Knowledge-Based Systems 203 (2020) 106105

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

A robust time series prediction method based on empirical mode decomposition and high-order fuzzy cognitive maps

Zongdong Liu, Jing Liu *

School of Artificial Intelligence, Xidian University, Xi'an 710071, China

ARTICLE INFO

Article history:
Received 23 December 2019
Received in revised form 13 May 2020
Accepted 3 June 2020
Available online 4 June 2020

Keywords:
Time series prediction
High-order fuzzy cognitive maps
Empirical mode decomposition
Bayesian ridge regression

ABSTRACT

Fuzzy cognitive maps (FCMs) have been widely used in time series prediction due to the excellent performance in dynamic system modeling. However, existing time series prediction methods based on FCMs have some defects, such as low precision and sensitivity to hyper parameters. Therefore, more accurate and robust methods remain to be proposed for handling non-stationary and large-scale time series. To address this issue, in this paper, a novel time series prediction method based on empirical mode decomposition (EMD) and high-order FCMs (HFCMs) is proposed, termed as EMD-HFCM. First, EMD is applied to extract features from the original sequence to obtain multiple sequences to represent the nodes of HFCM. To learn HFCM efficiently and accurately, a robust learning method based on Bayesian ridge regression is employed, which can estimate the regular parameters from data instead of being set manually. Then, prediction can be performed based on the iterative characteristics of HFCM. To compare EMD-HFCM with existing methods, extensive experiments are conducted on eight benchmark datasets and the results validate the performance of the proposal in handling large-scale and non-stationary time series. Furthermore, the experiments also show that the proposed method is much more robust and insensitive to hyper parameters than the state of art methods. Finally, non-parametric statistical tests are carried out and the superiority of the proposed method is verified in the statistical sense.

1. Introduction

Time series prediction is an important issue in scientific research with an extensive range of practical applications, including economics [1,2], environment sciences [3,4], traffic [5,6] and energy sciences [7]. In a general sense, a time series is a sequence of observations of a system in chronological order, denoted as $\boldsymbol{x} = \{x_1, x_2, \ldots, x_t\}$. The time series prediction task is to estimate future data based on the observed historical data. Various methods have been developed for this task, such as autoregressive moving average model (ARMA) and its extensions [8–10], artificial neural networks (ANN) [11,12], support vector machines [13] and random forests [14]. Of late years, as numerous efficient learning methods being introduced, the application of fuzzy cognitive maps (FCMs) for time series forecasting has attracted more and more attention [15].

Proposed as a tool for uncertainty expression and causality modeling, FCMs have been found pervasive applications in complex system analysis and modeling, such as system control [16], decision making [17], game system modeling [18], time series prediction [19], and gene regulation network reconstruction [20]. FCMs have two advantages in handling time series prediction tasks. First, the capability of fuzzy modeling allows FCMs to not only predict at the numerical level, but also at the linguistic level, meaning that people can use the provided data to predict the next values in the space of amplitude and changes [21]. On the other hand, the forecasting stability and accuracy can be enhanced by mining the causal relationship hidden in the data [22].

Although FCMs have achieved some successful research results in time series prediction, the existing methods still have some limitations: (1) Time series data observed in practical applications tend to be non-stationary, which means those data may contain trend and seasonality. However, the comparison between FCMs and several other time series prediction approaches in [19] yielded the conclusion that FCMs perform weakly in non-stationary data and the authors suggested using it only for time series that are linear and tend to be stationary. Therefore, developing FCM-based methods that can handle non-stationary data or eliminate the influence of trend and seasonality is a more complex and noteworthy issue. (2) FCMs utilize the influence relationship among concepts for inference. Hence, for univariate sequence, feature extraction must be conducted first to form multiple sequences. Each sequence can be regarded as a sequence of state values of a conceptual node of FCM. Feature extraction can be considered as information mining for observation data.

* Corresponding author.
E-mail address: neouma@mail.xidian.edu.cn (J. Liu).

https://doi.org/10.1016/j.knosys.2020.106105
0950-7051/

Therefore, the specific method of feature extraction has a great effect on the prediction accuracy. Fuzzy c-means clustering and redundant wavelet transform were used in [23,24], respectively, as feature extraction methods. However, the number of decomposition sequences needed to be defined in advance for both of them, reducing the robustness of them. (3) The widely used population-based FCM learning methods are of low efficiency when dealing with large-scale time series, because each evaluation requires a complete simulation of the entire system, which is time consuming.

To tackle the issues mentioned above, in this paper, a novel two-stage univariate time series prediction method is proposed, combining empirical mode decomposition (EMD) [25] and high-order FCMs (HFCMs) [23]. The proposed approach, termed as EMD-HFCM, employs EMD as a feature extraction technique first to decompose the original time series into several intrinsic mode function (IMF) sequences and a residual sequence. The sequences formed can be regarded as the iterative state sequences of the concept nodes of FCMs. Then in the second stage, in order to process large-scale time series more efficiently, a Bayesian ridge regression [26] learning method is designed to optimize the weight matrix of the HFCM formed by the decomposed sequences. Next, the obtained weight matrix can be used to predict the future values through the iterative characteristics of HFCMs. The main contributions of this work can be summarized as follows.

(1) To the best of the authors' knowledge, this is the first research that combines EMD and FCMs for time series prediction. By using EMD, a set of stationary sequences can be obtained. As a consequence, the proposal can compensate for the limitations of FCMs in nonlinear and non-stationary time series modeling.

(2) This study proposes an HFCM learning method based on Bayesian ridge regression, which can estimate regular parameters from data and is more robust than ridge regression.

(3) Systematic experiments have carried out and validated the performance of EMD-HFCM on eight public time series datasets. The experimental results show that EMD-HFCM is efficient and accurate when applied in large-scale and non-stationary time series. Comparisons between EMD-HFCM and existing approaches also indicate the advantages of the proposal.

The remainder of this paper is organized as follows. Section 2 gives the review of related work of existing time series prediction methods based on FCMs. Section 3 describes some basic knowledge related to the proposed method, including FCMs, HFCM and EMD. The proposed method is introduced from the whole to the concrete in Section 4. Section 5 presents the experimental study of EMD-HFCM. Finally, the conclusion of this work is made in Section 6.

2. Related work

Several FCM-based time series prediction methods have been proposed [19,21–24,27–32], which share a similar framework. Firstly, the original numerical time series is fuzzified to obtain the fuzzy-valued series. Then, the series is transformed into the nodes of FCMs by some feature extraction techniques. Finally, after the weight matrix of FCMs is obtained by the learning method, the prediction can be performed.

Stach et al. in [21] put forward a prediction method based on granular FCMs, which can be used for prediction at both numerical and linguistic levels. The original time series was transformed into granular and fuzzy inputs, and the FCM was learned by real-code genetic algorithm. Song et al. in [27] introduced a technique of combining FCMs and automatic determined membership functions to improve the accuracy of time series prediction. Froelich et al. in [22] applied evolutionary FCMs to the task of prediction of prostate cancer. Froelich et al. in [28] proposed a time series prediction method based on fuzzy gray cognitive maps (FGCM) for climatological prediction, which employed evolutionary algorithms (EAs) to learn FCMs from multivariate interval-valued time series. Lu et al. in [23] proposed an approach to model and predict time series in the granular level based on high-order FCM and fuzzy c-means clustering, and particle swarm optimization (PSO) was used to learn the parameters. Salmeron et al. in [19] introduced a dynamic optimization method for FCMs, which can retrain the FCMs dynamically, and the concepts and transfer function can be also selected by the algorithm during the training process. Papageorgiou et al. in [29,30] applied a two-stage model in multivariate time series prediction based on FCMs and artificial neural network, and the structure optimization genetic algorithm (SOGA) was used to learn FCMs. Pedrycz et al. in [31] introduced a mechanism to represent a numeric time series in terms of information granules by fuzzy-c means clustering and the formed FCM was optimized with the use of PSO. Hajek et al. in [32] proposed an interval-valued intuitionistic representation of FCMs for stock index forecasting. Yang et al. in [24] proposed an efficient prediction method based on hybrid combination of HFCM with the redundant wavelet transform for non-stationary time series forecasting.

In addition, some rapid learning methods for FCMs have been proposed recently [33,34]. Feng et al. in [33] proposed a rapid and robust learning method for FCMs with maximum entropy, which transformed the learning of FCMs into a convex optimization problem with constraints. L1 regularization and the entropy of the weights were used as the penalty items of objective function for getting sparse solution and PSO was used to solve the problem. Shen et al. in [34] introduced the evolutionary multitasking framework to learn FCMs. The decomposition strategy based multiobjective optimization algorithm was used as the based learning method and they also adopted the memetic algorithm and lasso initialization operator to accelerate the convergence of the learning process.

Comparing these methods reveals that most of them focus on two aspects. One is the fuzzification of the original time series, and the other is the learning of the weight matrix of FCMs. Granularity [21,28] and membership function [27] are the most frequently used fuzzification techniques. In addition, fuzzy-c means clustering [23] and wavelet transform [24] also show good performance. In fact, they can be regarded as the feature extraction of the original series. In the aspect of FCM learning, almost all the work mentioned above used population-based methods to learn FCMs, except for the method proposed by Yang et al. in [24], which is based on ridge regression.

3. Prerequisite

3.1. Empirical mode decomposition

Empirical mode decomposition is a self-adaptive signal processing method proposed by Huang et al. [25], the essence of which is data smoothing, so it is quite suitable for dealing with non-stationary and nonlinear series. Since its introduction, EMD has been widely used in signal and time series processing [35–39]. Unlike priori basis function decomposition methods, such as Fourier decomposition and wavelet decomposition, EMD decomposes a signal according to the time scale characteristics of the signal itself and there is no need for a basis function. Theoretically, EMD can be applied to the decomposition of any type of time series.

Huang et al. gave the opinion that all signals are composed of several intrinsic mode functions (IMFs). In other words, a signal can comprise a finite number of IMFs. An IMF should satisfy the following two conditions:

(1) The number of extreme points and zero-crossings must be equal or at most one difference in the whole dataset;

(2) At any time, the mean value of the upper envelope defined by the local maxima and the lower envelope defined by the local minima is zero.

The purpose of EMD is to extract the IMFs of the original series and obtain a result of multiple IMFs and a residual sequence. The IMFs contain local information of the original sequence on different time scales. And EMD is based on the following assumptions:

(1) There are at least one maximum and one minimum over the entire length of the series;

(2) The characteristic time scale is uniquely determined by the time scale between extreme points;

(3) If the data has no extreme points but contains inflection points, the extreme points can be revealed by data differentiation once or more times.

The process of EMD can be summarized as Algorithm 1 in line with Huang's definition [25]. Assuming that the outer loop is performed $n + 1$ times in Algorithm 1, then n IMFs and one residue can be obtained.

Algorithm 1: EMD

Input:
$x(t)$: Original time series;
Output:
IMFs and the residual sequence;

$i \leftarrow 0$, $r_i(t)=x(t)$;
while (the number of extremums of $r_i(t)$ is greater than 2) ***do***
  $j \leftarrow 0$, $h_j(t)=r_i(t)$;
  repeat
    Find all local extreme points of $h_j(t)$;
    Perform cubic spline interpolation on the maxima and minima points of $h_j(t)$ to obtain the upper envelope $e_{max}(t)$ and the lower envelope $e_{min}(t)$, respectively;
    Calculate the average of the upper and lower envelope, $m_j(t)= (e_{max}(t)+e_{min}(t))/2$;
    $j \leftarrow j+1$;
    Calculate the difference of $h_{j-1}(t)$ and $m_{j-1}(t)$ as $h_j(t)= h_{j-1}(t)-m_{j-1}(t)$;
  until ($h_j(t)$ is an IMF);
  $i \leftarrow i+1$, $imf_i(t)= h_j(t)$;
  Calculate the residue, $r_i(t)= r_{i-1}(t)- imf_i(t)$;
end while.

Algorithm 1 gives the procedure of EMD, in which the cubic spline interpolation is a key technique used to interpolate between every two adjacent local extreme points to get the envelope sequence with the same length as the original sequence. In other words, cubic spline interpolation is used to obtain an envelope capable of performing point-to-point difference operations with the original sequence from the local extremums. More specifically, cubic spline interpolation computes a cubic function between every two adjacent extreme points, according to which all missing values between corresponding extreme points are obtained. Then the points on the curve made up of all functions of the adjacent extreme points constitute the envelope of the original time series.

As shown in Fig. 1, v_i and v_{i+1} are two adjacent local maximum points of the original series. The interpolation function between v_i and v_{i+1} calculated by cubic spline interpolation is given in the following form.

$$S_i(x) = a_i x^3 + b_i x^2 + c_i x + d_i \tag{1}$$

where a_i, b_i, c_i and d_i are the coefficients. $S_i(x)$ can be obtained by many existing software packages. All the interpolation functions $S_i(x)$ obtained from the interval of local maximum points constitute the upper envelope $e_{\max}(t)$ of the series and the lower envelope $e_{\min}(t)$ can be obtained in the same way.

After conducting the procedure of EMD on a time series data, the original data $x(t)$ can be expressed as follows,

$$x(t) = \sum_{i=1}^{n} imf_i(t) + r_n(t) \tag{2}$$

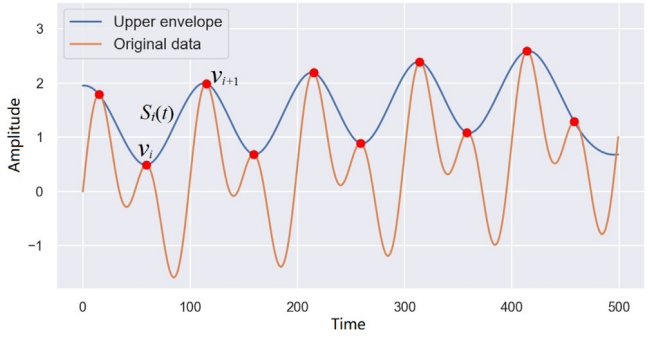

Fig. 1. Cubic spline interpolation for getting the upper envelope of a time series.

where $imf_i(t)$ denotes the ith IMF and n is the number of total IMFs. $r_n(t)$ is the final residue, which is either the mean trend or a constant.

3.2. High-order fuzzy cognitive maps

Representing the influence relationship between concepts in the system by weight matrix of real-valued elements in the interval $[-1, 1]$, FCMs can describe a system more accurately than cognitive maps of three-valued logic and are widely used to model complex systems [40].

Fig. 2 shows an FCM with 7 nodes and the corresponding relation matrix [41]. Through the representation of weighted directed graph on the left, the causal relationship between nodes can be seen clearly. Nevertheless, in actual modeling, the representation of weight matrix on the right is more preferred on account of easy operation and simulation. Each non-zero element in the matrix denotes an edge between two nodes, the actual meaning of which is the degree to which one node affects another. For instance, in Fig. 2(b), $w_{35} = 0.8$ means node C_3 has a positive effect on node C_5. Homoplastically, $w_{61} = -0.3$ indicates that node C_6 has a negative effect on node C_1.

Each node of an FCM has its state value at each iteration, denoted as $C_i(t)$. And $C_i(t)$ is located in the state space $L = [0, 1]$ (or $L = [-1, 1]$). Assuming that $\mathbf{C}(t)=[C_1(t), C_2(t), \ldots, C_N(t)]^T$ denotes the state vector of an FCM at the tth iteration, $C_i(t) \in L$. Then the state vector $\mathbf{C}(t + 1)$ at the $(t + 1)$th iteration can be inferred by the weight matrix $\mathbf{W}$ and $\mathbf{C}(t)$. This is the iterative property of FCMs, which can be expressed by the following equation.

$$\forall i \in \{1, \ldots, N\}, C_i(t+1) = f\left(\sum_{j=1}^{N} w_{ji} C_j(t)\right) \tag{3}$$

where $C_i(t)$ and $C_i(t + 1)$ denote the state value of node i at the tth and $(t + 1)$th iteration, respectively. w_{ji} denotes the weight of the influence of node j on node i and is in the range of $[-1, 1]$. N is the total number of nodes. $f(\bullet)$ is the transfer function used to map the activation level of a node into the state space. Eq. (3) can be vectorized as follows.

$$\mathbf{C}(t+1) = f(\mathbf{W} \times \mathbf{C}(t)) \tag{4}$$

Through the above iterative equation, state values of all nodes at the $(t + 1)$th iteration can be generated from state values at the tth iteration, and then state values at the $(t + 2)$th iteration can be generated from state values at the $(t + 1)$th iteration. In the same way, the future state values can be obtained as well. For node i, the iteratively generated state values can be also recorded as a sequence $\mathbf{C}_i = [C_i(1), C_i(2), \ldots, C_i(T)]$, where T is the total number of iterations. Thus, the state values generated by an FCM

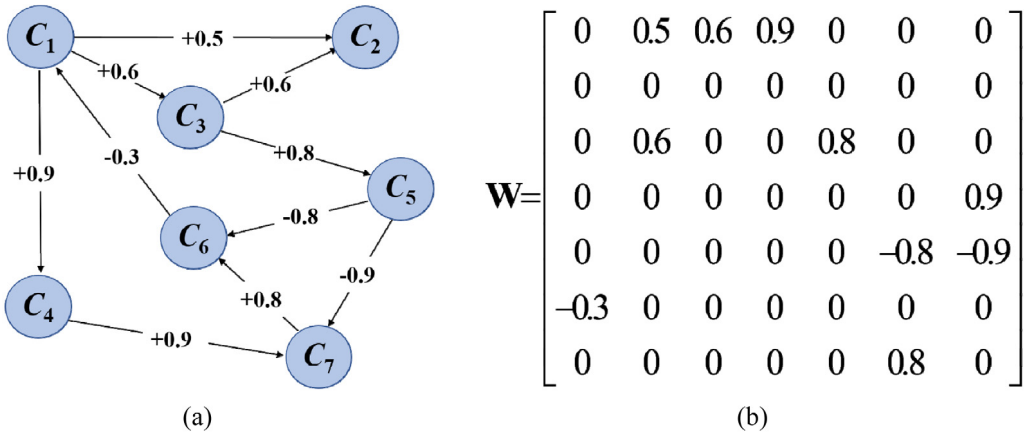

Fig. 2. An example FCM with 7 nodes: (a) representation of graph; (b) representation of weight matrix.

during T iterations can be denoted as the following state matrix.

$$\mathbf{C} = \begin{bmatrix} C_1(1) & C_1(2) & \cdots & C_1(T) \\ C_2(1) & C_2(2) & \cdots & C_2(T) \\ \vdots & \vdots & \ddots & \vdots \\ C_N(1) & C_N(2) & \cdots & C_N(T) \end{bmatrix} \tag{5}$$

The purpose of the transfer function $f(\bullet)$ in Eqs. (3) and (4) is to map the activation level of a node into L. Basically, sigmoid function (Eq. (6)) and hyperbolic tangent function (Eq. (7)) are the most widely used ones for FCMs whose state value is continuous [42]. The sigmoid function produces state values in [0, 1], while hyperbolic tangent function, also called tanh function, corresponds to $[-1, 1]$.

(1) sigmoid function

$$f(x) = \sigma(x) = \frac{1}{1 + e^{-\mu x}} \tag{6}$$

(2) hyperbolic tangent function

$$f(x) = \tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{7}$$

Traditionally, the state value of a node in FCMs at a certain iteration depends only on the state values of all nodes pointing to it at the previous iteration. But for some tasks such as time series prediction, earlier data may also have impacts on the current system state, hence the traditional configuration would limit the modeling capabilities of FCMs. To compensate for the limitation, in [23], the authors modified the iterative equation of FCMs as follows,

$$C_i(t+1) = f\left(\sum_{j=1}^{N} w_{ji}^{1} C_j(t) + w_{ji}^{2} C_j(t-1) + \cdots + w_{ji}^{k} C_j(t-k+1) + w_{i0}\right) \tag{8}$$

Eq. (8) describes the dynamic behavior of HFCMs with an order of k, where $w_{ji}^{s} \in [-1, 1] (s = 1,2,\ldots, k)$ is the weight of node i to node j with a time step of s and w_{i0} is the constant bias associated with node i, located in [0, 1].

4. Proposed method

4.1. Overall framework

When modeling a system using FCMs, what are really exploited are the interactions between nodes that represent meaningful concepts. In other words, nodes and relationship weights are two key factors. However, for univariate time series, the one-dimensional numerical sequence cannot directly constitute the structure of multiple nodes of FCMs. Therefore, EMD is used in this paper to extract different time-scale features of the original time series to obtain multiple time series, and then forecasting can be made based on the FCM constructed by the multivariate time series.

The overall framework of the proposed time series prediction method is shown in Fig. 3. First, the original univariate time series, which is usually nonlinear and non-stationary, needs to be normalized to the interval [-1, 1] by min–max normalization because EMD needs to use the zero-crossing information during the decomposition process. Next, EMD is performed to decompose the univariate time series into multiple IMF series and a residue series as shown in Eq. (2) and each subsequence represents the characteristics of original time series on different time scales. Then an FCM can be constructed where the total number of subsequences generated by EMD is exactly the number of nodes of the constructed FCM and each sequence represents the iterative sequence of a node. Further, taking k greater than 1, a k-order HFCM can be formed according to Eq. (8). Next Bayesian ridge regression is employed to optimize the weight matrix of HFCM composed of the multivariate time series. Using the iterative property of HFCM, the values of the next moment can be generated based on the values of previous k moments. Finally, by adding up all nodes' values, the predicted value at the next moment can be obtained. The following subsections give the details of the proposed EMD-HFCM.

4.2. Feature extraction of time series by EMD

The purpose of feature extraction is to drive multiple meaningful sequences from the original univariate time series to represent nodes of FCMs. Fuzzy c-mean clustering and redundant wavelet decomposition were used in [23] and [24], respectively, to achieve the purpose. However, both of them need to set the number of categories in advance.

In this paper, EMD is employed as the feature extraction method for time series decomposition at the first stage, the details of which have been shown in the previous section and Algorithm 1. Compared with fuzzy c-mean clustering and wavelet transform, EMD is a posteriori and self-adaptive technique. It is based on the local time characteristics of the original data and does not require human intervention. The obtained IMFs and residual sequence via EMD are of specific physical meanings, and each of them denotes the different time scale characteristics of original time series [25]. Fig. 4 shows an example of the decomposition of the Mackey-Glass time series obtained by EMD.

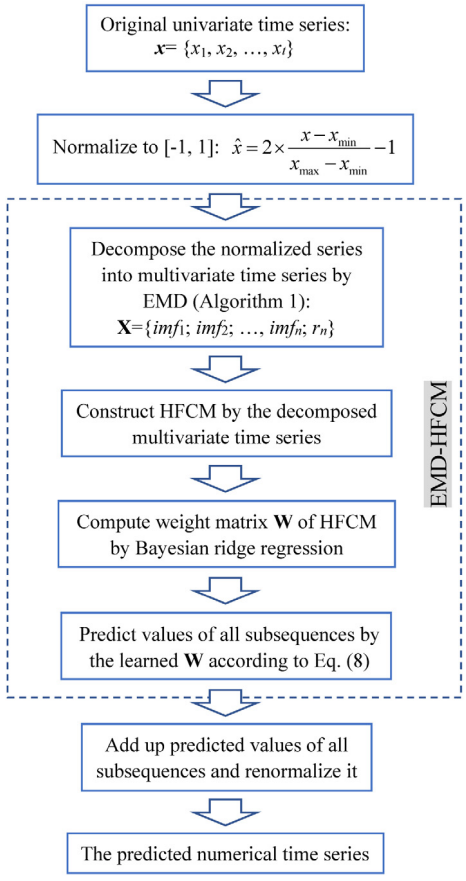

Fig. 3. Overall framework of the proposed EMD-HFCM.

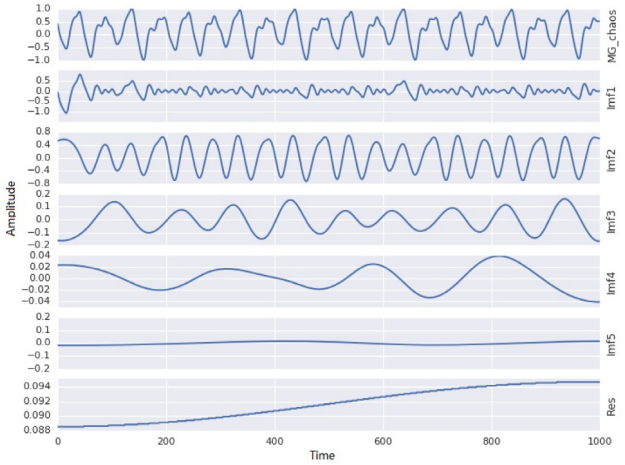

Fig. 4. An example of EMD for MG time series. From top to bottom are: original time series, imf1, imf2, imf3, imf4, imf5, and the residue, respectively.

4.3. HFCM learning via Bayesian ridge regression

After feature extraction by EMD and obtaining multiple time series, the remaining task is to learn the weight matrix of the HFCM composed by the time series extracted. Among numerous proposed automatic learning methods for FCMs, population-based learning is one of the most popular algorithms [20,43,44]. However, population-based methods are time-consuming and therefore inefficient when dealing with large-scale time series [24,45,46]. Chen et al. in [47] first introduced a decomposition optimization strategy for FCMs, as shown in Fig. 5. The authors employed EAs to learn the local connections node-by-node and finally merged them into an entire weight matrix. This decomposition strategy proved to greatly reduce the difficulty of learning weights of FCMs. Wu et al. used this decomposition strategy in [45] and [46], but the more efficient methods based on compressed sensing and lasso regression were used respectively, instead of EAs. Yang et al. also used this strategy and ridge regression for prediction [24].

In this paper, the superior Bayesian ridge regression with this decomposition strategy is used to learn the weight matrix of HFCM for the reason of automatic parameter estimation. Eq. (8) shows the general form of dynamic iterative process of HFCM, which is nonlinear and can be transformed into the following linear form.

$$f^{-1}(C_i(t+1)) = \sum_{j=1}^{N} w_{ji}^1 C_j(t) + w_{ji}^2 C_j(t-1) + \cdots + w_{ji}^k C_j(t-k+1) \quad (9)$$

where $f^{-1}(\bullet)$ represents the inverse function of the transfer function $f(\bullet)$. The bias term in Eq. (8) is removed since it does not have a practical meaning and even increases the error of the model. Then, the above formula can be vectorized into the following form.

$$\mathbf{Y}_i = \mathbf{X}\mathbf{W}_i \quad (10)$$

where $\mathbf{Y}_i$ consists of $f^{-1}(C_i(t+1))$ at different t, $\mathbf{X}$ is made up of state values $C_i(t)$ of all nodes and $\mathbf{W}_i$ is the weight vector of weights of all nodes to node i with different time step, as expressed in Box I,

Specific to 2-order HFCM, the dynamic equation (Eq. (9)) can be written as Eq. (14) and $\mathbf{Y}_i$, $\mathbf{X}$ and $\mathbf{W}_i$ are shown as Eqs. (15)–(17), respectively.

$$f^{-1}(C_i(t+1)) = \sum_{j=1}^{N} w_{ji}^1 C_j(t) + w_{ji}^2 C_j(t-1) \quad (14)$$

$$\mathbf{Y}_i = \begin{bmatrix} f^{-1}(C_i(3)) \\ f^{-1}(C_i(4)) \\ \vdots \\ f^{-1}(C_i(L)) \end{bmatrix} \quad (15)$$

$$\mathbf{X} = \begin{bmatrix} C_1(2) & C_1(1) & C_2(2) & C_2(1) & \cdots & C_N(2) & C_N(1) \\ C_1(3) & C_1(2) & C_2(3) & C_2(2) & \cdots & C_N(3) & C_N(2) \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ C_1(L-1) & C_1(L-2) & C_2(L-1) & C_2(L-2) & \cdots & C_N(L-1) & C_N(L-2) \end{bmatrix} \quad (16)$$

$$\mathbf{W}_i = \begin{bmatrix} w_{1i}^1 & w_{1i}^2 & w_{2i}^1 & w_{2i}^2 & \cdots & w_{Ni}^1 & w_{Ni}^2 \end{bmatrix}^T \quad (17)$$

where L is the length of the time series.

In Eq. (10), both $\mathbf{Y}_i$ and $\mathbf{X}$ come from provided time series data and the task is to solve $\mathbf{W}_i$ for each node i. In [46] and [24], lasso regression and ridge regression are used respectively to solve the problem. Their objective functions are shown in Eq. (18) and Eq. (19), respectively.

$$\min_{\mathbf{W}_i} \left\{ \frac{1}{2L} \|\mathbf{Y}_i - \mathbf{X}\mathbf{W}_i\|_2^2 + \lambda \|\mathbf{W}_i\|_1 \right\} \quad (18)$$

$$\min_{\mathbf{W}_i} \left\{ \frac{1}{2L} \|\mathbf{Y}_i - \mathbf{X}\mathbf{W}_i\|_2^2 + \lambda \|\mathbf{W}_i\|_2^2 \right\} \quad (19)$$

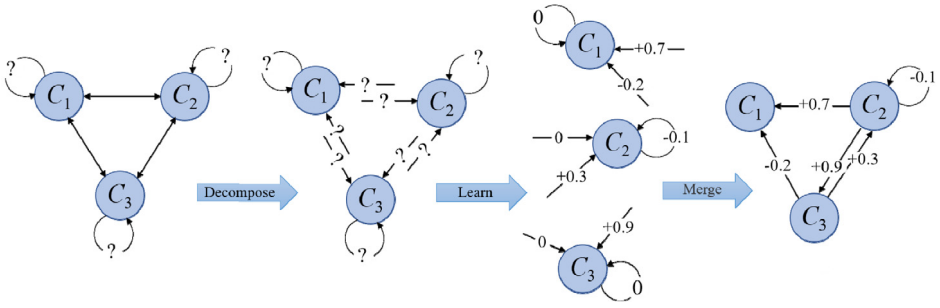

Fig. 5. Framework of decomposition method for learning FCMs.

$$\mathbf{Y}_i = \begin{bmatrix} f^{-1}(C_i(k+1)) \\ f^{-1}(C_i(k+2)) \\ \vdots \\ f^{-1}(C_i(L)) \end{bmatrix} \tag{11}$$

$$\mathbf{X} = \begin{bmatrix} C_1(k) & \cdots & C_1(2) & C_1(1) & C_2(k) & \cdots & C_2(1) & \cdots & C_N(k) & \cdots & C_N(1) \\ C_1(k+1) & \cdots & C_1(3) & C_1(2) & C_2(k+1) & \cdots & C_2(2) & \cdots & C_N(k+1) & \cdots & C_N(2) \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ C_1(L-1) & \cdots & C_1(L-k+1) & C_1(L-k) & C_2(L-1) & \cdots & C_2(L-k) & \cdots & C_N(L-1) & \cdots & C_N(L-k) \end{bmatrix} \tag{12}$$

$$\mathbf{W}_i = \begin{bmatrix} w_{1i}^1 & w_{1i}^2 & \cdots & w_{1i}^k & w_{2i}^1 & \cdots & w_{2i}^k & \cdots & w_{Ni}^1 & \cdots & w_{Ni}^k \end{bmatrix}^T \tag{13}$$

Box I.

where $\|\mathbf{W}_i\|_1 = \sum_j |w_{ji}|$ represents L1 regularization and $\|\mathbf{W}_i\|_2^2 = \sum_j w_{ji}^2$ is L2 regularization. λ >0 is a regular parameter that needs to be configured manually.

But in fact, this parameter can be estimated based on data. On the other hand, lasso regression is usually used in learning some sparse structures to obtain sparse solutions. Whereas, here in the prediction task, the sparsity of the node connections of HFCM is not necessary. So, neither the lasso nor basic ridge but Bayesian ridge regression is adopted to learn HFCMs. Bayesian ridge regression can automatically estimate the value of the regularization parameter based on the data during the training phase, thus eliminating the subjectivity of artificial settings and making the algorithm more stable.

First, in order to introduce Bayesian estimation and obtain a probabilistic model, it is assumed that the output values of the model obey Gaussian distribution. That is, each actual value y_k in $\mathbf{Y}_i$ is a Gaussian random variable with $\mathbf{X}_k\mathbf{W}_i$ as mean and α^{-1} as the variance, as shown in Eq. (20).

$$p(y_k|\mathbf{X}_k, \mathbf{W}_i, \alpha) = N(y_k|\mathbf{X}_k\mathbf{W}_i, \alpha^{-1}) \tag{20}$$

where $\mathbf{X}_k$ is the vector consisting of the kth row of $\mathbf{X}$ and $\mathbf{X}_k\mathbf{W}_i$ is the predicted value of the model corresponding to y_k. And let the prior distribution of $\mathbf{W}_i$ be given by a spherical Gaussian.

$$p(\mathbf{W}_i|\beta) = N(\mathbf{W}_i|0, \beta^{-1}\mathbf{I}) \tag{21}$$

The priors over α and β are gamma distributions; that is, $\alpha^{-1} \sim G(\alpha_1, \alpha_2)$, $\beta \sim G(\beta_1, \beta_2)$, and α>0, β>0. α, β and $\mathbf{W}_i$ can be estimated from the data during training process [48,49].

Then, the posterior probability of $\mathbf{W}_i$ can be obtained according to Bayesian rules.

$$p(\mathbf{W}_i|\mathbf{X}_k, y, \alpha, \beta) = p(y|\mathbf{X}_k, \mathbf{W}_i, \alpha)p(\mathbf{W}_i, \beta) \tag{22}$$

For the entire time series data, it is easy to derive the log likelihood probability of the above equation as follows,

$$\ln p(\mathbf{W}_i|\mathbf{Y}_i) = -\frac{\alpha}{2L}\|\mathbf{Y}_i - \mathbf{X}\mathbf{W}_i\|_2^2 - \frac{\beta}{2}\|\mathbf{W}_i\|_2^2 + const \tag{23}$$

where *const* denotes the constant term unrelated to $\mathbf{W}_i$. Obviously, maximizing Eq. (23) is the purpose of the training process. As a result, the objective function of Bayesian ridge regression is shown in Eq. (24). It can be seen that the regular term is generated by introducing the Bayesian probability model, and the obtained objective function is similar to that of ridge regression. But compared with the basic ridge regression, the coefficients α and β of Bayesian ridge regression are estimated from the data rather than be set manually.

$$\min_{\mathbf{W}_i}\left\{\frac{\alpha}{2L}\|\mathbf{Y}_i - \mathbf{X}\mathbf{W}_i\|_2^2 + \frac{\beta}{2}\|\mathbf{W}_i\|_2^2\right\} \tag{24}$$

Rezaee et al. in [50] proposed a method based on extended Delta rule to learn multi-stage cognitive map, which can be applied to the circumstance that inputs are independent but not orthogonal. The most striking difference between the method in [50] and Bayesian ridge regression is that the former updates weights according to the gradient of square error just like ANN while the method in this work tries to find weights that maximize the posterior probability $p(\mathbf{W}_i|\mathbf{Y}_i)$ according to Bayes criterion. In addition, the extended Delta rule method does not have regular term which means the method may not consider the problem of overfitting.

By solving Eq. (24), relative weights of node i can be obtained. Therefore, the complete weight matrix of HFCM can be obtained by repeating this process N times, where N is the total number of nodes. It is convenient to solve Eq. (24) by applying the Bayesian ridge regression procedure of Scikit-learn toolkit in

Python and the parameters can be inferred by using Expectation–Maximization (EM) algorithm. On the whole, it takes a time complexity of $O(N^3kLT)$ to solve the weight vector of one node. Here T is the number of iterations of learning process of Bayesian ridge regression and is set to 300 as a default value in Scikit-learn toolkit. Thus, it needs a complexity of $O(N^4kLT)$ to achieve solutions of all nodes totally.

5. Experiments

In this section, experiments are conducted on eight datasets to validate the performance of the proposed method. All experiments are performed using a Python program on a computer with a 3.3 GHz CPU and 32 GB of memory.

5.1. Datasets

Eight frequently used datasets were used to verify the validity of the proposed method. The first one is the time series of yearly number of sunspots from 1700 to 1988, which contains 289 observations and is regarded as a non-linear series. The second dataset records the daily open price of S&P 500 stock index from June 1, 2016 to June 1, 2017, with a total of 251 records. The third dataset contains 168 observations in total, which records the monthly milk production from January 1962 to December 1975. The fourth dataset concerns the closing of the Dow–Jones industrial index from August 1968 to August 1981 in month. The fifth dataset records the monthly Lake Erie level from 1921 to 1970, consisting of 600 observations. The sixth dataset includes returns of daily Istanbul stock exchange national 100 index from June 5, 2009 to February 22, 2011 with a total of 536 records. The seventh dataset consists of 40 days temperature data from a monitor system of a room in hours, which contains 1034 observations. The last dataset, Mackey-Glass chaos time series [51], is generated by the MG equation defined in Eq. (25). It shows well-understood chaos behavior with dimensionality dependent on the chosen value of the delay parameter and has become a benchmark for evaluating the performance of a prediction model. In the experiment, a series is generated based on Eq. (25) by using the 4-order Runge–Kutta algorithm, which contains 1000 observations from $t = 124$ to $t = 1123$. The initial value $x(0)$ is set to 1.2 and τ is set to 17 according to the literature.

$$\dot{x}(t) = \frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)} - 0.1x(t) \tag{25}$$

5.2. Experimental setup

Fig. 3 shows the overall framework of the proposed method. Firstly, the data should be normalized into $[-1, 1]$ before feeding them into EMD-HFCM because EMD uses zero-crossing information, which means that both positive and negative values are needed. So, the min-max normalization method is used, as shown in Eq. (26), where $x_{\max}$ and $x_{\min}$ denotes the maximum and minimum of the original time series, respectively. x is the true value and $\hat{x}$ is the normalized value. Then, considering that the normalized data is in the interval $[-1, 1]$, the tanh function is adopted as the transfer function of HFCM to match the input data.

$$\hat{x} = 2 \times \frac{x - x_{\min}}{x_{\max} - x_{\min}} - 1 \tag{26}$$

As mentioned earlier, EMD is self-adaptive and the number of nodes of HFCM need not be set manually. However, there are still three hyper-parameters, the order k, α and β of Bayesian ridge regression, that may affect the performance of the prediction model. Therefore, the dataset is divided into three subsets: training dataset for learning the weight matrix of HFCM, validation dataset for selecting the best model and test dataset for evaluating the performance of the obtained model. For the objectivity of the results, the same division is used as the existing literature. Details of the division are shown in Table 1.

Table 1
Division of subsets.

Dataset	Training	Validation	Test
Sunspot	177	44	67
MG time series	400	100	500
S&P 500 index	120	30	101
Milk production	108	26	34
Dow–Jones index	175	43	73
Lake Erie Level	368	92	140
Istanbul stock exchange index	343	85	108
Temperature of monitor system	620	155	259

For the evaluation of results, the classic root-mean-square error (RMSE) is adopted as the performance criteria.

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{t=1}^{L}(x(t) - \hat{x}(t))^2}{L}} \tag{27}$$

where L is the length of the time series, and x and $\hat{x}$ denote the true and predicted values, respectively.

5.3. Analysis of hyper-parameters

In this subsection, the three hyper-parameters and the robustness of EMD-HFCM are studied by taking the sunspot time series and the S&P500 index time series as examples.

(1) *Effect of Bayesian ridge regression parameters α and β*: First, the two parameters α and β of Bayesian ridge regression are analyzed, which are different from the regular parameter of the basic ridge regression. As mentioned in Section 4, α and β are random numbers obeying the gamma distribution, $\alpha^{-1} \sim G(\alpha_1, \alpha_2)$, $\beta \sim G(\beta_1, \beta_2)$, but they both can be estimated from data during the training process. Thus, although α and β do not need to be set manually, their distribution parameters still need to be specified. As a result, the task is transformed into the study of the hyper-parameters of α and β.

For the sake of convenience, the experiments study the hyper-parameters of α and β separately. Mackay in [26] suggested that $\alpha_1 = \alpha_2 = \beta_1 = \beta_2 = 1e{-}6$. Therefore, in the experiments, β_1 and β_2 are set to 1e−6 and the hyper-parameters of α are studied and then α_1 and α_2 are set to 1e−6 and the hyper-parameters of β are studied. Specifically, the grid search method is adopted in this part. Taking α as an example, β_1 and β_2 are set to the recommended value and α_1 and α_2 are arranged as different values among [1e−1, 1e−3, 1e−6, 1e−8, 1e−10, 1e−14, 1e−18, 1e−20] to observe the change of RMSE. The variation of RMSE with four hyper-parameters is shown in Figs. 6 and 7.

It can be seen from Figs. 6 and 7 that, in most cases, RMSE of the prediction result is a little higher only when a value of 1e−1 is given. Fig. 7(b) shows that RMSE is the smallest when β_1 equals 1e−1. But even when β_1 is smaller than 1e−3, RMSE does not change too much. For all the four hyper-parameters, when they are set to the values less than or equal to 1e−3, RMSE are stable around a small value and with little change. The result indicates that Bayesian ridge regression is insensitive to hyper-parameter settings, although more hyper-parameters are introduced compared with ridge regression. Therefore, any small values less than 1e−3 are applicable for hyper-parameters of Bayesian ridge regression when using it in learning HFCM. In the follow-up experiments the Mackay's recommendation is adopted,

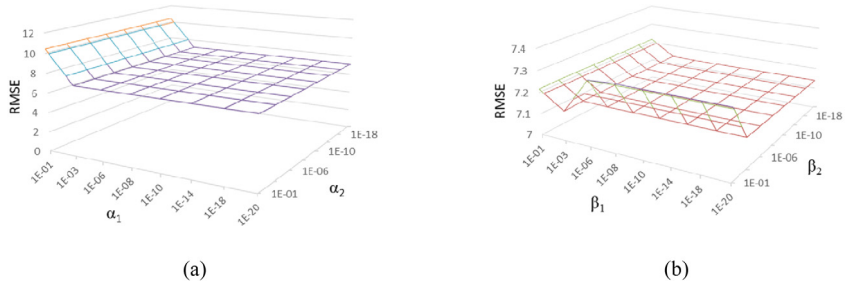

Fig. 6. Variation of RMSE with the distribution of α and β on the sunspot time series.

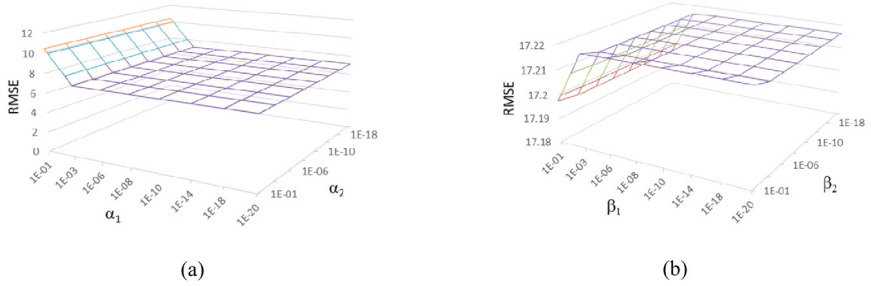

Fig. 7. Variation of RMSE with the distribution of α and β on the S&P500 time series.

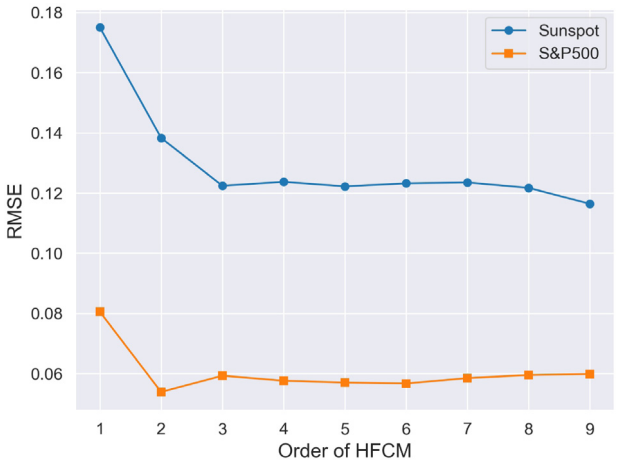

Fig. 8. Variation of RMSE with order k of HFCM on the sunspot and S&P500 datasets.

i.e. $\alpha_1 = \alpha_2 = \beta_1 = \beta_2 =$ 1e−6, which is also the default setting in the Scikit-learn toolkit.

(2) *Effect of k on EMD-HFCM*: According to the previous analysis of the hyper-parameters of Bayesian ridge regression, α_1, α_2, β_1 and β_2 are all set to 1e−6 when analyzing the influence of the order k on the prediction results. Fig. 8 shows how RMSE varies with order k on the two datasets. The errors are recorded in the validation phase, so RMSE is calculated with normalized data. It can be seen that for both of the two datasets, RMSE is the highest at $k = 1$, which indicates that HFCM can indeed achieve better prediction performance by using more historical data. As the order increases, RMSE fluctuates around a small value, on the whole it does not change too much. This indicates that the model is not sensitive to the order k, either. For generality, the authors suggest setting k to a value from 4 to 6 as a suitable choice.

5.4. Comparison of Bayesian ridge regression with ridge regression and lasso regression

Both ridge regression and lasso regression have been used to learn FCMs and showed a good performance. However, the choice of regularization coefficient is tough but predominant in the quality of result. In order to verify the advantages of Bayesian ridge regression, both the two regression approaches are used to learn HFCMs and their results are compared with that of Bayesian ridge regression. The hyper-parameters of Bayesian ridge regression are still set as the default values, and regular parameter λ of ridge and lasso regression is set to 1e−2, 1e−4, 1e−8, 1e−12 in each set of experiments, respectively. Fig. 9 shows the results of comparisons.

It can be seen from Fig. 9 that the value of λ has a great influence on the prediction result when learning HFCMs via ridge regression or lasso ridge. Concretely speaking, Bayesian ridge regression can reach a good result even with the default parameters, whereas RMSE of both ridge and lasso regression fluctuate greatly versus varying λ and there is no fixed trend. RMSE of lasso regression of all datasets gets very high when $\lambda =$ 1e−2. On the contrary, RMSE of ridge regression of four datasets gets higher when $\lambda =$ 1e−12. Even though ridge regression and lasso regression show better results than Bayesian ridge regression in a few cases, neither of them can get good results on all datasets with a fixed λ. From an overall perspective, the result of Bayesian ridge regression is better than either of them on almost all eight datasets. Thus, combined with results of the previous experiments, it can be concluded that the proposed EMD-HFCM by Bayesian ridge regression is very robust.

5.5. Comparison with existing methods and statistical verification

Fig. 10 shows the original time series and predicted time series obtained by EMD-HFCM. More detailed results are shown

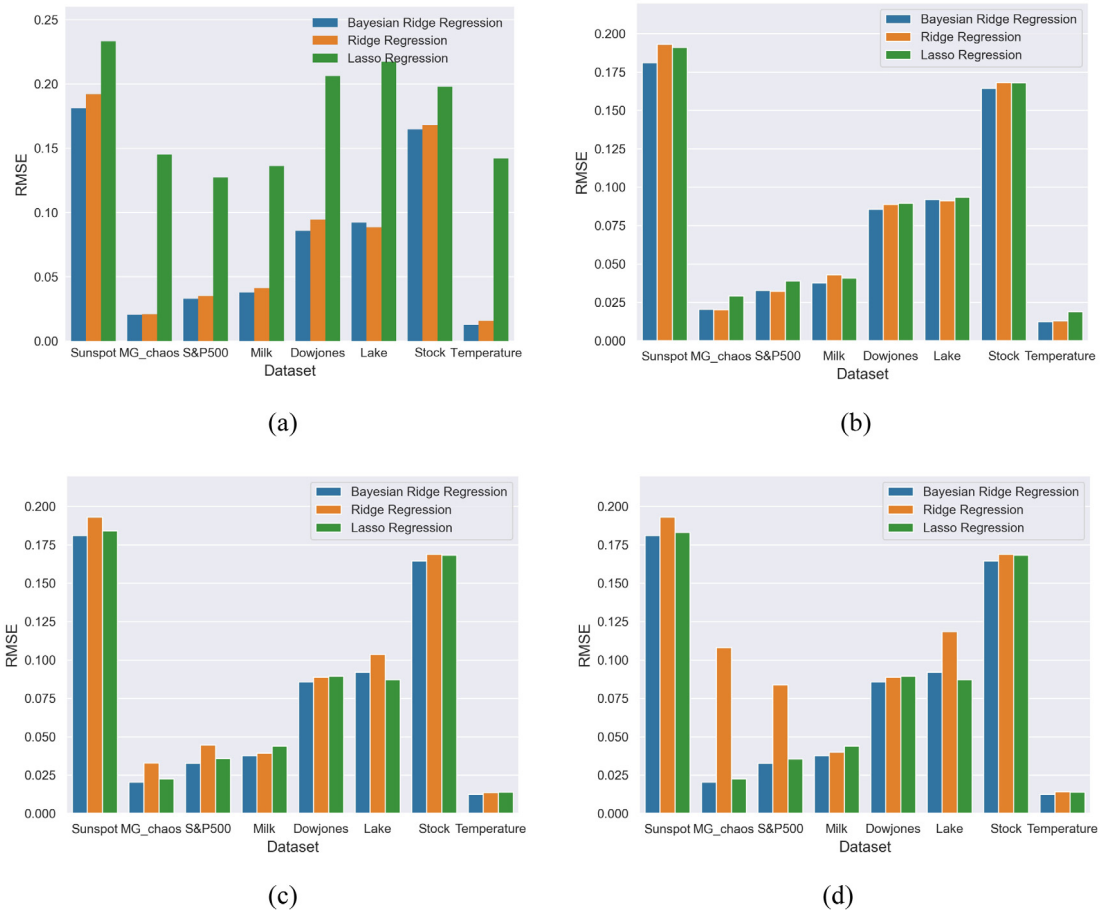

Fig. 9. Comparison in terms of RMSE among Bayesian ridge regression, ridge, and lasso with varying λ: (a) λ = 1e−2; (b) λ = 1e−4; (c) λ = 1e−8; (d) λ = 1e−12.

in Table 2, which gives the RMSE of the training part, validation part, test part and the corresponding optimal order k of each dataset. Finally, comparisons with five existing prediction methods, including Wavelet-HFCM, ANFIS [52], AR model [53], Multiresolution AR [54] model and ANN [55], are shown in Table 3 to verify the superiority of EMD-HFCM. As can be seen from the comparison, EMD-HFCM outperforms other methods in six of eight cases. Even on the remaining two datasets, it also shows an acceptable prediction accuracy. In terms of efficiency, since EM algorithm is needed to estimate the parameters of Bayesian ridge regression, the solution efficiency is slightly slower than that of ridge regression. However, the time consumption of hyper-parameters optimization is reduced due to the robustness of EMD-HFCM for regular parameters, hence the increased time complexity is reasonable and acceptable.

Likewise, the learning method based on Bayesian ridge regression has many advantages compared with population-based methods. First, Bayesian ridge regression is still faster than population-based methods although it is a little slower than ridge regression. In each iteration, population-based methods need a complexity of $O(N^2kL)$ to evaluate each individual, which means it needs at least $O(N^2kLPT)$ for any population-based methods to evaluate individuals, where P is the size of population and T is the number of generations. Note that population-based methods usually need hundreds of individuals and tens of thousands of generations to find a good solution. Besides, considering the evolution operators and the local search operators designed to avoid falling into local optima, the process will be more time-consuming. As a result, population-based methods are much slower than Bayesian ridge regression though the order of N is smaller than that of Bayesian ridge regression. Secondly, most of the population-based methods use the square error as evaluation function, which is easy to cause overfitting. And if the regular term is added to the evaluation function, a similar problem like ridge and lasso regression will arise, *i.e.* the setting of the regularization parameters. Therefore, according to the above analysis, the proposed learning method based on Bayesian ridge regression is more competitive than population-based learning methods, especially in large-scale time series modeling.

Table 2
RMSE of the training part, validation part, test part and the corresponding optimal order k of EMD-HFCM on each dataset.

Dataset	Training	Validation	Test	k
Sunspot	12.017	11.066	17.216	9
MG time series	0.009	0.010	0.009	22
S&P 500 index	8.728	11.781	7.139	2
Milk production	4.668	5.095	7.403	11
Dow–Jones index	12.473	11.523	18.875	2
Lake Erie Level	0.357	0.365	0.436	6
Istanbul stock exchange index	0.013	0.009	0.011	2
Temperature of monitor system	0.177	0.139	0.116	6

In order to further demonstrate the superiority of the proposed method compared to existing methods, statistical verification is conducted to compare the performance of different methods. The non-parametric test methods introduced in [56] are used here.

First, the multiple sign-test is used to determine whether there is a performance difference between other methods and the control one. In order to eliminate the effect of the numerical

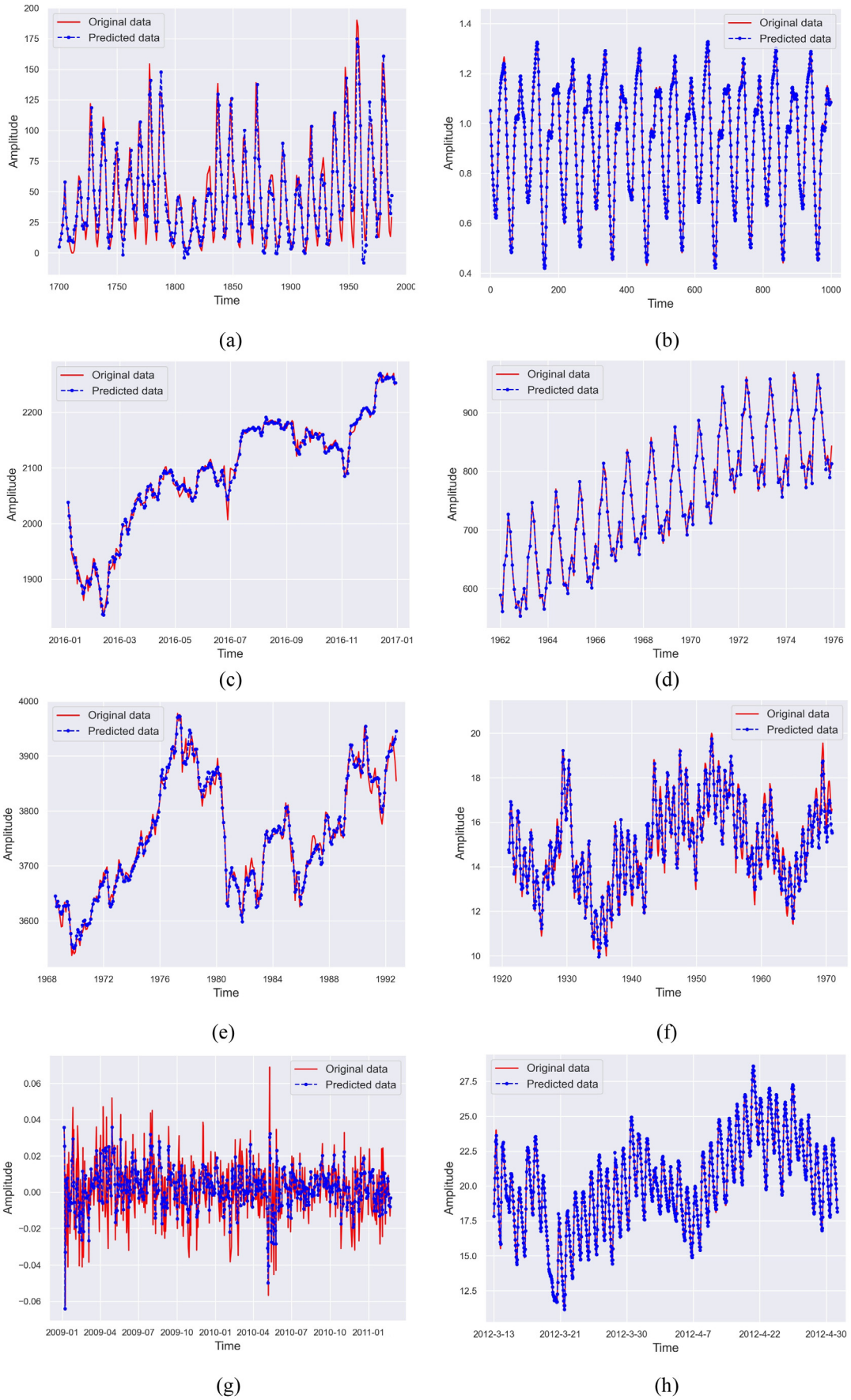

Fig. 10. Original time series and predicted time series of (a) sunspot time series, (b) MG chaos time series, (c) S&P 500 index, (d) monthly milk production, (e) Dow–Jones industrial index, (f) Lake Erie levels, (g) Istanbul stock exchange index, (h) temperature of monitor system.

Table 3
Comparison in terms of RMSE between EMD-HFCM and existing methods.

Dataset	EMD -HFCM	Wavelet -HFCM	ANFIS	AR model	Multiresolution AR model	ANN
Sunspot	**17.216**	18.916	22.753	35.262	19.186	19.901
MG time series	0.009	0.004	**0.001**	0.035	0.002	0.005
S&P 500 index	**7.139**	16.105	14.935	17.897	16.041	17.696
Milk production	**7.403**	8.258	9.578	57.717	37.838	27.113
Dow–Jones index	**18.875**	23.159	27.526	29.822	26.733	28.52
Lake Erie Level	0.436	**0.377**	0.458	0.638	0.390	0.402
Istanbul stock exchange index	**0.011**	0.015	0.014	0.018	0.015	0.015
Temperature of monitor system	**0.116**	0.224	0.157	0.579	0.124	0.421

differences of different datasets on the results, values in Table 3 are normalized to [0,1] by row and the signed differences among the methods are counted and shown in Table 4. The control method is the proposed EMD-HFCM. Then with the hypotheses to be H_0: the control method is better than the other one and H_1: the control method is not better than the other one at a level of significance 0.05, Table A.1 in [56] reveals that the critical value of r_j is 0 for six methods and eight datasets. Thus, it means that EMD-HFCM has a better performance than AR model since the number of pluses of it is no more than 0. However, for Wavelet-HFCM, ANFIS, Multiresolution AR model and ANN, the null hypothesis is rejected and further tests are needed.

Thus, the contrast estimation based on medians is conducted to examine the performance of the rest methods. For brevity, EMD-HFCM, Wavelet-HFCM, ANFIS, Multiresolution AR model and ANN are numbered 1, 2, 3, 4, 5 and the dataset are numbered 1 to 8 in the order from top to bottom in Table 4. Then, the difference between the performances of every pair of the five methods on each of the eight datasets can be calculated as follows,

$$D_{i(uv)} = x_{iv} - x_{iu} \tag{28}$$

where $D_{i(uv)}$ is the difference of normalized RMSE between the vth method and uth method. The result is shown in Table 5. The last row in the table is the median Z_{uv} of each set of differences and is called as the unadjusted estimator.

By Z_{uv}, the mean values of each set of unadjusted medians of each method can be calculated as Eq. (29), where N_m is the number of methods. With M_i being the median response of the results of method i, the estimator of M_i-M_j is m_i-m_j then. Finally, the estimators of each pair of methods are computed and shown in Table 6.

$$m_u = \frac{\sum_{j=1}^{N_m} Z_{uj}}{N_m} \tag{29}$$

As shown in Table 6, the proposed EMD-HFCM always has a positive difference value with respect to other methods, indicating that EMD-HFCM has a best performance among these methods. Therefore, the superior of EMD-HFCM over other methods is verified by multiple sign-test and contrast estimation.

6. Conclusions

This paper presents a univariate time series prediction method based on EMD and HFCM. A self-adaptive feature extraction technique, EMD, is used to decompose the original time series into multiple meaningful sequences, and then the HFCM composed of the obtained multivariate time series is employed as a prediction tool. To learn the weight matrix of HFCM, a ridge regression learning method based on Bayesian rules is designed, which is more robust than pure ridge regression. Finally, the future value can be predicted by using the structure of the learned HFCM and its iterative property. And the numerical predictive value can be obtained by adding up and de-normalizing the values at each time step.

In the experimental section, the performance of the proposed method is tested on eight publicly available datasets. In order to analyze the influence of hyper-parameters on EMD-HFCM, the hyper-parameters of Bayesian ridge regression and the order of HFCM is studied orderly. The results indicate that Bayesian ridge can estimate parameters α and β during the training process from an arbitrarily given small distribution parameters. That is to say, Bayesian ridge regression does not require manual parameter selection though it introduces more hyper-parameters than ridge regression. The study also shows that EMD-HFCM is insensitive to order k and the suggestion values are 4, 5 and 6. Next, in order to compare Bayesian ridge regression with basic ridge regression and lasso regression, the comparison in terms of the prediction errors of EMD-HFCM learned by the three methods is made. The result shows that both ridge regression and lasso regression are sensitive to the regular parameter λ and Bayesian ridge regression is more effective and more robust. In the experiment, the optimal order is selected by the validation set and then the optimal RMSE can be obtained. Comparison of results of EMD-HFCM and several existing methods also demonstrate the usefulness of the proposed method. To be more convincing, nonparametric tests are carried out on the experimental results, which also show that EMD-HFCM is the best one.

The feature extraction method is an important ingredient that would affect the prediction accuracy when using FCMs for time series prediction. Although the use of EMD in this paper has yielded a satisfactory result. The authors believe that there are more effective ways to be discovered and investigated in the future. In addition, due to the convergence characteristics of FCMs, long-term prediction based on FCMs is still a difficult problem and is worthy of study.

CRediT authorship contribution statement

Zongdong Liu: Conceptualization, Methodology, Software, Validation, Formal analysis, Writing - original draft, Writing - review & editing. **Jing Liu:** Writing - review & editing, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported by the General Program of National Natural Science Foundation of China (NSFC) under Grant 61773300.

Table 4
Signs of comparison in terms of normalized RMSE between EMD-HFCM and existing methods.

Dataset	EMD -HFCM	Wavelet -HFCM	ANFIS	AR model	Multiresolution AR model	ANN
Sunspot	0.488	0.536(+)	0.645(+)	1.000(+)	0.544(+)	0.564(+)
MG time series	0.257	0.114(−)	0.029(−)	1.000(+)	0.057(−)	0.143(−)
S&P 500 index	0.399	0.900(+)	0.834(+)	1.000(+)	0.896(+)	0.989(+)
Milk production	0.128	0.143(+)	0.166(+)	1.000(+)	0.656(+)	0.470(+)
Dow–Jones index	0.633	0.777(+)	0.923(+)	1.000(+)	0.896(+)	0.956(+)
Lake Erie Level	0.683	0.591(−)	0.718(+)	1.000(+)	0.611(−)	0.630(−)
Istanbul stock exchange index	0.611	0.833(+)	0.778(+)	1.000(+)	0.833(+)	0.833(+)
Temperature of monitor system	0.200	0.387(+)	0.271(+)	1.000(+)	0.214(+)	0.727(+)
Number of pluses		6	7	8	6	6
Number of minuses		2	1	0	2	2
r_j		2	1	0	2	2

Table 5
Differences in terms of normalized RMSE between each pair of methods on each dataset.

DataSet	$D_{i(12)}$	$D_{i(13)}$	$D_{i(14)}$	$D_{i(15)}$	$D_{i(23)}$	$D_{i(24)}$	$D_{i(25)}$	$D_{i(34)}$	$D_{i(35)}$	$D_{i(45)}$
1	0.048	0.157	0.056	0.076	0.109	0.008	0.028	−0.101	−0.081	0.020
2	−0.143	−0.229	−0.200	−0.114	−0.086	−0.057	0.029	0.029	0.114	0.086
3	0.501	0.436	0.497	0.590	−0.065	−0.004	0.089	0.062	0.154	0.092
4	0.015	0.038	0.527	0.341	0.023	0.513	0.327	0.490	0.304	−0.186
5	0.144	0.290	0.263	0.323	0.146	0.120	0.180	−0.027	0.033	0.060
6	−0.092	0.034	−0.072	−0.053	0.127	0.020	0.039	−0.107	−0.088	0.019
7	0.222	0.167	0.222	0.222	−0.056	0.000	0.000	0.056	0.056	0.000
8	0.187	0.071	0.014	0.527	−0.116	−0.173	0.340	−0.057	0.456	0.513
Median	0.096	0.114	0.139	0.273	−0.016	0.004	0.064	0.001	0.085	0.040

Table 6
Contrast estimation based on medians among methods.

Dataset	EMD -HFCM	Wavelet -HFCM	ANFIS	Multiresolution AR model	ANN
EMD-HFCM	0	0.1332	0.1268	0.1452	0.2168
Wavelet-HFCM	−0.1332	0	−0.0064	0.0120	0.0836
ANFIS	−0.1268	0.0064	0	0.0184	0.0900
Multiresolution AR model	−0.1452	−0.0120	−0.0184	0	0.0716
ANN	−0.2168	−0.0836	−0.0900	−0.0716	0

References

[1] E. Kayacan, B. Ulutas, O. Kaynak, Grey system theory-based models in time series prediction, Expert Syst. Appl. 37 (2) (2010) 1784–1789.

[2] G. Jha, K. Sinha, Time-delay neural networks for time series prediction: An application to the monthly wholesale price of oilseeds in India, Neural Comput. Appl. 24 (3–4) (2014) 563–571.

[3] D. Faruk, A hybrid neural network and ARIMA model for water quality time series prediction, Eng. Appl. Artif. Intell. 23 (4) (2010) 586–594.

[4] C. Wu, K. Chau, Prediction of rainfall time series using modular soft computing methods, Eng. Appl. Artif. Intell. 26 (3) (2013) 997–1007.

[5] Y. Lv, Y. Duan, W. Kang, Z. Li, F. Wang, Traffic flow prediction with big data: A deep learning approach, IEEE Trans. Intell. Transp. Syst. 16 (2) (2014) 865–873.

[6] H. Yang, H. Xu, Wavelet neural network with improved genetic algorithm for traffic flow time series prediction, Int. J. Light Electron Opt. 127 (19) (2016) 8103–8110.

[7] F. Alvarez, A. Troncoso, J. Riquelme, J. Ruiz, Energy time series forecasting based on pattern sequence similarity, IEEE Trans. Knowl. Data Eng. 23 (8) (2011) 1230–1243.

[8] I. Rojas, O. Valenzuela, F. Rojas, A. Guillen, L.J. Herrera, H. Pomares, L. Marquez, M. Pasadas, Soft-computing techniques and ARMA model for time series prediction, Neurocomputing 71 (4) (2008) 519–537.

[9] O. Valenzuela, I. Rojas, F. Rojas, H. Pomares, L.J. Herrera, A. Guillen, L. Marquez, M. Pasadas, Hybridization of intelligent techniques and ARIMA models for time series prediction, Fuzzy Sets and Systems 159 (7) (2008) 821–845.

[10] C. Liu, S. Hoi, P. Zhao, J. Sun, Online ARIMA algorithms for time series prediction, in: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, 2016, pp. 1867–1873.

[11] F. Gaxiola, P. Melin, F. Valdez, O. Castillo, Generalized type-2 fuzzy weight adjustment for backpropagation neural networks in time series prediction, Inform. Sci. 325 (2015) 159–174.

[12] M. Khashei, M. Bijari, An artificial neural network (p, d, q) model for timeseries forecasting, Expert Syst. Appl. 37 (1) (2010) 479–489.

[13] G. Rubio, H. Pomares, I. Rojas, L. Herrera, A heuristic method for parameter selection in LS-SVM: Application to time series prediction, Int. J. Forecast. 27 (3) (2011) 725–739.

[14] M. Kane, N. Price, M. Scotch, P. Rabinowitz, Comparison of ARIMA and random forest time series models for prediction of avian influenza H5N1 outbreaks, BMC Bioinformatics 15 (1) (2014) 1–9.

[15] G. Felix, G. Nápoles, R. Falcon, W. Froelich, K. Vanhoof, Rafael Bello, A review on methods and software for fuzzy cognitive maps, Artif. Intell. Rev. 52 (2019) 1707–1737.

[16] J. Salmeron, E. Papageorgiou, Fuzzy grey cognitive maps and nonlinear hebbian learning in process control, Appl. Intell. 41 (1) (2014) 223–234.

[17] S. Sacchelli, S. Fabbrizzi, Minimisation of uncertainty in decision-making processes using optimised probabilistic Fuzzy Cognitive Maps: A case study for a rural sector, Socio Econ. Plann. Sci. 52 (2015) 31–40.

[18] P.K. Dhanji, S.K. Singh, Fuzzy cognitive maps based game balancing system in real time, Indonesian J. Electr. Eng. Comput. Sci. 9 (2) (2018) 335–341.

[19] J. Salmeron, W. Froelich, Dynamic optimization of fuzzy cognitive maps for time series forecasting, Knowl.-Based Syst. 105 (2016) 29–37.

[20] J. Liu, Y. Chi, C. Zhu, A dynamic multiagent genetic algorithm for gene regulatory network reconstruction based on fuzzy cognitive maps, IEEE Trans. Fuzzy Syst. 24 (2) (2016) 419–431.

[21] W. Stach, L. Kurgan, W. Pedrycz, Numerical and linguistic prediction of time series with the use of fuzzy cognitive maps, IEEE Trans. Fuzzy Syst. 16 (1) (2008) 61–72.

[22] W. Froelich, E. Papageorgiou, M. Samarinas, K. Skriapas, Application of evolutionary fuzzy cognitive maps to the long-term prediction of prostate cancer, Appl. Soft Comput. 12 (12) (2012) 3810–3817.

[23] W. Lu, J. Yang, X. Liu, The modeling and prediction of time series based on synergy of high-order fuzzy cognitive map and fuzzy c-means clustering, Knowl.-Based Syst. 70 (2014) 242–255.

[24] S. Yang, J. Liu, Time-series forecasting based on high-order fuzzy cognitive maps and wavelet transform, IEEE Trans. Fuzzy Syst. 26 (6) (2018) 3391–3402.

[25] N. Huang, Z. Shen, S. Long, M. Wu, H. Shih, Q. Zheng, N. Yen, C. Tung, H. Liu, The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis, Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci. (1998) 903–995.

[26] D. Mackay, Bayesian interpolation, Neural Comput. 4 (3) (1992) 415–447.
[27] H. Song, C. Miao, W. Roel, Z. Shen, F. Catthoor, Implementation of fuzzy cognitive maps based on fuzzy neural network and application in prediction of time series, IEEE Trans. Fuzzy Syst. 18 (2) (2010) 233–250.
[28] W. Froelich, J. Salmeron, Evolutionary learning of fuzzy grey cognitive maps for the forecasting of multivariate, interval-valued time series, Int. J. Approx. Reason. 55 (6) (2014) 1319–1335.
[29] E. Papageorgiou, K. Poczęta, Application of fuzzy cognitive maps to electricity consumption prediction, in: Proceedings of the Fuzzy Information Processing Society, 2015, pp. 1–6.
[30] E. Papageorgiou, K. Poczta, A two-stage model for time series prediction based on fuzzy cognitive maps and neural networks, Neurocomputing 232 (5) (2017) 113–121.
[31] W. Pedrycz, A. Jastrzebska, W. Homenda, Design of fuzzy cognitive maps for modeling time series, IEEE Trans. Fuzzy Syst. 24 (1) (2015) 120–130.
[32] P. Hajek, O. Prochazka, W. Froelich, Interval-valued intuitionistic fuzzy cognitive maps for stock index forecasting, in: Proceedings of IEEE Conference on Evolving and Adaptive Intelligent Systems, 2018, pp. 1–7.
[33] G. Feng, W. Lu, W. Pedrycz, J. Yang, X. Liu, The learning of fuzzy cognitive maps with noisy data: A rapid and robust learning method with maximum entropy, IEEE Trans. Cybern. (2019) http://dx.doi.org/10.1109/TCYB.2019.2933438.
[34] F. Shen, J. Liu, K. Wu, Evolutionary multitasking fuzzy cognitive map learning, Knowl.-Based Syst. (2019) http://dx.doi.org/10.1016/j.knosys.2019.105294.
[35] W. Wang, K. Chau, L. Qiu, Y. Chen, Improving forecasting accuracy of medium and long-term runoff using artificial neural network based on EEMD decomposition, Environ. Res. 139 (2015) 46–54.
[36] D. Mandic, N. Rehman, Z. Wu, N. Huang, Empirical mode decomposition-based time-frequency analysis of multivariate signals: The power of adaptive data analysis, IEEE Signal Process. Mag. 30 (6) (2013) 74–86.
[37] B. Mijović, M.D. Vos, I. Gligorijević, J. Taelman, S.V. Huffel, Source separation from single-channel recordings by combining empirical-mode decomposition and independent component analysis, IEEE Trans. Biomed. Eng. 57 (9) (2010) 2188–2196.
[38] C. Cheng, L. Wei, A novel time-series model based on empirical mode decomposition for forecasting TAIEX, Econ. Model. 36 (1) (2014) 136–141.
[39] A. Zeiler, R. Faltermeier, C. Puntonet, A. Brawanski, E.W. Lang, Sliding empirical mode decomposition for on-line analysis of biomedical time series, in: Proceedings of International Work-conference on Artificial Neural Networks, 2011, pp. 299-306.
[40] B. Kosko, Fuzzy cognitive maps, Int. J. Man-Mach. Stud. 24 (1) (1986) 65–75.
[41] A. Tsadiras, Using fuzzy cognitive maps for e-commerce strategic planning, in: Proceedings of the 9th Panhellenic Conference on Informatics, 2003.
[42] S. Bueno, J. Salmeron, Benchmarking main activation functions in fuzzy cognitive maps, Expert Syst. Appl. 36 (3) (2009) 5221–5229.
[43] X. Zou, J. Liu, A mutual information-based two-phase memetic algorithm for large-scale fuzzy cognitive map learning, IEEE Trans. Fuzzy Syst. 26 (4) (2017) 2120–2134.
[44] Y. Chi, J. Liu, Learning of fuzzy cognitive maps with varying densities using a multiobjective evolutionary algorithm, IEEE Trans. Fuzzy Syst. 24 (1) (2016) 71–81.
[45] K. Wu, J. Liu, Learning large-scale fuzzy cognitive maps based on compressed sensing and application in reconstructing gene regulatory networks, IEEE Trans. Fuzzy Syst. 25 (6) (2017) 1546–1560.
[46] K. Wu, J. Liu, Robust learning of large-scale fuzzy cognitive maps via the lasso from noisy time series, Knowl.-Based Syst. 113 (2017) 23–38.
[47] Y. Chen, L. Mazlack, A. Minai, L. Lu, Inferring causal networks using fuzzy cognitive maps and evolutionary algorithms with application to gene regulatory network reconstruction, Appl. Soft Comput. 37 (2015) 667–679.
[48] R. Neal, Bayesian Learning for Neural Networks, Springer, 1996.
[49] C. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.
[50] M.J. Rezaee, S. Yousefi, M. Babaei, Multi-stage cognitive map for failures assessment of production processes: An extension in structure and algorithm, Neurocomputing 232 (2017) 69–82.
[51] M. Mackey, L. Glass, Oscillation and chaos in physiological control systems, Science 197 (4300) (1977) 287–289.
[52] J. Jang, ANFIS: Adaptive-network-based fuzzy inference system, IEEE Trans. Syst. Man Cybern. 23 (3) (1993) 665–685.
[53] G. Zheng, J. Starck, J. Campbell, F. Murtagh, Multiscale transforms for filtering financial data streams, J. Comput. Intell. Finance 7 (1999) 18–35.
[54] O. Renaud, J. Starck, F. Murtagh, Wavelet-based combined signal filtering and prediction, IEEE Trans. Syst. Man Cybern. B 35 (6) (2005) 1241–1251.
[55] A. Geva, Scalenet-multiscale neural-network architecture for timeseries prediction, IEEE Trans. Neural Netw. 9 (6) (1998) 1471–1482.
[56] S. Garcia, A. Fernandez, J. Luengo, F. Herrera, Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power, Inform. Sci. 180 (10) (2010) 2044–2064.