

5 时间序列的线性和非线性预测

5	时间序列的线性和非线性预测	1
5.1	引言.....	2
5.2	时间序列的线性预测方法.....	3
5.2.1	线性预测方法.....	3
5.2.2	基于 <i>AR</i> 模型的线性预测方法	5
5.2.3	基于 <i>ARMA</i> 模型的线性预测方法.....	8
5.3	基于相空间重构的预测方法.....	9
5.3.1	全域法预测.....	10
5.3.2	零阶局域法预测.....	11
5.3.3	一阶局域法预测.....	12
5.4	基于径向基神经网络的预测器.....	14
5.5	Volterra自适应预测器	17
5.6	海杂波的线性和非线性预测.....	22
5.7	本章小结.....	22
5.8	后记.....	23

5.1 引言

时间序列的建模和预测是信号处理研究领域的一个重要方向，其中预测方法主要有线性和非线性预测方法两种。所谓预测就是对未来作出估计和推断，这往往要对研究对象进行模仿或抽象，这一过程称为建模；用建模的方式获得对象的一种表示和体现就称为模型。混沌时间序列虽然貌似随机信号，但它却是由确定系统产生的，因此我们可以对混沌时间序列作出短期预测；同时由于混沌对初始条件微小偏差的高度敏感性，我们无法对混沌时间序列作出长期预测，每一步预测带来的微小偏差都会使得下一步预测的偏差扩大，最终导致预测序列与原始序列完全不同。

为了比较各种线性和非线性预测方法的性质，比较预测数据序列与观测数据的差异，文献[1]提出了一些度量预测效果好坏的 S_p 误差评价函数。将数据做归一化处理后，设数据点个数为 N ， i 为数据序号，则有 p_i 为预测数据值， p_m 为预测数据的平均值， O_i 为观测数据值， O_m 为观测数据的平均值， S_p 为预测数据的标准差， O_p 为观测值的标准差。以下四种误差评价函数从不同侧面和角度描述了误差特征^[1]。

■ 均方根误差 E_{RMSE}

$$E_{RMSE} = \left(\frac{1}{N-1} \sum_{i=1}^N [P_i - O_i]^2 \right)^{\frac{1}{2}}$$

均方根误差反映了预测值对观测值的平均偏离程度，取值大于或等于零，预测无误差时等于零。

■ 偏差 E_{Bias}

$$E_{Bias} = \frac{1}{N} \sum_{i=1}^N [P_i - O_i]$$

偏差反映了预测值对观测值的平均系统性的同向偏离程度，取值可正可负，取值为零表明平均系统性偏离为零，预测无误差时取零值。

■ 预测精度 E_{PA}

$$E_{PA} = \frac{\sum_{i=1}^N [(P_i - P_m)(O_i - O_m)]}{(N-1)S_p S_o}$$

预测精度反映了预测值在其均值附近的偏离与观测值在其均值附近的偏离之间的相关性，取值在 $[-1, +1]$ 之间。当 $E_{PA} = 1$ 时，表明预测曲线与观测值曲线分别对自身的均值和标准差归一化后是一样的。预测精度是预测曲线与观测曲线之间形状相似程度。

■ 决定度系数 E_{R^2}

$$E_{R^2} = \frac{\sum_{i=1}^N [P_i - O_m]^2}{\sum_{i=1}^N [O_i - O_m]^2}$$

决定度系数反映了预测数据对观测数据均值的偏离程度。预测无误差时等于 1，取值始终大于等于零。

本文主要研究了时间序列预测中的集中常用线性和非线性预测算法，其中重点是时间序列相空间后，如何使用线性和非线性方法进行预测^[2-4]。本章中各节主要内容如下：5.2节介绍了时间序列的线性预测方法，主要是基于 *AR* 和 *ARMA* 模型的预测；5.3节介绍了在相空间重构后的几种常用预测方法，包括全域法和局域法，局域法包括零阶和一阶模型；5.4节介绍了基于径向基神经网络的预测方法；5.5节介绍了基于Voterra级数自适应预测的方法；5.6节为各种预测方法在海杂波建模和目标检测中的应用；5.7节为本章小结。

5.2 时间序列的线性预测方法

信号处理通常是解决如何在噪声中提取信号的问题。因此需要设计一个模型，该模型有最佳线性过滤特性，当信号与噪声同时输入时，在输出端口能将信号尽可能精确地恢复出来，而噪声得到了最大的抑制。线性预测法是一种信号建模方法，它将信号的当前值看成是以前若干时刻采样值的线性组合。线性预测法主要用于滤波器设计，如语音信息编码、系统辨识和频谱分析等领域。

5.2.1 线性预测方法^[5]

线性预测的基本任务就是根据已知的前 p 次观测值来预测当时时刻信号 $x(n)$ 的估计值 $\hat{x}(n)$ 。这里，若假设预测值 $\hat{x}(n)$ 只是根据前一次采样数据 $x(n-1)$ 作出的，即称为一阶预测：

$$\hat{x}(n) = -ax(n-1) \quad (5.1)$$

其中 $-a$ 为一阶预测系数，这样预测误差可以写成

$$e(n) = x(n) - \hat{x}(n) = x(n) + ax(n-1) \quad (5.2)$$

根据最小均方误差准则，应使预测误差满足

$$E\{e^2(n)\} = E\{[x(n) + ax(n-1)]^2\} = \min error \quad (5.3)$$

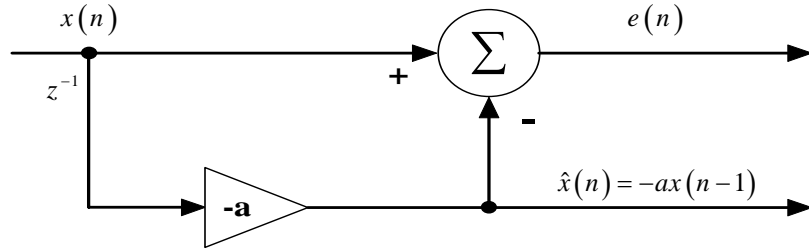
满足上式，获得的系数称为最佳预测系数。为此，对 a 求微分，并令

$$\frac{\partial E\{e^2(n)\}}{\partial a} = 2E\left[e(n)\frac{\partial e(n)}{\partial a}\right] = 2E[e(n)x(n-1)] \quad (5.4)$$

则 $2E[e(n)x(n-1)] = 0$ 时误差最小，并进一步写成

$$2E[e(n)x(n-1)] = E[\{x(n) + ax(n-1)\} \cdot x(n-1)] = \varphi_{xx}(1) + a\varphi_{xx}(0) = 0 \quad (5.5)$$

得到最佳预测系数为 $a_0 = -\varphi_{xx}(1)/\varphi_{xx}(0)$ 。这样一阶线性预测滤波器如下图所示：



图表 5-1：一阶线性预测器示意图

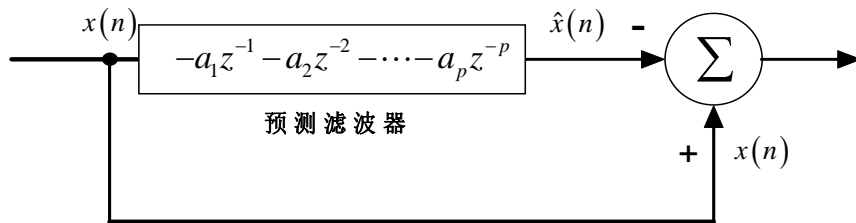
如果线性预测 $\hat{x}(n)$ 还参考了过去 p 个取值 $x(n-1), x(n-1), \dots, x(n-p)$ 来预测当前 $x(n)$ ，则构成 p 阶线性预测，表示为：

$$\hat{x}(n) = -a_{p1}x(n-1) - a_{p2}x(n-2) - \dots - a_{pp}x(n-p) = -\sum_{k=0}^p a_{pk}x(n-k) \quad (5.6)$$

上式称为预测滤波器，表明在预测当前值 $x(n)$ 时，参考了过去 p 个取样值， $-a_{pk}$ 称为第 k 个 p 阶预测系数，系数第一个下标 p 表示 p 阶。信号的预测值 $\hat{x}(n)$ 和真实值 $x(n)$ 之间存在一个预测误差，可以表示为：

$$e_p = x(n) - \hat{x}(n) = -\sum_{k=0}^p a_{pk}x(n-k) \quad (5.7)$$

上式为预测误差滤波器， a_{p0} 是常数 1。这样，任务就是根据输入信号序列 $x(n)$ 来产生预测序列 $\hat{x}(n)$ 和预测误差 $e_p(n)$ 。所以 p 阶线性预测器如图表 5-3 所示。



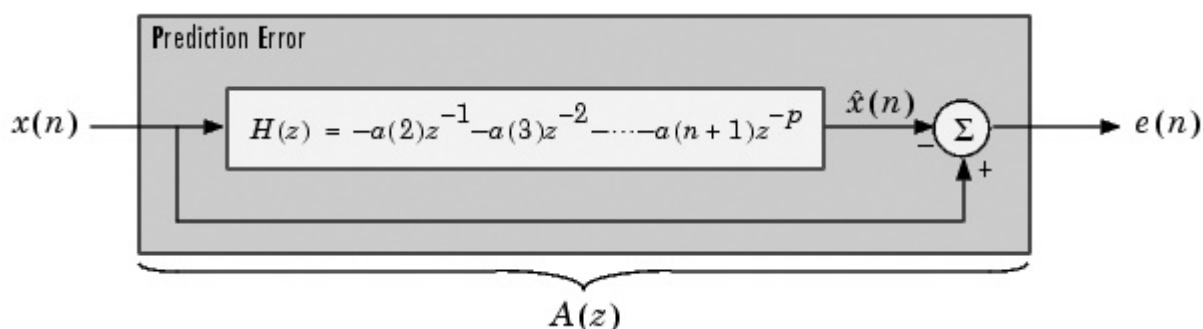
图表 5-2: p 阶线性预测器示意图

5.2.2 基于 AR 模型的线性预测方法

假设系统为 AR 模型，则每个输出 $x(n)$ 是过去的 p 个输出的线性组合：

$$y(n) = -\sum_{i=1}^p a_i x(n-i) + w(n) \quad (5.8)$$

其中 p 为 AR 模型阶数， $w(n)$ 为均值为 0，方差为 σ_w^2 的高斯白噪声，如图表 5-3 所示。



图表 5-3 基于 AR 模型的线性估计框图

由给定的时间序列求解 AR 模型的参数 a_i 和阶数 p 是进行线性预测的两个关键问题。对于给定一个系统输出序列，如果采用最小二乘法，就是要找出系统的参数 a_i 、阶数 p ，使得偏差序列 $w(n)$ 和使实际序列不包含任何相关性，是一个白噪声化的线性预测误差序列。设我们已经得到 $a_i, i=0,1,\dots,p$ ，计算得到估计值

$$\hat{x}(n) = -\sum_{i=1}^p a_i x(n-i) \quad (5.9)$$

然后使均方误差

$$E\{e^2(n)\} = E\{[x(n) - \hat{x}(n)]^2\} \quad (5.10)$$

取最小值。为此，令

$$\frac{\partial E\{e^2(n)\}}{\partial a_l} = 0, \quad l=1,2,\dots,p \quad (5.11)$$

得到

$$2E\left\{\left[x(n) + \sum_{i=1}^p a_i x(n-i)\right]x^*(n-l)\right\} = 0, \quad l=1,2,\dots,l \quad (5.12)$$

即

$$E\{e(n)\hat{x}^*(n)\} = 0 \quad (5.13)$$

于是得到：

$$\begin{aligned}
E\{e^2(n)\}_{\min} &= E\{e(n)[x^*(n) - \hat{x}^*(n)]^2\} = E\{e(n)x^*(n)\} \\
&= E\left\{\left[x(n) + \sum_{i=1}^p a_i x(n-i)\right]x^*(n)\right\}
\end{aligned} \tag{5.14}$$

可以将上式和下式

$$E\left\{\left[x(n) + \sum_{i=1}^p a_i x(n-i)\right]x^*(n-l)\right\} = 0, \quad l=1,2,\dots,p \tag{5.15}$$

用自相关函数表示, 有

$$\begin{cases} R_x(0) + \sum_{i=1}^p a_i R_x(i) = E\{e^2(n)\}_{\min} \\ R_x(l) + \sum_{i=1}^p a_i R_x(l-i) = 0, \quad l=1,2,\dots,p \end{cases} \tag{5.16}$$

写成矩阵形式, 得到关于系数 a_i 的方程就是功率谱估计的 Yule-Walker 方程, 形式如下:

$$\begin{bmatrix} R_x(0) & R_x(-1) & \cdots & R_x(-p) \\ R_x(1) & R_x(0) & \cdots & R_x(-(p-1)) \\ \vdots & \vdots & \cdots & \vdots \\ R_x(p) & R_x(p-1) & \cdots & R_x(0) \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} \sigma_w^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{5.17}$$

式中的 $R_x(n)$ 为信号序列的自相关序列。

Levinson-Durhlk 对 Yule-walker 方程提出了高效的递推算法, 该算法首先以 $AR(0)$ 和 $AR(1)$ 的模型参数作为初始条件, 计算 $AR(2)$ 的模型参数, 然后根据这些参数计算 $AR(3)$ 模型参数, 一直递推到 $AR(p)$ [6]。在上式中, $AR(1)$ 的模型参数为

$$a_{11} = -\frac{R_x(1)}{R_x(0)}, \quad \sigma_1^2 = (1 - |a_{11}|^2)R_x(0) \tag{5.18}$$

依此类推 $AR(k)$, $k=2,3,\dots,p$ 的参数为

$$a_{kk} = \frac{R_x(k) + \sum_{i=1}^{k-1} a_{k-1,i} R_x(k-i)}{\sigma_{k-1}^2} \tag{5.19}$$

其中有

$$a_{ki} = a_{k-1,i} + a_{kk} a_{k-1,k-i}, \quad i=1,2,\dots,k-1 \tag{5.20}$$

$$\sigma_k^2 = (1 - |a_{kk}|^2) \sigma_{k-1}^2, \quad \sigma_0^2 = R_x(0) \tag{5.21}$$

对于 AR 模型阶数 p 的选取, 由于实际的杂波功率并非理想的 AR 模型, 不能保证

前向预测均方误差 σ_k^2 一定收敛。此外根据Levison递推算法的性质，若功率谱函数符合真实 p 阶 AR 模型，则 $k \geq p$ 时前向预测均方误差 σ_k^2 等于 σ_p^2 不变，且在正确模型阶数时首先达到它的最小值。因此有文献提出，选择一个收敛因子 $\alpha (\alpha \leq 1)$ 作为约束条件，若 $\sigma_k^2 / \sigma_{k-1}^2 > \alpha$ ，则可认为 k 等于 p ，从而得到 AR 模型阶数的估计值^[7]。这种方法虽然能够保证滤波器收敛，但收敛因子的选择没有明确的准则，不同的收敛因子确定的阶数相差很大，存在着不确定性和主观性。

目前选取阶数 p 的方法主要是采用基于信息量准则的各种方法，信息量准则不需要选择主观参数来限定阶数的选择，并且具有明确的物理意义。各种信息量准则不仅监视了预测功率的变化情况，也考虑了模型阶数增加后谱估计方差的增大，是比较常用的阶数选择方法。下面介绍几种常用信息量准则以及它们的物理意义^[6]。

(1) Akaike 信息量准则(AIC)

$$AIC(k) = N \ln \sigma_k^2 + 2k \quad (5.22)$$

式中 σ_k^2 为 k 阶预测误差功率的估计值， N 是观察数据长度。当 $AIC(k)$ 取得最小值时，确定最佳自回归模型阶数 p 。

(2) 最终预测误差准则(FPE)

$$FPE(k) = \sigma_k^2 \left(\frac{N+k+1}{N-k+1} \right) \quad (5.23)$$

FPE 的基本思想是选择一个阶数，使得一步预测的平均误差最小，对于长观测数据 AIC 与 FPE 将得到相同的模型阶数。

(3) 最短描述准则(MDL)

$$MDL(k) = N \ln \sigma_k^2 + k \ln N \quad (5.24)$$

MDL 准则是根据使模型对应过程的概率分布最大可能逼近实际过程概率分布这一思想，用 Kullback_Leibler 信息测度来度量两个概率分布之间的距离。

(4) 自回归传递函数准则(CAT)

$$CAT(k) = \frac{1}{N} \sum_{i=1}^k \frac{1}{\tilde{\rho}_i} - \frac{1}{\tilde{\rho}_k}, \quad \text{其中 } \tilde{\rho}_i = \frac{N}{N-i} \rho_i \quad (5.25)$$

这个准则是把实际预测误差滤波器和相应的估计滤波器的均方误差之间的差值最小所对应的阶次作为最佳阶数。

5.2.3 基于 ARMA 模型的线性预测方法^[5]

在工程领域，AR 模型的应用比 MA 模型和 ARMA 模型更为广泛，这是因为 AR 模型参数辨识比较简单，实时性较好，且已证明 MA 模型和 ARMA 模型可由 AR 模型逼近。AR 模型的解法已经相当成熟，从早期的自相关法到后来的 Burg 方法等，而且许多实际系统大都可以用全极点模型来描述，即使有零点，也可以用增加极点数目的方法来接近。但对于有较多零点的系数，ARMA 模型(零极点模型)仍有广泛应用。

若离散随机过程 $\{x(n)\}$ 服从线性差分方程：

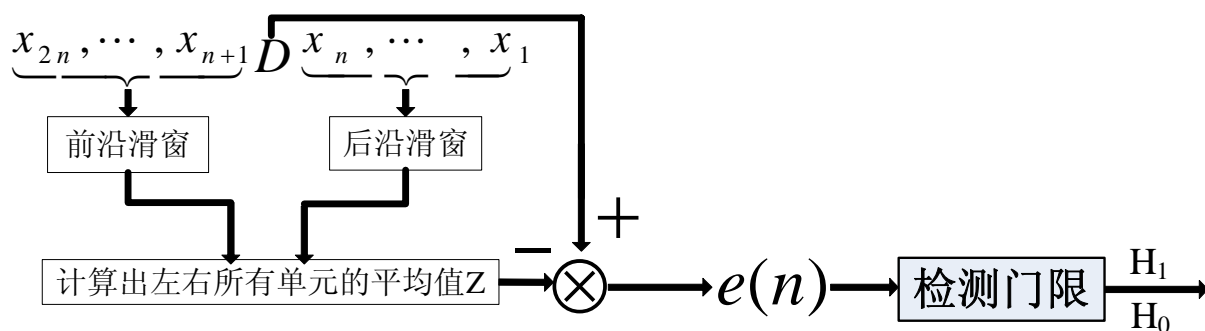
$$x(n) + \sum_{i=1}^p a_i x(n-i) = e(n) + \sum_{j=1}^q b_j e(n-j) \quad (5.26)$$

上式中的 $e(n)$ 是一离散白噪声，则称 $\{x(n)\}$ 为 ARMA 过程，而上式表示的差分方程被称为 ARMA 模型。系数 a_1, \dots, a_p 和 b_1, \dots, b_q 分别称为自回归(AR)参数和滑动平均(MA)参数， p 和 q 分别叫做 AR 阶数和 MA 阶数。

ARMA 模型描述的时不变系统的传递函数定义为

$$H(z) = \frac{B(z)}{A(z)} = \sum_{i=-\infty}^{\infty} h_i z^{-i} \quad (5.27)$$

式中 h_i 为系统的冲激响应系数，可见系统的极点 $A(z)=0$ 贡献为自回归，而系统零点 $B(z)=0$ 贡献为滑动平均。对于 ARMA 模型参数的估计算法，早期多采用非线性算法，目前较为成熟的方法是将 ARMA 问题转化为 AR 模型，求解 AR 模型后，再获得 ARMA 模型参数的估计。我们这里仅使用最简单的基于左右单元平均的 ARMA 模型预测器，这种预测方法和统计检测领域的单元平均类恒虚警检测器^[8]的设计相似，如下图表 5-4 所示



图表 5-4：左右单元平均预测和目标检测器

将前后各 n 个单元的平均值作为待求单元的平均值是一种非常简单的预测方法，也

可看做一种最简单的基于 $ARMA$ 模型的预测，则所求单元的预测值为

$$y(i) = \frac{1}{2 * n} \left(\sum_{p=-n}^{-1} x(i-p) + \sum_{q=1}^n x(i-q) \right) \quad (5.28)$$

由于基于单元平均的预测方法由于它的权值和阶数均为设定的固定值，因此它的预测效果可能不是非常理想，我们仅用它与各种线性和非线性预测算法作性能比较。

5.3 基于相空间重构的预测方法

传统的建模和预测方法已有广泛地应用，如自适应滤波器等，但它们都是基于线性系统的，并不适合于非线性的混沌时间序列。在鉴别时间序列性质以后，如果判定序列以随机性为主，应采用从 AR 或 $ARMA$ 等随机模型来预测，如果判定以混沌为主，则应采用非线性确定性的方法进行预测^[4]。非线性时间序列的预报是经过相空间重构后，对相空间中的动力轨迹进行预报。由于混沌系统具有被称为“蝴蝶效应”的初始值敏感性，所以混沌时间序列是不能长期预测的。但对于短时间内，混沌系统的轨道在相空间中具有一定的规律，它们的可预测性是可获得的，并比一般的基于统计方法的预测能力要好。

时间序列数据是时间离散的采样值，用 $\{x_n\}$ 表示，将它“嵌入”到一个 m (正整数) 维的空间中，进行相空间重构后得到嵌入向量

$$\mathbf{x}_n = [x_{n-(m-1)\tau}, x_{n-(m-2)\tau}, \dots, x_{n-\tau}, x_n] \quad (5.29)$$

由嵌入向量 \mathbf{x}_n ，可张成嵌入空间 \mathbf{E} ，由Takens嵌入定理可知，存在光滑映射 $f: \mathbf{R}^m \rightarrow \mathbf{R}$ 满足：

$$x_{n+\tau} = f(x_n, x_{n-\tau}, \dots, x_{n-(m-1)\tau}) \quad (5.30)$$

该式表明在拓朴等价意义上可以完全恢复和预测吸引子的特征，上式也可表示为：

$$\mathbf{x}_{n+1} = f(\mathbf{x}_n) \quad (5.31)$$

结合(5.29)和(5.31)式，但由于实际数据是有限的，并且往往含有噪声，因此不可能得到真正的 f ，而只能根据观测数据构造映射 $\hat{f}: \mathbf{R}^m \rightarrow \mathbf{R}$ 来拟合映射 f

$$x_{n+1} = \hat{f}(\mathbf{x}_n) \quad (5.32)$$

其中函数 \hat{f} 是 f 的一个估计，式(5.32)表明可以用 $x_{n-(m-1)\tau}, x_{n-(m-2)\tau}, \dots, x_{n-\tau}, x_n$ 来预测 x_{n+1} 。

在此基础上，人们提出了很多预测混沌时间序列的非线性方法。这些方法大致可分为：基于相空间的局部预测法和基于模型的全域预测法。

相空间局部预测法的基本思想是：在式(5.31)中映射 f 未知的情况下，假设映射 f 是连续变化的，也就是说 \mathbf{x}_{n+1} 和 \mathbf{x}_n 在相空间中是充分靠近的。给定当前状态 \mathbf{x}_N ，为了预

测下一个状态 \mathbf{x}_{N+1} ，我们遍历过去的所有状态，即 $\mathbf{x}_n (n < N)$ ，只要我们对系统已经观测了足够长的时间，那么在过去的状态中必然存在一个或多个充分接近 \mathbf{x}_N 的状态。假设 n_0 时刻的状态充分接近当前状态(在相空间中)，则存在一充分小的正实数 ε 使得

$$\|\mathbf{x}_N - \mathbf{x}_{n_0}\| < \varepsilon \quad (5.33)$$

且由于映射 \mathbf{g} 的连续变化性，有

$$\|\mathbf{x}_{N+1} - \mathbf{x}_{n_0+1}\| < \varepsilon \quad (5.34)$$

其中 $\|\bullet\|$ 算子表示范数。此时我们就可以对 \mathbf{x}_{N+1} 作出估计

$$\hat{\mathbf{x}}_{N+1} = \mathbf{x}_{n_0+1} \quad (5.35)$$

从而进一步估计出 x_{N+1} 。基于相空间的局部预测法的优点是无需有关原始动力学系统的先验知识，实现算法简单。但缺点是为保证相空间的状态遍历，需要的数据量较大，目前主要使用零阶和一阶局域法。基于模型的全域预测法的原理是使用所有的数据点直接估计式(5.31)中的函数 f 。从目前的应用看，主要有多项式、神经网络法和Voterra级数法等。

5.3.1 全域法预测

对 f 最简单的逼近就是取 \hat{f} 为 m 元 p 阶全域多项式模型拟合。多项式模型为

$$x_{n+\tau} = \hat{f}(\mathbf{x}_n) = \sum_i \mathbf{w} \Phi_i(\mathbf{x}_n) \quad (5.36)$$

其中 $\Phi_i(\bullet)$ 是向量幂乘和叉乘函数，权系数向量 \mathbf{w} 的微小变化将引起映射 \hat{f} 几乎处处都有变化，所以称这种方法为全域方法。多项式模型的缺点是独立参数的个数等于 $\binom{m+p}{p}$ ，为了提高预测精度，就要增大 p ，这时 $\binom{m+p}{p}$ 的值将会非常大，导致计算量大大增加；而且太多的独立参数又会使模型把噪声也拟合了，预测效果反倒不好。因此当维数 m 较低时，可以考虑使用较高阶的多项式进行全域拟合；而当维数 m 较高时，用高阶多项式就不再适用，我们更经常使用一阶多项式拟合。由于式(5.36)关于参数 \mathbf{w} 是完全线性的，所以这些参数可通过最小二乘拟合确定，一阶多项式逼近就是上节提到的自回归(AR)模型^[9]

$$x_{n+\tau} = \sum_{i=1}^m a_i x_{n-(m-1)\tau} + A \quad (5.37)$$

由于系统往往是非线性的，因此常令

$$A = k \xi_{n+\tau} \quad (5.38)$$

这里 A 表示系统未知因素的影响, $\xi_{n+\tau}$ 看成是一个服从 $N(0,1)$ 分布的高斯随机向量, k 是较小的定常因子, 用来调整随机性引入的强度。从时间序列本身来求出 $a_i (i=1,2,\dots,m)$, 并且尽量使 A 未知部分的影响最小, 则要求系统的 a_i 使得下面的误差平方和最小

$$Q_m = \sum_{n=m}^N \left(x_n - \sum_{i=1}^m a_i x_{n-i} \right)^2 \quad (5.39)$$

对每一个 a_i 求偏导, 并令它等于零, 则有

$$\sum_{i=1}^m a_i \left(\sum_{n=m}^N x_{n-i} x_{n-j} \right) = \sum_{i=m}^N x_i x_{i-j}, \quad j=1,2,\dots,m \quad (5.40)$$

将上式变形可得

$$\sum_{i=1}^m a_i C(i-j) = C(j), \quad j=1,2,\dots,m \quad (5.41)$$

其中 $C(i-j)$, $C(j)$ 是自相关函数, 根据式(5.41)可以求出系数 a_i 。再回到式(5.38)中考虑 A 部分, 可令

$$k = \sqrt{\frac{Q_m}{N-m}} \quad (5.42)$$

同时当取 ξ_n 服从正态分布的随机数时, 式(5.37)的预测能力较高。

全域法除了计算复杂, 高阶系统中计算量大的缺点外; 同时随着嵌入维的增大, 它的预测精度会迅速下降。原因在于嵌入维越高, 预测算法所利用的信息历史就越长, 而对于混沌时间序列, 历史对未来的影响是呈指数衰减的。因此, 全域法一般用在嵌入维较低、 f 不是很复杂, 同时噪声干扰比较小的情况下。在其他情况下, 可以用该模型为标准来评价和比较其他非线性预测技术的预测效果, 而在实际预测中使用局域法等更合适的预测方法。

5.3.2 零阶局域法预测

局域化方法的基本原理是只利用有限的 u 个邻近点处的信息, 而不是利用全部的样本点来拟 f 。局域法预测是把相空间轨迹的最后一点作为预报点, 对离预报点邻近的若干相点的关系作出拟和, 再对预测相点的未来演化规律进行预测。局域化方法有两种: 一种是直接将邻近样本点应用于预测; 另一种方法是在邻近点处拟合一个函数, 并用此函数来预测。根据邻近样本点预测 $x_{n+\tau}$ 的最简单的方法是寻找嵌入空间 \mathbf{R}^m 中距离 \mathbf{x}_n 最近的点。假定距离 \mathbf{x}_n 最近的点是 \mathbf{x}_{n1} , 那么它的下一个样本点

$$x_{n1+\tau} = f(\mathbf{x}_{n1}) \quad (5.43)$$

就可作为 $x_{n+\tau}$ 的预测值, 这种预测最为简单的方法是零阶局域法。Lorenz 曾首先使用这一思想来预报天气, 他通过分析数据(当时仅有 4000 幅天气图), 找出历史上与今天天气

最相近的某天的天气图，然后根据该天的第 2 天的天气图来预报明天的天气。这种方法等价于构造一个状态映射表，对原始的时间序列来说，就是寻找一个长为 m 且与 $x_n, x_{n-\tau}, \dots, x_{n-(m-1)\tau}$ 最“相似”的片段，用这个片段的下一个样本点 $x_{n+\tau}$ 作为的预测，所以这种方法又称为类比法。作为上述方法的改进，可以利用 S 个邻近点(或 u_n 邻域内)的映射平均值作为预测：

$$\hat{x}_{n+\tau} = \frac{1}{|u_n|} \sum_{x_j \in u_n} \psi_i \mathbf{x}_{ni+\tau} = \frac{1}{S} \sum_{i=1}^S \psi_i \mathbf{x}_{ni+\tau} \quad (5.44)$$

其中 ψ_i 为权值，除了简单取平均值以外，还可以采用距离加权或指数加权^[10]。上述方法为加权零阶局域法预测，多个数据点的平均效果降低了预测的不稳定性和噪声的影响，它的效果要好于简单的零阶局域法。

5.3.3 一阶局域法预测

实际应用和数值实验表明：一般情况下，局域法的预测效果要好于全域法；一阶局域法的预测效果要好于零阶局域法^[11, 12]。我们在这里仅讨论加权一阶局域法，设中心点 \mathbf{x}_n 的邻近点为 $\mathbf{x}_{ni}, i=1, 2, \dots, q$ ，并且到 \mathbf{x}_n 的距离为 d_i ，设 d_{\min} 是 d_i 中的最小值，定义点 \mathbf{x}_{ni} 的权值为：

$$P_i = \frac{\exp(-(d_i - d_{\min}))}{\sum_{i=1}^q \exp(-(d_i - d_{\min}))} \quad (5.45)$$

a 为参数，一般取 $a=1$ ，则一阶局域法线性拟合为：

$$\mathbf{x}_{ni+1} = a\mathbf{e} + b\mathbf{x}_{ni}, i=1, 2, \dots, q \quad (5.46)$$

其中 a 、 b 为拟合所需的实系数， \mathbf{e} 为一个 q 维向量， $\mathbf{e} = (1, 1, \dots, 1)^T$ ， \mathbf{x}_{ni+1} 是 \mathbf{x}_{ni} 演化一步后的相点。当嵌入维数 $m=1$ 时，应用加权最小二乘法有

$$\sum_{i=1}^q P_i (x_{ni+1} - a - bx_{ni})^2 = \min \quad (5.47)$$

解上述方程组即可得到系数 a 和 b ，将 a ， b 代入一步预测公式为 $\mathbf{x}_{n+1} = a\mathbf{e} + b\mathbf{x}_n$ ，即可得到演化一步后的相点预测值 \mathbf{x}_{n+1} ：

$$\mathbf{x}_{n+1} = (x_{n+1}, x_{n+1-\tau}, \dots, x_{n+1-(m-1)\tau}) \quad (5.48)$$

这里， \mathbf{x}_{n+1} 中前 $m-1$ 个元素为原序列中已知值，其第 m 个元素 x_{n+1} 即为原序列的一步预测值 \hat{x}_{n+1} ，此即加权一阶局域法一步预报模型。

加权一阶局域法一步预报模型的实质是在重构的相空间中找到与参考点最相似的 q 个相点，并根据这 q 个相点演化一步的规律进行一步预报。对于混沌时间序列，相空间

中一对最近邻随时间演化遵循的是一种指数规律 $e^{-\lambda t}$ ，参数 λ 是最大 Lyapunov 指数，体现了系统初始闭轨道的指数发散速率。因此，当嵌入维数 $m > 1$ ，且需进行 $k(k > 1)$ 步预测时，可以类似加权一阶局域法一步预报模型，根据这 q 个相点演化 k 步的规律进行 k 步预报。

具体推导如下^[11, 13]：

设中心点 \mathbf{x}_n 的参考向量集 $\{\mathbf{x}_{ni}\}, i=1, 2, \dots, q$ ，其演化 k 步后的相点集为 $\{\mathbf{x}_{ni+k}\}$ ，一阶局域线性拟合为：

$$\mathbf{x}_{ni+k} = a_k \mathbf{e} + b_k \mathbf{x}_{ni}, i=1, 2, \dots, q \quad (5.49)$$

根据加权最小二乘法有：

$$\sum_{i=1}^q P_i \left[\sum_{j=1}^m (x_{ni+k}^j - a_k - b_k x_{ni}^j)^2 \right] = \min \quad (5.50)$$

将上式看成是关于未知数 a_k ， b_k 的二元函数，两边求偏导得：

$$\begin{cases} \sum_{i=1}^q P_i \sum_{j=1}^m (x_{ni+k}^j - a_k - b_k x_{ni}^j) = 0 \\ \sum_{i=1}^q P_i \sum_{j=1}^m (x_{ni+k}^j - a_k - b_k x_{ni}^j) x_{ni}^j = 0 \end{cases} \quad (5.51)$$

化简得：

$$\begin{cases} a_k \sum_{i=1}^q P_i \sum_{j=1}^m x_{ni}^j + b_k \sum_{i=1}^q P_i \sum_{j=1}^m (x_{ni}^j)^2 = \sum_{i=1}^q P_i \sum_{j=1}^m x_{ni+k}^j x_{ni}^j \\ a_k m + b_k \sum_{i=1}^q P_i \sum_{j=1}^m x_{ni}^j = \sum_{i=1}^q P_i \sum_{j=1}^m x_{ni+k}^j \end{cases} \quad (5.52)$$

$$\text{写成矩阵形式为:} \quad \begin{pmatrix} coe1 & coe2 \\ m & coe1 \end{pmatrix} \begin{pmatrix} a_k \\ b_k \end{pmatrix} = \begin{pmatrix} e_k \\ f_k \end{pmatrix} \quad (5.53)$$

$$\text{其中:} \quad coe1 = \sum_{i=1}^q P_i \sum_{j=1}^m x_{ni}^j \quad (5.54)$$

$$coe2 = \sum_{i=1}^q P_i \sum_{j=1}^m (x_{ni}^j)^2 \quad (5.55)$$

$$e_k = \sum_{i=1}^q P_i \sum_{j=1}^m x_{ni+k}^j x_{ni}^j \quad (5.56)$$

$$f_k = \sum_{i=1}^q P_i \sum_{j=1}^m x_{ni+k}^j \quad (5.57)$$

则：

$$\begin{pmatrix} a_k \\ b_k \end{pmatrix} = \begin{pmatrix} coe1 & coe2 \\ m & coe1 \end{pmatrix}^{-1} \begin{pmatrix} e_k \\ f_k \end{pmatrix} \quad (5.58)$$

在编程计算中，系数 $coe1$ 中的项 $\sum_{j=1}^m x_{ni}^j$ 是参考向量 \mathbf{x}_{ni} 的元素和，

系数 $coe2$ 中的项为

$$\sum_{j=1}^m (x_{ni}^j)^2 = \mathbf{x}_{ni} \mathbf{x}_{ni}^T \quad (5.59)$$

e_k 中的项为

$$\sum_{j=1}^m x_{ni+k}^j x_{ni}^j = \mathbf{x}_{ni+k} \mathbf{x}_{ni}^T \quad (5.60)$$

f_k 中的项为

$$\sum_{j=1}^m x_{ni+k}^j \quad (5.61)$$

这样求取 a_k ， b_k 的公式为：

$$\begin{pmatrix} a_k \\ b_k \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^q P_i \sum_{j=1}^m x_{ni}^j & \sum_{i=1}^q P_i \mathbf{x}_{ni} \mathbf{x}_{ni}^T \\ m & \sum_{i=1}^q P_i \sum_{j=1}^m x_{ni}^j \end{pmatrix}^{-1} \begin{pmatrix} \sum_{i=1}^q P_i \mathbf{x}_{ni+k} \mathbf{x}_{ni}^T \\ \sum_{i=1}^q P_i \sum_{j=1}^m x_{ni+k}^j \end{pmatrix} \quad (5.62)$$

根据求得的 a_k 、 b_k ，代入 k 步预测公式(5.49)，即可得到演化 k 步后的相点预测值

$$\mathbf{x}_{n+k} = (x_{n+k}, x_{n+k-\tau}, \dots, x_{n+k-(m-1)\tau}) \quad (5.63)$$

这里， \mathbf{x}_{n+k} 中的第 m 个元素 x_{n+k} 即为原序列的 k 步预测值 \hat{x}_{n+k} 。

加权一阶局域法预报模型是根据参考相点预报下一个相点，对于一步预报而言，所得到的预测相点中前 $m-1$ 个元素实际上都是已知时间序列中的点，只有第 m 个元素才是未知的，这样可以将前 $m-1$ 个元素与已知时间序列中的相应点进行比较，得出 $m-1$ 个误差数据构成误差序列，将误差序列的一步平滑外推值(可用最小二乘法实现)作为预测相点第 m 个元素的误差并进行修正，即可得到精度更高的预测值。对于多步预报，需将得到的预测值添加到已知时间序列中，然后按上述思路进行误差修正即可。

由于重建相空间本身就是一种非线性的变换，所以即使局域预报函数为线性形式，本质上仍然是一种非线性预报。由于重建相空间本身就是一种非线性的变换，所以即使局域预报函数为线性形式，本质上仍然是一种非线性预报。

5.4 基于径向基神经网络的预测器

各类神经网络由于具有普遍的非线性函数逼近能力，而成为非线性时间序列预测的

主要工具之一。神经网络的基本模型有前馈型、自组织型和随机型网络等，而常用的前馈型网络有径向基网络(RBFN, Radial Basis Function Network)因其具有最佳逼近特性、训练方法快速易行、不存在局部极小等优点而得到更为广泛的实际应用。

径向基函数网络利用具有局部隆起的所谓径向基函数来做逼近或分类问题^[14]。它可以看作是一种前馈网络，所处理的信息在工作过程中逐层向前流动。虽然它也可以像BP网络那样利用训练样本作有教师学习，但是其更典型更常用的学习方法则与BP网络有所不同，综合利用了有教师学习和无教师学习两种方法。对于某些问题，径向基函数网络可能比BP网络精度更高。径向基函数可以写成

$$g(\mathbf{x}) = \sum_{i=1}^P \lambda_i \varphi(\|\mathbf{x} - c_i\|) \quad (5.64)$$

其中 $\mathbf{x} \in R^N$ 表示模式向量； $\{c_i\}_{i=1}^P \subset R^N$ 是基函数中心； λ_j 是权系数； φ 是选定的非线性基函数。式(5.64)可以看作是一个神经网络，输入层有 N 个单元，输入模式向量由此进入网络。隐层有 P 个单元，第 i 个单元的输入为 $h_i = \|\mathbf{x} - c_i\|$ ，输出为 $\varphi(h_i)$ 。输出层 1 个单元，输出为 $g(\mathbf{x})$ 。假设给定了一组训练样本 $\{x^j, y^j\}_{j=1}^J \subset R^N \times R^1$ ，当 y^j 只取有限个值(例如，取 0, 1 或 ± 1)时，可以认为是分类问题；而当 y^j 可取任意实数时，视为逼近问题。网络学习(或训练)的任务就是利用训练样本来确定输入层到隐层的权向量 c_i 和隐层到输出层的权系数 λ_i ，使得 $g(x^j) = y^j$, $j = 1, \dots, J$ 为此，当 $J = P$ 时，可以简单令 $c_i = x^i$, $i = 1, \dots, P$ ，这时式(5.64)成为关于 $\{\lambda_i\}$ 的线性方程组，其系数矩阵通常可逆，因此有唯一解。在实践中更多的情况是 $J > P$ ，这时式(5.64)一般无解，只能求近似解。

常用的非线性基函数有以下几种：

◆ 高斯基函数

$$g(\mathbf{x}) = \sum_{i=1}^P \lambda_i \varphi_i(\mathbf{x}) \quad (5.65)$$

其中

$$\varphi_i(x) = \frac{R_i(x)}{\sum_{q=1}^P R_q(x)}, \quad R_i(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x} - c_i\|^2}{2\sigma_i^2}\right) \quad (5.66)$$

确定了 $\{c_i\}$ 后，可以选取如上面的高斯基函数来构造径向基函数，这里参数 σ_i 是第 i 个高斯基函数 $R_i(x)$ 的“宽度”或“平坦程度”。 σ_i 越大，则以 c_i 为中心的等高线越稀疏， $R_i(x)$ 越平坦，对其它 $\varphi_q(x)$ 的影响也就越大。 σ_i 的一种选法是

$$\sigma_i^2 = \frac{1}{M_i} \sum_{x \in \theta_i} \|x - c_i\|^2 \quad (5.67)$$

即 θ_i 类所含的样本点与中心 c_i 的平均距离越大，则 $R_i(x)$ 应该越平坦。

◆ 薄板样条函数

$$\varphi(v) = v^2 \lg(v) \quad (5.68)$$

◆ 多二次函数

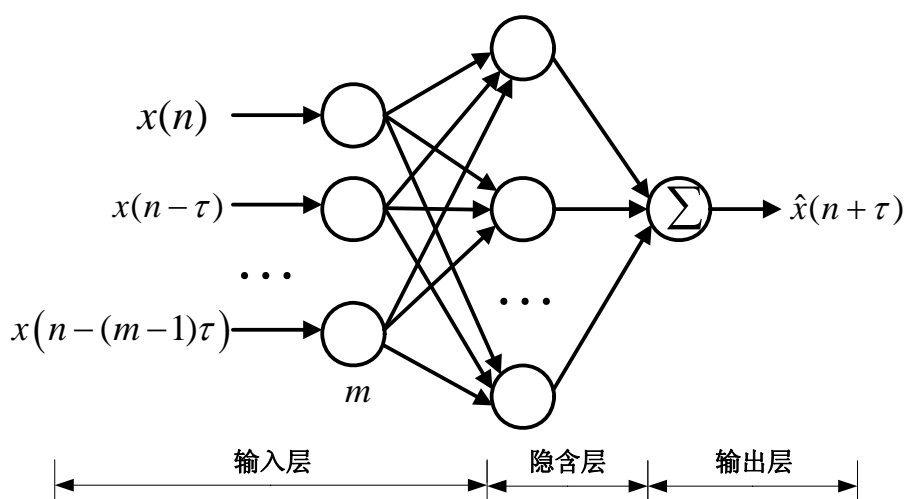
$$\varphi(v) = (v^2 + c)^{\frac{1}{2}}, \quad c > 0 \quad (5.69)$$

◆ 逆多二次函数

$$\varphi(v) = (v^2 + c)^{-1/2}, \quad c > 0 \quad (5.70)$$

一般认为，非线性函数 φ 的具体形式对网络性能的影响不大。

我们以含一个隐层的前馈式网络为例，径向基神经网络结构如图表 5-5所示



图表 5-5: 径向基神经网络结构图

其中输入层有 m 个单元， m 即为嵌入维数；各输入单元对应的输入

$$x(n-(i-1)\tau), i=1, 2, \dots, m \quad (5.71)$$

表示嵌入向量 \mathbf{x}_n 的第 i 个分量； $S_j (j=1, 2, \dots, K)$ 表示隐层单元，也就是神经元，神经元的特性函数一般是光滑的非线性递增函数，如 sigmoid 函数；输出层则只有一个单元。

若神经元特性函数为

$$S_j(\mathbf{y}) = \frac{1}{1 + e^{\mathbf{y} - c_j}}, \text{ 其中 } \mathbf{y} = \mathbf{b}_j^T \mathbf{x}, j=1, 2, \dots, K \quad (5.72)$$

其中向量 \mathbf{b}_j 的各分量为神经元各输入的加权系数， c_j 为神经元的偏置。则整个神经网络的输出可表示为

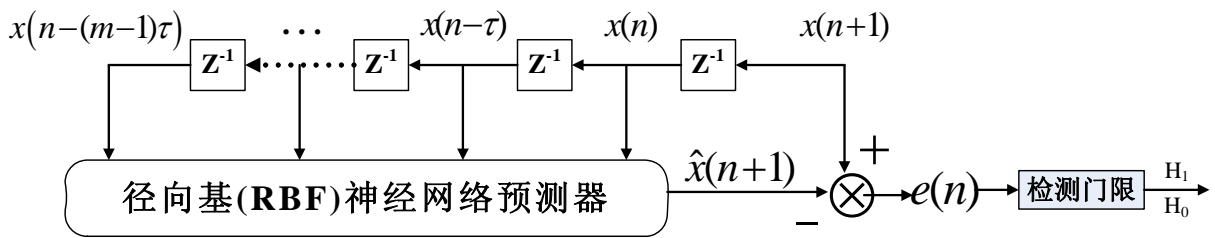
$$f(\mathbf{x}) = \sum_{j=1}^K \frac{a_j}{1 + e^{\mathbf{b}_j^T \mathbf{x} - c_j}} \quad (5.73)$$

其中的 a_j 为各神经元输出的加权系数。

用神经网络作非线性时间序列预测时，首先由时间序列 $\{x_i\}$ 延迟重构得到嵌入向量 $\{\mathbf{x}_i\}$ ，将嵌入向量 \mathbf{x}_i 与对应的时间序列下一步的值 x_{i+1} 组成训练对训练神经网络，得到网络参数。对于训练好的网络，以嵌入向量 \mathbf{x}_i 作为神经网络的输入，则神经网络的输出为预测值 \hat{x}_{n+1} ，即 $\hat{x}_{n+1} = \hat{g}(\mathbf{x})$ 。

混沌时间序列具有长期不可预测而短期可预测的特点。这样在检测混沌背景下的目标信号时可以利用神经网络先对混沌时间序列建模，然后对接收到的信号进行预测。混沌背景下基于神经网络的目标检测方法可简单归结为如下几个步骤^[15-18]：

- 1) 用神经网络对混沌时间序列建模，将神经网络作为一个预测器，即用混沌背景信号产生模式对训练神经网络，使其预测误差较小，且对时间序列有较好的短期预测性能。否则改变网络的拓扑结构重新进行训练，直到满足要求为止。
- 2) 神经网络训练好以后，对接收到的信号（训练过程中未使用过的数据） $x(i), i=1, 2, \dots, n$ 进行单步预测（ n 为观测阶段所获得的数据点的总长度），得到预测值 $\hat{x}(i)$ ，并计算预测误差 $\varepsilon(i) = x(i) - \hat{x}(i)$
- 3) 计算出总误差的均方值为： $R_\varepsilon = \frac{1}{n} \sum_{i=1}^n \varepsilon^2(i)$
- 4) 根据一定的准则选择一门限 η ，比较误差 R_ε 和门限 η 的大小，如果 $R_\varepsilon > \eta$ ，则认为接收信号中存在目标，否则就判为没有目标。



图表 5-6: 径向基(RBF)神经网络预测器框图

5.5 Volterra自适应预测器^[19]

非线性函数逼近法主要是基于函数逼近理论，利用非线性函数去逼近原始的非线性系统，模拟其行为。常用的非线性函数有Sigmoid函数、径向基函数，Volterra级数等。

Volterra级数是由意大利数学家Vito Volterra于1880年首先提出的一种泛函级数，当时是作为对Taylor级数的推广而提出的。在1912年，Volterra将这种泛函级数用于研

究某些积分方程和积分—微分方程的求解。到了1942年，控制论的奠基人—美国著名科学家Wiener开始将Volterra泛函级数用于发展非线性算子理论以及非线性系统分析。在二十世纪七十年代以后，Volterra级数开始受到人们的普遍重视。法国学者Fliess等人建立了由Lie群表示的动力学系统Volterra级数分析理论，Brockett研究了Volterra泛函级数与几何控制论的关系。美国学者Sandberg利用Volterra级数研究了一大类非线性动力系统。而且随着计算机技术的不断发展，Volterra级数显现出更多的应用价值和巨大潜力。

Volterra级数之所以具有如此大的吸引力，其根本原因是它和幂级数之间有着天然的密切联系和相似之处，很容易被广大工程技术人员和科技工作者接受。Volterra技术核具有鲜明的物理意义，可以根据实际情况应用与工程技术领域。Volterra级数使得人们能够象利用Laplace变换和线性传递函数法分析线性系统那样分析一般的非线性动力系统，推动和促进了非线性科学的研究与发展。它不仅提供了一套新的理论框架，而且为解决实际的非线性问题提供了一种有力的方法和工具。

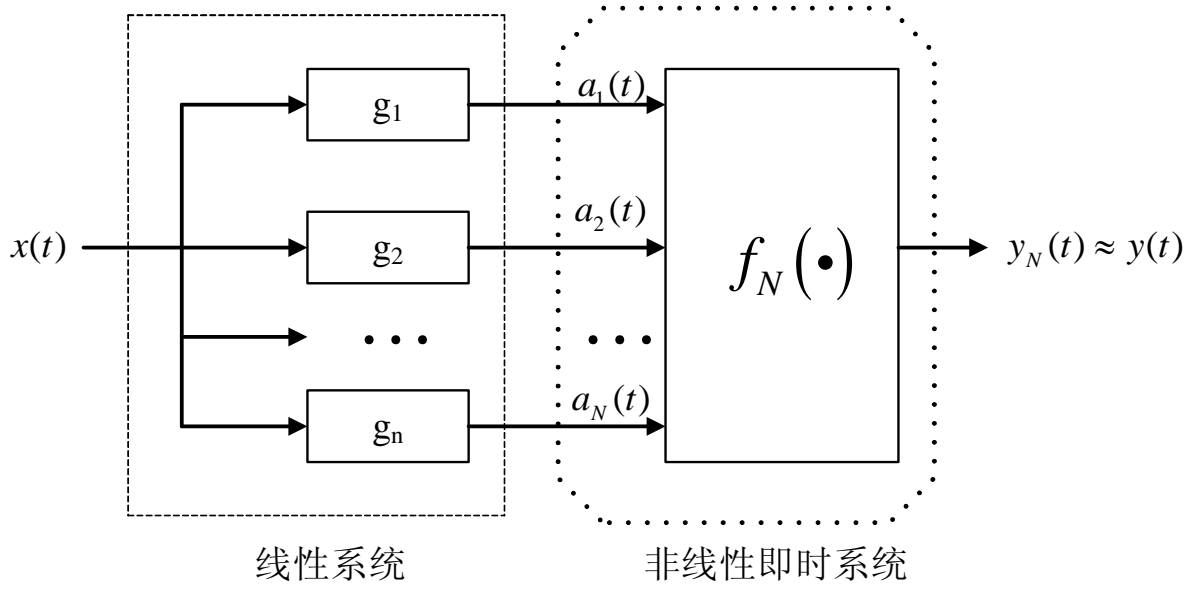
一个时不变的非线性系统，其输入 $x(t)$ 与输出 $y(t)$ 之间的关系可用一个非线性算子 F (或称为泛函) 表示为

$$y(t) = F[x(t)] \quad (5.74)$$

并作以下假设：

- 输入信号 $x(t)$ 的能量是有限的，即 $\|x(t)\| = \left[\int_{-\infty}^{+\infty} x^2(t) dt \right]^{\frac{1}{2}} < \infty$
- 算子 F 是连续的，即对于任意两个满足能量有限条件的信号 $x_1(t)$ 和 $x_2(t)$ ，当 $\|x_2(t) - x_1(t)\| \rightarrow 0$ 时，恒有 $\|F[x_2(t)] - F[x_1(t)]\| \rightarrow 0$

则系统总可以任意准确地分解为一组线性系统和一个非线性即时系统，这就是非线性系统分解定理。如图表 5-7所示，左边框图内的函数 g_1, g_2, \dots, g_N 组成线性系统，而函数 $f_N(a_1, a_2, \dots, a_N)$ 为非线性“即时系统”，即函数 f_N 对输入 a_1, a_2, \dots, a_N 只进行代数运算，而不进行求导、积分等“动态”运算。当泛函 F 连续时，函数 f_N 也是连续的。



图表 5-7: 非线性系统的分解示意图

非线性系统分解定理具有重要而深刻的含义。它说明:满足一定条件的非线性系统总可以进行分解,分解的结果可将系统把所有的记忆因素都归于线性系统,而把所有的非线性因素都归于即时系统。这样的分解把复杂的非线性系统分解成两类比它简单的子系统。

根据函数逼近理论中的Weierstrass定理,任何定义于一个闭区间的连续函数可以用多项式任意准确地逼近。显然函数 $f_N: R^N \mapsto R$ 可以在一个闭区间内近似展开为

$$f_N(a_1, a_2, \dots, a_N) = \beta_0 + (\beta_1 a_1 + \beta_2 a_2 + \dots + \beta_N a_N) + (\beta_{11} a_1 a_1 + \beta_{12} a_1 a_2 + \dots + \beta_{NN} a_N a_N) + \dots \quad (5.75)$$

其中
$$a_i = \int_{-\infty}^{+\infty} g_i(\tau) x(t-\tau) d\tau \quad (i=1, 2, \dots, N) \quad (5.76)$$

由式(5.75)和(5.76)可得

$$\begin{aligned} y(t) = F[x(t)] &= f_N(a_1, a_2, \dots, a_N)|_{N \rightarrow \infty} = h_0 + \int_{-\infty}^{+\infty} h_1(\tau) x(t-\tau) d\tau \\ &+ \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} h_2(\tau_1, \tau_2) \prod_{i=1}^2 x(t-\tau_i) d\tau_i + \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} h_3(\tau_1, \tau_2, \tau_3) \prod_{i=1}^3 x(t-\tau_i) d\tau_i + \dots \\ &+ \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} h_n(\tau_1, \tau_2, \dots, \tau_n) \prod_{i=1}^n x(t-\tau_i) d\tau_i + \dots = \sum_{n=0}^{\infty} y_n(t) \end{aligned} \quad (5.77)$$

其中
$$y_n(t) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} h_n(\tau_1, \tau_2, \dots, \tau_n) \prod_{i=1}^n x(t-\tau_i) d\tau_i \quad (5.78)$$

$$\begin{aligned}
h_0 &= \beta_0 \\
h_1(\tau) &= \beta_1 g_1(\tau) + \beta_2 g_2(\tau) + \cdots = \sum_{i=0}^{\infty} \beta_i g_i(\tau) \\
h_2(\tau_1, \tau_2) &= \beta_{11} g_1(\tau_1) g_1(\tau_2) + \beta_{12} g_1(\tau_1) g_2(\tau_2) + \cdots = \sum_{i,j=0}^{\infty} \beta_{ij} g_i(\tau_i) g_j(\tau_j) \\
h_3(\tau_1, \tau_2, \tau_3) &= \cdots = \sum_{i,j,k=0}^{\infty} \beta_{ijk} g_i(\tau_i) g_j(\tau_j) g_k(\tau_k)
\end{aligned} \tag{5.79}$$

式(5.77)即是非线性系统的Volterra泛函级数表示，其中 $y_n(t)$ 称为系统的 n 阶输出， $h_n(\tau_1, \tau_2, \cdots, \tau_n)$ 称为系统的 n 阶Volterra核或 n 阶冲激响应。由式(5.78)可知 $y_n(t)$ 本身也是一个泛函，因而用Volterra泛函级数来表征非线性系统，从本质上讲是用一个泛函序列来逼近一个连续泛函，这与用一个函数序列来逼近一个连续函数是相似的。函数通常可以用多种函数序列来逼近，例如幂级数或正交函数族。连续泛函的分解也不是唯一的，Volterra泛函级数表示法只是其中之一。例如在得到Volterra级数分解的过程中，我们采用的是幂级数来逼近非线性函数 $y_n(t)$ ，如果采用的不是幂级数而是Hermite(正交)函数族，那么结果就是非线性系统的另一种表征方式—Wiener级数。

总之，在满足输入信号能量有限的条件下，由式(5.74)描述的非线性系统可以用Volterra级数逼近到任意程度。换句话说，一大类非线性系统都有其固有的Volterra核，Volterra基数完全表征了这类非线性系统。这里有两种意思：第一，若系统的各阶Volterra核是已知的，那么就可以根据当前状态来确定下一个状态；第二，当且仅当两个系统的各阶Volterra核对应相等时，它们才是等价的。此外，当系统的二阶以上Volterra核为零时，非线性系统就退化为线性系统：

$$y(t) = h_0 + \int_{-\infty}^{+\infty} h_1(\tau) x(t-\tau) d\tau \tag{5.80}$$

设式(5.77)中的Volterra级数核为

$$h_n(\tau_1, \tau_2, \cdots, \tau_n) = a_n \delta(\tau_1) \delta(\tau_2) \cdots \delta(\tau_n) \tag{5.81}$$

将其代入式(5.77)，得到

$$y(t) = \sum_{n=0}^{\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} a_n \delta(\tau_1) \delta(\tau_2) \cdots \delta(\tau_n) \prod_{i=1}^n x(t-\tau_i) d\tau_i = \sum_{n=0}^{\infty} a_n x^n(t) \tag{5.82}$$

从而Volterra级数退化为一个幂级数。因此可以说Volterra级数是幂级数的推广。如前所述，Volterra级数可以描述一大类非线性系统。利用这个特点，我们可以将Volterra级数用作基于模型的非线性混沌时间序列预测。

对于离散的时不变非线性系统，我们必须将式(5.74)所示的连续Volterra级数离散

化。用求和替代积分，系统对输入 $x(n)$ 的响应 $y(n)$ 可用离散Volterra级数可以表示为

$$y(n) = h_0 + \sum_{k=1}^{\infty} y_k(n) \quad (5.83)$$

其中

$$y_n(k) = \sum_{i_1, i_2, \dots, i_k=0}^{\infty} h_k(i_1, i_2, \dots, i_k) x(n-i_1) \cdots x(n-i_k) \quad (5.84)$$

显然，在实际应用中式(5.83)和(5.84)所示的无穷级数是无法实现的，必须采用有限阶输出求和 ($k \in [0, p]$)，且各阶响应为有限项求和 ($i_1, i_2, \dots, i_k \in [0, m-1]$) 的形式：

$$y(n) = h_0 + \sum_{k=1}^p y_k(n) \quad (5.85)$$

其中

$$y_n(k) = \sum_{i_1, i_2, \dots, i_k=0}^{m-1} h_k(i_1, i_2, \dots, i_k) \prod_{j=1}^k x(n-i_j) \quad (5.86)$$

将式(5.83)中将求和项截断为 m ，而式(5.84)将阶数截断为 p 。这里 $h_k(i_1, i_2, \dots, i_k)$ 称为 k 阶Volterra核， p 为Volterra滤波器阶数。实际应用中，一般取二阶截断 $p=2$ 次求和^[20]，则用于混沌时间序列预测的滤波器为

$$x(n+1) = h_0 + \sum_{i_1=0}^{m-1} h_1(i_1) x(n-i_1\tau) + \sum_{i_1, i_2=0}^{m-1} h_2(i_1, i_2) x(n-i_1\tau) x(n-i_2\tau) \quad (5.87)$$

有了预测模型，余下的问题是如何根据已知的时间序列来估计模型参数，即式(5.87)中的各阶Volterra级数核^[21]。根据式(5.85)，将已知的时间序列嵌入到 m 维空间后，组成输入-输出对 $\{x_n, s_{n+1}\}$ ，下面就可以利用数值算法对各阶Volterra级数核作出最小二乘估计。展开式(5.87)，Volterra滤波器即是非线性自适应FIR滤波器，滤波器的滤波系数矢量和输入信号矢量分别为

$$W(n) = [h_0, h_1(0), h_1(1), \dots, h_1(m-1), h_2(0,0), h_2(0,1), \dots, h_2(m-1, m-1)]^T \quad (5.88)$$

$$Z(n) = [1, x(n), x(n-\tau), \dots, x(n-(m-1)\tau), x^2(n), x(n)x(n-\tau), \dots, x^2(n-(m-1)\tau)]^T \quad (5.89)$$

因此，式(5.87)可表示为

$$x(n+1) = Z^T(n)W(n) \quad (5.90)$$

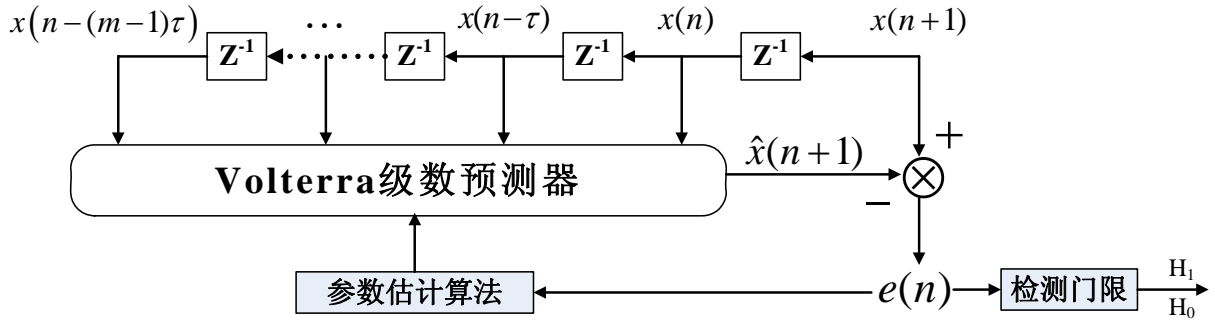
对于上式描述的Volterra自适应滤波器，一般采用归一化最小均方(Normalized Least Mean Square, NLMS)自适应算法来求解^[22]。当滤波系数矢量 $W(n)$ ，输入信号矢量 $Z(n)$ ，NLMS算法可描述如下：

$$e(n) = x(n+1) - Z^T(n)W(n) \quad (5.91)$$

$$W(n+1) = W(n) - \frac{\mu}{Z^T(n)Z(n)} e(n)Z(n) \quad (5.92)$$

其中 μ 为收敛步长 ($0 < \mu < 2$)。

对混沌时间序列的预测而言，我们首先要将混沌时间序列 $\{x_n\}$ 以延迟时间 τ (正整数) 嵌入到 m 维的相空间中，再根据相空间的当前状态来作出预测。由此可以得到混沌时间序列的离散Volterra级数预测模型称为混沌时间序列的 m 维 p 阶离散Volterra级数预测模型，或简称为Volterra级数预测模型，如图表 5-8所示。



图表 5-8: Volterra 基数预测与检测预测器

5.6 海杂波的线性和非线性预测

内容 (略)

5.7 本章小结

本章介绍了混沌时间序列的几种常见预测方法，分析了这几种方法进行预测的过程和优缺点。混沌时间序列相对来说是个比较新的研究方向，随着非线性系统辨识和预测的许多研究方法推广到混沌系统中，预测的精度和时间均会有所提高。另外实际时间序列中含有噪声，如何解决预测模型对噪声的抗干扰能力也是一个重要课题。

5.8 后记

本版本为“时间序列的线性和非线性预测”的初稿，写的过程中才发现对一些基本概念的理解还是很模糊，时间序列预测方面的研究者很多，欢迎大家告诉我初稿里面的错误和误解，这是我进行时间序列预测研究的第一步。

你的关注是我前进的最大动力☺。

任何意见、建议、批评和讨论我都热烈欢迎。

我的信箱：xuxkboy@newmail.dlmu.edu.cn

欢迎大家到研学论坛(<http://bbs.matwav.com/>)混沌分形版进行讨论

本文版权目前归本人所有，引用格式如下：

许小可. 海杂波的非线性分析与建模: (博士学位论文). 大连: 大连海事大学, 2007

参考文献:

- [1] 简相超, 郑君里. 混沌神经网络预测算法评价准则与性能分析. 清华大学学报(自然科学版). 2001, 41(07): 43-46.
- [2] 温权, 张勇传, 程时杰. 混沌时间序列预测技术. 水电能源科学. 2001, 19(03): 76-78.
- [3] <http://arxiv.org/abs/chao-dyn/9401003>.
- [4] Hegger R, Kantz H, Schreiber T, et al. Practical implementation of nonlinear time series methods: The TISEAN package. chaos. 1999, 9: 413-435.
- [5] 罗军辉, 罗勇江, 白义臣, 等. matlab7.0在数字信号处理中的作用. 北京: 机械工业出版社, 2005.
- [6] 曹兰兰, 许小可, 索继东. AR模型仿真相关高斯序列的阶数选择. 2007, inpress.
- [7] 朱灿焰, 何佩琨, 高梅国. 相关杂波的AR模型及其研究. 现代雷达. 1998, 20(5): 36-43.
- [8] 何友, 关键, 彭应宁, 等. 雷达自动检测与恒虚警处理. 北京: 清华大学出版社, 1999.
- [9] 吕金虎, 陆君安, 陈士华. 混沌时间序列分析及其应用. 武汉: 武汉大学出版社, 2002.
- [10] 李恒超, 张家树. 混沌时间序列局域零阶预测法性能比较. 西南交通大学学报. 2004, 39(03): 328-331.
- [11] 吕金虎, 张锁春. 加权一阶局域法在电力系统短期负荷预测中的应用. 控制理论与应用. 2002, 19(05): 767-770.
- [12] Farmer J D, Sidorowich J J. Predicting chaotic time series. Physical Review Letters. 1987, 59(8): 845-848.
- [13] 蔡烽, 万林, 石爱国. 基于相空间重构技术的舰船摇荡极短期预报. 水动力学研究与进展A辑. 2005, 20(06): 780-784.
- [14] 吴微. 神经网络计算. 北京: 高等教育出版社, 2006.
- [15] Haykin S, Xiao B L. Detection of signals in chaos. Proceedings of the IEEE. 1995, 83(1): 95-122.
- [16] 杜鹏飞, 王永良, 孙文峰. 混沌海杂波背景下的弱信号检测. 系统工程与电子技术. 2002, 24(07): 65-67.
- [17] 朱丽莉, 张永顺, 李兴成. 基于RBF神经网络的混沌背景下瞬态弱信号检测. 空军工程大学学报(自然科学版). 2006, 7(02): 61-63.
- [18] 陈晓楠, 许小可, 索继东. 基于径向基函数的海杂波时间序列预测方法. 大连海事大学学报. 2007, 33(01): 115-118.
- [19] 朱昀. 水声信号非线性分析方法研究: 博士学位论文. 西安: 西北工业大学, 2002.
- [20] 张家树, 肖先赐. 混沌时间序列的Volterra自适应预测. 物理学报. 2000, 49(03): 403-407.
- [21] <http://luzhenbo.88uu.com.cn/>.
- [22] Douglas S C, Meng T H. Normalized data nonlinearities for LMS adaptation. IEEE Transactions on Signal Processing. 1994, 42(6): 1352-1365.