

# 一种错误率可控的混沌时间序列区间预测算法

王迪<sup>†</sup>, 王萍, 石君志

(天津大学 电气自动化与信息工程学院, 天津 300072)

**摘要:** 针对高风险背景下的混沌时间序列区间预测问题, 首次将回声状态网络与一致性预测框架相结合, 提出基于两者的混沌时间序列区间预测算法. 该算法将回声状态网络的拟合能力与一致性预测区间的可靠性相结合, 使得最终的预测区间包含被预测值的频率或概率可以被显著性水平参数所控制, 即预测区间具有极高的可信度. 同时, 由于使用岭回归学习回声状态网络的输出权重, 使得算法在学习阶段对样本的留一交叉估计可以被快速地计算, 极大地缩短了一致性预测的学习时间. 理论分析表明, 所提出算法的时间复杂度等价于原始回声状态网络算法的时间复杂度, 即算法具有较快的计算速度. 实验表明, 所提出算法能够较精确地控制预测的错误率, 对噪声具有鲁棒性, 且预测区间比基于高斯过程的预测区间更加准确地刻画了被预测值的波动范围.

**关键词:** 递归神经网络; 回声状态网络; 一致性预测; 刀切法一致性预测; 混沌时间序列; 区间预测

中图分类号: TP273

文献标志码: A

## An interval prediction algorithm for chaos time series with controllable error rate

WANG Di<sup>†</sup>, WANG Ping, SHI Jun-zhi

(School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China)

**Abstract:** To tackle the chaos time series interval prediction problem with high risk, this paper proposes an algorithm combining the ideas of echo state network and conformal prediction. While inheriting the learning ability from echo state network, the algorithm is capable of predicting reliable intervals for chaos time series with the help of the learning framework of conformal prediction, as the error rate of the prediction intervals can be controlled by the preset parameter named significance level. Meanwhile, due to the use of ridge regression to train the output weights of echo state network, the leave-one-out estimates in the training set can be computed efficiently, which accelerates the learning process of the algorithm effectively. As such, from the point of view of algorithm complexity, it is shown that the time complexity of the whole algorithm is equivalent to that of echo state network, implying that the algorithm is as fast and of practical use as echo state network. The experimental results show that the proposed algorithm is empirically valid as an interval predictor, robust to noise, and the prediction intervals output by the proposed algorithm are more related to the variance of the predicted values than those output by Gaussian process.

**Keywords:** recurrent neural network; echo state network; conformal prediction; jackknife conformal prediction; chaos time series; interval prediction

## 0 引言

混沌是自然界中普遍存在的貌似随机的、非周期的、具有短期可预测而长期不可预测的现象. 虽然理想的混沌系统是确定性的, 但在混沌系统的分析结果和遍历理论的启发下, 诸多学者开始建议并尝试用概率和统计的语言描述确定的混沌系统中的“不确定性”<sup>[1-4]</sup>. 同时, 现实中的混沌系统在自身演化的过

程中, 往往会受到动力学噪声的干扰, 且为了分析和研究混沌系统所进行的测量和记录结果, 也会由于测量精度、截断误差等因素含有随机噪声<sup>[5]</sup>. 这些不确定性使得点预测(Point prediction)方法在诸如电力、机械系统的故障检测和金融混沌时间序列预测等具有高风险低容错的混沌序列预测当中很难被广泛应用. 因为在这些领域, 预测的可靠性是非常重要的, 不

收稿日期: 2017-11-15; 修回日期: 2018-03-30.

基金项目: 天津市青年基金项目(2016FH-0011).

责任编委: 曹进德.

作者简介: 王迪(1989—), 男, 博士生, 从事模式识别、神经网络、一致性预测的研究; 王萍(1955—), 女, 教授, 博士生导师, 从事模式识别、图像识别、运动对象跟踪等研究.

<sup>†</sup>通讯作者. E-mail: wangdi2015@tju.edu.cn.

可靠的预测常常会伴随着巨大的损失.这就要求预测器的每一个预测结果的错误率都能够被事先指定的参数所控制,而点预测无法满足这样的需求.在这种背景下,需要构造可靠的预测区间进行预测.所有以上因素均表明,用统计语言描述混沌时间序列的预测问题,并构造具有高可信度的预测区间,有其重要的理论和实践意义.

常见的预测区间的构造方法或过度依赖于模型假设,或只在渐近的情况下才能保证预测的可靠性,使得预测的结果在实际应用中可信度较低.而作为近些年来新提出的统计学习框架,一致性预测(Conformal prediction)很好地克服了这一问题.

一致性预测框架由 Vovk 等提出,其萌芽起源于20世纪90年代,受到 Kolmogorov 复杂性和半监督支持向量机的启发<sup>[6-7]</sup>.该框架可以建立在传统的点预测算法(称为底层算法)之上,以作为对底层算法的补充和提升.一致性预测框架的主要的理论成果成型于2005年,之后在理论和应用上得到了长足发展<sup>[8-9]</sup>.目前,一致性预测区间的可靠性已被许多实验和应用所检验和证实,是构造高可信度预测区间的非常实用和有效的非参数方法.在一致性预测的框架下,一些统计上的非参数预测问题得到了突破<sup>[10-12]</sup>.

一致性预测框架的提出主要是为了解决分类和回归问题.为了将其可靠的预测结果应用到混沌时间序列的预测中,需要一种模型将时序预测问题转化为回归问题,进而通过一致性预测框架给出被预测值的区间预测,而回声状态网络正是这样一种模型.

回声状态网络(Echo state network, ESN)是训练递归神经网络的一种方法,最早由 Jaeger 于2001年提出,自提出后便成为混沌时间序列建模的主流方法之一<sup>[13-14]</sup>.其基本思想是将原始混沌时间序列信号的时序特征转换成递归神经网络隐藏层高维状态向量的空间特征,进而通过隐藏层状态向量分量的线性加权构建出输出信号,得到输入信号与输出信号之间的复杂关系.即 ESN 将时间序列预测转换成以当前时刻的隐藏层状态向量为自变量,对应的输出信号为因变量回归模型,并利用岭回归算法进行拟合,得到最终的预测模型.通过这种视角,本文将 ESN 最后的岭回归过程作为一致性预测的底层计算过程,将 ESN 模型与一致性预测相结合,即得到本文的混沌时间序列区间预测算法.

综上,为了建立具有高可信度的混沌时间序列区间预测模型,本文首次将回声状态网络与一致性预测相结合,并利用岭回归快速计算留一交叉估计的特

性,极大地缩短整个算法的计算时间.通过算法复杂度分析,算法学习过程的计算复杂度等价于原始 ESN 的计算复杂度,具有较快的学习速度.经实验检验,所提出算法能够较精确地控制预测的错误率,且预测结果能够同时刻画被预测点的位置和波动信息,是一个可靠和有效的混沌时间序列区间预测算法.

### 1 回声状态网络

设由  $N$  个时间点得到的训练数据集为  $\{(\mathbf{u}(t), y(t)), t = 1, 2, \dots, N\}$ , 其中  $\mathbf{u}(t) \in \mathbf{R}^q$  为  $t$  时刻输入,  $y(t) \in \mathbf{R}$  为其对应输出.

回声状态网络通过如下方程组对时间序列进行建模:

$$\begin{cases} \mathbf{x}(t+1) = f(W_{in}\mathbf{u}(t+1) + W\mathbf{x}(t)), \\ y(t+1) = W_{out}^T\mathbf{x}(t+1). \end{cases} \quad (1)$$

其中:  $\mathbf{x}(t) \in \mathbf{R}^p$  为 ESN 在  $t$  时刻的隐层状态向量;  $W_{in} \in \mathbf{R}^{p \times q}$  为输入权值矩阵;  $W \in \mathbf{R}^{p \times p}$  为内部权值矩阵,  $W_{out} \in \mathbf{R}^p$  为输出权值矩阵,  $f$  为神经网络的激活函数,一般取双曲正切函数.

ESN 训练时,首先随机生成  $W_{in}$  和  $W$ . 其中,  $W_{in}$  的各元素在  $[-\gamma, \gamma]$  中随机取值,  $\gamma$  为输入变换系数;  $W$  的各元素先在  $[-1, 1]$  之间随机取值,同时保持一定程度的稀疏性,然后通过乘以一个尺度系数使得最终得到的  $W$  的谱半径小于 1<sup>[15]</sup>. 然后设置初始状态  $\mathbf{x}(0)$  (如令  $\mathbf{x}(0) = 0$ ), 并利用方程组 (1) 的第 1 个方程和  $\{\mathbf{u}(t), t = 1, 2, \dots, N\}$  得到隐层状态向量  $\{\mathbf{x}(t), t = 1, 2, \dots, N\}$ . 为了消除初始状态的影响,舍弃  $T_0$  时刻及之前的状态向量,仅用余下的状态向量构成矩阵  $X = [\mathbf{x}(T_0 + 1), \mathbf{x}(T_0 + 2), \dots, \mathbf{x}(N)]^T$ , 对应的输出构成向量  $\mathbf{y} = [y(T_0 + 1), y(T_0 + 2), \dots, y(N)]^T$ . 通过解如下岭回归问题:

$$\begin{aligned} \min \lambda \|W_{out}\|^2 + \sum_{t=T_0+1}^N \|\xi_t\|^2; \\ \text{s.t. } W_{out}^T\mathbf{x}(t) = y(t) - \xi_t, t = T_0 + 1, \dots, N. \end{aligned}$$

最终得到输出权值矩阵

$$\hat{W}_{out} = (X^T X + \lambda I)^{-1} X^T \mathbf{y}.$$

其中  $\lambda$  为正则化参数.

由于学习的过程使用了岭回归,使得上述回归模型的留一交叉估计存在闭式解,并可以很快得到.记  $\hat{f}_i$  为  $X$  去掉第  $i$  行  $\mathbf{x}(T_0 + i)^T$  且  $\mathbf{y}$  去掉  $y(T_0 + i)$  后得到的回归函数在  $\mathbf{x}(T_0 + i)$  上的估计值,即第  $i$  个留一交叉估计.  $\text{hat}_{ii}$  为  $X(X^T X + \lambda I)^{-1} X^T$  的第  $i$  行第  $i$  列的值.由文献[15]中定义 5.17 和定理 5.34 可知,由

于岭回归是线性光滑器 (Linear smoother), 其留一交叉估计可由如下公式得到:

$$\hat{f}_i = \frac{\hat{W}_{\text{out}}^T \mathbf{x}(T_0 + i) - y(T_0 + i) \times \text{hat}_{ii}}{1 - \text{hat}_{ii}}. \quad (2)$$

由式(2)可以看出, ESN 只需训练一次即可得到  $N - T_0$  个训练数据的留一交叉估计, 而不需要训练  $N - T_0$  次, 这大大加快了局部加权刀切法一致性预测的计算速度.

预测时, 对于  $\{\mathbf{u}(t), t = T+1, T+2, \dots, T+M\}$ , 若  $T = N$ , 则直接利用  $\mathbf{u}(t)$  和  $\mathbf{x}(t-1)$  通过方程组(1)得到  $\mathbf{x}(t)$  并预测  $t$  时刻输出为  $\hat{W}_{\text{out}}^T \mathbf{x}(t)$ ; 若  $T \neq N$ , 则首先重新设置初始状态  $\mathbf{x}(T)$ , 然后利用方程组(1)的第1个方程进行迭代, 并从  $T_1 + 1$  时刻开始进行预测, 这样做是为了消除初始状态对预测的影响.

## 2 一致性预测

本节介绍原始一致性预测框架及其变种——局部加权刀切法一致性预测<sup>[12]</sup>.

设训练集为  $\mathbf{z}^l = \{\mathbf{z}_i, i = 1, 2, \dots, l\} = \{(\mathbf{x}_i, y_i), i = 1, 2, \dots, l\}$ , 其中  $\mathbf{x}_i \in \mathbf{R}^p$  为输入,  $y_i \in \mathbf{R}$  为其对应的输出. 记  $\mathbf{z}_{(j)}^l = \{\mathbf{z}_i, i = 1, 2, \dots, j-1, j+1, \dots, l\} = \{(\mathbf{x}_i, y_i), i = 1, 2, \dots, j-1, j+1, \dots, l\}$ . 对于一个新来的测试输入  $\mathbf{x}_{l+1}$ , 一致性预测的目标是输出一个标签的集合  $C(\mathbf{x}_{l+1})$ , 使得  $y_{l+1}$  属于  $C(\mathbf{x}_{l+1})$  的概率大于  $1 - \epsilon$ , 其中  $\epsilon$  为人为设定的参数, 称为显著性水平参数, 用来控制预测的错误率.

### 2.1 原始一致性预测框架

原始一致性预测构造预测标签集合的方法如下.

首先对  $\mathbf{x}_{l+1}$  每一个可能的标签  $\tilde{y}$ , 产生增广数据集  $\mathbf{z}^l \cup \{(\mathbf{x}_{l+1}, \tilde{y})\}$ , 然后计算  $l + 1$  个不一致性得分

$$\alpha_i^{\tilde{y}} = A(\mathbf{z}_{(i)}^l \cup \{(\mathbf{x}_{l+1}, \tilde{y})\}, (\mathbf{x}_i, y_i)),$$

$$\alpha_{l+1}^{\tilde{y}} = A(\mathbf{z}^l, (\mathbf{x}_{l+1}, \tilde{y})),$$

其中  $i = 1, 2, \dots, l$ .

上式中  $A(S, \mathbf{z})$  是一个度量数据  $\mathbf{z}$  与数据集  $S$  的不一致程度的函数, 该函数值越大, 表明  $\mathbf{z}$  与数据集  $S$  越不一致.  $A(S, \mathbf{z})$  的选取往往与具体的应用有关.

对每一个  $\tilde{y}$ , 计算完不一致性得分后还要计算其  $p$  值, 公式如下:

$$p^{\tilde{y}} = \frac{|\{i \in \{1, 2, \dots, l\} | \alpha_i^{\tilde{y}} \geq \alpha_{l+1}^{\tilde{y}}\}| + 1}{l + 1}.$$

对于给定的显著性水平参数  $\epsilon$ , 一致性预测输出的预测集合为  $\{\tilde{y} | p^{\tilde{y}} > \epsilon\}$ , 即  $C(\mathbf{x}_{l+1}) = \{\tilde{y} | p^{\tilde{y}} > \epsilon\}$ . 一致性预测的相关理论表明, 若数据满足独立同分布假设, 则  $y_{l+1}$  在  $\{\tilde{y} | p^{\tilde{y}} > \epsilon\}$  中的概率大于  $1 - \epsilon$ .

对于以上的原始一致性预测框架, 每计算一次

$A(S, \mathbf{z})$ , 就要训练一次底层算法, 所以当预测可能的输出值个数较多时, 该计算框架的计算过程将非常耗时. 而对于回归问题, 由于可能的输出值个数是无穷的, 原始一致性预测需要对不同的底层算法额外设计不同的计算过程, 且计算过程都非常耗时.

为了解决原始计算框架计算速度慢的问题, 一些原始框架的变种被提出, 包括归纳一致性预测、交叉一致性预测<sup>[16]</sup>、基于 Bootstrap 的一致性预测<sup>[17]</sup> 和刀切法一致性预测<sup>[12]</sup>. 由于刀切法一致性预测的计算框架用到了训练样本的留一交叉估计值, 而基于岭回归的 ESN 可以快速地计算留一交叉估计, 将两者结合可以使整个算法的计算速度得到巨大提升.

### 2.2 刀切法一致性预测

刀切法一致性预测是最近被提出的原始一致性预测框架的变种, 目的是想要在尽量利用训练样本信息的情况下克服原始计算框架速度慢的缺陷<sup>[12]</sup>. 由刀切法一致性预测得到的预测区间利用了底层算法的留一交叉估计.

设  $\mu$  为一种回归算法,  $\hat{\mu}(\cdot)$  是该算法在训练集中得到的回归函数,  $\hat{\mu}_i$  是第  $i$  个样本的留一交叉验证估计. 对于回归问题, 一般取  $A(S, \mathbf{z})$  为  $S$  上得到的回归函数  $\hat{\mu}(\cdot)$  在  $\mathbf{x}$  上的估计值  $\hat{\mu}(\mathbf{x})$  与  $y$  的绝对误差, 即  $A(S, \mathbf{z}) = |y - \hat{\mu}(\mathbf{x})|$ . 故刀切法一致性预测则用留一交叉绝对误差  $r_i = |y_i - \hat{\mu}_i|$  作为  $(\mathbf{x}_i, y_i)$  上的不一致性得分的估计值  $\alpha_i$ . 以下为刀切法一致性预测的计算流程.

#### 算法1 刀切法一致性预测算法.

输入: 训练数据  $\mathbf{z}^l$ , 显著性水平  $\epsilon$ , 回归算法  $\mu$ ;

输出: 对于任意的  $\mathbf{x} \in \mathbf{R}^p$ , 输出预测区间  $C(\mathbf{x})$ .

Step 1: 在训练集  $\mathbf{z}^l$  上利用回归算法  $\mu$  得到回归函数  $\hat{\mu}(\cdot)$ ;

Step 2: 计算留一交叉绝对误差  $r_i = |y_i - \hat{\mu}_i|$ , 其中  $i = 1, 2, \dots, l$ ;

Step 3: 令  $\alpha_i = r_i$ ;

Step 4: 记  $d$  为  $\{\alpha_i, i = 1, 2, \dots, l\}$  中第  $k$  个最小的值, 其中  $k = \lceil l(1 - \epsilon) \rceil$ ;

Step 5: 对于任意的  $\mathbf{x} \in \mathbf{R}^p$ , 输出预测区间  $C(\mathbf{x}) = [\hat{\mu}(\mathbf{x}) - d, \hat{\mu}(\mathbf{x}) + d]$ .

刀切法一致性预测有两个优点: 一是其计算效率高于原始的一致性预测框架; 二是其产生的预测区间相对于其他一致性预测的改进算法更短<sup>[12]</sup>, 预测结果蕴含有更多的信息量. 一般而言, 由于用到了留一交叉估计, 刀切法一致性预测需要在训练集上训练  $l$  次底层算法. 但如果回归算法  $\mu$  是岭回归, 则只需要一次训练即可通过式(2)得到留一交叉估计, 加快

计算速度.

### 2.3 局部加权刀切法一致性预测

从刀切法一致性预测的计算流程可以看出,其输出区间的长度为定值,没有考虑输出结果的波动程度(即方差)随输入变化的情况.如果输出结果的方差依赖于输入,则输出区间的长度应与该方差相关,即方差越大,输出区间越大.局部加权刀切法一致性预测则是考虑了输出方差变化情况的刀切法一致性预测<sup>[12]</sup>.

本文算法采用局部加权刀切法一致性预测来输出预测区间.令 $\mu$ 、 $\hat{\mu}(\cdot)$ 和 $\hat{\mu}_i$ 的内涵同上, $\mu'$ 为另一个回归算法,用于估计预测结果的方差. $\hat{\mu}'(\cdot)$ 是用 $\mu'$ 在数据集 $\{(\mathbf{x}_i, \ln(r_i)^2), i = 1, 2, \dots, l\}$ 上得到的回归函数, $\hat{\mu}'_i$ 是对应的第 $i$ 个留一交叉估计.以下为局部加权刀切法一致性预测的计算流程.

#### 算法2 局部加权刀切法一致性预测算法.

输入: 训练数据 $\mathbf{z}^l$ ,显著性水平 $\epsilon$ ,两个回归算法 $\mu$ 、 $\mu'$ ;

输出: 对每个 $\mathbf{x} \in R^p$ ,输出预测区间 $C(\mathbf{x})$ .

Step 1: 在数据集 $\mathbf{z}^l$ 上用算法 $\mu$ 得到回归函数 $\hat{\mu}(\cdot)$ ;

Step 2: 计算留一交叉绝对误差 $r_i = |y_i - \hat{\mu}_i|$ ,其中 $i = 1, 2, \dots, l$ ;

Step 3: 在数据集 $\{(\mathbf{x}_i, \ln(r_i)^2), i = 1, 2, \dots, l\}$ 上用回归算法 $\mu'$ 得到回归函数 $\hat{\mu}'(\cdot)$ ;

Step 4: 计算 $\hat{\mu}'_i$ ,其中 $i = 1, 2, \dots, l$ ;

Step 5: 令 $\alpha_i = r_i \sqrt{e^{-\hat{\mu}'_i}}$ ;

Step 6: 记 $d$ 为 $\{\alpha_i, i = 1, 2, \dots, l\}$ 中第 $k$ 个最小值,其中 $k = \lceil l(1 - \epsilon) \rceil$ ;

Step 7: 对每一个 $\mathbf{x} \in R^n$ ,输出 $C(\mathbf{x}) = [\hat{\mu}(\mathbf{x}) - \sqrt{e^{\hat{\mu}'(\mathbf{x})}}d, \hat{\mu}(\mathbf{x}) + \sqrt{e^{\hat{\mu}'(\mathbf{x})}}d]$ .

以上计算过程中,回归算法 $\mu'$ 用于估计输出方差的对数,从而间接估计方差.之所以不直接估计方差是因为方差总是正的,而回归算法得到的回归函数可能会输出负值,从而无法估计方差的大小.通过取对数可以避免这类问题<sup>[15]</sup>,得到 $\hat{\mu}'(\cdot)$ 后,用 $e^{\hat{\mu}'(\cdot)}$ 即可估计方差.

与刀切法一致性预测不同之处在于,局部加权刀切法一致性预测用留一交叉绝对误差 $r_i = |y_i - \hat{\mu}_i|$ 与 $y_i$ 的标准差的估计值 $\sqrt{e^{\hat{\mu}'_i}}$ 的比值 $r_i \sqrt{e^{-\hat{\mu}'_i}}$ 作为不一致性得分的估计值 $\alpha_i$ ,相当于对不一致性得分进行了规范化.

局部加权刀切法一致性预测输出区间的长度是正比于 $\sqrt{e^{\hat{\mu}'(\mathbf{x})}}$ ,即正比于标准差的估计值,所以方差越大,输出区间越大.输出区间的长度还受到显著性

水平参数 $\epsilon$ 的控制, $\epsilon$ 越小,输出区间越大.

## 3 基于回声状态网络和局部加权刀切法一致性预测的时间序列区间预测算法

### 3.1 算法流程

由第2节可知,回声状态网络将时序建模问题转换成以隐层状态向量 $\mathbf{x}(t)$ 为自变量, $y(t)$ 为因变量的回归建模问题,故只需在回归建模过程中引入局部加权刀切法一致性预测,即可使得最终的模型是区间预测模型.

对于局部加权刀切法一致性预测中的两个回归算法 $\mu$ 、 $\mu'$ ,本文均使用岭回归算法.其中由 $\mu$ 得到的回归函数即为原始ESN算法得到的岭回归函数,其权重参数为 $\hat{W}_{out}$ ,对应的超参数为 $\lambda$ ,是需要调的参数.由 $\mu'$ 得到的回归函数其权重参数记为 $\hat{W}_r$ ,对应的岭参数取1.

下面令 $\mathbf{x}_i = \mathbf{x}(T_0 + i)$ , $y_i = y(T_0 + i)$ , $l = N - T_0$ ,回声状态网络与局部加权刀切法一致性预测结合后的算法流程如下.

算法3 基于回声状态网络和局部加权刀切法一致性预测的时间序列区间预测算法(本文算法).

输入: 训练数据 $\{(\mathbf{u}(t), y(t)), t = 1, 2, \dots, N\}$ ,显著性水平 $\epsilon$ ,隐层状态向量维数 $p$ ,正则化参数 $\lambda$ ;

输出: 对于输入序列 $\{\mathbf{u}(t), t = T + 1, T + 2, \dots, T + M\}$ ,给出各时刻 $y(t)$ 的预测区间 $C(t)$ .

Step 1: 随机生成 $W_{in}$ 和 $W$ ;

Step 2: 收集 $T_0$ 时刻后的隐层状态向量构成的矩阵 $\mathbf{x} = [\mathbf{x}(T_0 + 1), \mathbf{x}(T_0 + 2), \dots, \mathbf{x}(N)]^T$ 和对应的输出构成向量 $\mathbf{y} = [y(T_0 + 1), y(T_0 + 2), \dots, y(N)]^T$ ;

Step 3:  $\hat{W}_{out} = (X^T X + \lambda I)^{-1} X^T \mathbf{y}$ ;

Step 4: 利用式(2)计算留一交叉绝对误差

$$r_i = \left| y(T_0 + i) - \frac{\hat{W}_{out}^T \mathbf{x}(T_0 + i) - y(T_0 + i) \times \text{hat}_{ii}}{1 - \text{hat}_{ii}} \right|,$$

其中 $i = 1, 2, \dots, N - T_0$ ;

Step 5: 得到向量 $\mathbf{r} = [\ln(r_1^2), \ln(r_2^2), \dots, \ln(r_l^2)]^T$ ,并计算 $\hat{W}_r$

$$\hat{W}_r = (X^T X + I)^{-1} X^T \mathbf{r};$$

Step 6: 仍利用式(2)得到

$$\hat{\mu}'_i = \frac{\hat{W}_r^T \mathbf{x}(T_0 + i) - \ln(r_i^2) \times \text{hat}'_{ii}}{1 - \text{hat}'_{ii}},$$

其中 $i = 1, 2, \dots, l$ ,  $\text{hat}'_{ii}$ 为 $X(X^T X + I)^{-1} X^T$ 在第 $i$ 行第 $i$ 列的元素;

Step 7: 令 $\alpha_i = r_i \sqrt{e^{-\hat{\mu}'_i}}$ ;

Step 8: 记 $d$ 为 $\{R_i, i = 1, 2, \dots, l\}$ 中第 $k$ 个最小

值,其中  $k = \lceil l(1 - \epsilon) \rceil$ ;

Step 9: 当隐层状态向量  $\mathbf{x}(t)$  能够预测  $y(t)$  时(见第1节末),输出如下预测区间  $C(t)$ :

$$[\hat{W}_{\text{out}}^T \mathbf{x}(t) - \sqrt{e^{\hat{W}_{\text{out}}^T \mathbf{x}(t)} d}, \hat{W}_{\text{out}}^T \mathbf{x}(t) + \sqrt{e^{\hat{W}_{\text{out}}^T \mathbf{x}(t)} d}].$$

### 3.2 时间复杂度分析

以乘法为基本运算,分析本文算法学习和预测阶段的时间复杂度<sup>[18]</sup>.

在学习阶段,Step 3 和 Step 5 以及计算  $\text{hat}_{ii}$ ,  $\text{hat}'_{ii}$  的时间复杂度为  $\Theta(lp^2 + p^3)$ ; Step 4 和 Step 6 的时间复杂度为  $\Theta(lp)$ ; Step 7 的时间复杂度为  $\Theta(l)$ . 可知学习阶段的时间复杂度为  $\Theta(lp^2 + p^3) + \Theta(lp) + \Theta(l) = \Theta(lp^2 + p^3)$ ,与 Step 3 的时间复杂度等价,即与原始 ESN 算法学习阶段的时间复杂度等价.

在预测阶段,Step 9 的时间复杂度为  $\Theta(p)$ ,等价于原始 ESN 算法预测阶段的时间复杂度.

以上分析表明,本文算法在学习和预测阶段的时间复杂度等价于原始 ESN 算法的时间复杂度. 由于原始 ESN 算法计算速度较快,本文算法同样具有较快的计算速度.

综合算法流程和时间复杂度的分析可知,本文算法在不增加计算复杂度的同时,将 ESN 的预测结果扩展为区间预测,有助于已有的基于 ESN 的时间序列预测系统的性能提升.

## 4 实验分析

评价区间预测算法的指标主要有两个,预测错误率和预测区间的平均长度. 预测错误率是指在测试数据集上被预测的值不在预测区间的比率,预测区间的平均长度是指在测试数据集上预测区间的长度的平均值. 如果预测错误率小于或接近显著性水平参数  $\epsilon$ ,则表明预测结果能被  $\epsilon$  很好地控制,称具有这样性质的区间预测器是可靠的(Valid),具有可靠性(Validity). 在区间预测算法是可靠的前提下,若预测区间的平均长度越短,则表明预测区间所含有的被预测值的信息量越大,预测越有效(Efficient),此时的区间预测器具有有效性(Efficiency)<sup>[9]</sup>.

本节通过实验测试本文算法,并检验其可靠性和有效性.

实验分为两个部分. 实验1在 Mackey-Glass 混沌时间序列上对本文算法进行检验,检验内容包括算法的显著性水平参数  $\epsilon$  对预测错误率的控制能力以及预测错误率对噪声的鲁棒性. 实验2在多个混沌时间序列、含噪混沌时间序列和太阳黑子数时间序列资料上对本文算法进行测试,并与基于高斯过程(Gaussian process, GP)<sup>[19]</sup>的 ESN(GP-ESN)进行对比.

所有实验中的参数  $\lambda \in \{2^{-20}, 2^{-19}, \dots, 2^{20}\}$ ,并在训练上利用留一交叉验证进行选取. 对于 Stock 数据集,取  $T_0 = 10$ ,  $p = 1000$ ,其余数据集均取  $T_0 = 100$ ,  $p = 1000$ .  $W_{\text{in}}$  的各个元素值取自  $[-1, 1]$  上的均匀分布.  $W$  中 5% 的元素非零,谱半径为 0.9. 实验中的时间序列记为  $\{m(t)\}$ ,并令  $u(t) = [1 \ m(t)]^T$ ,  $y(t) = m(t + 1)$ ,即用序列在  $t$  时刻及其之前的值预测下一时刻的值,其中  $u(t)$  的第 1 个分量表示输入到隐层的信号有一个偏置项. 每组时间序列数据在用于测试算法之前,均被归一化到  $[0, 1]$ . 经过这样的预处理,预测区间的平均长度便能够直观地体现预测区间的大小与时间序列波动范围的比例关系.

### 4.1 利用 Mackey-Glass 混沌系统对本文算法进行检验(实验1)

本文采用的 Mackey-Glass 混沌时间序列对应的微分方程如下:

$$\frac{dm(t)}{dt} = \frac{0.3m(t-30)}{1 + m^{10}(t-30) - 0.1m(t)}.$$

该时间序列是检验混沌预测算法的常用序列<sup>[20]</sup>. 对该序列,所有实验均取  $m(1) = 1.2$ .

#### 4.1.1 显著性水平参数在本文算法中所起的作用

取 4000 个时间序列数据,其中前 2000 个数据用于训练,后 2000 个数据用于测试. 图 1 和图 2 分别为本文算法的预测错误率和预测区间的平均长度与显著性水平参数  $\epsilon$  的关系.

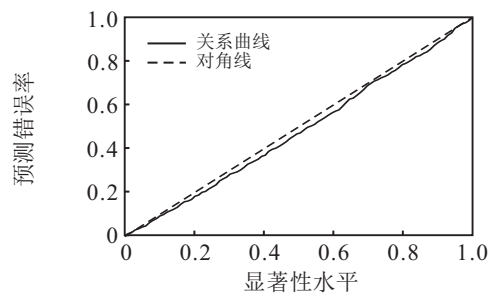


图1 预测错误率与显著性水平的关系

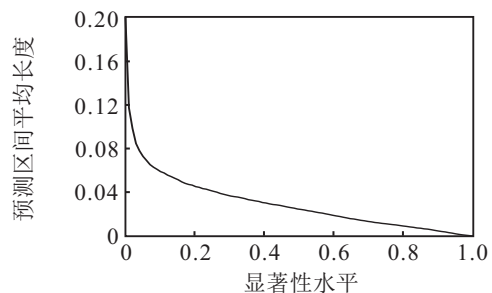


图2 预测区间的平均长度与显著性水平的关系

从图1可以看出,本文算法的预测错误率与  $\epsilon$  的关系曲线在对角线的下方靠近对角线,表明  $\epsilon$  对预测错误率有较精确的控制能力,本文算法的预测结果是

可靠的.从图2可以看出,为了保证预测错误率被 $\epsilon$ 所控制,预测区间的长度会随着显著性水平的减小而增大.同时,参数 $\epsilon$ 的控制作用具有强制性,即当 $\epsilon$ 增大时,预测区间长度会减小以保证错误率增大.实际应用本文算法时要对预测错误率和预测区间长度做出权衡,一般可取 $\epsilon$ 为0.1、0.05或0.01,以保证预测错误率被控制在10%、5%甚至是1%以下.

4.1.2 噪声对本文算法的影响

为研究噪声对本文算法的影响,分别对上一小节产生的时间序列加入噪声水平<sup>[5]</sup>为1%、1.5%、2%、...、5%的高斯白噪声,产生9组含噪时间序列.算法在该9组时间数列上的预测结果见表1和表2.其中MG(1.0%)指在Mackey-Glass混沌时间序列上加上噪声水平为1%的高斯白噪声得到的序列,其余记号以此类推.

表1 预测错误率对噪声的鲁棒性

	$\epsilon = 0.1$	$\epsilon = 0.05$	$\epsilon = 0.01$
MG(1.0%)	0.087	0.049	0.011
MG(1.5%)	0.084	0.039	0.010
MG(2.0%)	0.080	0.048	0.009
MG(2.5%)	0.102	0.040	0.011
MG(3.0%)	0.108	0.052	0.006
MG(3.5%)	0.105	0.050	0.011
MG(4.0%)	0.096	0.043	0.006
MG(4.5%)	0.097	0.044	0.008
MG(5.0%)	0.103	0.056	0.014

从表1中可以看出,本文算法的预测错误率对噪声具有抗性,预测错误率均被控制在 $\epsilon$ 左右,体现了预测错误率对噪声的鲁棒性.同时,表2表明为保证预测错误率接近 $\epsilon$ 或小于 $\epsilon$ ,预测区间的平均长度会随

噪声水平的增加而增加.这表明在 $\epsilon$ 给定的情况下,本文算法的预测区间长度能够反映被预测值的波动大小.

表2 噪声对预测区间平均长度的影响

	$\epsilon = 0.1$	$\epsilon = 0.05$	$\epsilon = 0.01$
MG(1.0%)	0.062	0.075	0.117
MG(1.5%)	0.072	0.087	0.124
MG(2.0%)	0.090	0.104	0.142
MG(2.5%)	0.095	0.114	0.155
MG(3.0%)	0.109	0.130	0.185
MG(3.5%)	0.122	0.149	0.201
MG(4.0%)	0.136	0.163	0.221
MG(4.5%)	0.154	0.188	0.239
MG(5.0%)	0.166	0.194	0.249

4.2 本文算法在多个混沌时间序列上的测试(实验2)

为更全面地测试本文算法,本节实验将其应用到多个混沌时间序列的预测问题中,并将实验结果与基于线性高斯过程的ESN的预测结果进行比较.其中基于线性高斯过程的ESN的 $W_{in}$ 、 $W$ 和正则化参数 $\lambda$ 与本文算法相同,而 $W_{out}$ 的确定方式及最终的输出区间是在高斯过程的框架下得到的.该算法的实现使用了python的机器学习程序包sklearn,这样的实验设计可以在尽量控制其他因素的条件下较全面地检验本文算法的可靠性和有效性.

4.2.1 实验数据集

实验用到4个模拟混沌时间序列和1个实测混沌时间序列,除了Mackey-Glass混沌时间序列生成自Matlab外,其余数据分别来源于R语言的自带数据集和R语言的fNonlinear包.所用数据相关细节见表3.

表3 实验所用数据集

数据名称	简记形式	数据说明	数据来源	训练数据个数	测试数据个数
Henon混沌时间序列	Henon	参见文献[21]	R语言fNonlinear包	2000	2000
Logistic混沌时间序列	Logistic	参见文献[21]	R语言fNonlinear包	2000	2000
Lorenz混沌时间序列	Lorenz	参见文献[20]	R语言fNonlinear包	2000	2000
Mackey-Glass混沌时间序列	MG	见5.1节	同5.1节	2000	2000
每月太阳黑子数	Sunspot	1749年~1997年每月太阳黑子数	R语言自带数据集	2000	1177
股票交易记录	Stock	石油和矿业股票交易记录	Time Series Data Library <sup>[1]</sup>	200	156

来源于R语言fNonlinear包的数据均取于混沌时间序列对应的生成函数,函数的参数取默认值.对于多元混沌系统,令每个分量的演化过程作为一组时间序列,如Lorenz混沌序列的 $x$ 分量的演化过程即为一

组时间序列,简记为Lorenz. $x$ ,其他多元序列各分量序列简记形式以此类推.为模拟含噪情况,分别在模拟产生的时间序列上加上噪声水平为5%的高斯白噪声.

4.2.2 实验结果与分析

表4、表5和表6分别为显著性水平 $\epsilon$ 取0.1、0.05和0.01时对应的实验结果.其中高斯过程的预测

结果是在假设被预测值是高斯分布的情况下得到的.两个算法的点预测值均取预测区间的中值,对于本文算法,该值为原始ESN的预测值.

表4 本文算法与GP-ESN的比较( $\epsilon = 0.1$ )

时间序列	本文算法			GP-ESN		
	预测错误率	预测区间平均长度	均方误差	预测错误率	预测区间平均长度	均方误差
Henon. $x$	0.092	$1.29 \times 10^{-4}$	$1.97 \times 10^{-9}$	0.629	0.054	$1.86 \times 10^{-9}$
Henon. $x$ (5%)	0.099	0.084	0.001	0.689	0.044	0.001
Henon. $y$	0.091	$1.24 \times 10^{-4}$	$1.84 \times 10^{-9}$	0.724	0.034	$1.78 \times 10^{-9}$
Henon. $y$ (5%)	0.112	0.082	0.001	0.252	0.135	0.001
Logistic	0.088	$1.41 \times 10^{-4}$	$2.17 \times 10^{-9}$	0.409	0.093	$2.07 \times 10^{-9}$
Logistic (5%)	0.100	0.129	0.002	0.786	0.033	0.002
Lorenz. $x$	0.117	$3.06 \times 10^{-4}$	$1.21 \times 10^{-8}$	0.471	0.076	$1.19 \times 10^{-8}$
Lorenz. $x$ (5%)	0.103	0.048	$2.20 \times 10^{-4}$	0.463	0.081	$2.20 \times 10^{-4}$
Lorenz. $y$	0.097	$5.70 \times 10^{-4}$	$3.10 \times 10^{-8}$	0.306	0.106	$3.11 \times 10^{-8}$
Lorenz. $y$ (5%)	0.103	0.044	$1.83 \times 10^{-4}$	0.484	0.081	$1.83 \times 10^{-4}$
Lorenz. $z$	0.101	$1.49 \times 10^{-4}$	$2.39 \times 10^{-9}$	0.308	0.110	$2.28 \times 10^{-9}$
Lorenz. $z$ (5%)	0.084	0.039	$1.31 \times 10^{-4}$	0.660	0.041	$1.31 \times 10^{-4}$
MG	0.091	0.056	$3.80 \times 10^{-4}$	0.587	0.061	$3.80 \times 10^{-4}$
MG (5%)	0.097	0.107	0.001	0.503	0.085	0.001
Sunspot	0.098	0.202	0.005	0.413	0.107	0.005
Stock	0.115	0.384	0.030	0.250	0.486	0.030

表5 本文算法与GP-ESN的比较( $\epsilon = 0.05$ )

时间序列	本文算法			GP-ESN		
	预测错误率	预测区间平均长度	均方误差	预测错误率	预测区间平均长度	均方误差
Henon. $x$	0.045	$1.47 \times 10^{-4}$	$1.97 \times 10^{-9}$	0.629	0.064	$1.86 \times 10^{-9}$
Henon. $x$ (5%)	0.051	0.100	0.001	0.681	0.052	0.001
Henon. $y$	0.049	$1.41 \times 10^{-4}$	$1.84 \times 10^{-9}$	0.724	0.041	$1.78 \times 10^{-9}$
Henon. $y$ (5%)	0.057	0.100	0.001	0.241	0.160	0.001
Logistic	0.047	$1.51 \times 10^{-4}$	$2.17 \times 10^{-9}$	0.409	0.111	$2.07 \times 10^{-9}$
Logistic (5%)	0.047	0.160	0.002	0.773	0.040	0.002
Lorenz. $x$	0.048	$3.96 \times 10^{-4}$	$1.21 \times 10^{-8}$	0.471	0.091	$1.19 \times 10^{-8}$
Lorenz. $x$ (5%)	0.054	0.057	$2.20 \times 10^{-4}$	0.459	0.097	$2.20 \times 10^{-4}$
Lorenz. $y$	0.043	$7.66 \times 10^{-4}$	$3.10 \times 10^{-8}$	0.306	0.126	$3.11 \times 10^{-8}$
Lorenz. $y$ (5%)	0.055	0.052	$1.83 \times 10^{-4}$	0.481	0.097	$1.83 \times 10^{-4}$
Lorenz. $z$	0.045	$2.22 \times 10^{-4}$	$2.39 \times 10^{-9}$	0.308	0.131	$2.28 \times 10^{-9}$
Lorenz. $z$ (5%)	0.035	0.047	$1.31 \times 10^{-4}$	0.658	0.049	$1.31 \times 10^{-4}$
MG	0.044	0.071	$3.80 \times 10^{-4}$	0.583	0.073	$3.80 \times 10^{-4}$
MG (5%)	0.046	0.128	0.001	0.489	0.101	0.001
Sunspot	0.058	0.246	0.005	0.352	0.128	0.005
Stock	0.076	0.487	0.030	0.212	0.579	0.030

表6 本文算法与GP-ESN的比较( $\epsilon = 0.01$ )

时间序列	本文算法			GP-ESN		
	预测错误率	预测区间平均长度	均方误差	预测错误率	预测区间平均长度	均方误差
Henon.x	0.007	$1.80 \times 10^{-4}$	$1.97 \times 10^{-9}$	0.629	0.084	$1.86 \times 10^{-9}$
Henon.x (5%)	0.010	0.134	0.001	0.672	0.069	0.001
Henon.y	0.009	$1.81 \times 10^{-4}$	$1.84 \times 10^{-9}$	0.724	0.053	$1.78 \times 10^{-9}$
Henon.y (5%)	0.009	0.139	0.001	0.230	0.211	0.001
Logistic	0.007	$1.70 \times 10^{-4}$	$2.17 \times 10^{-9}$	0.409	0.146	$2.07 \times 10^{-9}$
Logistic (5%)	0.005	0.241	0.002	0.758	0.052	0.002
Lorenz.x	0.008	$7.20 \times 10^{-4}$	$1.21 \times 10^{-8}$	0.471	0.120	$1.19 \times 10^{-8}$
Lorenz.x (5%)	0.014	0.074	$2.20 \times 10^{-4}$	0.456	0.128	$2.20 \times 10^{-4}$
Lorenz.y	0.006	0.001	$3.10 \times 10^{-8}$	0.306	0.166	$3.11 \times 10^{-8}$
Lorenz.y (5%)	0.004	0.075	$1.83 \times 10^{-4}$	0.478	0.127	$1.83 \times 10^{-4}$
Lorenz.z	0.003	$3.32 \times 10^{-4}$	$2.39 \times 10^{-9}$	0.308	0.172	$2.28 \times 10^{-9}$
Lorenz.z (5%)	0.008	0.062	$1.31 \times 10^{-4}$	0.656	0.064	$1.31 \times 10^{-4}$
MG	0.010	0.122	$3.80 \times 10^{-4}$	0.578	0.096	$3.80 \times 10^{-4}$
MG (5%)	0.009	0.175	0.001	0.474	0.133	0.001
Sunspot	0.012	0.352	0.005	0.260	0.168	0.005
Stock	0.019	0.87	0.030	0.16	0.76	0.030

从实验结果中可以得到如下结论:

1) 本文算法对预测错误率具有很好的控制能力,并对噪声具有鲁棒性,这些结果验证了算法的可靠性.

2) 本文算法和GP-ESN的拟合能力几乎相同,但区间预测的结果有很大差异.由于高斯过程对先验信息的过度假设,使得预测区间无法真实反映被预测值的波动大小,预测错误率较高.与此相反,本文算法对被预测值的波动信息来源于对训练样本留一交叉绝对误差的估计以及对训练样本方差的估计,能通过统计信息更客观地反映被预测值的波动大小.

3) 在可靠性被验证的前提下,本文算法在某些时间序列上的预测区间的平均长度小于GP-ESN算法的预测区间的平均长度,体现了本文算法的有效性.

4) 本文算法对预测错误率的控制能力说明,一旦增大或减小显著性参数,就会有一些边缘点被移出或移入预测区间,这侧面反映了算法的预测区间是比较紧致地围绕在被预测值附近的.

为进一步检验以上结论,图3对本文算法( $\epsilon = 0.05$ )在每月太阳黑子数时间序列中的预测结果进行了可视化,其中的数据取自第2500~2800个序列值.可以看出:被预测值基本在预测区间内;预测区间的上下限随着数据的波动而波动;预测区间的上下限与数据的包络线很贴近,体现了预测区间的有效

性;被预测值即便不再预测区间内,仍较近地分布在预测区间周围.以上观察均支持上述论点,其余的时间序列预测的结果与图3类似,这里不予列出.

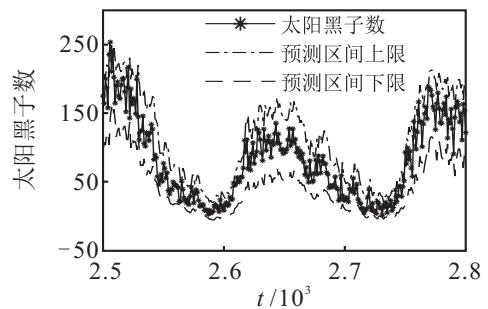


图3 每月太阳黑子数的预测区间

以上实验结果表明了本文算法具有可靠性,算法的预测区间可以同时刻画被预测值的位置和波动信息,算法在某些时间序列上的预测区间更短,预测结果更有效.

### 5 结论

针对混沌时间序列区间预测建模问题,本文首次将回声状态网络与一致性预测相结合,提出了基于回声状态网络和一致性预测的混沌时间序列区间预测算法.通过算法的复杂度分析可知,所提出算法具有较快的学习和预测速度.在多个混沌时间序列上的实验结果表明,所提出算法继承了一致性预测的可靠性特征,其预测错误率能够被较精确地控制,且对噪声

具有鲁棒性. 同时, 算法给出的预测区间能够较准确地刻画被预测值位置和波动的相关信息. 以上分析和实验结果表明, 所提出算法对混沌时间序列区间预测的研究和建模具有重要参考意义, 并在诸如电力、机械系统的故障检测和金融混沌时间序列预测等具有高风险低容错的混沌序列预测中具有极高的应用价值.

#### 参考文献(References)

- [1] Berliner L M. Statistics, probability and chaos[J]. *Statistical Science*, 1992, 7(1): 69-90.
- [2] González J A, Pino R. Chaotic and stochastic functions[J]. *Physica A: Statistical Mechanics and its Applications*, 2000, 276(3): 425-440.
- [3] González J A, Reyes L I, Suárez J J, et al. Chaos-induced true randomness[J]. *Physica A: Statistical Mechanics and its Applications*, 2002, 316(1): 259-288.
- [4] McGoff K, Mukherjee S, Pillai N. Statistical inference for dynamical systems: A review[J]. *Statistics Surveys*, 2015, 9(1): 209-252.
- [5] 韩敏. 混沌时间序列预测理论与方法[M]. 北京: 中国水利水电出版社, 2007: 103-104.  
(Han M. Prediction theory and method of chaotic time series[M]. Beijing: China Waterpower Press, 2007: 103-104.)
- [6] Vovk V, Gammerman A, Saunders C. Machine-learning applications of algorithmic randomness[C]. *Proc of the 6th Int Conf on Machine Learning*. Slovenia: Morgan Kaufmann Publishers Inc, 1999: 444-453.
- [7] Saunders C, Gammerman A, Vovk V. Transduction with confidence and credibility[C]. *Proc of the 6th Int Joint Conf on Artificial Intelligence*. San Francisco: Morgan Kaufmann Publishers Inc, 1999: 722-726.
- [8] Vovk V, Gammerman A, Shafer G. Algorithmic learning in a random world[M]. New York: Springer Science Business Media, 2005: 1-16.
- [9] Balasubramanian V, Ho S S, Vovk V. Conformal prediction for reliable machine learning: Theory, adaptations and applications[M]. Waltham: Morgan Kaufmann Publishers Inc, 2014: 1-46.
- [10] Lei J, Robins J, Wasserman L. Distribution-free prediction sets[J]. *J of the American Statistical Association*, 2013, 108(501): 278-287.
- [11] Lei J, Wasserman L. Distributionfree prediction bands for nonparametric regression[J]. *J of the Royal Statistical Society: Series B (Statistical Methodology)*, 2014, 76(1): 71-96.
- [12] Lei J, G'Sell M, Rinaldo A, et al. Distribution-free predictive inference for regression[J]. *J of the American Statistical Association*, 2017, DOI:10.1080/01621459.2017.1307116.
- [13] Jaeger H, Haas H. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication[J]. *Science*, 2004, 304(5667): 78-80.
- [14] Lukoeviius M, Jaeger H, Schrauwen B. Reservoir computing trends[J]. *KI-Künstliche Intelligenz*, 2012, 26(4): 365-371.
- [15] Wasserman L. All of nonparametric statistics[M]. New York: Springer, 2006: 66-85.
- [16] Vovk V. Cross-conformal predictors[J]. *Annals of Mathematics and Artificial Intelligence*, 2015, 74(1/2): 9-28.
- [17] Johansson U, Boström H, Löfström T, et al. Regression conformal prediction with random forests[J]. *Machine learning*, 2014, 97(1/2): 155-176.
- [18] Cormen T H, Leiserson C E, Rivest R L, et al. Introduction to algorithms[M]. London: The MIT Press, 2009: 43-65.
- [19] Rasmussen C E, Williams C K I. Gaussian processes for machine learning[M]. London: The MIT Press, 2006: 8-30.
- [20] 韩敏, 穆大芸. 回声状态网络LM算法及混沌时间序列预测[J]. *控制与决策*, 2011, 26(10): 1469-1472.  
(Han M, Mu D Y. LM algorithm in echo state network for chaotic time series prediction[J]. *Control and Decision*, 2011, 26(10): 1469-1472.)
- [21] 王新迎, 韩敏, 王亚楠. 含噪混沌时间序列预测误差分析[J]. *物理学报*, 2013, 62(5): 050504.  
(Wang X Y, Han M, Wang Y N. Analysis of noisy chaotic time series prediction error[J]. *Acta Physica Sinica*, 2013, 62(5): 050504.)
- [22] Hyndman R J. Time series data library[DB/OL]. (2018-01-01)[2018-01-15]. URL <https://datamarket.com/data/list/?q=provider:tsdl>.

(责任编辑: 郑晓蕾)