



第6章 极大似然法及其它辨识算法

6.3 随机逼近法

6.4 递推参数估计算法的收敛性描述

作者：李鹏波、胡德文

单位：国防科技大学机电工程与自动化学院

Email: gaoxia73@163.com

中国水利水电出版社版权所有



6.3 随机逼近法

2. 随机逼近法

3. 随机牛顿法

6.4 递推参数估计算法...

Home Page

Title Page

◀◀ ▶▶

◀ ▶

Page 1 of 18

Go Back

Full Screen

Close

Quit



6.3 随机逼近法
2. 随机逼近法
3. 随机牛顿法
6.4 递推参数估计算法...

第6章 极大似然法及其它辨识算法

6.3 随机逼近法

1. 随机逼近原理

2. 随机逼近法

3. 随机牛顿法

6.4 递推参数估计算法的收敛性描述

Home Page

Title Page

◀ ▶

◀ ▶

Page 2 of 18

Go Back

Full Screen

Close

Quit

6.3 随机逼近法

1. 随机逼近原理

考虑如下模型的辨识问题

$$y(k) = \varphi^T(k)\theta + e(k) \quad (1)$$

其中 $e(k)$ 是零均值的噪声。显然，这种模型的参数辨识问题可以通过极小化 $e(k)$ 的方差来实现，即求参数 θ 的估值，使准则函数 J 达到极小值。

$$J(\theta) = \frac{1}{2}E\{e^2(k)\} = \frac{1}{2}E\{[y(k) - \varphi^T(k)\theta]^2\} \quad (2)$$



6.3 随机逼近法

2. 随机逼近法

3. 随机牛顿法

6.4 递推参数估计算法...

Home Page

Title Page



Page 3 of 18

Go Back

Full Screen

Close

Quit



6.3 随机逼近法
2. 随机逼近法
3. 随机牛顿法
6.4 递推参数估计算法...

6.3 随机逼近法

1. 随机逼近原理

考虑如下模型的辨识问题

$$y(k) = \varphi^T(k)\theta + e(k) \quad (1)$$

其中 $e(k)$ 是零均值的噪声。显然，这种模型的参数辨识问题可以通过极小化 $e(k)$ 的方差来实现，即求参数 θ 的估值，使准则函数 J 达到极小值。

$$J(\theta) = \frac{1}{2}E\{e^2(k)\} = \frac{1}{2}E\{[y(k) - \varphi^T(k)\theta]^2\} \quad (2)$$

该准则函数的一阶负梯度为

$$\left[-\frac{\partial J(\theta)}{\partial \theta} \right]^T = E\{\varphi(k)[y(k) - \varphi^T(k)\theta]\} \quad (3)$$

Home Page

Title Page



Page 3 of 18

Go Back

Full Screen

Close

Quit



6.3 随机逼近法

1. 随机逼近原理

考虑如下模型的辨识问题

$$y(k) = \varphi^T(k)\theta + e(k) \quad (1)$$

其中 $e(k)$ 是零均值的噪声。显然，这种模型的参数辨识问题可以通过极小化 $e(k)$ 的方差来实现，即求参数 θ 的估值，使准则函数 J 达到极小值。

$$J(\theta) = \frac{1}{2}E\{e^2(k)\} = \frac{1}{2}E\{[y(k) - \varphi^T(k)\theta]^2\} \quad (2)$$

该准则函数的一阶负梯度为

$$\left[-\frac{\partial J(\theta)}{\partial \theta} \right]^T = E\{\varphi(k)[y(k) - \varphi^T(k)\theta]\} \quad (3)$$

令其梯度为零，即

$$E\{\varphi(k)[y(k) - \varphi^T(k)\theta]\}_{\theta=\hat{\theta}} = 0 \quad (4)$$



原则上说，由式(4)可以求得使 $J(\theta) = \min$ 的参数估计值 $\hat{\theta}$ ，但因为 $e(k)$ 的统计性质不知道，式(4)实际上是无法解得。

- 6.3 随机逼近法
- 2. 随机逼近法
- 3. 随机牛顿法
- 6.4 递推参数估计算法...

Home Page

Title Page

◀◀ ▶▶

◀ ▶

Page 4 of 18

Go Back

Full Screen

Close

Quit



- 6.3 随机逼近法
- 2. 随机逼近法
- 3. 随机牛顿法
- 6.4 递推参数估计算法...

原则上说，由式(4)可以求得使 $J(\theta) = \min$ 的参数估计值 $\hat{\theta}$ ，但因为 $e(k)$ 的统计性质不知道，式(4)实际上是无法解得。

如果左边的数学期望用平均值来近似，即式(4)近似写为

$$\frac{1}{N} \sum_{k=1}^N \varphi(k)[y(k) - \varphi^T(k)\hat{\theta}] = 0 \quad (5)$$

则有

$$\hat{\theta} = \left[\sum_{k=1}^N \varphi(k)\varphi^T(k) \right]^{-1} \left[\sum_{k=1}^N \varphi(k)y(k) \right] \quad (6)$$

[Home Page](#)[Title Page](#)[Page 4 of 18](#)[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)



- 6.3 随机逼近法
- 2. 随机逼近法
- 3. 随机牛顿法
- 6.4 递推参数估计算法...

原则上说，由式(4)可以求得使 $J(\theta) = \min$ 的参数估计值 $\hat{\theta}$ ，但因为 $e(k)$ 的统计性质不知道，式(4)实际上是无法解得。
如果左边的数学期望用平均值来近似，即式(4)近似写为

$$\frac{1}{N} \sum_{k=1}^N \varphi(k)[y(k) - \varphi^T(k)\hat{\theta}] = 0 \quad (5)$$

则有

$$\hat{\theta} = \left[\sum_{k=1}^N \varphi(k)\varphi^T(k) \right]^{-1} \left[\sum_{k=1}^N \varphi(k)y(k) \right] \quad (6)$$

显然，这种近似使问题退化成最小二乘问题，式(6)即是最小二乘解 $\hat{\theta}_{LS} = (\Phi^T \Phi)^{-1} \Phi^T Y$ 。下面推导式(4)的随机逼近解。

[Home Page](#)[Title Page](#)[◀◀](#) [▶▶](#)[◀](#) [▶](#)[Page 4 of 18](#)[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)



设是 x 标量， $y(x)$ 是对应的随机变量， $p(y|x)$ 是条件概率密度函数，则随机变量 y 关于 x 的条件数学期望为

$$E\{y|x\} = \int y dp(y|x) \quad (7)$$

它是以为 x 变量的回归函数。

- 6.3 随机逼近法
- 2. 随机逼近法
- 3. 随机牛顿法
- 6.4 递推参数估计算法...

[Home Page](#)

[Title Page](#)

[◀◀](#) [▶▶](#)

[◀](#) [▶](#)

Page 5 of 18

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)



- 6.3 随机逼近法
- 2. 随机逼近法
- 3. 随机牛顿法
- 6.4 递推参数估计算法...

设是 x 标量， $y(x)$ 是对应的随机变量， $p(y|x)$ 是条件概率密度函数，则随机变量 y 关于 x 的条件数学期望为

$$E\{y|x\} = \int ydp(y|x) \quad (7)$$

它是以为 x 变量的回归函数。

对于给定的 α ，设方程

$$f(x) = E\{y|x\} = \alpha \quad (8)$$

具有唯一解。当 $f(x)$ 函数的形式及条件概率密度函数 $p(y|x)$ 都不知道时，求方程式(8)的解析解是困难的，这时可由随机逼近法来求解。所谓随机逼近法(RAA)，就是利用 x_1, x_2, \dots 变量及其对应的随机变量 $y(x_1), y(x_2), \dots$ ，通过迭代计算，逐步逼近方程式(8)的解。这种迭代算法常用的有Robbins–Monro算法和Kiefer–Wolfowitz算法。

[Home Page](#)[Title Page](#)[Page 5 of 18](#)[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)



- 6.3 随机逼近法
- 2. 随机逼近法
- 3. 随机牛顿法
- 6.4 递推参数估计算法...

求方程式(8)解的Robbins–Monro算法为

$$x(k+1) = x(k) + \rho(k)[\alpha - y(x(k))] \quad (9)$$

其中 $y(x(k))$ 是对应于 $x(k)$ 的 y 值， $\rho(k)$ 称为收敛因子。

Home Page

Title Page

◀◀ ▶▶

◀ ▶

Page 6 of 18

Go Back

Full Screen

Close

Quit



- 6.3 随机逼近法
- 2. 随机逼近法
- 3. 随机牛顿法
- 6.4 递推参数估计算法...

求方程式(8)解的Robbins–Monro算法为

$$x(k+1) = x(k) + \rho(k)[\alpha - y(x(k))] \quad (9)$$

其中 $y(x(k))$ 是对应于 $x(k)$ 的 y 值， $\rho(k)$ 称为收敛因子。如果收敛因子满足以下条件

$$\begin{cases} \rho(k) > 0, & \forall k; & \lim_{k \rightarrow \infty} \rho(k) = 0, \\ \sum_{k=1}^{\infty} \rho(k) = \infty; & \sum_{k=1}^{\infty} \rho^2(k) < \infty; \end{cases} \quad (10)$$

则 $x(k)$ 在均方意义下收敛于方程式(8)的解。

Home Page

Title Page

◀ ▶

◀ ▶

Page 6 of 18

Go Back

Full Screen

Close

Quit



6.3 随机逼近法

2. 随机逼近法

3. 随机牛顿法

6.4 递推参数估计算法...

求方程式(8)解的Robbins–Monro算法为

$$x(k+1) = x(k) + \rho(k)[\alpha - y(x(k))] \quad (9)$$

其中 $y(x(k))$ 是对应于 $x(k)$ 的 y 值， $\rho(k)$ 称为收敛因子。如果收敛因子满足以下条件

$$\begin{cases} \rho(k) > 0, & \forall k; \quad \lim_{k \rightarrow \infty} \rho(k) = 0, \\ \sum_{k=1}^{\infty} \rho(k) = \infty; & \sum_{k=1}^{\infty} \rho^2(k) < \infty; \end{cases} \quad (10)$$

则 $x(k)$ 在均方意义下收敛于方程式(8)的解。调和级数 $\{\frac{1}{k}\}$ 满足式(10)的条件，可作为收敛因子，但一般取

$$\rho(k) = \frac{b}{k+a} \quad (11)$$

或

$$\rho(k) = \frac{1}{k^q}, \quad q \in [0.5, 1] \quad (12)$$

Home Page

Title Page



Page 6 of 18

Go Back

Full Screen

Close

Quit



Robbins–Monro算法一般是用来求方程式(8)的根。Kiefer和Wolfowitz后来引用它来确定回归函数 $f(x)$ 的极值。如果回归函数 $f(x)$ 存在极值，则极值处的 x 使得 $\frac{df(x)}{dx} = 0$ 。根据Robbins–Monro算法，Kiefer和Wolfowitz给出了求回归函数极值的迭代算法。

- 6.3 随机逼近法
- 2. 随机逼近法
- 3. 随机牛顿法
- 6.4 递推参数估计算法...

[Home Page](#)[Title Page](#)[◀◀](#)[▶▶](#)[◀](#)[▶](#)[Page 7 of 18](#)[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)



Robbins–Monro算法一般是用来求方程式(8)的根。Kiefer和Wolfowitz后来引用它来确定回归函数 $f(x)$ 的极值。如果回归函数 $f(x)$ 存在极值，则极值处的 x 使得 $\frac{df(x)}{dx} = 0$ 。根据Robbins–Monro算法，Kiefer和Wolfowitz给出了求回归函数极值的迭代算法。

$$x(k+1) = x(k) - \rho(k) \left. \frac{dy}{dx} \right|_{x=x(k)} \quad (13)$$

式中 $\rho(k)$ 是收敛因子，必须满足式(10)的条件，则 $x(k)$ 的收敛值将使 $f(x(k))$ 达到极值。

- 6.3 随机逼近法
- 2. 随机逼近法
- 3. 随机牛顿法
- 6.4 递推参数估计算法...

[Home Page](#)[Title Page](#)[◀◀](#) [▶▶](#)[◀](#) [▶](#)[Page 7 of 18](#)[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)



6.3 随机逼近法

2. 随机逼近法

3. 随机牛顿法

6.4 递推参数估计算法...

Robbins–Monro算法一般是用来求方程式(8)的根。Kiefer和Wolfowitz后来引用它来确定回归函数 $f(x)$ 的极值。如果回归函数 $f(x)$ 存在极值，则极值处的 x 使得 $\frac{df(x)}{dx} = 0$ 。根据Robbins–Monro算法，Kiefer和Wolfowitz给出了求回归函数极值的迭代算法。

$$x(k+1) = x(k) - \rho(k) \left. \frac{dy}{dx} \right|_{x=x(k)} \quad (13)$$

式中 $\rho(k)$ 是收敛因子，必须满足式(10)的条件，则 $x(k)$ 的收敛值将使 $f(x(k))$ 达到极值。

Kiefer–Wolfowitz算法可以推广到多维的情况。考虑标量函数 $J(\theta)$ 的极值问题，如果 θ 在点 $\hat{\theta}$ 上使得 $J(\theta)$ 取极值，那么求 $\hat{\theta}$ 的随机逼近迭代算法为

$$\hat{\theta}(k+1) = \hat{\theta}(k) - \rho(k) \left[\frac{\partial J(\theta)}{\partial \theta} \right]^T \bigg|_{\theta=\hat{\theta}(k)} \quad (14)$$

我们下面讨论的随机逼近法，就是以Kiefer–Wolfowitz算法为基础的。

Home Page

Title Page



Page 7 of 18

Go Back

Full Screen

Close

Quit



- 6.3 随机逼近法
- 2. 随机逼近法
- 3. 随机牛顿法
- 6.4 递推参数估计算法...

2. 随机逼近法

对模型式(1)的参数辨识问题，如果取式(2)作为准则函数，则由Kiefer–Wolfowitz算法得到其随机逼近法解为

$$\hat{\theta}(k) = \hat{\theta}(k-1) + \rho(k)\varphi(k)[y(k) - \varphi^T(k)\hat{\theta}(k-1)] \quad (15)$$

Home Page

Title Page

◀ ▶

◀ ▶

Page 8 of 18

Go Back

Full Screen

Close

Quit



2. 随机逼近法

对模型式(1)的参数辨识问题，如果取式(2)作为准则函数，则由Kiefer–Wolfowitz算法得到其随机逼近法解为

$$\hat{\theta}(k) = \hat{\theta}(k-1) + \rho(k)\varphi(k)[y(k) - \varphi^T(k)\hat{\theta}(k-1)] \quad (15)$$

如果准则函数取为更一般的形式，即

$$J(\theta) = E\{h(\theta, \mathbf{D}(k))\} \quad (16)$$

其中 $h(\cdot)$ 为某个标量函数， $\mathbf{D}(k)$ 表示 k 时刻以前的输入输出数据集合。

- 6.3 随机逼近法
- 2. 随机逼近法
- 3. 随机牛顿法
- 6.4 递推参数估计算法...

Home Page

Title Page

◀ ▶

◀ ▶

Page 8 of 18

Go Back

Full Screen

Close

Quit



2. 随机逼近法

对模型式(1)的参数辨识问题，如果取式(2)作为准则函数，则由Kiefer–Wolfowitz算法得到其随机逼近法解为

$$\hat{\theta}(k) = \hat{\theta}(k-1) + \rho(k)\varphi(k)[y(k) - \varphi^T(k)\hat{\theta}(k-1)] \quad (15)$$

如果准则函数取为更一般的形式，即

$$J(\theta) = E\{h(\theta, \mathbf{D}(k))\} \quad (16)$$

其中 $h(\cdot)$ 为某个标量函数， $\mathbf{D}(k)$ 表示 k 时刻以前的输入输出数据集。显然，准则函数的一阶负梯度为

$$\begin{aligned} \left[-\frac{\partial J(\theta)}{\partial \theta}\right]^T &= \left[E\left\{-\frac{\partial}{\partial \theta}h(\theta, \mathbf{D}(k))\right\}\right]^T \\ &= E\{q(\theta, \mathbf{D}(k))\} \end{aligned} \quad (17)$$

6.3 随机逼近法

2. 随机逼近法

3. 随机牛顿法

6.4 递推参数估计算法...

Home Page

Title Page

◀ ▶

◀ ▶

Page 8 of 18

Go Back

Full Screen

Close

Quit



- 6.3 随机逼近法
- 2. 随机逼近法
- 3. 随机牛顿法
- 6.4 递推参数估计算法...

那么，模型式(1)的辨识问题归结成求如下方程的解

$$E\{q(\theta, \mathbf{D}(k))\} = 0 \quad (18)$$

利用随机逼近原理，有

$$\hat{\theta}(k) = \hat{\theta}(k-1) + \rho(k)q[\hat{\theta}(k-1), \mathbf{D}(k)] \quad (19)$$

其中 $\rho(k)$ 为收敛因子，必须满足式(10)的条件。

Home Page

Title Page

◀◀ ▶▶

◀ ▶

Page 9 of 18

Go Back

Full Screen

Close

Quit



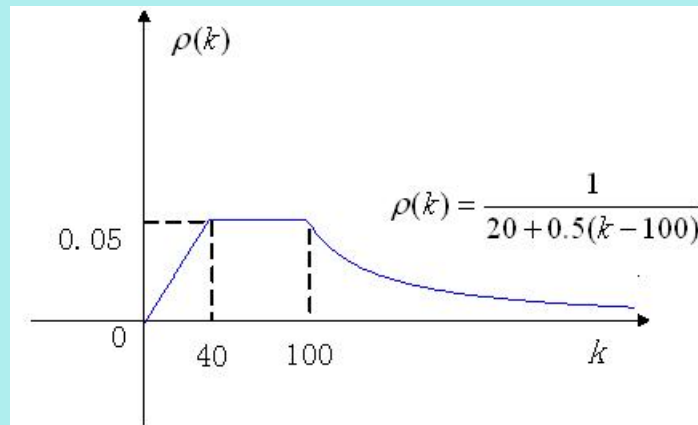
那么，模型式(1)的辨识问题归结成求如下方程的解

$$E\{q(\theta, \mathbf{D}(k))\} = 0 \quad (18)$$

利用随机逼近原理，有

$$\hat{\theta}(k) = \hat{\theta}(k-1) + \rho(k)q[\hat{\theta}(k-1), \mathbf{D}(k)] \quad (19)$$

其中 $\rho(k)$ 为收敛因子，必须满足式(10)的条件。一般来说， $\rho(k)$ 随着 k 的增加要有足够的下降速度，但 $\rho(k)$ 不能下降得太快，否则被处理的数据总量太少。Isermann推荐一种选择 $\rho(k)$ 的方法，如图6.1所示。



- 6.3 随机逼近法
- 2. 随机逼近法
- 3. 随机牛顿法
- 6.4 递推参数估计算法...

Home Page

Title Page

◀ ▶

◀ ▶

Page 9 of 18

Go Back

Full Screen

Close

Quit



- 6.3 随机逼近法
- 2. 随机逼近法
- 3. 随机牛顿法
- 6.4 递推参数估计算法...

3. 随机牛顿法

这里所讨论的随机逼近法实质上就是沿着准则函数的一阶负梯度方向去搜索极小值点，其作用和最速下降法基本上是一致的。但是，当搜索点接近极小值点时，这种算法的收敛速度变成很慢。要想获取较高的辨识精度，辨识时间将很长。为了加快辨识速度，可采用如下的牛顿算法。

[Home Page](#)

[Title Page](#)

[◀](#) [▶](#)

[◀](#) [▶](#)

Page 10 of 18

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)



- 6.3 随机逼近法
- 2. 随机逼近法
- 3. 随机牛顿法
- 6.4 递推参数估计算法...

3. 随机牛顿法

这里所讨论的随机逼近法实质上就是沿着准则函数的一阶负梯度方向去搜索极小值点，其作用和最速下降法基本上是一致的。但是，当搜索点接近极小值点时，这种算法的收敛速度变成很慢。要想获取较高的辨识精度，辨识时间将很长。为了加快辨识速度，可采用如下的牛顿算法。

$$\hat{\theta}(k) = \hat{\theta}(k-1) - \left[\frac{\partial J^2(\theta)}{\partial \theta \partial \theta^T} \right]^{-1} \left[\frac{\partial J(\theta)}{\partial \theta} \right]^T \bigg|_{\hat{\theta}(k-1)} \quad (20)$$

其中 $\frac{\partial J^2(\theta)}{\partial \theta \partial \theta^T}$ 表示准则函数 $J(\theta)$ 关于 θ 的二阶导数，称为Hessian矩阵。显然，Hessian矩阵是对称阵，而且在递推计算过程中必须保证它的正定性。

[Home Page](#)[Title Page](#)[◀◀](#) [▶▶](#)[◀](#) [▶](#)[Page 10 of 18](#)[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)



- 6.3 随机逼近法
- 2. 随机逼近法
- 3. 随机牛顿法
- 6.4 递推参数估计算法...

3. 随机牛顿法

这里所讨论的随机逼近法实质上就是沿着准则函数的一阶负梯度方向去搜索极小值点，其作用和最速下降法基本上是一致的。但是，当搜索点接近极小值点时，这种算法的收敛速度变成很慢。要想获取较高的辨识精度，辨识时间将很长。为了加快辨识速度，可采用如下的牛顿算法。

$$\hat{\theta}(k) = \hat{\theta}(k-1) - \left[\frac{\partial^2 J(\theta)}{\partial \theta \partial \theta^T} \right]^{-1} \left[\frac{\partial J(\theta)}{\partial \theta} \right]^T \bigg|_{\hat{\theta}(k-1)} \quad (20)$$

其中 $\frac{\partial^2 J(\theta)}{\partial \theta \partial \theta^T}$ 表示准则函数 $J(\theta)$ 关于 θ 的二阶导数，称为Hessian矩阵。显然，Hessian矩阵是对称阵，而且在递推计算过程中必须保证它的正定性。

如果准则函数是确定性函数，那么牛顿算法式(20)将有较快的收敛速度和较好的辨识精度。但是，这里所用的准则函数一般为回归函数，如式(16)所示，牛顿算法基本上不能适用，而且Hessian矩阵的表达式也难求。L.Ljung提出了改善这个算法的随机牛顿算法(SNA)

[Home Page](#)[Title Page](#)

Page 10 of 18

[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)



$$\hat{\theta}(k) = \hat{\theta}(k-1) + \rho(k)R^{-1}(k)q(\theta, \mathbf{D}(k))|_{\hat{\theta}(k-1)} \quad (21)$$

其中 $q(\theta, \mathbf{D}(k))$ 是准则函数的一阶负梯度方向， $R(k)$ 是Hessian矩阵在 $\hat{\theta}(k-1)$ 点上的近似形式，在特定的准则函数下，它可以再次用随机逼近法来确定。下面，利用随机牛顿法来讨论模型式(1)的参数辨识问题，其准则函数取式(2)。根据式(17)，有

$$q(\theta, \mathbf{D}(k)) = \varphi(k)[y(k) - \varphi^T(k)\theta] \quad (22)$$

6.3 随机逼近法

2. 随机逼近法

3. 随机牛顿法

6.4 递推参数估计算法...

Home Page

Title Page

◀ ▶

◀ ▶

Page 11 of 18

Go Back

Full Screen

Close

Quit



$$\hat{\theta}(k) = \hat{\theta}(k-1) + \rho(k)R^{-1}(k)q(\theta, \mathbf{D}(k))|_{\hat{\theta}(k-1)} \quad (21)$$

其中 $q(\theta, \mathbf{D}(k))$ 是准则函数的一阶负梯度方向， $R(k)$ 是Hessian矩阵在 $\hat{\theta}(k-1)$ 点上的近似形式，在特定的准则函数下，它可以再次用随机逼近法来确定。下面，利用随机牛顿法来讨论模型式(1)的参数辨识问题，其准则函数取式(2)。根据式(17)，有

$$q(\theta, \mathbf{D}(k)) = \varphi(k)[y(k) - \varphi^T(k)\theta] \quad (22)$$

且Hessian矩阵为

$$\frac{\partial^2 J^2(\theta)}{\partial \theta \partial \theta^T} = E\{\varphi(k)\varphi^T(k)\} \quad (23)$$

6.3 随机逼近法

2. 随机逼近法

3. 随机牛顿法

6.4 递推参数估计算法...

Home Page

Title Page

◀◀ ▶▶

◀ ▶

Page 11 of 18

Go Back

Full Screen

Close

Quit



$$\hat{\theta}(k) = \hat{\theta}(k-1) + \rho(k)R^{-1}(k)q(\theta, \mathbf{D}(k))|_{\hat{\theta}(k-1)} \quad (21)$$

其中 $q(\theta, \mathbf{D}(k))$ 是准则函数的一阶负梯度方向， $R(k)$ 是Hessian矩阵在 $\hat{\theta}(k-1)$ 点上的近似形式，在特定的准则函数下，它可以再次用随机逼近法来确定。下面，利用随机牛顿法来讨论模型式(1)的参数辨识问题，其准则函数取式(2)。根据式(17)，有

$$q(\theta, \mathbf{D}(k)) = \varphi(k)[y(k) - \varphi^T(k)\theta] \quad (22)$$

且Hessian矩阵为

$$\frac{\partial^2 J^2(\theta)}{\partial \theta \partial \theta^T} = E\{\varphi(k)\varphi^T(k)\} \quad (23)$$

显然，Hessian矩阵是回归函数，其准确表达式难以确定。设 $R(k)$ 是Hessian矩阵在 k 时刻的估计值，则有

$$E\{\varphi(k)\varphi^T(k) - R(k)\} = 0 \quad (24)$$

6.3 随机逼近法

2. 随机逼近法

3. 随机牛顿法

6.4 递推参数估计算法...

Home Page

Title Page

◀ ▶

◀ ▶

Page 11 of 18

Go Back

Full Screen

Close

Quit



根据式(9), 可得 $R(k)$ 的随机逼近算法为

$$R(k) = R(k-1) + \rho(k)[\varphi(k)\varphi^T(k) - R(k-1)] \quad (25)$$

6.3 随机逼近法

2. 随机逼近法

3. 随机牛顿法

6.4 递推参数估计算法...

Home Page

Title Page

◀◀ ▶▶

◀ ▶

Page 12 of 18

Go Back

Full Screen

Close

Quit



根据式(9), 可得 $R(k)$ 的随机逼近算法为

$$R(k) = R(k-1) + \rho(k)[\varphi(k)\varphi^T(k) - R(k-1)] \quad (25)$$

于是, 模型式(1)的随机牛顿算法归结为

$$\hat{\theta}(k) = \hat{\theta}(k-1) + \rho(k)R^{-1}(k)\varphi(k)[y(k) - \varphi^T(k)\hat{\theta}(k-1)] \quad (26)$$

$$R(k) = R(k-1) + \rho(k)[\varphi(k)\varphi^T(k) - R(k-1)] \quad (27)$$

- 6.3 随机逼近法
- 2. 随机逼近法
- 3. 随机牛顿法
- 6.4 递推参数估计算法...

[Home Page](#)

[Title Page](#)

[◀◀](#) [▶▶](#)

[◀](#) [▶](#)

Page 12 of 18

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)



根据式(9)，可得 $R(k)$ 的随机逼近算法为

$$R(k) = R(k-1) + \rho(k)[\varphi(k)\varphi^T(k) - R(k-1)] \quad (25)$$

于是，模型式(1)的随机牛顿算法归结为

$$\hat{\theta}(k) = \hat{\theta}(k-1) + \rho(k)R^{-1}(k)\varphi(k)[y(k) - \varphi^T(k)\hat{\theta}(k-1)] \quad (26)$$

$$R(k) = R(k-1) + \rho(k)[\varphi(k)\varphi^T(k) - R(k-1)] \quad (27)$$

随机牛顿算法是很有用的算法形式。以它为基础，可以揭示许多辨识算法的内在联系。式(26)中，当取 $\rho(k) = \frac{1}{k}$ ， $R(k) = \frac{1}{k}P^{-1}(k)$ 时，即得到最小二乘递推算法。当准则函数取为 $J(\theta) = \frac{1}{2}E\{\nu^2(\theta)\}$ 的随机逼近法(或随机牛顿法)就是递推预报误差法(或递推极大似然法)。

6.3 随机逼近法

2. 随机逼近法

3. 随机牛顿法

6.4 递推参数估计算法...

Home Page

Title Page

◀ ▶

◀ ▶

Page 12 of 18

Go Back

Full Screen

Close

Quit



6.4 递推参数估计算法的收敛性描述

RLS、RELS、RIV、RML、SA等递推参数估计算法可以统一地表示如下

$$\hat{\theta}_{N+1} = \hat{\theta}_N + K_{N+1}\hat{e}(N+1) \quad (28)$$

$$K_{N+1} = F_{N+1}P_N\varphi_N \quad (29)$$

$$\hat{e}(N+1) = y(N+1) - \varphi_N^T \hat{\theta}_N \quad (30)$$

它们的差别只在于参数向量 $\hat{\theta}_N$ 、数据向量 φ_N 和修正向量 K_{N+1} 具有不同的形式。

- 6.3 随机逼近法
- 2. 随机逼近法
- 3. 随机牛顿法
- 6.4 递推参数估计算法...

[Home Page](#)

[Title Page](#)

[◀](#) [▶](#)

[◀](#) [▶](#)

Page 13 of 18

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)



6.4 递推参数估计算法的收敛性描述

RLS、RELS、RIV、RML、SA等递推参数估计算法可以统一地表示如下

$$\hat{\theta}_{N+1} = \hat{\theta}_N + K_{N+1}\hat{e}(N+1) \quad (28)$$

$$K_{N+1} = F_{N+1}P_N\varphi_N \quad (29)$$

$$\hat{e}(N+1) = y(N+1) - \varphi_N^T \hat{\theta}_N \quad (30)$$

它们的差别只在于参数向量 $\hat{\theta}_N$ 、数据向量 φ_N 和修正向量 K_{N+1} 具有不同的形式。

递推算法是时变的非线性差分方程，对于递推算法的收敛性，L.Ljung应用定常确定性微分方程的稳定性解，即ODE法进行了研究。这里仅给出一个启发性的描述，并引证一些关键性的收敛结果。

6.3 随机逼近法

2. 随机逼近法

3. 随机牛顿法

6.4 递推参数估计算法...

Home Page

Title Page

◀ ▶

◀ ▶

Page 13 of 18

Go Back

Full Screen

Close

Quit

从研究增广矩阵法的随机逼近法开始。由递推估计算法，取 $P_{N+1} = 1/(N + 1)$ ，则

$$\hat{\theta}_{N+1} = \hat{\theta}_N + \frac{1}{N+1} \varphi_N \hat{e}(N+1) \quad (31)$$



6.3 随机逼近法

2. 随机逼近法

3. 随机牛顿法

6.4 递推参数估计算法...

Home Page

Title Page



Page 14 of 18

Go Back

Full Screen

Close

Quit

从研究增广矩阵法的随机逼近法开始。由递推估计算法，取 $P_{N+1} = 1/(N+1)$ ，则

$$\hat{\theta}_{N+1} = \hat{\theta}_N + \frac{1}{N+1} \varphi_N \hat{e}(N+1) \quad (31)$$

注意到 φ_N 和 $\hat{e}(N+1)$ 是 $N+1$ 时刻以前所有参数估计 $\{\hat{\theta}_N, \hat{\theta}_{N-1}, \hat{\theta}_{N-2}, \dots\}$ 的函数。前面已经分析，当时间增加时，修正向量 K_{N+1} 下降， $\hat{\theta}_N$ 将与 $\hat{\theta}_{N-1}, \hat{\theta}_{N-2}$ 等接近。所以，直观上可以用平稳过程 $\bar{\varphi}_N$ 和 $\bar{e}(N+1, \hat{\theta})$ 来代替式(31)中的 φ_N 和 $\hat{e}(N+1)$ (即当 N 充分大时， $\bar{\varphi}_N$ 和 $\bar{e}(N+1, \hat{\theta})$ 对 $\hat{\theta}$ 的依赖关系不随时间而改变)。其中 $\bar{e}(N+1, \hat{\theta})$ 和 $\bar{\varphi}_N$ 定义如下

$$\bar{e}(N+1, \hat{\theta}_N) = y(N+1) - \bar{\varphi}_N^T(\hat{\theta}_N) \hat{\theta}_N$$

$$\bar{\varphi}_N(\hat{\theta}_N) = [-y(N) \cdots -y(N-n) \ u(N) \cdots u(N-n) \ \bar{e}(N, \hat{\theta}) \cdots \bar{e}(N-n, \hat{\theta})]^T$$



6.3 随机逼近法

2. 随机逼近法

3. 随机牛顿法

6.4 递推参数估计算法...

Home Page

Title Page



Page 14 of 18

Go Back

Full Screen

Close

Quit

从研究增广矩阵法的随机逼近法开始。由递推估计算法，取 $P_{N+1} = 1/(N+1)$ ，则

$$\hat{\theta}_{N+1} = \hat{\theta}_N + \frac{1}{N+1} \varphi_N \hat{e}(N+1) \quad (31)$$

注意到 φ_N 和 $\hat{e}(N+1)$ 是 $N+1$ 时刻以前所有参数估计 $\{\hat{\theta}_N, \hat{\theta}_{N-1}, \hat{\theta}_{N-2}, \dots\}$ 的函数。前面已经分析，当时间增加时，修正向量 K_{N+1} 下降， $\hat{\theta}_N$ 将与 $\hat{\theta}_{N-1}, \hat{\theta}_{N-2}$ 等接近。所以，直观上可以用平稳过程 $\bar{\varphi}_N$ 和 $\bar{e}(N+1, \hat{\theta})$ 来代替式(31)中的 φ_N 和 $\hat{e}(N+1)$ (即当 N 充分大时， $\bar{\varphi}_N$ 和 $\bar{e}(N+1, \hat{\theta})$ 对 $\hat{\theta}$ 的依赖关系不随时间而改变)。其中 $\bar{e}(N+1, \hat{\theta})$ 和 $\bar{\varphi}_N$ 定义如下

$$\bar{e}(N+1, \hat{\theta}_N) = y(N+1) - \bar{\varphi}_N^T(\hat{\theta}_N) \hat{\theta}_N$$

$$\bar{\varphi}_N(\hat{\theta}_N) = [-y(N) \cdots -y(N-n) \ u(N) \cdots u(N-n) \ \bar{e}(N, \hat{\theta}) \cdots \bar{e}(N-n, \hat{\theta})]^T$$

于是，式(31)可写为

$$\hat{\theta}_{N+1} = \hat{\theta}_N + \frac{1}{N+1} \bar{\varphi}_N(\hat{\theta}_N) \bar{e}(N+1, \hat{\theta}_N) \quad (32)$$

或

$$\hat{\theta}_{N+M} = \hat{\theta}_N + \sum_{k=N+1}^{N+M} \frac{1}{k} \bar{\varphi}_k(\hat{\theta}_k) \bar{e}(k+1, \hat{\theta}_k) \quad (33)$$



6.3 随机逼近法

2. 随机逼近法

3. 随机牛顿法

6.4 递推参数估计算法...

Home Page

Title Page



Page 14 of 18

Go Back

Full Screen

Close

Quit

上式中总体平均由时间平均所代替，总体平均为

$$\hat{\theta}_{N+M} = \hat{\theta}_N + E \left[\varphi_N(\hat{\theta}_N) e(N+1, \hat{\theta}_N) \right] \sum_{k=N+1}^{N+M} \frac{1}{k} \quad (34)$$



- 6.3 随机逼近法
- 2. 随机逼近法
- 3. 随机牛顿法
- 6.4 递推参数估计算法...

[Home Page](#)

[Title Page](#)

[◀◀](#) [▶▶](#)

[◀](#) [▶](#)

Page 15 of 18

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)



上式中总体平均由时间平均所代替，总体平均为

$$\hat{\theta}_{N+M} = \hat{\theta}_N + E \left[\varphi_N(\hat{\theta}_N) e(N+1, \hat{\theta}_N) \right] \sum_{k=N+1}^{N+M} \frac{1}{k} \quad (34)$$

上式中 $\sum_{k=N+1}^{N+M} \frac{1}{k}$ ，当 $N \rightarrow \infty$ 时，为一小量。如果用 Δt 来表示这个小量，即选取时间尺度，置 $N = t, N + M = t + \Delta t$ ，则式(34)为

$$\lim_{\Delta t \rightarrow 0} \frac{\hat{\theta}_{N+M} - \hat{\theta}_N}{\Delta t} = \frac{\partial \hat{\theta}}{\partial t} = R^{-1} f(\hat{\theta}) \quad (35)$$

其中

$$R = \lim_{N \rightarrow \infty} R_N, \quad \hat{\theta} = \lim_{N \rightarrow \infty} \hat{\theta}_N$$

$$f(\hat{\theta}) = E \left[\bar{\varphi}_N(\hat{\theta}) \bar{e}(N, \hat{\theta}) \right]$$

那么，式(34)可以看成是式(35)的微分方程的解。

- 6.3 随机逼近法
- 2. 随机逼近法
- 3. 随机牛顿法
- 6.4 递推参数估计算法...

Home Page

Title Page

◀ ▶

◀ ▶

Page 15 of 18

Go Back

Full Screen

Close

Quit



又，由协方差阵的递推公式

$$R_{N+1} = R_N + \frac{1}{N+1} [\bar{\varphi}_{N+1} \bar{\varphi}_{N+1}^T - R_N] \quad (36)$$

按照同样的推导过程，可得当 $N \rightarrow \infty$ 时，上式可看成如下微分方程的解

$$\frac{dR}{dt} = G(\hat{\theta}) - R \quad (37)$$

其中

$$G(\hat{\theta}) = E \left[\bar{\varphi}_N(\hat{\theta}) \bar{\varphi}_N^T(\hat{\theta}) \right]$$

- 6.3 随机逼近法
- 2. 随机逼近法
- 3. 随机牛顿法
- 6.4 递推参数估计算法...

[Home Page](#)

[Title Page](#)

[◀◀](#) [▶▶](#)

[◀](#) [▶](#)

Page 16 of 18

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)



又，由协方差阵的递推公式

$$R_{N+1} = R_N + \frac{1}{N+1} [\bar{\varphi}_{N+1} \bar{\varphi}_{N+1}^T - R_N] \quad (36)$$

按照同样的推导过程，可得当 $N \rightarrow \infty$ 时，上式可看成如下微分方程的解

$$\frac{dR}{dt} = G(\hat{\theta}) - R \quad (37)$$

其中

$$G(\hat{\theta}) = E \left[\bar{\varphi}_N(\hat{\theta}) \bar{\varphi}_N^T(\hat{\theta}) \right]$$

这样，式(35)和式(37)描述了递推辨识算法的渐近轨线，可以把它用于前面很大的一类递推算法上。于是，可以把辨识算法的收敛性与微分方程的渐近稳定性质联系起来。

6.3 随机逼近法

2. 随机逼近法

3. 随机牛顿法

6.4 递推参数估计算法...

Home Page

Title Page

◀ ▶

◀ ▶

Page 16 of 18

Go Back

Full Screen

Close

Quit

定理6.3 考虑常微分方程

$$\begin{cases} \dot{\theta} = R^{-1}f(\theta) \\ \dot{R} = G(\theta) - R \end{cases} \quad (38)$$

其中 $f(\hat{\theta}) = E[\bar{\varphi}_N(\hat{\theta})\bar{e}(N, \hat{\theta})]$, $G(\hat{\theta}) = E[\bar{\varphi}_N(\hat{\theta})\bar{\varphi}_N^T(\hat{\theta})]$ 。其稳定性分析的主要结果可叙述如下:

(1)如果 $[\hat{\theta}, R] = [\theta^*, G(\theta^*)]^T$ 是上述微分方程的一个总体平衡点, 则

$$\hat{\theta}_N \longrightarrow \theta^* \quad w.p.1$$

即以概率1收敛。

(2)递推算法所有收敛点的集合是

$$D_s = \{\theta | f(\theta) = 0\}$$

的一个子集。

(3) $\hat{\theta}_N$ 收敛到 θ^* 的必要条件是式(38)在 θ^* 的一个邻域内局部稳定, 即要求矩阵

$$G^{-1}(\theta^*) \left. \frac{df(\theta)}{d\theta} \right|_{\theta=\theta^*}$$

的所有特征值具有负实部。



6.3 随机逼近法

2. 随机逼近法

3. 随机牛顿法

6.4 递推参数估计算法...

Home Page

Title Page

◀ ▶

◀ ▶

Page 17 of 18

Go Back

Full Screen

Close

Quit



6.3 随机逼近法

2. 随机逼近法

3. 随机牛顿法

6.4 递推参数估计算法...

将上述定理用于各种递推算法中，可得到各种递推算法收敛性的结论：

(1) 递推最小二乘法总收敛于真值。

(2) 递推辅助变量法也总是收敛于真值。

(3) 如果信噪比高，递推广义最小二乘法收敛于参数真值；如果信噪比很低，那么收敛到非真值的情况可能出现。

(4) 如果系统是一个滑动平均过程，或是一阶自回归过程，增广最小二乘法收敛于参数真值。

(5) 如果系统是一个自回归滑动平均过程，近似递推极大似然法收敛于参数真值；对于受控的自回归滑动平均模型，如果信噪比高，可收敛于真值；在这两种情况下，都是收敛于似然函数的局部极大值。

Home Page

Title Page

◀

▶

◀

▶

Page 18 of 18

Go Back

Full Screen

Close

Quit