MA 模型和 AR 模型的数值模拟与阶数估计*

统计 91 董晟渤, 2193510853

西安交通大学数学与统计学院

日期：2022 年 3 月 23 日

摘　要

本文分别生成了 MA 模型和 AR 模型的模拟数据, 绘制了时间序列图, 并且对于 MA 模型和 AR 模型, 分别计算了自回归函数与偏自回归函数, 据此进行阶数估计.

关键词: MA 模型, AR 模型, 数值模拟, 阶数估计.

目录

1　MA 模型 **2**

1.1　理论基础 2

1.2　数值模拟 2

2　AR 模型 **4**

2.1　理论基础 4

2.2　数值模拟 5

附录: 数值模拟的 MATLAB 实现 **7**

A　MA 模型的数值模拟 7

B　AR 模型的数值模拟 9

*2021-2022 学年第二学期, 课程: 时间序列与金融统计, 指导老师: 惠永昌.

1 MA 模型

1.1 理论基础

在介绍时间序列模型时, 首先引入的是 MA(q) 模型, 其中 $q \geq 1$ 是给定的整数.

定义 1.1 (MA 模型)**.** 设 $\varepsilon_t \sim \mathrm{WN}(0, \sigma^2)$, 正整数 $q \geq 1$, 若对常数 $\mu, a_1, \cdots, a_q$, 有

$$X_t = \mu + \varepsilon_t + a_1\varepsilon_{t-1} + \cdots + a_q\varepsilon_{t-q},$$

则记 $X_t \sim \mathrm{MA}(q)$, 称 q 为阶数. 对于整数 k, 称

$$\gamma(k) = \mathrm{Cov}(X_{t+k}, X_t), \quad \rho(k) = \frac{\gamma(k)}{\gamma(0)}.$$

分别为自协方差函数和自回归函数.

命题 1.1. 设 $X_t \sim \mathrm{MA}(q)$, 则当 $|k| > q$ 时, 有 $\gamma(k) = \rho(k) = 0$.

对于来自某个模型的实际数据 $\{X_1, X_2, \cdots, X_T\}$, 我们通常用

$$\hat{\gamma}(k) = \frac{1}{T}\sum_{t=k+1}^{T}(X_t - \bar{X})(X_{t-k} - \bar{X}), \quad \hat{\rho}(k) = \frac{\hat{\gamma}(k)}{\hat{\gamma}(0)}$$

来作为 $\gamma(k)$ 和 $\rho(k)$ 的估计. 从而, 我们可以看 $\hat{\rho}(k)$ 与 0 的接近程度, 据此估计 MA(q) 模型的阶数 q.

定理 1.2 (Fan & Yao, 2003)**.** 设 $X_t \sim \mathrm{MA}(q)$, $\varepsilon_t \sim \mathrm{IID}(0, \sigma^2)$ 且 $\mathbb{E}\varepsilon_t^4 < \infty$, 则

$$\sqrt{T} \cdot \hat{\rho}(k) \xrightarrow{d} \mathcal{N}\left(0, 1 + 2 \cdot \sum_{j=1}^{q} \rho^2(j)\right), \quad k > q.$$

定理1.2得到了 $\hat{\rho}(k)(k > q)$ 的中心极限定理, 并据此得到了一个 $\hat{\rho}(k)(k > q)$ 的近似置信区间 (取置信水平 $\alpha = 0.95$):

$$\left[-\sqrt{1 + 2 \cdot \sum_{j=1}^{k-1} \rho^2(j)} \cdot \frac{1.96}{\sqrt{T}}, \sqrt{1 + 2 \cdot \sum_{j=1}^{k-1} \rho^2(j)} \cdot \frac{1.96}{\sqrt{T}}\right].$$

这是进行阶的估计的基础.

1.2 数值模拟

为了进一步理解时间序列, 并且在数值上验证阶的估计的合理性, 我们编写程序, 完成以下工作:

- 从给定的 MA 模型, 产生模拟数据;
- 由实际数据, 估计模型的阶数 q.

本节中, 我们考虑的四个模型如表所示 (设 $\varepsilon_t \sim \mathcal{N}(0, 1)$, 取 $T = 100$):

编号	模型	表达式
模型 1.1	MA(1)	$X_t = \varepsilon_t + 0.7\varepsilon_{t-1}$
模型 1.2	MA(1)	$X_t = \varepsilon_t - 0.7\varepsilon_{t-1}$
模型 1.3	MA(2)	$X_t = \varepsilon_t + 0.7\varepsilon_{t-1} - 0.4\varepsilon_{t-2}$
模型 1.4	MA(4)	$X_t = \varepsilon_t + 0.7\varepsilon_{t-1} - 0.4\varepsilon_{t-2} + 0.6\varepsilon_{t-3} + 0.8\varepsilon_{t-4}$

对于模型 1.1, 结果如图1所示. 对结果进行分析, 可以发现除了 $\hat{\rho}(1)$ 以外, 剩下的所有的 $\hat{\rho}(k)$ 都落在置信区间内, 从而可以认为 $\rho(k) = 0(k \geq 1)$. 据此估计得到 $\hat{q} = 1$.

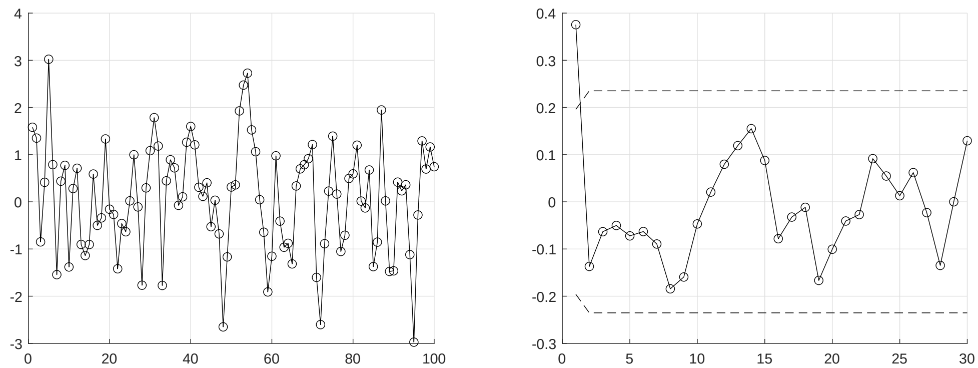

图 1: 模型 1.1 的模拟结果 (左图为时间序列, 右图为自回归函数与置信区间)

对于模型 1.2, 结果如图2所示. 发现自回归函数与图1中的自回归函数有点对称, 并且也和图1一样, 只有 $\hat{\rho}(1)$ 不在置信区间以内. 据此估计得到 $\hat{q} = 1$.

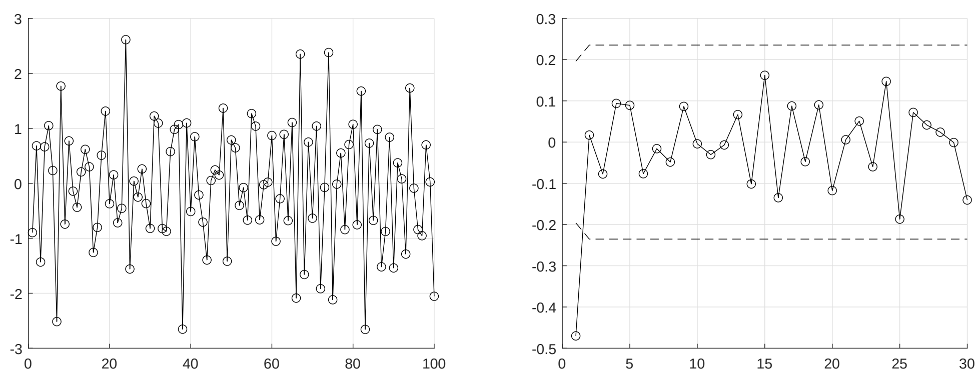

图 2: 模型 1.2 的模拟结果 (左图为时间序列, 右图为自回归函数与置信区间)

对于模型 1.3, 结果如图3所示. 此时自回归函数的估计值共有两项不在置信区间内, 可以认为 $\rho(k) = 0(k \geq 2)$, 据此估计得到 $\hat{q} = 2$.

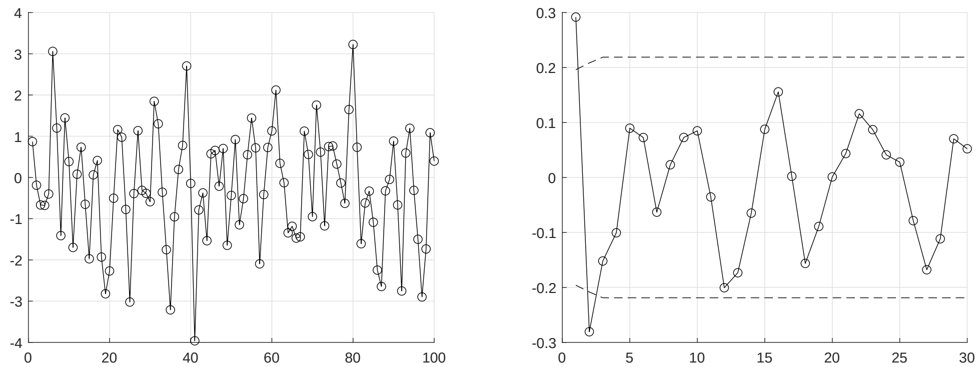

图 3: 模型 1.3 的模拟结果 (左图为时间序列, 右图为自回归函数与置信区间)

对于模型 1.4, 结果如图4所示. 此时虽然 $\hat{\rho}(1)$ 在置信区间内, 但是 $\hat{\rho}(2)$、$\hat{\rho}(3)$ 与 $\hat{\rho}(4)$ 均不在置信区间内, 因此在这里我们认为 $\rho(k)=0(k\geq 4)$, 据此估计得到 $\hat{q}=4$.

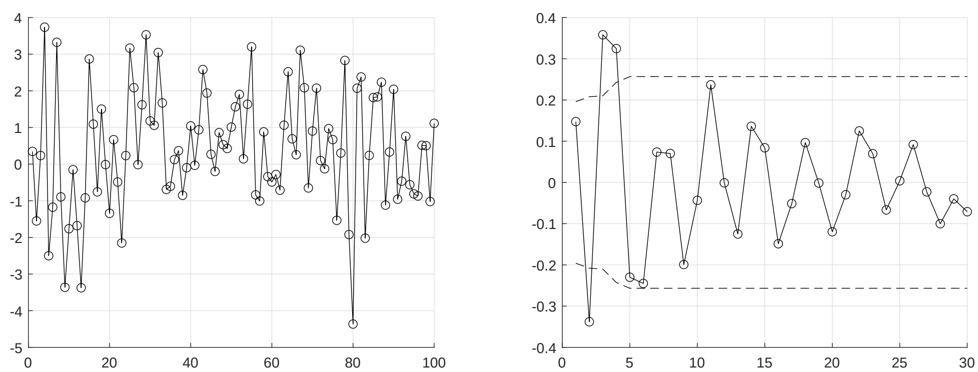

图 4: 模型 1.4 的模拟结果 (左图为时间序列, 右图为自回归函数与置信区间)

2 AR 模型

2.1 理论基础

另外一类重要的模型是 $\mathrm{AR}(p)$ 模型, 其中 $p\geq 1$ 是给定的整数.

定义 2.1 (AR 模型)**.** 设 $\varepsilon_t \sim \mathrm{WN}(0,\sigma^2)$, 正整数 $q\geq 1$, 若对常数 $c, b_1, b_2, \cdots, b_p$, 有

$$X_t = c + b_1 X_{t-1} + \cdots + b_p X_{t-q} + \varepsilon_t,$$

记 $X_t \sim \mathrm{AR}(p)$, 称 p 为阶数. 对于整数 $k\geq 0$, 设

$$(b_{k0}, b_{k1}, \cdots, b_{kk}) = \underset{b_1, b_2, \cdots, b_k}{\operatorname{argmin}}\ \mathbb{E}(X_{k+1} - b_0 - b_1 X_k - \cdots - b_k X_1)^2,$$

称 $\pi(k) = b_{kk}$ 为偏自相关函数.

命题 2.1. 设 $X_t \sim \mathrm{AR}(p)$, 则当 $k > p$ 时, 有 $\pi(k) = 0$.

对于来自某个模型的实际数据 $\{X_1, X_2, \cdots, X_T\}$, 设

$$\left(\hat{b}_{k1}, \hat{b}_{k2}, \cdots, \hat{b}_{kk}\right) = \underset{b_1, b_2, \cdots, b_k}{\operatorname{argmin}} \sum_{t=k+1}^{T} (X_t - b_1 X_{t-1} - \cdots - b_k X_{t-k})^2,$$

称 $\hat{\pi}(k) = \hat{b}_{kk}$ 为样本偏自相关函数. 和 MA 模型的处理方式相同, 我们只要看 $\hat{\pi}(k)$ 与 0 的接近程度, 就可以估计 $\mathrm{AR}(p)$ 模型的阶数 p.

定理 2.2 (Fan & Yao, 2003)**.** 设 $X_t \sim \mathrm{AR}(p)$, $\varepsilon_t \sim \mathrm{IID}(0, \sigma^2)$ 且 $\mathbb{E}\varepsilon_t^4 < \infty$, 则

$$\sqrt{T} \cdot \hat{\pi}(p) \xrightarrow{d} \mathcal{N}(0,1), \quad k > p.$$

定理2.2形式和1.2类似, 得到的是 $\hat{\pi}(k)(k > p)$ 的中心极限定理. 同样地, 在这里我们构造置信水平 $\alpha = 0.95$ 的置信区间 $\left[-\dfrac{1.96}{\sqrt{T}}, \dfrac{1.96}{\sqrt{T}}\right]$, 用于进行阶的估计.

2.2 数值模拟

进行类似上节的数值模拟, 我们考虑的四个模型如表所示 (设 $\varepsilon_t \sim \mathcal{N}(0,1)$, 取 $T = 100$):

编号	模型	表达式
模型 2.1	$\mathrm{AR}(1)$	$X_t = 0.7X_{t-1} + \varepsilon_t$
模型 2.2	$\mathrm{AR}(1)$	$X_t = -0.7X_{t-1} + \varepsilon_t$
模型 2.3	$\mathrm{AR}(2)$	$X_t = 0.26 + 0.5X_{t-1} + 0.24X_{t-2} + \varepsilon_t$
模型 2.4	$\mathrm{AR}(4)$	$X_t = 0.5X_{t-1} + 0.24X_{t-2} + 0.2X_{t-3} - 0.8X_{t-4} + \varepsilon_t$

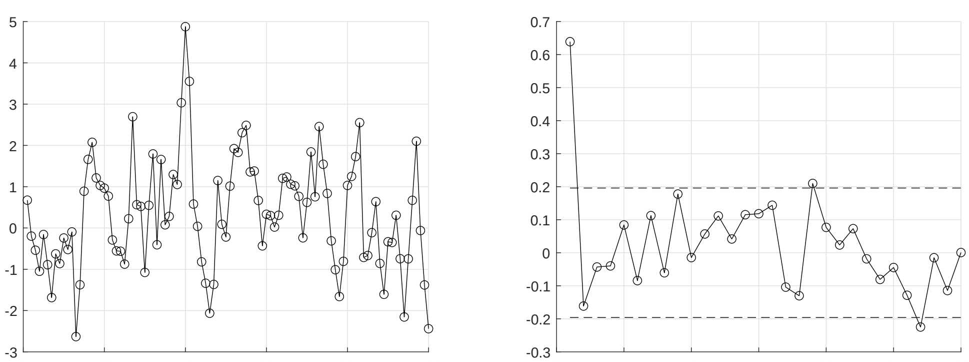

图 5: 模型 2.1 的模拟结果 (左图为时间序列, 右图为偏自回归函数与置信区间)

对于模型 2.1, 结果如图5所示. 对结果进行分析, 可以发现 $\hat{\pi}(1)$ 在置信区间之外, 而从

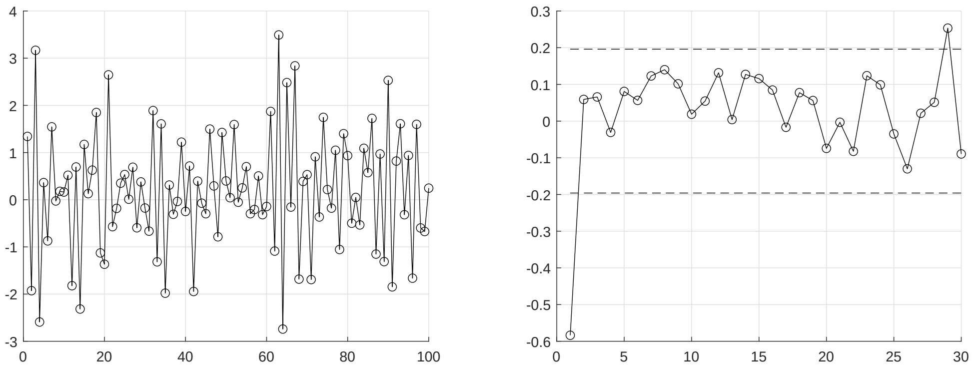

图 6: 模型 2.2 的模拟结果 (左图为时间序列, 右图为偏自回归函数与置信区间)

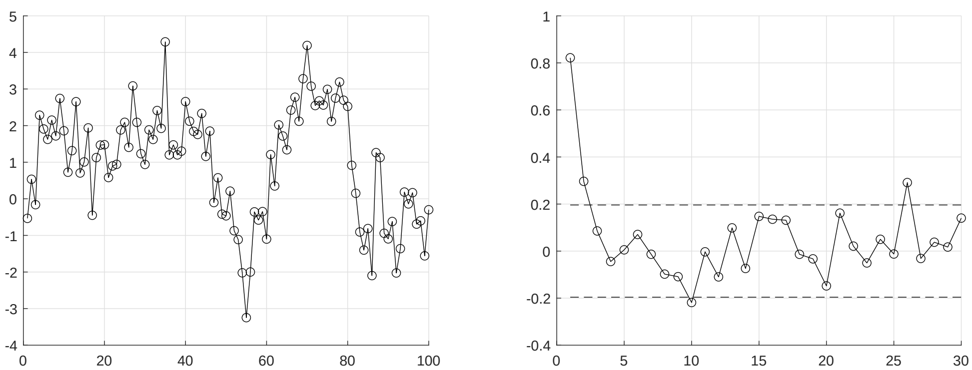

图 7: 模型 2.3 的模拟结果 (左图为时间序列, 右图为偏自回归函数与置信区间)

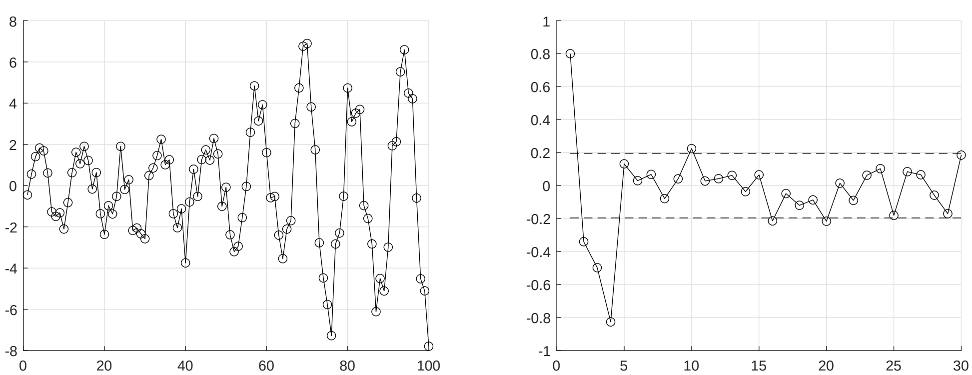

图 8: 模型 2.4 的模拟结果 (左图为时间序列, 右图为偏自回归函数与置信区间)

$\hat{\pi}(2)$ 开始的点, 几乎都在置信区间以内. 据此, 我们判断得到 $\hat{p}=1$.

对于模型 2.2, 结果如图6所示. 考虑到模型 2.1 和模型 2.2 是对称的, 我们发现这两个模型的偏自回归函数也是对称的, 并且根据结果得到 $\hat{p}=1$.

对于模型 2.3, 结果如图7所示, 注意到 $\hat{\pi}(1)$ 和 $\hat{\pi}(2)$ 都不在置信区间内, 并且根据结果, 可以认为 $\hat{\pi}(k)=0(k\geq 3)$, 因此得到 $\hat{p}=2$.

对于模型 2.4, 结果如图8所示, 其中偏自回归函数的前四个点都不在置信区间内, 从第五个点开始都在置信区间内, 因此认为 $\hat{p}=4$.

附录: 数值模拟的 MATLAB 实现

A MA 模型的数值模拟

```
%% 初始化

clc;
clear;
close;

%% 初始化

T = 100;
e = randn(1, T);
model = 4;

%% 模型

X = zeros(1, T); % 模拟数据
r = zeros(1, T); % 理论自回归函数
if model == 1 % 模型1
  X(1) = e(1);
  for i = 2 : T
    X(i) = e(i) + 0.7 * e(i - 1);
  end
  r(1) = 0.7 / (1 + 0.7^2);
elseif model == 2 % 模型2
  X(1) = e(1);
  for i = 2 : T
    X(i) = e(i) - 0.7 * e(i - 1);
```

```
    end
    r(1) = - 0.7 / (1 + 0.7^2);
elseif model == 3 % 模型3
    X(1) = e(1);
    X(2) = e(2) + 0.7 * e(1);
    for i = 3 : T
        X(i) = e(i) + 0.7 * e(i - 1) - 0.4 * e(i - 2);
    end
    r(1) = (0.7 - 0.4 * 0.7) / (1 + 0.4^2 + 0.7^2);
    r(2) = - 0.4 / (1 + 0.4^2 + 0.7^2);
elseif model == 4 % 模型4
    X(1) = e(1);
    X(2) = e(2) + 0.7 * e(1);
    X(3) = e(3) + 0.7 * e(2) - 0.4 * e(3);
    X(4) = e(4) + 0.7 * e(3) - 0.4 * e(2) + 0.6 * e(1);
    for i = 5 : T
        X(i) = e(i) + 0.7 * e(i - 1) - 0.4 * e(i - 2) + 0.6 * e(i - 3) + 0.8
            * e(i - 4);
    end
    r(1) = (0.7 - 0.7 * 0.4 - 0.4 * 0.6 + 0.6 * 0.8) / (1 + 0.8^2 + 0.6^2
        + 0.4^2 + 0.7^2);
    r(2) = (- 0.4 + 0.7 * 0.6 - 0.4 * 0.8) / (1 + 0.8^2 + 0.6^2 + 0.4^2 +
        0.7^2);
    r(3) = (0.6 + 0.7 * 0.8) / (1 + 0.8^2 + 0.6^2 + 0.4^2 + 0.7^2);
    r(4) = (0.8) / (1 + 0.8^2 + 0.6^2 + 0.4^2 + 0.7^2);
end

%% 置信区间
U = zeros(1, T); % 置信上限
U(1) = 1.96 / sqrt(T);
for k = 2 : T
    U(k) = 1.96 * sqrt(1 + 2 * sum(r(1 : k - 1).^2)) / sqrt(T);
end
L = -1 * U; % 置信下限

%% 自回归函数

gamma0 = sum((X - mean(X)).^2) / T; % 方差
gamma = zeros(1, T); % 自协方差函数
rho = zeros(1, T); % 自回归函数
```

```
for k = 1 : T
  sum = 0;
  for i = k + 1 : T
    sum = sum + (X(i) - mean(X)) * (X(i - k) - mean(X));
  end
  gamma(k) = sum / T;
  rho(k) = gamma(k) / gamma0;
end

%% 时间序列图

subplot(1, 2, 1);
scatter(1 : T, X, 'k');
hold on;
plot(1 : T, X, 'k');
grid on;

%% 自回归函数图

subplot(1, 2, 2);
t = 30;
scatter(1 : t, rho(1 : t), 'k');
hold on;
plot(1 : t, rho(1 : t), 'k');
hold on;
plot(1 : t, U(1 : t), '--k', 1 : t, L(1 : t), '--k');
grid on;

set(gcf, 'unit', 'centimeters', 'position', [10 10 30 10]);
```

B AR 模型的数值模拟

```
%% 初始化

clc;
clear;
close;

%% 初始化
```

```matlab
T = 100;
e = randn(1, T);
model = 4;

%% 模型

X = zeros(1, T); % 模拟数据
if model == 1 % 模型1
  X(1) = e(1);
  for i = 2 : T
    X(i) = 0.7 * X(i - 1) + e(i);
  end
elseif model == 2 % 模型2
  X(1) = e(1);
  for i = 2 : T
    X(i) = - 0.7 * X(i - 1) + e(i)
  end
elseif model == 3 % 模型3
  X(1) = 0.26 + e(1);
  X(2) = 0.26 + 0.5 * X(1) + e(2);
  for i = 3 : T
    X(i) = 0.26 + 0.5 * X(i - 1) + 0.24 * X(i - 2) + e(i);
  end
elseif model == 4 % 模型4
  X(1) = e(1);
  X(2) = 0.5 * X(1) + e(2);
  X(3) = 0.5 * X(2) + 0.24 * X(1) + e(3);
  X(4) = 0.5 * X(3) + 0.24 * X(2) + 0.2 * X(1) + e(4);
  for i = 5 : T
    X(i) = 0.5 * X(i - 1) + 0.24 * X(i - 2) + 0.2 * X(i - 3) - 0.8 * X(i
        - 4) + e(i);
  end
end

%% 置信区间

U = 1.96 * ones(1, T) / sqrt(T); % 置信上限
L = -1 * U; % 置信下限
```

```matlab
%% 偏自回归函数

pi = zeros(1, T);
for k = 1 : T
  b = zeros(1, k);
  fun = @(b) f(X, T, b, k);
  [b fval] = fminunc(fun, zeros(1, k));
  pi(k) = b(k);
end

%% 时间序列图

subplot(1, 2, 1);
scatter(1 : T, X, 'k');
hold on;
plot(1 : T, X, 'k');
grid on;

%% 自回归函数图

subplot(1, 2, 2);
t = 30;
scatter(1 : t, pi(1 : t), 'k');
hold on;
plot(1 : t, pi(1 : t), 'k');
hold on;
plot(1 : t, U(1 : t), '--k', 1 : t, L(1 : t), '--k');
grid on;

set(gcf, 'unit', 'centimeters', 'position', [10 10 30 10]);

%% 目标函数

function y = f(X, T, b, k)
  s1 = 0;
  for t = k + 1 : T
    s2 = 0;
    for i = 1 : k
      s2 = s2 + b(i) * X(t - i);
    end
```

```
    s1 = s1 + (X(t) - s2)^2;
  end
  y = s1;
end
```