The architecture of the DBDCs

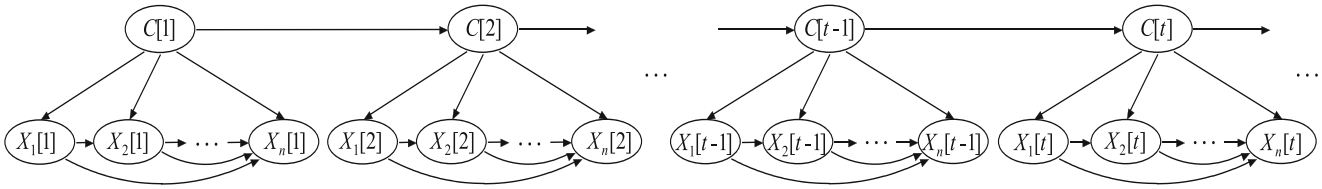


Fig. 5 The structure of the DFBC

classifier (DFBC) shown in Fig. 6(c), $C[t-1]$ and $X_5[t]$ provide only transitive dependency information for $C[t]$, whereas $X_1[t]$, $X_2[t]$, $X_3[t]$ and $X_4[t]$ provide all the three kinds of dependency information.

3.4 Classification accuracy criterion of time-series progressiveness

For the time-series dataset $D[T]$, the threshold of initial prediction time point (or time slice) T_0 can be determined according to the time-series size, class probability validity, and attribute density estimation, or actual needs. We use $D[t-1]$ as the training set, $x_1[t]$, …, $x_n[t]$ as the input and $\boldsymbol{\rho} = (\rho_1, \cdots, \rho_n)$ as the smoothing parameter vector, and denote the classification accuracy, classification result, and true numerical result of $c[t]$ in DFBC as $accuracy(\boldsymbol{\rho}, D[T], T_0)$, $c_{prediction}[t]$, and $c_{true}[t]$, respectively. Then we have:

$$accuracy(dfbc, \boldsymbol{\rho}, D[T], T_0) = \frac{1}{T-T_0+1}\sum_{t=T_0}^{T} signb\left(c_{prediction}[t], c_{true}[t]\right) \tag{9}$$

where

$$signb\left(c_{prediction}[t], c_{true}[t]\right) = \begin{cases} 1, c_{prediction}[t] = c_{true}[t] \\ 0, c_{prediction}[t] \neq c_{true}[t] \end{cases}.$$

3.5 Optimization of smoothing parameters

When dealing with time-series data, since a close relationship exists between the same class variable over different time points (or time slices), the classifier that can accurately classify the most adjacent class values in time sequences should be the most reliable. Inspired by this fact, the smoothing parameter configuration tree (or forest) is established over adjacent time points (or slices), and the smoothing parameters are optimized based on the configured tree (or forest).

The smoothing parameters that shape the curve (or surface) of a Gaussian function will directly affect the performance of a classifier. The smaller the value of the smoothing parameter and the steeper the density curve is, the better fit between classifier and data is achieved, although the generalization ability will appear worse. To trade-off between the fit and generalization ability of the classifier, we construct a smoothing parameter configuration tree (or forest) and use it to optimize the smoothing parameter configuration. The depth-first search method is adopted to build the smoothing parameter configuration tree (or forest), and the optimal synchronous change parameter is used to initialize all smoothing parameters. We take the latest class value as the starting point to search under the cumulative classification accuracy criterion of time-series progressiveness. If the cumulative classification $accuracy(dfbc, \boldsymbol{\rho}, D[T], T_0) = 1$, then $\rho^* = argmax_{\rho}\{accuracy(dfbc, \boldsymbol{\rho}, D[T], T_0) = 1\}$, that is, the generalization ability of the classifier is improved by taking the maximum value of smoothing parameter configuration; otherwise, the branch search ends.

When it is necessary to search space for the smoothing parameters, the final set-up construction is a smoothing parameter configuration tree (or forest) with or without repeated searches, which depends on whether, or not, the search space includes the smoothing parameters that have assigned values. Although more search space is needed to build a smoothing parameter configuration tree (or forest) with repeated searches, the results gained by repeated searches present a better investment. By repeated search space experiments, we found that the interval of smoothing parameters with the greatest influence on the DFBC classification accuracy is (0, 1].

We use $H = \{\rho^1, \rho^2, \ldots, \rho^L\}$ to denote the set of values for each smoothing parameter and $\rho_i^j (1 \le i \le j, 1 \le j \le L)$ to denote the jth value of the smoothing parameter ρ_i of the attribute X_i, where L denotes the numbers of values obtained by discretizing each smoothing parameter with interval (0, 1] in the step (step size 0.001) In the following, we give the algorithm to construct a smoothing parameter configuration tree by combining classification accuracy criterion of time-series progressiveness and non-repeated search.

Algorithm 1. Algorithm for constructing smoothing parameter configuration tree

Input: Time-series dataset $D[T]$, the threshold T_0 of smoothing parameter initialization and the smoothing parameter value set H
Output: A smoothing parameter configuration tree
Calculate $\rho^* = \underset{\rho=\rho_1=,\cdots,\rho_n\in H}{argmax}\{accuracy(dfbc, \boldsymbol{\rho}, D[T], T_0\}$ // Find initial smoothing parameter
smoothing_parameter$[] = \rho^*$, *search*$[] = 0$ // Initialize smoothing parameter vector and smoothing parameter search array
for $t = T$ to 1
  if $accuracy(dfbc$,smoothing_parameter$[], D[T], t) \le 1$ **then**
    store $T^* = t$
    exit for
  end if
end for
Create a node N (root node)，store T^* and ρ^* into the data domain of node N, $T' = T^*$
for $t = T'$ to 1 // Set up a smoothing parameter configuration tree by hierarchies
  Traverse the tree in depth-first
  if the terminal node of a path is a non-leaf node **then**
    According to the path from the root node to the terminal node, configure smoothing parameter vector *smoothing_parameter*[] and smoothing parameter search *array search*[] once again, set the path terminal node to the current node, $sign = 0$
    for $i = 1$ to n
      if *search*$[i] = 0$ then //The smoothing parameter ρ_i of attribute X_j has not been optimized
        for $j = L$ to 1
          if $accuracy(dfbc, (\ldots, \rho_i^j, \ldots), D[T], t) = 1$ **then** //Set *smoothing_parameter*$[i] = \rho_i^j$
            Update smoothing parameter vector *smoothing_parameter*[], *search*$[i] = 1$, $sign = 1$, $mark = 0$
            for $u = t$ to 1 // Found critical value T''
              if $accuracy(dfbc$,smoothing_parameter$[], D[T], t) < 1$ **then**
                Store$T'' = t$, create a new node N, store T'' and ρ_i^j into the data domain of node N, $mark = 1$
                exit for
              end if
            end for
            if $mark = 0$ **then**
              Set the newly created node as a leaf node
            end if
          end if
        end for
      end if
    end for
    if $sign = 0$ **then**
      Set the current node as a leaf node
    end if
  end if
end for

L is independent of n or T. Based on the classification accuracy estimation equation, the time complexity of the algorithm constructing a smoothing parameter configuration tree (or forest) is $O(nT^2)$, and that of the algorithm calculating a Gaussian function is $O(n^2T^3)$.

The smoothing parameter configuration tree (or forest) formed by repeated search will be more complex than that of a non-repeated search. A new configuration tree can be derived by resetting the smoothing parameter in an existing configuration tree. Therefore, the final smoothing parameter

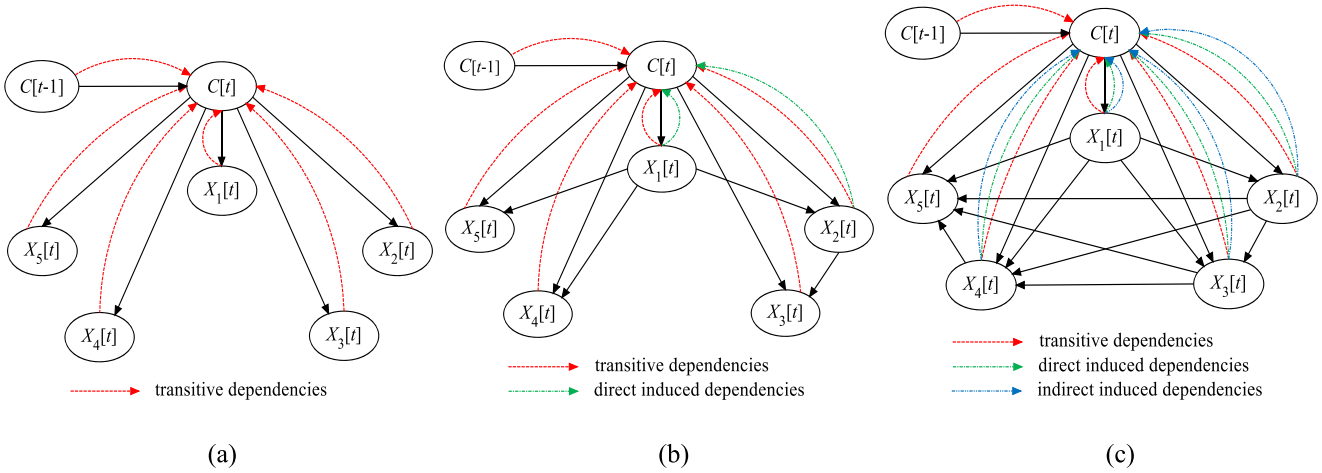


Fig. 6 The dependency information provided by attributes in dynamic naïve, dynamic tree and dynamic full Bayesian classifiers. **a** dynamic naïve Bayesian classifier, **b** dynamic tree Bayesian classifier, **c** dynamic full Bayesian classifier

configuration formed by repeated search is commonly denoted a forest. When constructing a smoothing parameter configuration tree (or forest) by repeated search, we avoid the possible looping situation (although the possibility is small) by limiting the number of generated subtrees. Additionally, by limiting the number of generated subtrees, the time complexity of the algorithm constructing the smoothing parameter configuration tree (or forest) based on repeated search will be the same as that based on a non-repeated search. We can also improve the smoothing parameter configuration tree (or forest) by using a repeated search and a non-repeated search strategy and adjust the classification accuracy criterion from completely cumulative accurate into partially accurate, which forms the basis for future work that we will do.

3.6 Ensemble DFBC by model averaging

Let $\boldsymbol{\rho}^1, \cdots, \boldsymbol{\rho}^U$ denote the smoothing parameter configuration vector with minimum threshold and $DFBC_u$ denote the corresponding dynamic full Bayesian classifier with the parameter configuration $\boldsymbol{\rho}^u (1 \le u \le U)$. Dynamic full Bayesian ensemble classifier (DFBEC) is established by averaging these classifiers, the structure of DFBEC is shown in Fig. 7.

DFBEC can be expressed as:

$$\underset{c[t](c[t-1],x_1[t],\ldots,x_n[t])}{argmax} \frac{1}{U}\sum_{u=1}^{U}\left\{\frac{N(c[t],c[t-1])}{(2\pi)^{n/2}N(c[t])N(c[t-1])\left(\rho_1^u\right)^2\cdots\left(\rho_n^u\right)^2}\sum_{v=1}^{t}\left\{signa(c[v])\prod_{i=1}^{n}exp\left[-\frac{(x_i[t]-x_i[v])^2}{2(\rho_i^u)^2}\right]\right\}\right\} \tag{10}$$

where ρ_i^u is the ith component of $\boldsymbol{\rho}^u$.

4 Experiment and analysis

We use time-series datasets from the UCI [40] Machine Learning Repository and the Wind Economic Database for experiments. The indexes are selected as class variables, and the related factors that affect these indexes are selected as attribute variables. Each index $g(t)$ is binary discretized over the time-series and fall into the class labeled $c[t]$: if $g(t)$ reaches an extremum at t_j, then $c[t] = 1$, and t_j is the turning point or extreme point; or else $c[t] = 0$, and t_j is the non-turning or non-extreme point. We repair records for the missing data using a moving average method, discretizing the class variables over the time-series according to the turning points and normalizing the attribute variables over the time-series. The experiment is comprised of five experimental modules: (1) the construction of a smoothing parameter configuration tree (or forest); (2) the comparison of classification accuracy (or error rate); (3) the influence of smoothing parameter changes on classification accuracy; (4) the influences of the minimum and maximum

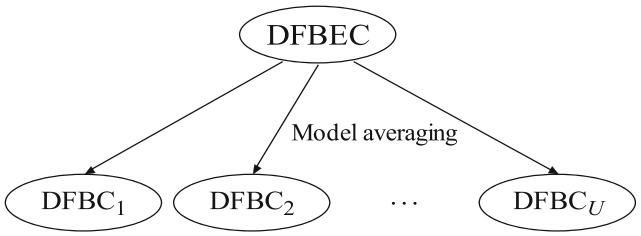


Fig. 7 The structure of DFBEC

smoothing parameter configurations on classification accuracy and (5) the anti-disturbance of DFBEC.

4.1 Construction of smoothing parameter configuration tree and forest

We choose the GDP annual data to create the smoothing parameter configuration tree (or forest). Setting the threshold for the initial perdition time point or time slice (threshold for short) $T_0 = 15$ and $H = \{0.002 + 0.002k | 0 \le k \le 500\}$, we obtain the initial configuration of smoothing parameters $\rho_0 = 1$, and $T_0 = 23$. Next, we present the smoothing parameter configuration tree with non-repeated searches and repeated searches, respectively.

(a) Construction of smoothing parameter configuration tree with non-repeated searches

The smoothing parameter configuration tree of GDP on one path with non-repeated searches is shown in Fig. 8.

In the smoothing parameter configuration tree, each path from the root to the leaf corresponds to a configuration vector. We form a DFBC from each smoothing parameter configuration vector. The minimum threshold $T_0 = 17$ is found by traversing the smoothing parameter configuration tree, and the corresponding configuration vectors are as follows:

$(1, 1, 1, 1, 0.036, 1, 0.016, 1, 0.088, 1, 1, 1, 0.09, 1, 1, 1, 1)$

$(0.132, 1, 1, 1, 0.036, 1, 0.016, 1, 1, 1, 1, 1, 0.052, 1, 1, 1, 1)$

$(1, 1, 1, 1, 0.036, 1, 0.016, 0.03, 1, 1, 1, 1, 0.052, 1, 1, 1, 1)$

$(1, 1, 1, 1, 0.022, 1, 1, 0.01, 0.036, 1, 1, 1, 1, 1, 1, 1, 1)$

We can obtain 4 DFBCs from the above 4 smoothing parameter configuration vectors, and the DFBEC is formed by averaging these classifiers. The restriction of the minimum threshold can also be extended. If we select $T_0 = 19$ as the threshold, we will obtain 13 smoothing parameter configuration vectors and the corresponding 13 DFBCs.

(b) Construction of smoothing parameter configuration forest with repeated searches

When we repeat the search space for a smoothing parameter, we can obtain a smoothing parameter configuration forest. One of the configuration trees of the forest is shown in Fig. 9.

Where the nodes with "*" will produce the new trees. The tree derived by $(*)\rho_7 = 0.076(6)$ is shown in Fig. 10(a). The trees derived by $(*)\rho_{14} = 1(61)$ and $(*)\rho_{16} = 0.034(62)$ from Fig. 10(a) are shown in Fig. 10(b) and 10(c), respectively.

We find the minimum threshold $T_0 = 12$ by traversing the smoothing parameter configuration forest, and the corresponding configuration vectors are as follows:

$(0.136, 1, 0.064, 1, 0.036, 1, 0.076, 1, 1, 1, 1, 1, 1, 0.056, 0.034, 1)$

$(0.116, 1, 1, 1, 0.036, 1, 0.052, 1, 0.246, 1, 1, 1, 0.254, 1, 0.072, 1, 1)$

$(0.116, 1, 1, 1, 0.036, 1, 1, 1, 0.08, 1, 1, 1, 1, 1, 0.048, 1, 1)$

$(0.116, 1, 1, 1, 0.036, 1, 0.052, 1, 0.142, 1, 1, 1, 1, 1, 0.072, 0.032, 1)$

The restriction of the minimum threshold can also be extended. If we select 19, 18, 17, 16, 15, 14 and 13 as the thresholds T_0, we will obtain 26, 19, 15, 12, 11, 9 and 6 smoothing parameter configuration vectors and DFBCs, respectively.

4.2 Comparison of classification accuracy

We choose 9 commonly used classifiers together with DFBEC classifier for comparing classification accuracy. 21 time-series datasets from the UCI and 24 time-series datasets from the Wind database are selected for experiments. Firstly, we select a dataset of 30 in front of the time-series dataset to establish the smoothing parameter configuration tree and determine the parameter configuration. Then, we take the latest 113 (or 103) data

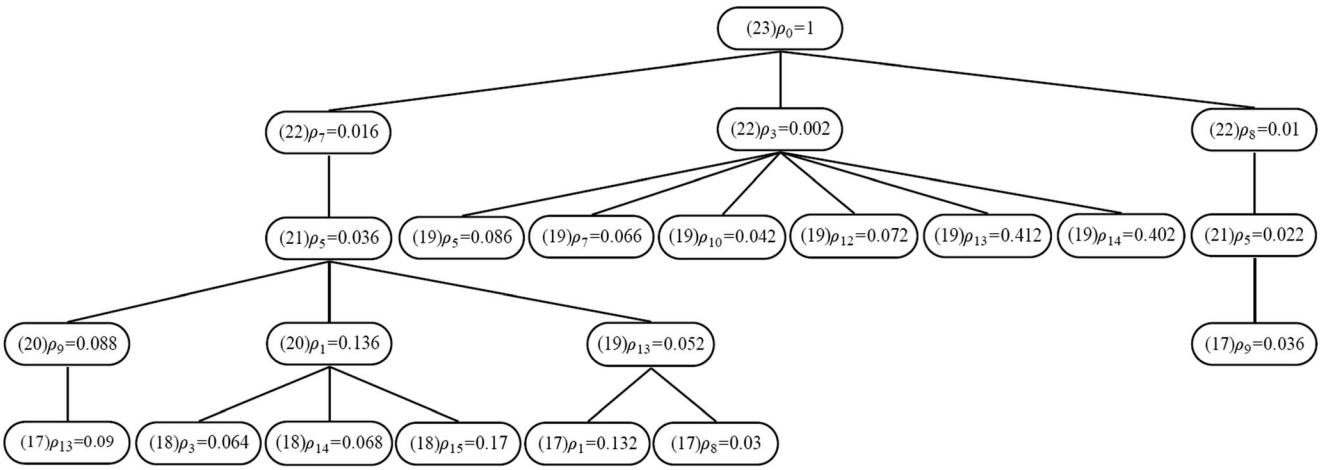


Fig. 8 The smoothing parameter configuration tree of GDP

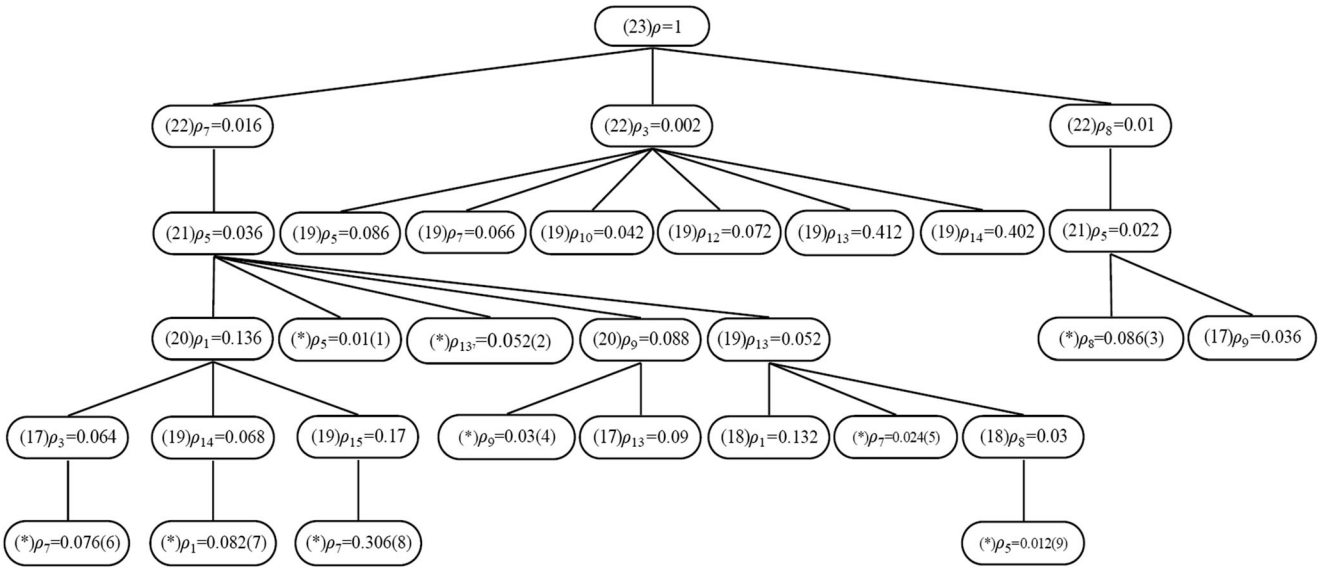


Fig. 9 The smoothing parameter configuration tree of GDP with repeated searches

in the time-series dataset as the testing set to carry out a sliding test, with a fixed window size (training set size) of 30.

Descriptions of the comparison classifiers are as follows:

- *GDNBC:dynamic naïve Bayesian classifier based on Gaussian density* [33]
- *KDNBC:dynamic naïve Bayesian classifier based on Gaussian kernel density* [6, 33]
- *FMDBN:a first-order Markov dynamic* Bayesian network classifier *based on Gaussian density* [6]
- *KDTBC:dynamic tree Bayesian classifier based on Gaussian kernel density* [6, 33]
- *GDOBC:dynamic One-dependence* [17] *Bayesian classifier based on Gaussian density* [33]
- *KDOBC:dynamic One-dependence* [17] *Bayesian classifier based on Gaussian kernel density* [6, 33]
- RNN: Recurrent Neural Network [41]
- LSTM: Long Short-Term Memory [42]
- GRU: Gated Recurrent Unit [43]
- DFBEC: Dynamic Full Bayesian Ensemble Classifier

Among these, the parameter configurations for RNN, LSTM and GRU are: (a) RNN, 1 hidden layer, units = 32, active_function = 'tanh', loss = 'mean_squared_error', optimizer = '

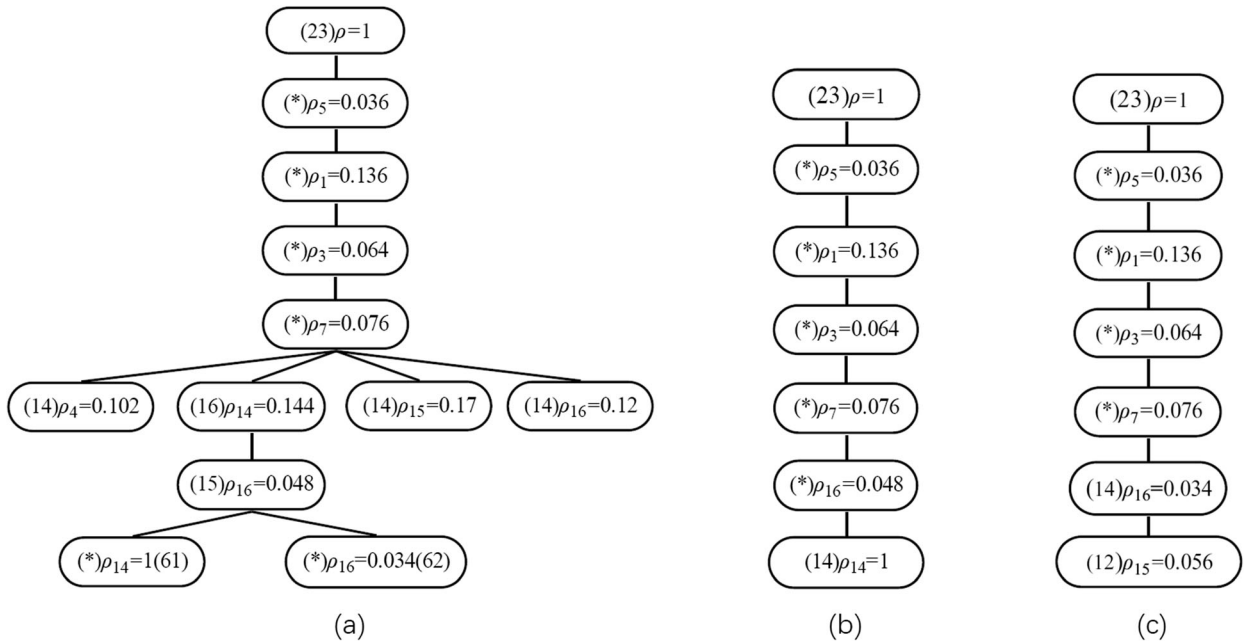


Fig. 10 Derivative trees from (*)$h_7 = 0.076(6)$. **a** Derivative tree of No. 6, **b**. Derivative tree of No. 61, **c** Derivative tree of No. 62

rmsprop', metrics = ['accuracy']; (b) LSTM: 1 hidden layer, units = 32, active_function = 'relu', loss = 'mean_squared_error', optimizer = 'adam', metrics = ['accuracy']; (c) GRU: 1 hidden layer, units = 32, active_function = 'tanh', loss = 'nn.CrossEntropyLoss', optimizer = 'optim.SGD', metrics = ['accuracy'].

A statistical comparison of the classifiers' performances over error rate (error rate = 1-accuracy) is conducted by taking the Friedman Test with the post-hoc Bonferroni-Dunn method and Wilcoxon Signed-Ranks Test [44]. The differences amongst the 10 classifiers' performances are first examined via the Friedman Test, and the pairwise comparisons are then realized by the post-hoc Bonferroni-Dunn method, showing the critical value at a significance level of 0.05 to be 2.773. If the difference between average ranks for pairwise classifiers is greater than the critical value, then we say the performance of the two classifiers is significantly different. We then use the Wilcoxon Signed-Ranks Test to examine the difference between each pairwise comparison (other classifier vs. DFBEC classifier). The test results listed at the bottom of Table 1 show significant differences in classification error between DFBEC and other classifiers. Considering the overall average classification performance, the classification accuracy of DFBEC is 15.81%, 12.95%, 9.47%, 14.13%, 15.33%, 18.62%, 12.62%, 18.46% and 13.79% larger than the other 9 classifiers by simple calculation.

The classification error rates of DFBEC and the other classifiers are shown in Fig. 11 as scatter-plot graphs, where the coordinates of each point represent the error rates of the two compared classifiers. The points above each diagonal line show that the error rates of DFBEC are smaller than those of the given classifier, and the points below the diagonal line show that the error rates of DFBEC are greater than those of the given classifier.

From Fig. 11, we can clearly see that the classification error rates of DFBEC are smaller than those of the other classifiers. Overall, the results of significance tests of difference, average value comparison and scatter diagrams show that DFBEC has significant advantages over the other 9 classifiers in terms of classification error rate.

Classification is one kind of human concept learning based on computer simulation. In human concept learning, to make full use of sample information (fitting data), rote learning (data overfitting) should be avoided as much as possible, and flexible learning (generalization ability) should be promoted when there are relatively few references. In our research, multivariate Gaussian kernel function is used to estimate attribute joint density, and the smoothing parameters are optimized based on the configured tree (or forest), so that DFBEC can fit data well. In addition, on the premise of maintaining the accuracy of classification, we reduce the overfitting by taking the maximum value of the smoothing parameter configuration, and thus improve the generalization ability of DFBEC.

4.3 Influence of smoothing parameter changes on classification accuracy

The smoothing parameters determine the shape of the Gaussian function curve, so any change to the smoothing parameter will affect the fit accuracy of the classifier to the data. We choose 6 time-series datasets for experiments: the Dow_jones_index (DJI), Drink_glass_model_1 (DGM), Energydata_complete (EDC), ME_BTSC3 (MEB), Gold_ICP (GIC) and Fund_MTUNV (FMA). We use $s1$, …, $s9$, $s10$, …, $s18$, $s19$, …, $s27$, $s28$ forρ when they are 0.001, …, 0.009,0.01, …, 0.09,0, 1, …, 0.9,1, correspondingly. The experiments and analysis of that the influence of smoothing parameter changes on classification accuracy are carried out under both temporal synchronous and asynchronous situations. The influence of smoothing parameter changes on classification accuracy is shown in Fig. 12, where the horizontal axis represents the value of smoothing parameter, and the vertical axis represents the classification accuracy. To reduce the influence of initial configuration on smoothing parameter changes, in the asynchronous change situation, the initial values of the smoothing parameters are set to 1 for the 6 time-series datasets. In the asynchronous change situation, the smoothing parameters ρ_{11}, ρ_7, ρ_{20}, ρ_6, ρ_3 and ρ_{19} are selected for the 6 indexes according to the sequence of the datasets, respectively.

In Fig. 12, we find that the smoothing parameter changes on the 6 time-series datasets all have significant influence on DFBECs' classification accuracies in both temporal synchronous and asynchronous situations. In the synchronous changes, the maximum classification accuracy differences are 23.01%, 29.20%, 25.66%, 33.63%, 23.01% and 12.39% (with an average of 24.48%); in the asynchronous changes, the maximum classification accuracy differences are 41.60%, 5.31%, 21.24%, 24.78%, 21.24% and 7.96% (with an average of 20.35%). The significance of the differences illustrates both that the smoothing parameters need to be optimized and the benefit of doing so.

4.4 Influence of the minimum and maximum smoothing parameter configurations on classification accuracy

In this part, we discuss the influence of the minimum and maximum smoothing parameter configurations on DFBECs' classification accuracy. These experiments and analyses are carried out based on the 45 datasets from the UCI and the Wind database and are discussed under both the temporal synchronous and asynchronous situations. The experimental results are shown in Fig. 13, where the horizontal axis represents the number of datasets, and the vertical axis represents the classification accuracy.

Table 1 Classification error rate of 10 classifiers on 45 time-series datasets

Dataset	GDNBC	KDNBC	FMDBN	KDTBC	GDOBC	KDOBC	RNN	LSTM	GRU	DFBEC
1. Absenteeism_at_work	**0.3629**	0.3983	0.3717	0.3717	0.3806	0.4514	0.4336	0.4159	0.4513	0.3732
2. Boy_building_buy	0.4337	0.4425	0.4071	0.4160	0.3983	0.4337	0.3982	0.4779	0.4690	**0.3540**
3. Brush_teeth_1	0.5133	0.4956	0.3806	0.3894	0.4868	0.4691	**0.3451**	0.4779	0.3717	0.3629
4. Change_mind_cold_come	0.4956	0.4602	0.4248	0.4956	0.4425	0.5133	0.4514	0.4956	0.5044	**0.3806**
5. Climb_stairs_1	0.4425	0.4514	0.3982	0.4514	0.4602	0.4779	0.4956	0.4867	0.4956	**0.3098**
6. CMHS_CE_1	0.4337	0.4248	**0.3540**	0.5399	0.4337	0.5399	0.5133	0.5310	0.5398	0.3629
7. CMHS_CE_2	0.3806	0.3540	0.5487	0.5576	**0.3275**	0.5576	0.5398	0.5221	0.5221	0.3629
8. CMHS_CP_1	0.4602	0.4425	0.4514	0.5399	0.4691	0.5399	0.5487	0.5752	0.5487	**0.3983**
9. Computer_cost_crazy	**0.3186**	0.3894	0.4425	0.3806	0.3717	0.3452	0.3982	0.4425	0.4159	0.3629
10. Dow_jones_index	0.2478	0.2655	0.2213	0.3629	0.2124	0.4337	0.2832	0.4602	0.3097	**0.1859**
11. Draw_drink_eat	0.4602	0.3983	0.4868	0.4602	0.3717	0.5222	0.5398	0.5133	0.5221	**0.3009**
12. Drink_glass_model_1	0.4160	0.4779	0.3363	0.3098	0.4071	0.3098	0.3097	0.3097	0.3097	**0.2743**
13. Drink_glass_model_2	0.5045	0.5133	0.3983	0.3894	0.5487	0.3629	0.3717	0.3805	0.3628	**0.3540**
14. EEG_steady_state_A001SB1	0.3540	0.4514	0.4160	0.5310	0.4602	0.5310	0.5221	0.4956	0.5133	**0.3452**
15. Energydata_complete	0.5310	0.3363	0.5044	0.5664	0.5310	0.4602	0.3363	0.5133	0.3363	**0.2567**
16. Exit_flash_light_forget	0.4867	0.4248	0.4337	0.4425	0.4956	0.5399	0.4867	0.6018	0.4867	**0.3806**
17. Girl_give_glove	0.4337	0.4691	0.4425	0.4248	0.4337	0.4514	0.4779	0.4867	0.4867	**0.3629**
18. GSAFM_1	**0.3009**	0.3186	0.3629	0.3275	0.4160	0.3717	0.3363	0.3629	0.3540	0.3452
19. head_hear_hello	0.4868	0.4779	0.4779	0.4691	0.4691	0.4868	0.4956	0.5221	0.5133	**0.4248**
20. her_hot_how	0.5399	0.4779	0.4513	0.4248	0.5222	0.4160	0.4602	0.4425	0.4513	**0.4071**
21. Istanbul_stock_exchange	0.4071	0.4779	0.4071	0.4160	0.4248	0.4779	0.3274	0.4513	**0.3186**	0.3806
22. ME_Oil_price	0.4956	0.4779	**0.4071**	0.4602	0.5133	0.4602	0.4336	0.4248	0.4248	0.4160
23. ME_BTSC1	0.3275	0.3363	0.2567	0.3009	0.3186	0.3363	0.3009	0.3186	0.3186	**0.1770**
24. ME_BTSC2	0.4514	0.3452	0.4514	0.3363	0.4248	0.4071	0.3628	0.3982	0.3540	**0.2566**
25. ME_BTSC3	0.5664	0.3363	0.3399	0.2390	0.4425	0.3098	0.2743	0.3186	0.2566	**0.2301**
26. ME_NMPMI	0.5576	0.4514	0.4757	0.4425	0.5930	0.4868	0.4690	0.4602	0.4602	**0.4159**
27. ME_MPMI	0.3894	0.4425	**0.3787**	0.4602	0.4160	0.4779	0.5575	0.5221	0.5664	0.4248
28. ME_Prosperity_index	0.4868	0.4691	0.4248	0.3983	0.5133	0.3717	**0.3274**	0.3628	0.3363	0.3451
29. ME_Global_indicators	0.4956	0.4956	0.4514	0.5222	0.5222	0.5310	0.5841	0.5044	0.5575	**0.4159**
30. StockEOP	0.4779	0.5045	0.4425	0.4248	0.5222	0.4513	0.4513	0.4602	0.4513	**0.4071**
31. Stock_EHP	0.5222	0.5310	0.5044	0.4868	0.4691	0.5664	0.5221	0.5310	0.5310	**0.3983**
32. Stock_ECP	0.4868	0.4691	0.4691	0.4956	0.5222	0.4868	0.5044	0.4956	0.4779	**0.4248**
33. Stock_MMP	0.4690	0.4868	0.4336	**0.4248**	0.5222	**0.4248**	0.4690	**0.4248**	0.4336	**0.4248**
34. Fund _BDGR	0.4425	0.4602	0.3981	0.4779	0.5045	0.4690	0.4425	0.4867	0.4690	**0.3363**
35. Fund_CDAY	0.4272	**0.3690**	0.4159	0.4272	0.4564	0.4758	0.4369	0.4563	0.4563	0.4078
36. Fund_MTUNV	0.5310	0.5222	0.2832	0.4337	0.2832	0.4956	0.4159	0.4336	0.4248	**0.2719**
37. Futures_D_OP	0.4956	0.5222	0.4956	0.5045	0.5222	0.5045	0.4602	0.4779	0.5044	**0.4337**
38. Futures_PM_MP	0.4779	0.4602	0.4159	0.4159	0.4337	**0.4071**	0.4425	0.4425	0.4248	0.4159
39. Gold_MCP	0.3363	0.3806	0.3186	0.3186	0.3806	0.3540	**0.2478**	0.3097	0.2655	0.3009
40. Gold_ICP	0.2921	0.2567	0.3274	0.4779	0.3452	0.5310	0.2124	0.4336	**0.1770**	0.3107
41. ER_CAD_MP	0.4868	0.4691	0.4159	0.4425	0.4868	0.4425	0.4425	0.4425	0.4336	**0.4071**
42. ER_ARGENTINA	0.4078	0.3981	0.4248	0.4467	**0.3884**	0.4369	0.4369	0.4369	0.4757	0.3981
43. ER_AUSTRALIA	0.3884	0.3496	0.3593	0.4369	**0.3399**	0.4952	0.4369	0.4466	0.4466	0.3496
44. ER_BRAZIL	0.4467	0.3981	0.4757	0.4272	0.4272	0.4467	0.4757	0.4369	0.4660	**0.3496**
45. ER_CANADA	0.4272	0.3787	0.3593	0.4564	0.3787	0.4272	0.4078	0.4660	0.4563	**0.3204**
Average	0.4421	0.4280	0.4098	0.4339	0.4398	0.4553	0.4263	0.4546	0.4322	**0.3539**
Average Rank (Friedman Test)	6.0111	5.6111	4.4333	5.4667	6.2556	7.0111	5.5889	6.8222	6.0333	1.7667
p-value (Wilcoxon SR Test)	7.71E-08	7.19E-08	7.91E-08	1.69E-08	2.14E-08	1.07E-08	6.40E-07	7.59E-09	3.43E-07	–

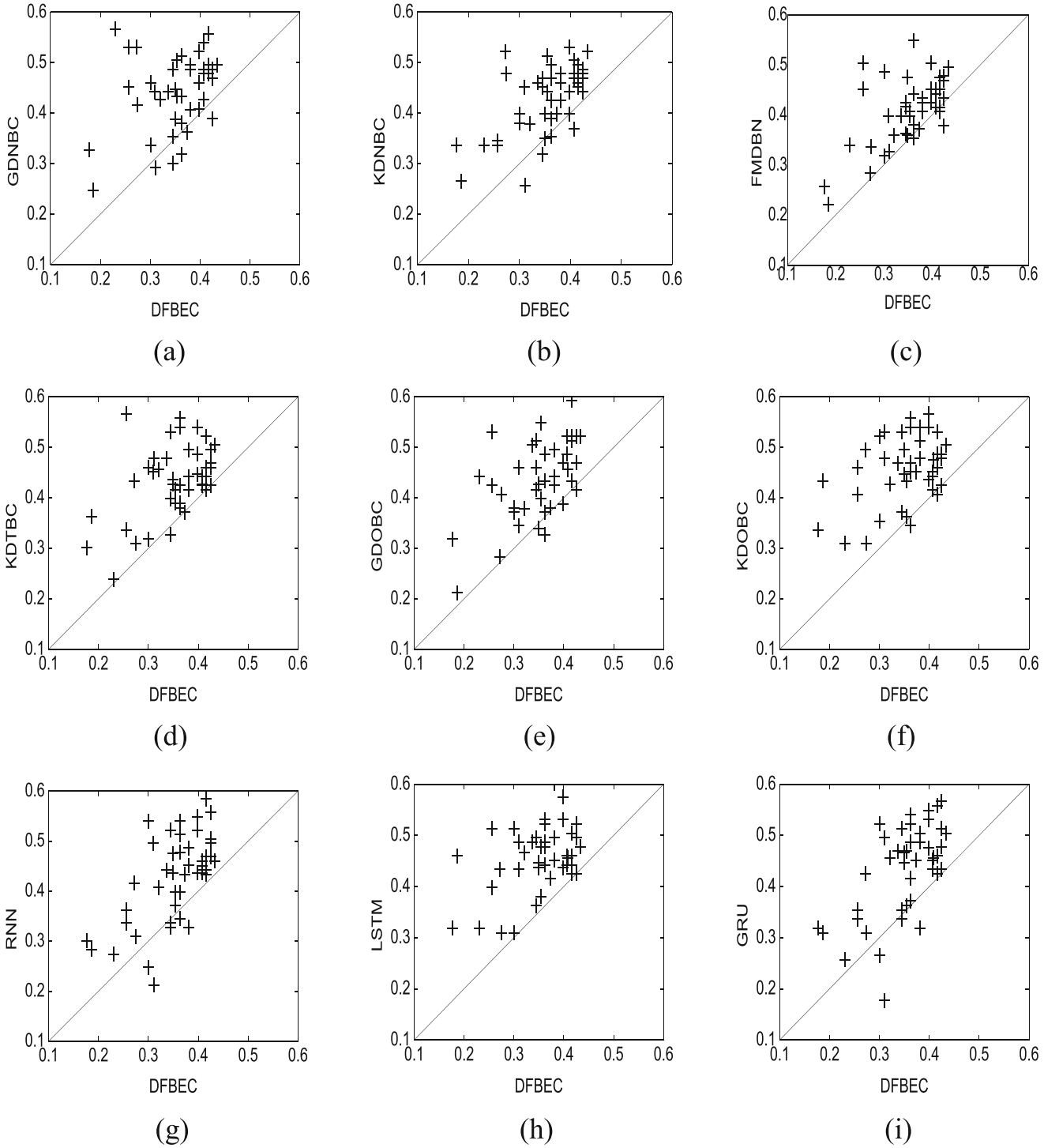
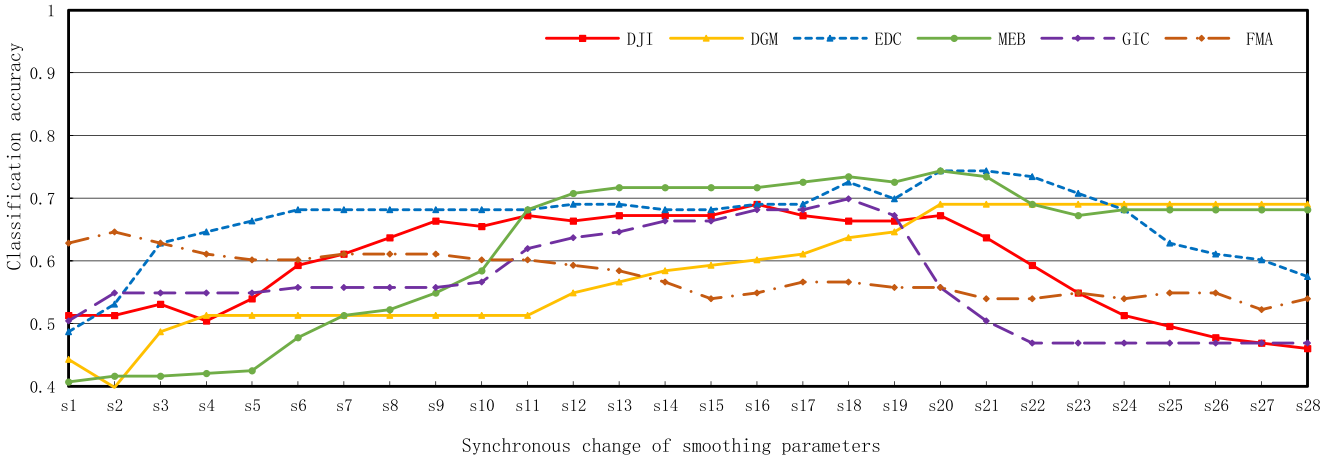
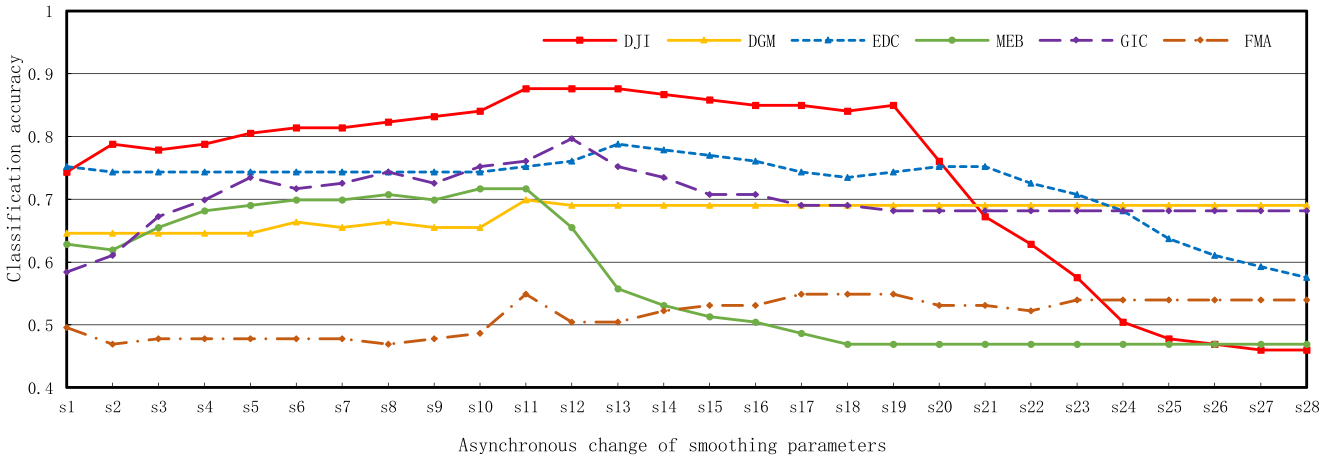


Fig. 11 Scatter plot graphs of classification error rates. **a** DFBEC vs GDNBC, **b** DFBEC vs KDNBC, **c** DFBEC vs FMDBN, **d** DFBEC vs KDTBC, **e** DFBEC vs GDOBC, **f** DFBEC vs KDOBC, **g** DFBEC vs RNN, **h** DFBEC vs LSTM, **i** DFBEC vs GRU

From both Fig. 13 (a) and (b), we can see that the classifiers under the maximum smoothing parameter configurations are more accurate than those under the minimum configurations in both synchronous changes and asynchronous changes, but especially for the asynchronous series. Among the 45 classification problems, in synchronous optimization, the classification accuracies for 23 out of 45 problems under the maximum configuration is greater than that under the minimum configuration, and the accuracies occur in 15 of the same problems. In asynchronous optimization, the classification accuracy of 39 problems under the maximum configuration is greater than that under the minimum configuration, and the accuracies occur in 3 of the



(a)



(b)

Fig. 12 The influence of smoothing parameter changes on classification accuracy. **a** The influence of temporal synchronous changes, **b** The influence of temporal asynchronous changes

same problems. The comparison results are shown in Table 2. For synchronous optimization and asynchronous optimization, the differences between the average classification accuracy of the maximum configuration and the minimum configuration are 1.78% and 4.84%, respectively.

We can conclude that, on the premise of maintaining the classification accuracy of the classifier, data overfitting can be reduced by selecting the maximum configuration of the smoothing parameter. This is especially important for the classification of small sample time-series data.

4.5 Anti-disturbance of DFBEC

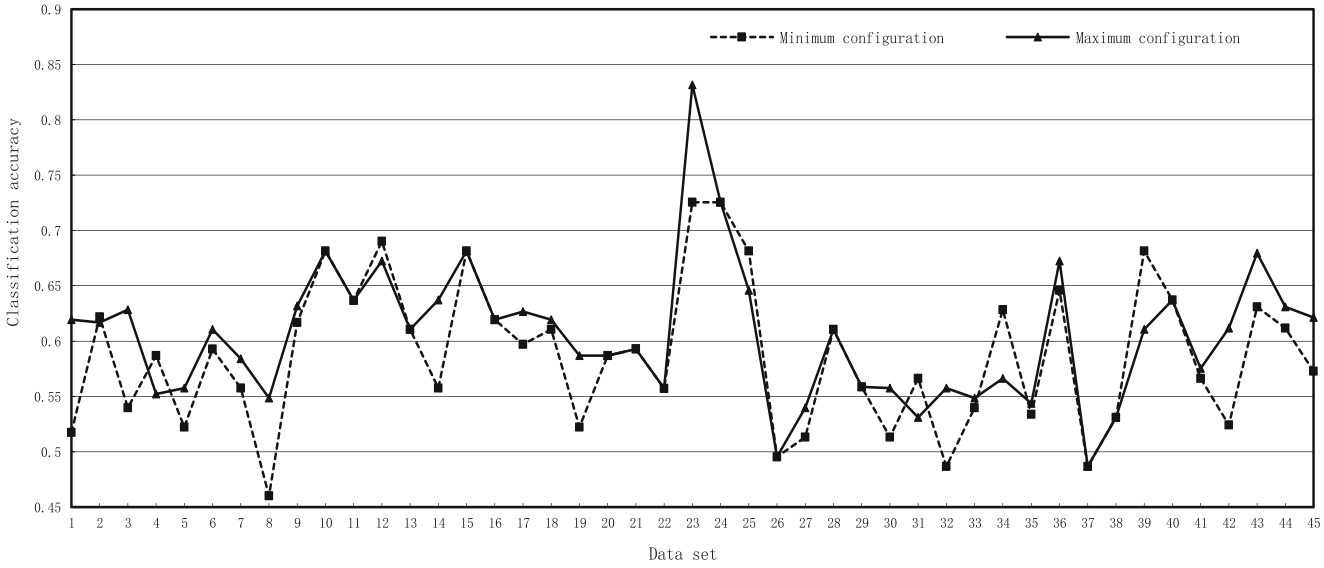
Additionally, the anti-disturbance ability of DFBEC based on asynchronous optimization is analyzed by using the 45 time-series datasets, under the disturbance of that 20% of the attribute data are randomly changed within the range of the attribute value. The experimental results are shown in Fig. 14.

After introducing noise into the attribute data, the classification accuracy for some time-series datasets decrease slightly, with an average decline of 0.012, which shows that DFBEC with the maximum smoothing parameter configuration has good anti-disturbance performance based on asynchronous optimization, and a similar conclusion can be drawn based on synchronous optimization.

5 Conclusions and future work

In this paper, we develop the DFBEC which is suitable for small sample time-series data by combining estimation of the multivariate Gaussian kernel function with a diagonal smoothing parameter matrix; the classification accuracy criterion of time-series progressiveness; the construction of smoothing parameter configuration tree (or forest) and classifier selection and averaging.

The smoothing parameters that shape the curve (or surface) of a Gaussian function have direct impact on the performance of a



(a)

(b)

Fig. 13 The influence of the minimum and maximum smoothing parameter configurations on classification accuracy. **a** The influence of synchronous optimizations, **b** The influence of asynchronous optimization

classifier. The smaller the value of the smoothing parameter and the steeper the density curve is, the better the fit between the classifier and data achieves, but the worse the generalization ability appears; and vice versa. To establish a small sample time-series data classifier, two problems must be solved: one is to make the classifier fit data well, and the other is to avoid data

Table 2 Comparison of classification accuracies under the maximum and minimum smoothing parameter configurations

	Problems under the maximum configuration vs. under the minimum configuration			
Comparison of classification accuracies	>	=	<	Total
In synchronous optimization	23	15	7	45
In asynchronous optimization	39	3	3	45
Total	62	18	10	90

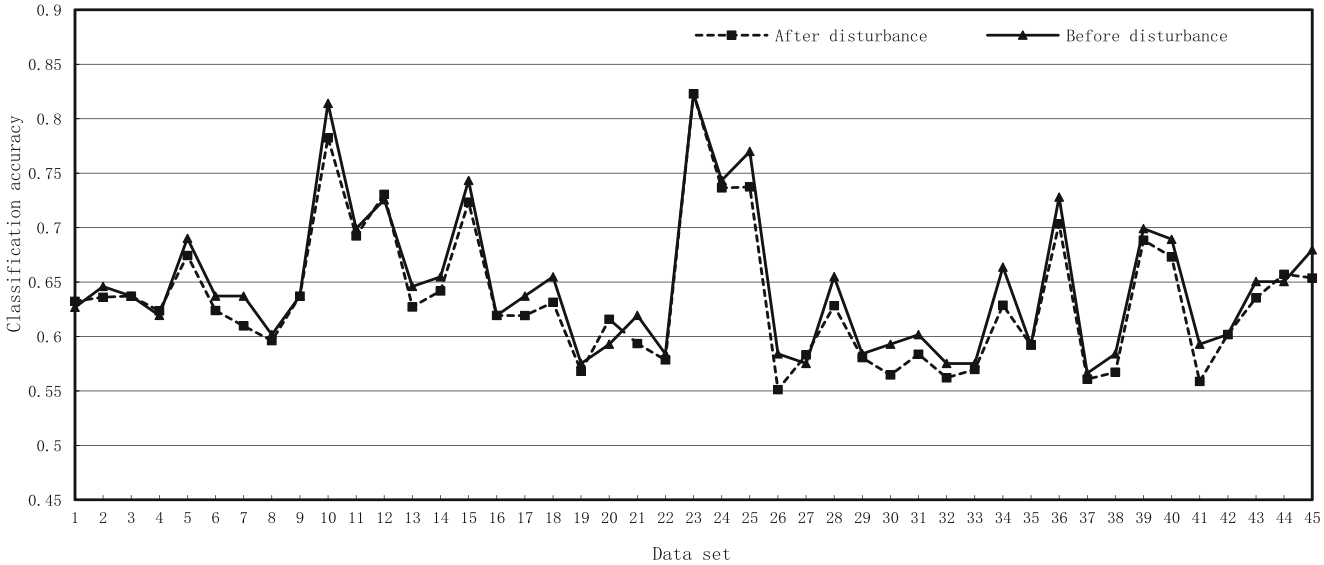


Fig. 14 The experimental results for anti-disturbance of DFBEC based on asynchronous optimization

overfitting. The latter is the more difficult problem to solve. In our research, multivariate Gaussian kernel function is used to estimate attribute joint density, and the smoothing parameters are optimized based on the configured tree (or forest), enabling the DFBEC to fit the data most advantageously. On the premise of maintaining the accuracy of classification, overfitting is reduced by taking the maximum value of the smoothing parameter configuration, thus improving the generalization ability of DFBEC. By combining classifier selection and averaging results, the generalization of DFBEC is further improved.

The experimental results based on the UCI and Wind datasets show that the DFBEC is more accurate with good anti-disturbance ability in the turning point classifications with small sample time-series data compared with the other nine commonly used classifiers. However, the DFBEC is limited in that it is only suitable for the classification of small time-series data. We propose further work to expand DFBEC by adjusting the classification accuracy criterion from completely cumulative accurate into partially accurate, and thus make it more suitable for general time-series data classification.

Acknowledgements This work is supported by the National Social Science Foundation of China [Grant number 18BTJ020]; the National Natural Science Foundation of China [Grant numbers 71771179, 72021002, 82072228]; the Foundation of National Key R&D Program of China [Grant number 2020YFC2008700]; and the Foundation of Shanghai 5G + Intelligent Medical Innovation Laboratory.

References

1. Vapnik V, Vapnik V, Vapnik VN (2003) Statistical learning theory. Ann Inst Stat Math 55(2):371–389
2. Chauvin Y, Rumelhart DE (1995) *Backpropagation: theory, structures, and applications.* L. Erlbaum Associates Inc
3. Breiman LJ, Friedman CJ, Olshen RA. *Classification and Regression Tree*. Chapman and Hall/CRC, Boca Raton, FL:1984
4. Hernandez-Matamoros A et al (2020) Forecasting of COVID19 per regions using ARIMA models and polynomial functions. *Applied Soft Computing* 96:106610
5. Hongsakulvasu N, Khiewngamdee C, Liammukda A (2020) Does COVID-19 crisis affects the spillover of oil Market's return and risk on Thailand's Sectoral stock return?: evidence from bivariate DCC GARCH-in-mean model. International Energy Journal 20(4):647–662
6. Wang S, Zhang S et al (2020) FMDBN: A first-order Markov dynamic Bayesian network classifier with continuous attributes. *Knowledge-Based Systems* 195:105638
7. Johansson A, Guillemette Y, Murtin F et al Looking to 2060: Long-term global growth prospects. Oecd Economic Policy Papers 8.4(2012):330–338
8. Naraidoo R, Paya I (2012) Forecasting monetary policy rules in South Africa. Int J Forecast 28(2):446–455
9. Duda RO, Hart PE. *Pattern classification and scene analysis*. Vol. 3. Wiley, New York: 1973
10. Chow C, Liu C (1968) Approximating discrete probability distributions with dependence trees. IEEE Trans Inf Theory 14(3):462–467
11. Friedman N, Geiger D, Goldszmidt M (1997) Bayesian network classifiers. Mach Learn 29(2–3):131–163
12. Domingos P, Pazzani M (1997) On the optimality of the simple Bayesian classifier under zero-one loss. Mach Learn 29(2–3): 103–130
13. de Campos CP, Corani G, Scanagatta M, Cuccu M, Zaffalon M (2016) Learning extended tree augmented naive structures. Int J Approx Reason 68:153–163
14. Cheng J, Greiner R (1999) "Comparing Bayesian network classifiers." *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence* Morgan Kaufmann Publishers Inc: 101–108
15. Petitjean F, Buntine W, Webb GI, Zaidi N (2018) Accurate parameter estimation for Bayesian network classifiers using hierarchical Dirichlet processes. Mach Learn 107(8–10):1303–1331
16. Yager RR (2006) An extension of the naive Bayesian classifier. Inf Sci 176(5):577–588
17. Webb GI, Boughton JR, Wang Z (2005) Not so naive Bayes: aggregating one-dependence estimators. Mach Learn 58(1):5–24

18. Flores MJ, Gámez JA, Martínez AM (2014) Domains of competence of the semi-naive Bayesian network classifiers. Inf Sci 260: 120–148
19. Berend D, Kontorovich A (2015) A finite sample analysis of the naive Bayes classifier. J Mach Learn Res 16(44):1519–1545
20. Wang SC, Xu GL, RuiJie D (2013) Restricted Bayesian classification networks. SCIENCE CHINA Inf Sci 56(7):1–15
21. Sathyaraj R, Prabu S (2018) A hybrid approach to improve the quality of software fault prediction using Naïve Bayes and k-NN classification algorithm with ensemble method. Int J Intell Syst Technol Appl 17(4):483
22. Yang Y, Ding M (2019) Decision function with probability feature weighting based on Bayesian network for multi-label classification. Neural Comput & Applic 31(9):4819–4828
23. Kuang L, Yan H, Zhu Y, Tu S, Fan X (2019) Predicting duration of traffic accidents based on cost-sensitive Bayesian network and weighted K-nearest neighbor. J Intell Transp Syst 23(2):161–174
24. Pérez A, Larrañaga P, and Inza I (2006) "Supervised classification with conditional Gaussian networks: Increasing the structure complexity from naive Bayes." *International Journal of Approximate Reasoning* 43.1: 1–25
25. Pérez A, Larrañaga P, Inza I (2009) Bayesian classifiers based on kernel density estimation: flexible classifiers. Int J Approx Reason 50(2):341–362
26. Salinas-Gutiérrez R, Aguirre AH, Rivera-Meraz MJJ et al. (2010) "Supervised Probabilistic Classification Based on Gaussian Copulas. " *Mexican International Conference on Artificial Intelligence, Advances in Soft Computing*: 104–115
27. Scott DW (2015) *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, New Jersey
28. John GH, Langley P (1995) "Estimating Continuous Distributions in Bayesian Classifiers. " *In Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence* (San Mateo, USA, 1995), 338–345
29. He Y-L, Wang R, Kwong S, Wang X-Z (2014) Bayesian classifiers based on probability density estimation and their applications to simultaneous fault diagnosis. Inf Sci 259:252–268
30. Gutiérrez L, Gutiérrez-Peña E, Mena RH (2014) Bayesian nonparametric classification for spectroscopy data. Computational Statistics & Data Analysis 78:56–68
31. Zhang W, Zhang Z, Chao H-C, Tseng F-H (2018) Kernel mixture model for probability density estimation in Bayesian classifiers. Data Min Knowl Disc 32(3):675–707
32. Xiang Z-L, Yu X-R, Kang D-K (2016) Experimental analysis of naïve Bayes classifier based on an attribute weighting framework with smooth kernel density estimations. Appl Intell 44(3):611–620
33. Wang S-c, Gao R, Wang L-m (2016) Bayesian network classifiers based on Gaussian kernel density. Expert Syst Appl 51:207–217
34. Dang Q, Gao F, Zhou Y (2016) Early detection method for emerging topics based on dynamic bayesian networks in micro-blogging networks. Expert Syst Appl 57:285–295
35. Xiao Q, Chaoqin C, Li Z (2017) Time series prediction using dynamic Bayesian network. Optik 135:98–103
36. Premebida C, Faria DR, Nunes U (2017) Dynamic bayesian network for semantic place classification in mobile robotics. Auton Robot 41(5):1161–1172
37. Chhabra R, Rama Krishna C, Verma S (2019) Smartphone based context-aware driver behavior classification using dynamic bayesian network. Journal of Intelligent & Fuzzy Systems 36(5):4399–4412
38. Song C, Xu Z, Zhang Y, et al. (2020) "Dynamic hesitant fuzzy Bayesian network and its application in the optimal investment port decision making problem of "twenty-first century maritime silk road"." *Applied Intelligence* 2
39. Pearl J (1988) Probabilistic reasoning in intelligent systems: networks of plausible inference. San Mateo, California
40. Murphy SL, Aha DW (2019) UCI repository of machine learning databases. https://archive.ics.uci.edu/ml/datasets.php
41. Zhang XY, Yin F, Zhang YM et al (2017) Drawing and recognizing Chinese characters with recurrent neural network. IEEE Trans Pattern Anal Mach Intell 40(4):849–862
42. Wijnands JS, Thompson J, Aschwanden GDPA et al (2018) "Identifying behavioural change among drivers using Long Short-Term Memory recurrent neural networks". *Transportation research, Part F. Traffic psychology and behaviour* 53.2: 34–49
43. Kim PS, Lee DG, Lee SW (2018) Discriminative context learning with gated recurrent unit for group activity recognition. Pattern Recogn 76(4):149–161
44. Demsar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7(1):1–30