第 56 卷 第 9 期 2007 年 9 月
1000-3290/2007/56(09)/5111-08

物 理 学 报
ACTA PHYSICA SINICA

Vol.56, No.9, September, 2007

基于 Bernstein 多项式的自适应混沌时间序列预测算法*

闫 华 魏 平 肖先赐
(电子科技大学电子工程学院，成都 610054)
(2005 年 12 月 27 日收到；2006 年 12 月 19 日收到修改稿)

提出了利用 Bernstein 多项式对混沌时间序列的动力学方程进行建模的方法，并将该方法与递推最小二乘(RLS)算法相结合，从而可以自适应地逼近混沌时间序列的动力学特性，以达到预测的目的. 理论分析和仿真实验表明该方法对一些常见的混沌时间序列具有较高的预测精度和较理想的准确预测率. 由于 RLS 算法的收敛速度较快，因此该方法比较适合于对短混沌时间序列进行实时预测.

关键词：混沌，预测，Bernstein 多项式，RLS 算法
PACC：0545

1. 引 言

混沌现象的发现可以追溯到 1963 年，当时美国气象学家 Lorenz 对一个完全确定的常微分方程组用计算机做数值计算时，发现因初始值的微小差异却导致完全不同的结果[1]，后来这个方程组就命名为 Lorenz 方程组，这样，混沌现象伴随着它对初值的敏感性而被人们发现了. 最早使用"混沌"一词的则是 Li 和 Yorke，他们首次提出了"chaos"这个词[2]，并得到了科学界的认可. 从此，混沌现象便引起了科学界和工程界越来越广泛的关注.

近年来，混沌预测这一课题越来越受到人们的重视，这在很大程度上是因为混沌已经在各个领域得到了广泛的应用[3—9]. 混沌时间序列预测主要分为局域预测法、全局预测法和自适应预测法. 全局预测方法最初由 He 及 Aguirre 等人所提出[10,11]，这类方法将相空间轨迹中全部点作为拟合对象，找出其规律进而对轨迹的走向进行预测[12]. 局域预测方法由 Farmer 与 Sidorowich 提出[13]，其核心思想是选出相空间轨迹的最后一点的若干个邻近点进行拟合，以估计轨迹下一点的位置，然后从轨迹点的坐标中分离出所需要的预测值[12]. 自适应预测方法则是将以上两种方法与自适应算法相结合而产生的. 张家树和甘建超分别将全局预测方法和局域预测方法与最小均方(LMS)算法相结合，从而构造出自适应混沌时间序列预测算法[14,15]，这些方法的优点在于可以根据序列的变化而自动地调整模型参数以适应其变化，这就给实时混沌时间序列预测提供了一种新的有效的思路，但算法的收敛速度不够快依然是将算法用于实时预测环境的一个主要限制.

Bernstein 多项式是函数逼近理论中的一个经典多项式，近年来，Bernstein 多项式这一经典工具在工程领域得到了比较广泛的应用[16,17]，但尚未见到将其用于混沌序列预测领域的相关报道，本文提出了一种利用 Bernstein 多项式对混沌序列的动力学方程进行建模，并采用 RLS 算法对 Bernstein 多项式诸参数作自适应估计的新方法. 理论分析和仿真实验表明，该方法可以对一些常见的混沌序列进行预测，其预测精度较高且收敛速度也较快.

本文首先介绍一元 Bernstein 多项式及其性质，在此基础上进一步介绍 Bernstein 多项式的多元形式，随后定义了待定参数的 Bernstein 多项式并对其进行了简化，然后将简化后的多项式模型与 RLS 算法相结合，从而给出了基于 Bernstein 多项式的 RLS 自适应混沌时间序列预测算法，最后通过仿真实验验证了该算法的有效性.

* 新世纪优秀人才支持计划(批准号：NCET-05-0803)资助的课题.

2. 一元 Bernstein 多项式及其性质

设 P_m 表示次数不超过 m 的代数多项式集合，$C[a,b]$ 表示 $[a,b]$ 上全体连续函数的集合.

定义 1[18] 设 $f(x)\in C[0,1]$，$f(x)$ 的 $m(m\geqslant 1)$ 阶 Bernstein 多项式由下式给出：

$$\begin{aligned}B_m(f) &= B_m(f;x)\\ &= \sum_{k=0}^{m} f\left(\frac{k}{m}\right)C_m^k x^k(1-x)^{m-k},\end{aligned} \tag{1}$$

其中 $C_m^k=\dfrac{m!}{(m-k)!\,k!}$，显然有 $B_m(f)\in P_m$.

定理 1[18] 设 $f(x)\in C[0,1]$，则 $B_m(f)$ 满足

$$\lim_{m\to\infty}\max_{0\leqslant x\leqslant 1}|f(x)-B_m(f)|=0. \tag{2}$$

定理 1 表明 Bernstein 多项式可以逼近 $[0,1]$ 上的任意连续函数，此外，Bernstein 多项式还具有以下一些重要的性质：

性质 1[18] 设 $f^{(P)}(x)\in C[0,1]$，那么

$$\lim_{m\to\infty}B_m^{(P)}(f)=f^{(P)}(x), \tag{3}$$

在 $[0,1]$ 上一致地成立，其中 P 是一个固定的非负整数.

性质 1 表明 Bernstein 多项式可以对函数及其各阶连续导数进行联合逼近.

性质 2[18] 设 $f(x)\in C[0,1]$，P 是一个固定的整数，且有 $0\leqslant P\leqslant m$，若

$$\min\leqslant f^{(P)}(x)\leqslant\max,\ 0\leqslant x\leqslant 1, \tag{4}$$

那么

$$\begin{aligned}\min &\leqslant \frac{m^P}{m(m-1)\cdots(m-P+1)}B_m^{(P)}(f)\\ &\leqslant \max, 0\leqslant x\leqslant 1,\end{aligned} \tag{5}$$

特别地，如果 $P=0$，那么上式中 $B_m^{(P)}(f)$ 前的因数应是 1.

性质 2 表明利用 Bernstein 多项式对连续函数进行逼近时，它可以‘适应’函数的值域变化范围，与被逼近的函数处于同一值域范围内. 并且 Bernstein 多项式还具有以下重要特点：

性质 3[18] 设 $f(x)\in C[0,1]$，则有

1) 若 $f^{(P)}(x)\geqslant 0$，$x\in[0,1]$，则有 $B_m^{(P)}(f)\geqslant 0$，$x\in[0,1]$；

2) 若 $f(x)$ 在 $[0,1]$ 上是非减的，那么 $B_m(f)$ 在 $[0,1]$ 上也是非减的；

3) 若 $f(x)$ 在 $[0,1]$ 上是凸的，那么 $B_m(f)$ 在 $[0,1]$ 上也是凸的.

性质 3 表明对 $[0,1]$ 上的连续函数而言，单调的和凸的函数分别产生单调的和凸的 Bernstein 多项式逼近.

下面我们将一元 Bernstein 多项式进行推广，使之可以逼近 $[a,b]$ 上的任意连续函数.

定理 2 设 $f(x)\in C[a,b]$，则可修正 Bernstein 多项式为

$$\begin{aligned}B_m(f) &= B_m(f;x)\\ &= \sum_{k=0}^{m} f\left(a+\frac{k}{m}(b-a)\right)\\ &\quad\times C_m^k\left(\frac{x-a}{b-a}\right)^k\left(1-\frac{x-a}{b-a}\right)^{m-k},\end{aligned} \tag{6}$$

此时有

$$\lim_{m\to\infty}\max_{a\leqslant x\leqslant b}|f(x)-B_m(f)|=0. \tag{7}$$

证明 令 $x=a+y(b-a)$，则有

$$f(x)=f(a+y(b-a))=\phi(y), \tag{8}$$

因为 $y=\dfrac{x-a}{b-a}$，所以 $\phi(y)$ 是定义在 $[0,1]$ 上的连续函数，于是由定理 1 知，对 $B_m(\phi)=B_m(\phi;y)=\sum_{k=0}^{m}\phi\left(\frac{k}{m}\right)C_m^k y^k(1-y)^{m-k}$，任给 $\varepsilon>0$，存在 M，使得当 $m>M$ 时有

$$|\phi(y)-B_m(\phi;y)|<\varepsilon,\ \forall y\in[0,1], \tag{9}$$

即

$$|f(x)-B_m(f;x)|<\varepsilon,\ \forall x\in[a,b], \tag{10}$$

其中

$$\begin{aligned}B_m(f;x) &= B_m\left(\phi;\frac{x-a}{b-a}\right)\\ &= \sum_{k=0}^{m} f\left(a+\frac{k}{m}(b-a)\right)\\ &\quad\times C_m^k\left(\frac{x-a}{b-a}\right)^k\left(1-\frac{x-a}{b-a}\right)^{m-k}.\end{aligned} \tag{11}$$

由定理 2 可知，利用修正的 Bernstein 多项式可以对 $[a,b]$ 上的任意连续函数进行逼近.

所有上面的关于 Bernstein 多项式的各种性质说明这种多项式模拟被逼近函数的特性可以达到一种比较理想的程度，而这也正是我们选择 Bernstein 多项式对混沌序列动力学方程进行建模，从而达到预测目的的原因之所在.

3. 预测算法

本文的主要思路是先利用 Bernstein 多项式进行建模，然后再利用 RLS 算法使模型参数自适应地收

敛，由于 RLS 算法的收敛速度较快，因而该方法的建模速度也较快，关于这一点，也可以从后面的仿真实验中看到.

3.1. 多元待定参数的 Bernstein 多项式

下面首先介绍具有 n 个自变量的 Bernstein 多项式.

定义 2[19]　设函数 $f(x_1,\cdots,x_n)$ 在 n 维单位立方体上连续，定义如下多项式为 $f(x_1,\cdots,x_n)$ 的 n 元 m 阶 Bernstein 多项式：

$$B_m(f;x_1,\cdots,x_n)=\sum_{k_1=0}^{m}\cdots\sum_{k_n=0}^{m}f\left(\frac{k_1}{m},\cdots,\frac{k_n}{m}\right)\times p_{mk_1}(x_1)\cdots p_{mk_n}(x_n),\quad(12)$$

其中

$$p_{mk_i}(x_i)=C_m^{k_i}x_i^{k_i}(1-x_i)^{m-k_i},i=1,2,\cdots,n.\quad(13)$$

定理 3[19]　对任意定义在 n 维单位立方体上的 n 元连续函数 $f(x_1,\cdots,x_n)$ 而言，均有

$$\lim_{m\to\infty}B_m(f;x_1,\cdots,x_n)=f(x_1,\cdots,x_n),$$
$$0\leqslant x_i\leqslant 1,i=1,\cdots,n.\quad(14)$$

显然，定理 3 的结论容易推广到对定义在任意 n 维长方体 $[a_1,b_1]\times[a_2,b_2]\times\cdots\times[a_n,b_n]$ 上的任意 n 元连续函数也成立，这与定理 2 的证明过程类似，这里不再重复.

由以上的分析可知，利用 n 元 m 阶 Bernstein 多项式 $B_m(f;x_1,\cdots,x_n)$ 可以对定义在任意 n 维长方体 $[a_1,b_1]\times[a_2,b_2]\times\cdots\times[a_n,b_n]$ 上的任意 n 元连续函数 $f(x_1,\cdots,x_n)$ 进行逼近，但在实际预测时，由于混沌序列的动力学方程是未知的，因此我们可以考虑利用待定参数的 n 元 m 阶 Bernstein 多项式并结合自适应算法使模型参数收敛，于是我们有：

定义 3　定义待定参数的 n 元 m 阶 Bernstein 多项式为

$$B_m(x_1,\cdots,x_n\mid \boldsymbol{a})=\sum_{k_1=0}^{m}\cdots\sum_{k_n=0}^{m}a_{k_1k_2\cdots k_n}p_{mk_1}(x_1)\cdots p_{mk_n}(x_n),\quad(15)$$

其中 $\boldsymbol{a}=(a_{k_1k_2\cdots k_n},k_i=0,\cdots,m,i=1,\cdots,n)$ 为待定参数矢量，$p_{mk_i}(x_i)(i=1,2,\cdots,n)$ 的定义同(13)式.

定义 3 中的 Bernstein 多项式共有 $(m+1)^n$ 个参数，虽然在其参数取适当值的情况下，可以逼近定义在任意 n 维长方体 $[a_1,b_1]\times[a_2,b_2]\times\cdots\times[a_n,b_n]$ 上的任意 n 元连续函数，但是它的主要缺点在于参数过多，实际实现时的运算量过大，这一点对于实时处理来说是一个必须考虑的因素，在这里我们利用去除变量交叉相乘项的方法来减少其参数的个数，具体的做法如下：

定义 4　定义去除变量交叉相乘项后的待定参数 n 元 m 阶 Bernstein 多项式为

$$B_m(x_1,\cdots,x_n\mid \boldsymbol{H})=\sum_{i=1}^{n}\sum_{k_i=0}^{m}h_{i,k_i}p_{mk_i}(x_i),\quad(16)$$

其中 $\boldsymbol{H}=(h_{i,k_i},k_i=0,\cdots,m,i=1,\cdots,n)$ 为待定参数矢量.

(16)式就是我们在实际建模预测时采用的 Bernstein 多项式，对比(15)式与(16)式可以看到，(16)式中去掉了不同变量的交叉相乘项，其参数个数变为 $(m+1)\cdot n$ 个，这样一来则大大减少了多项式的参数个数，从而也就提高了算法的速度，这一点从下面的图 1 中也可以看出来. 图 1 给出了(16)式的预测模型的一个结构示意图，图中 $x_1,\cdots,x_n$ 是输入值，$\hat{x}_{n+1}$ 表示输出的预测值. 从图中可以清楚地看到，每个输入值分别通过一个通道后，将得到的输出值相加，而不同的输入值相互之间不产生交叉相乘项，即所有的交叉相乘项均被去掉了. 当然，这样做的代价必然是降低了该模型对连续函数的逼近能力，但从后面的仿真实验中可以看到，其预测效果从工程的角度看仍然是比较令人满意的.

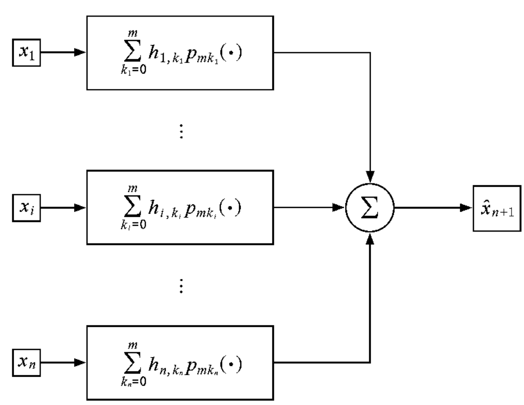

图 1　去除变量交叉相乘项后的 Bernstein 多项式结构示意图

3.2. 基于 Bernstein 多项式的 RLS 自适应混沌时间序列预测算法

RLS 算法是信号处理领域的经典收敛算法，由于它的收敛速度较快且收敛后的稳态误差较小而得

到了广泛的应用,本文将基于 Bernstein 多项式的建模方法与 RLS 算法相结合,得到了基于 Bernstein 多项式的 RLS 自适应混沌时间序列预测算法,其详细的算法步骤如下:

3.2.1. 给定关键参数及某些变量的初值

1)给定预测时所采用的过去的采样点的个数 n 以及每个采样点所对应的 Bernstein 多项式的阶数 m(对所有采样点均采用相同阶数的 Bernstein 多项式).

2)给定 RLS 算法中的两个重要参数 λ 与 δ 的取值,其中 λ 为遗忘因子.

3)构造 $1\times((m+1)\cdot n)$ 阶的加权行向量 $\boldsymbol{H}(i)$,将其初值 $\boldsymbol{H}(0)$ 置为零向量.

3.2.2. 利用 RLS 自适应算法调整加权行向量 $\boldsymbol{H}(i)$,以逼近混沌序列的未知动力学方程

1)利用得到的采样点构造 Bernstein 多项式的输入列向量 $\boldsymbol{U}(k,n)$,

$$\boldsymbol{U}(k,n)=[\boldsymbol{u}(k)\quad \boldsymbol{u}(k-1)\quad \cdots\quad \boldsymbol{u}(k-n+1)]^{\mathrm{T}},\tag{17}$$

其中

$$\begin{aligned}\boldsymbol{u}(i)=[&C_m^0x_i^0(1-x_i)^m\quad\cdots\quad C_m^{k_i}x_i^{k_i}(1-x_i)^{m-k_i}\\&\cdots C_m^mx_i^m(1-x_i)^0],\\&i=k,k-1,\cdots,k-n+1.\end{aligned}\tag{18}$$

将加权行向量 $\boldsymbol{H}(k)$ 与输入列向量 $\boldsymbol{U}(k,n)$ 相乘作为预测值:

$$\hat{x}_{k+1}=\boldsymbol{H}(k)\times\boldsymbol{U}(k,n).\tag{19}$$

在以上算法中,x_i 为 i 时刻的混沌序列真实值,$\hat{x}_{k+1}$ 表示对 $k+1$ 时刻的混沌序列取值的预测.

2)利用 RLS 算法调整权向量 $\boldsymbol{H}(i)$:

$$\boldsymbol{K}(k+1)=\frac{(1/\lambda)\bar{P}(k)\boldsymbol{U}(k,n)}{1+(1/\lambda)\boldsymbol{U}(k,n)^{\mathrm{T}}\bar{P}(k)\boldsymbol{U}(k,n)},\tag{20}$$

$$e_{k+1}=x_{k+1}-\hat{x}_{k+1},\tag{21}$$

$$\boldsymbol{H}(k+1)=\boldsymbol{H}(k)+e_{k+1}\boldsymbol{K}(k+1)^{\mathrm{T}},\tag{22}$$

$$\begin{aligned}\bar{P}(k+1)=&(1/\lambda)\bar{P}(k)-(1/\lambda)\boldsymbol{K}(k+1)\\&\times\boldsymbol{U}(k,n)^{\mathrm{T}}\bar{P}(k).\end{aligned}\tag{23}$$

在以上算法中,$\boldsymbol{K}(i)$ 为 $((m+1)\cdot n)\times 1$ 阶的列向量,$\bar{P}(i)$ 为 $((m+1)\cdot n)\times((m+1)\cdot n)$ 阶的方阵,其初值为

$$\bar{P}(0)=(1/\delta)\bar{I},\tag{24}$$

其中,$\bar{I}$ 为 $((m+1)\cdot n)\times((m+1)\cdot n)$ 阶的单位矩阵.

4. 仿真实验

4.1. 算法中四个重要参数的确定方法

该算法中所涉及的四个重要的参数分别是预测时所采用的过去的采样点的个数 n、每个采样点所对应的 Bernstein 多项式的阶数 m 以及 RLS 算法中的两个重要参数 λ 与 δ.

我们固然希望算法中的重要参数具有很好的鲁棒性,但是由于每一种非线性关系都具有很强的'个性",因而实际上很难做到使用统一的参数值来对各种不同的混沌序列进行有效的预测,这一点从后面给出的仿真结果也可以看到(见表 1 与表 2 中的参数参考取值),因此,也就有必要对如何确定算法中这四个重要的参数作一个简单的说明.

为了确定这四个重要参数的取值,一种比较简单的方法是逐步试探法,其具体做法如下:

步骤 1 将参数 λ 与 δ 分别设置为 0.99 与 0.1,这是因为从仿真的结果可以看出,这两个参数不是算法的敏感参数(见表 1 与表 2),因而可以直接取为固定的常数.

步骤 2 将参数 n 与 m 暂时均设置为 1,然后逐步增加其取值,每次改变它们的取值以后,就保持这两个参数的当前取值一段时间不变(如计算 200 个预测点所需的时间),同时计算这段时间的预测均方误差或准确预测率,以判断参数的改变是否改善了预测效果,直至其预测均方误差低于某一给定的门限或预测率超过某一给定的门限为止,即可停止调整这两个参数,否则应继续调整.

4.2. 一些常见混沌序列的参数参考取值及相应的仿真实验结果

首先给出仿真实验中将涉及到的混沌映射和混沌系统的方程.

仿真实验中涉及的离散混沌映射包括:

1)Hénon 映射

$$\begin{aligned}x_{n+1}&=1+y_n-1.4x_n^2,\\y_{n+1}&=0.3x_n.\end{aligned}\tag{25}$$

2)Kawakami 映射

$$\begin{aligned}x_{n+1}&=-0.1x_n+y_n,\\y_{n+1}&=x_n^2-1.6.\end{aligned}\tag{26}$$

3)Kent 映射

$$x_{n+1}=\beta-1-\beta|x_n|,\beta=1.9. \tag{27}$$

4) Logistic 映射

$$x_{n+1}=4x_n\cdot(1-x_n). \tag{28}$$

5) Lozi 映射

$$\begin{aligned} x_{n+1}&=1-1.75|x_n|+y_n,\\ y_{n+1}&=0.3x_n. \end{aligned} \tag{29}$$

6) Tent 映射

$$x_{n+1}=\begin{cases} x_n/a, & x_n\leqslant a\\ (1-x_n)/(1-a), & x_n>a \end{cases},$$
$$a=0.4. \tag{30}$$

仿真实验中涉及的连续混沌系统包括:

1) Chua 电路系统

$$\begin{aligned} &\dot{x}=\alpha(y-x-f(x)),\\ &\dot{y}=x-y+z,\\ &\dot{z}=-\beta y,\\ &f(x)=bx+0.5(a-b)(|x+1|-|x-1|),\\ &\alpha=10,\beta=14.87,a=-1.27,\\ &b=-0.68,\tau=0.1. \end{aligned} \tag{31}$$

2) Lorenz 混沌系统

$$\begin{aligned} &\dot{x}=a(y-x),\\ &\dot{y}=rx-xz-y,\\ &\dot{z}=xy-bz,\\ &a=10,b=8/3,r=28,\tau=0.09. \end{aligned} \tag{32}$$

3) Rŏssler 混沌系统

$$\begin{aligned} &\dot{x}=-y-z,\\ &\dot{y}=x+ay,\\ &\dot{z}=b+z(x-c),\\ &a=0.15,b=0.2,c=10,\tau=0.28. \end{aligned} \tag{33}$$

4) Wiem-Type 混沌系统

$$\begin{aligned} &\dot{x}=-x+2.5(y-z),\\ &\dot{y}=-x+1.5y-2.5z,\\ &\dot{z}=5u(y-1)-z,\\ &u(y)=\begin{cases}1, & y\geqslant 0,\\ 0, & y<0,\end{cases}\quad \tau=0.18. \end{aligned} \tag{34}$$

利用试探的方法,这里给出以上各种混沌映射与混沌系统产生的混沌序列所对应的参数参考取值.详见表 1 与表 2.

下面将利用表 1 与表 2 给出的参数的参考取值对这些常见的混沌序列作一步预测仿真实验.仿真时所有微分方程均采用 4 阶 Runge-Kutta 法求解,求解的步长见(31)式至(34)式中的参数 τ.所有的预测均针对 x 分量而言,并且 x 分量均已归一化至[0,1],每次实验均产生 2000 个真实值,表 3 给出了若干个利用后 1000 个真实值和预测值计算得到的反映预测效果的重要指标,其中均方误差(MSE)的计算公式为 $\mathrm{MSE}=\frac{1}{N}\sum_{k=1}^{N}(x_k-\hat{x}_k)^2$,相对均方误差(RMSE)的计算公式为

$$\mathrm{RMSE}=\left(\sum_{k=1}^{N}(x_k-\hat{x}_k)^2\right)\Big/\sum_{k=1}^{N}x_k^2,$$

准确预测率是以预测值 $\hat{x}_k$ 落入 $[x_k-0.025,x_k+0.025]$ 范围内就认为是准确预测而计算得到的,该区间长度占[0,1]长度的 5%.

表 1　一些常见离散混沌序列所对应的基于 Bernstein 多项式的 RLS 自适应混沌序列预测算法中四个重要参数的参考取值(混沌序列均已归一化至[0,1],若混沌序列有多个分量,则参数取值仅针对 x 分量而言)

混沌序列名称	参数 n	参数 m	参数 λ	参数 δ
Hénon	2	2	0.99	0.1
Kawakami	2	2	0.99	0.1
Kent	1	15	0.99	0.1
Logistic	1	2	0.99	0.1
Lozi	2	12	0.99	0.1
Tent	2	13	0.99	0.1

表 2　一些常见连续混沌序列所对应的基于 Bernstein 多项式的 RLS 自适应混沌序列预测算法中四个重要参数的参考取值(混沌序列均已归一化至[0,1],若混沌序列有多个分量,则参数取值仅针对 x 分量而言)

混沌序列名称	参数 n	参数 m	参数 λ	参数 δ
Chua 电路系统	2	3	0.99	0.1
Lorenz 混沌系统	3	3	0.99	0.1
Rŏssler 混沌系统	3	3	0.99	0.1
Wiem-Type 混沌系统	3	5	0.99	0.1

表 3　仿真实验结果中的一些重要指标

混沌序列名称	MSE	RMSE	准确预测率/%
Hénon	8.7723×10^{-15}	2.0090×10^{-14}	100
Kawakami	4.0055×10^{-15}	1.8425×10^{-14}	100
Kent	5.6135×10^{-5}	1.5444×10^{-4}	98.6
Logistic	7.5028×10^{-15}	1.9575×10^{-14}	100
Lozi	1.0462×10^{-4}	2.7913×10^{-4}	97.4
Tent	9.2661×10^{-5}	2.7455×10^{-4}	97.8
Chua 电路系统	9.7817×10^{-5}	2.6376×10^{-4}	96.4
Lorenz 混沌系统	1.3155×10^{-4}	4.7545×10^{-4}	96.3
Rŏssler 混沌系统	2.0072×10^{-4}	7.2175×10^{-4}	96.9
Wiem-Type 混沌系统	5.6443×10^{-5}	1.4974×10^{-4}	97.9

从表 3 中可以看到,本文提供的方法对这些常见的混沌序列均有较好的预测效果及较高的预测精度,从而验证了该方法用于混沌序列预测时的有效性.为了直观起见,下面的图 2 至图 4 将分别给出 Hénon,Tent 及 Chua 三种混沌系统的一步预测仿真图像及相应的误差平方曲线图.每次仿真共取 2000 个数据点,由于图像过密,故在所有的预测仿真图像中均是每隔 10 点标出 1 点作为示意(所有数据均已归一化至[0,1]),其中误差平方曲线图的纵坐标均设置为对数坐标.

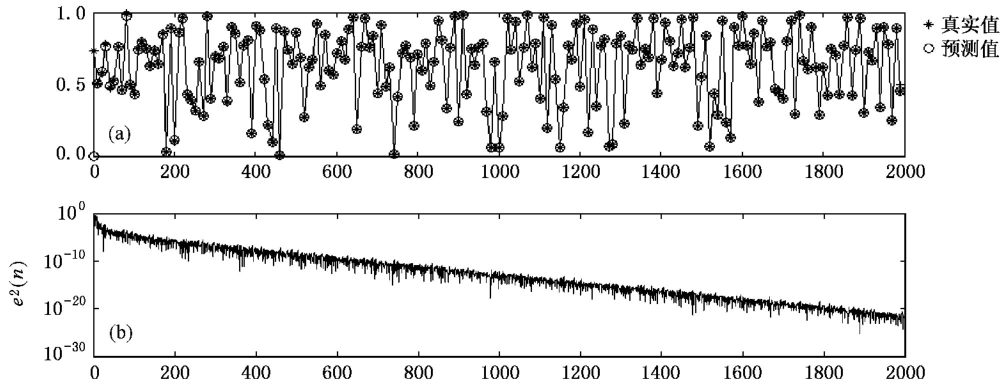

图 2 Hénon 混沌序列的一步预测仿真图像及相应的误差平方曲线图 (a)一步预测图像;(b)误差平方曲线

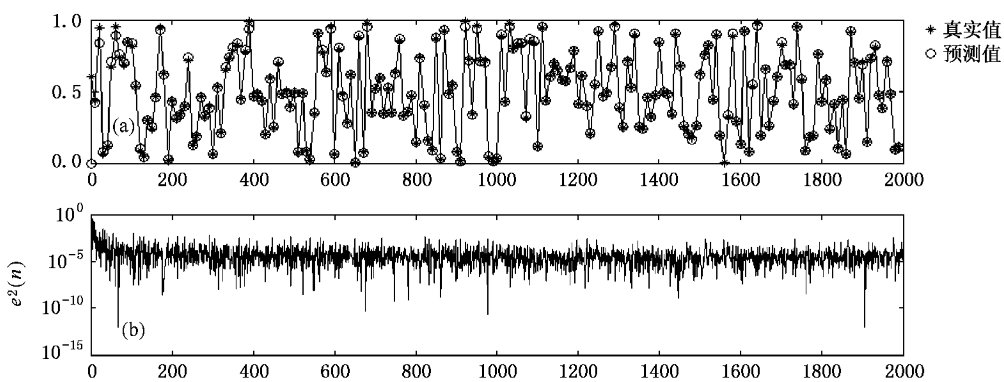

图 3 Tent 混沌序列的一步预测仿真图像及相应的误差平方曲线图 (a)一步预测图像;(b)误差平方曲线

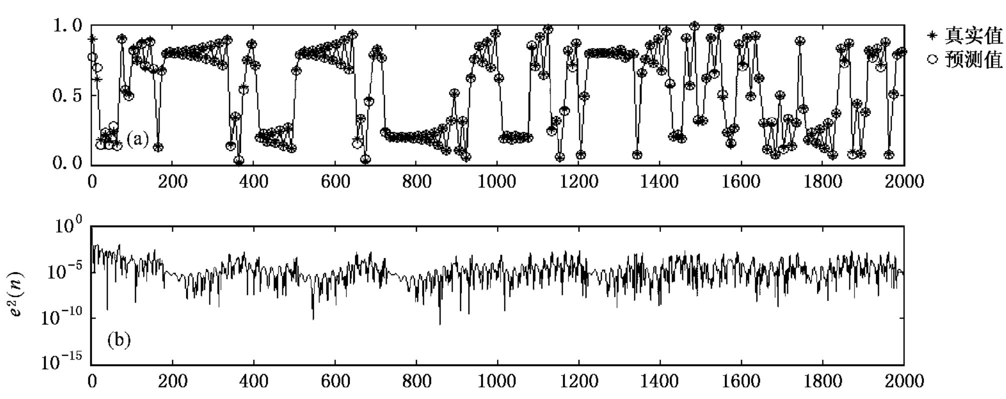

图 4 Chua 混沌序列的一步预测仿真图像及相应的误差平方曲线图 (a)一步预测图像;(b)误差平方曲线

从以上仿真实验的误差平方曲线图中可以看到,由于采用了 RLS 这种自适应收敛算法,该方法的收敛速度是比较快的,通常 200 个数据点左右即已经收敛,这也是由 RLS 算法自身的快速收敛特性

所决定的，因此该方法比较适合应用于实时混沌序列预测的环境中.

5. 结　　论

本文提出了基于 Bernstein 多项式的 RLS 自适应混沌时间序列预测算法. 该算法利用去除变量交叉相乘项的方法减少了 n 元 m 阶 Bernstein 多项式的参数个数，从而降低了算法的运算量. 仿真实验表明，该方法对一些常见的混沌序列提供了比较令人满意的预测效果，其准确预测率均可达 95% 以上，且 MSE 和 RMSE 均可以达到 10^{-4} 这一数量级水平. 该算法的收敛速度快，通常 200 个数据点左右即可收敛，因而即使是在数据点较少的情况下也可以对混沌时间序列的动力学方程进行逼近，以达到预测的目的. 该方法适合于对短混沌时间序列进行实时预测.

[1] Lorenz E N 1963 *J. Atmos. Sci.* **20** 130

[2] Li T Y，Yorke J A 1975 *Am. Math. Monthly* **82** 985

[3] Haykin S，Li X B 1995 *Proceedings of the IEEE* **83** 95

[4] Ling C，Sun S G 1998 *IEEE Trans. Comm.* **46** 1433

[5] Zhang J S，Xiao X C 2001 *Chin. Phys.* **10** 390

[6] Wang B Y 2004 *Chin. Phys.* **13** 329

[7] Lu J G，Xi Y G 2005 *Chin. Phys.* **14** 274

[8] Dou C X 2005 *Chin. Phys.* **14** 902

[9] Xie K，Lei M，Feng Z J 2005 *Acta. Phys. Sin.* **54** 1267（in Chinese）[谢　鲲、雷　敏、冯正进 2005 物理学报 **54** 1267]

[10] He X D，Lapedes A 1994 *Physica* D **70** 289

[11] Aguirre L A，Billings S A 1995 *Physica* D **85** 239

[12] Lü J H，Lu J A，Chen S H 2002 *Analysis and Applications of Chaotic Time Series*（Wuhan：Wuhan University Press）（in Chinese）[吕金虎、陆君安、陈士华 2002 混沌时间序列分析及其应用（武汉 武汉大学出版社）]

[13] Farmer J D，Sidorowich J J 1987 *Phys. Rev. Lett.* **59** 845

[14] Zhang J S，Xiao X C 2000 *Acta. Phys. Sin.* **49** 403（in Chinese）[张家树、肖先赐 2000 物理学报 **49** 403]

[15] Gan J C，Xiao X C 2003 *Acta. Phys. Sin.* **52** 1096（in Chinese）[甘建超、肖先赐 2003 物理学报 **52** 1096]

[16] Zarowski C J 1997 *IEEE Pacific Rim Conference on Communications，Computers and Signal Processing* **1** 477

[17] Mayer J 2002 *IEEE International Conference on Image Processing* **1** 824

[18] Mo G D，Liu K D 2003 *Methods of Approximation of Functions*（Beijing：Science Press）（in Chinese）[莫国端、刘开第 2003 函数逼近论方法（北京 科学出版社）]

[19] Liang X Z，Li Q 2005 *Multivariate Approximation*（Beijing：National Defense Industry Press）（in Chinese）[梁学章、李　强 2005 多元逼近（北京 国防工业出版社）]

An adaptive approach based on Bernstein polynomial to predict chaotic time series*

Yan Hua Wei Ping Xiao Xian-Ci
(*School of Electronic Engineering, University of Electronic Science and Technology of China, Chengdu* 610054, *China*)

(Received 27 December 2005; revised manuscript received 19 December 2006)

Abstract

In this paper, we propose an approach using the Bernstein polynomial to model the dynamics of chaotic time series. Combining it with RLS algorithm, we can predict the chaotic time series adaptively. Theoretical analysis and computer simulation have demonstrated that this approach can provide high precision and satisfactory percentage of prediction for some typical chaotic time series. Because of the fast convergence of RLS algorithm, this approach can be applied to predicting short record chaotic time series in real time.

Keywords: chaos, prediction, Bernstein polynomial, RLS algorithm
PACC: 0545

* Project supported by the Program for New Century Excellent Talents in University (Grant No. NCET-05-0803).