时间序列分析——数字特征估计方法

李逢君　2016060601010

初步平稳化数据

上次作业使用最小二乘法拟合多项式去除了时间序列的趋势项，使用滑动平均去除时间序列的周期项。得到以下数据：

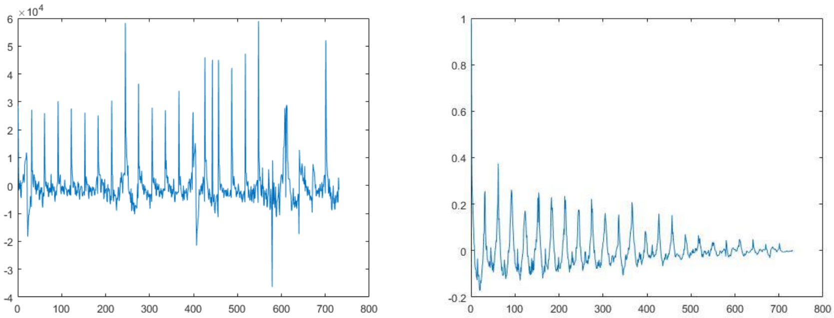

图 1　去除二次曲线趋势去周期项数据及自相关系数

在平稳化处理后的时间序列数据基础上对该序列的数字特征进行估计，并且自己实现相应的计算方法，而不仅仅是调用已有的函数。

均值函数的估计

随机变量总体 X 的样本为 $X_1,X_2,\cdots,X_N$，其均值 E(X)的估计量为

$$\overline{X_{\mathrm{n}}} = \frac{1}{n}\sum_{k=1}^{n} X_k$$

是均值的无偏、有效、相容估计。

在很多应用科学中，对时间序列$\{X_t\}$的观察是不能重复的，面对其一条现实：

$$x_1,x_2,\cdots,x_t,\cdots$$

取估计量

$$\hat{\mu}_t = \overline{X_{\mathrm{n}}} = \frac{1}{n}\sum_{k=1}^{n} X_k$$

作为 $E(X_t)=\hat{m}_t$的估计

实现代码为：

```
avg = sum(data)/length(data);
```

Matlab 代码验证：

```
mean(data)
```

零均值化

计算公式为：

$$\mathrm{X}' = X - mean(X)$$

实现代码为：

```
data = data - avg;
```

方差

计算公式为:

$$\mathrm{Var}(X) = \hat{\gamma}_0$$

实现代码为:

```
gama0 = var(data);
```

自协方差函数的估计

自协方差函数的样本估计量通常有两种类型:
一种是有偏估计量

$$\hat{\gamma}_k = \frac{1}{n}\sum_{t=1}^{N-h}(X_{t+k} - \overline{X})(X_t - \overline{X}), \qquad 0 \le k \le K$$

另一种是无偏估计量

$$\hat{\gamma}_k = \frac{1}{n-k}\sum_{t=1}^{N-h}(X_{t+k} - \overline{X})(X_t - \overline{X}), \qquad 0 \le k \le K$$

对比发现，对于较大的序列数N和较小的差数n，自相关函数的无偏估计和有偏估计相差不多。而有偏估计量收敛于令的速度更快，工程上常选用这种方法，因此在接下来的计算都使用有偏估计。

$\hat{\gamma}_k$表征了 X_t 和 X_{t+k} 之间的线性相关程度，对绝对值大的 h 值，$\hat{\gamma}_k$的精度会降低，根据 Box-Jenkins 原则，要求样本数量 N≥50，且 k≤$\frac{N}{4}$，一般 max(k)=$\frac{N}{10}$，结合实验数据我采用 k=50 来计算平稳数据的自相关函数和偏相关函数的前 50 项。

实现代码为:

```
L=50
% 自协方差函数
for i = 1:L
    gama1(i) = data(1:end-i)'*data(i+1:end)/data_length; % 有偏估计
end
```

自相关系数函数的估计

计算公式为:

$$\rho_k = \frac{\hat{\gamma}_k}{\hat{\gamma}_0}, \quad 0 \le k \le K$$

实现代码为:

```
rof1 = gama1 / gama0  % 有偏估计
autocorr(data, L); %matlab验证

figure;
plot([1 rho1],'o')
```

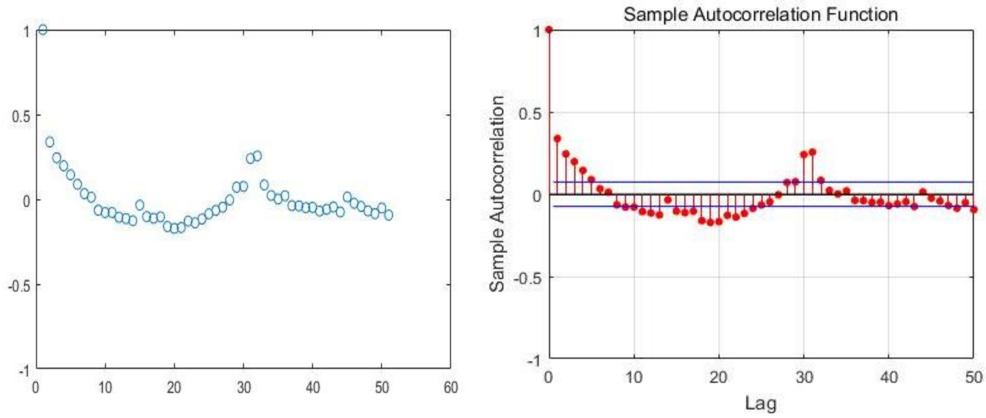

图 2　自相关函数图

偏相关函数的估计

使用递推公式

$$
\begin{cases}
\varphi_{1,1} = \rho_1 \\
\varphi_{k+1,k+1} = (\rho_{k+1} - \sum_{j=1}^{k} \rho_{k+1-j}\varphi_{k,j})(1 - \sum_{j=1}^{k} \rho_j \varphi_{k,j})^{-1} \\
\varphi_{k+1,j} = \varphi_{k,j} - \varphi_{k+1,k+1}\varphi_{k,k+1-j}, \quad j = 1,2,\cdots,k
\end{cases}
$$

实现结果为

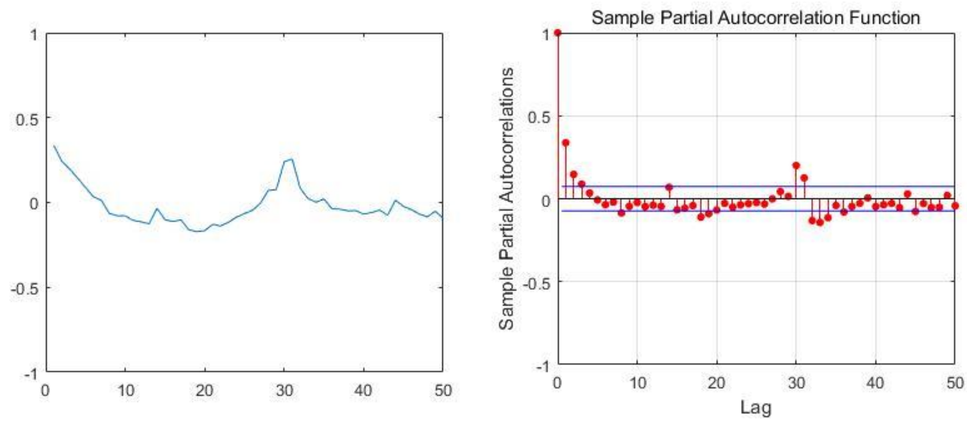

图 3　偏相关函数图

实现代码为:

```
% 偏相关函数
f(1,1) = rho1(1);
for k = 2:L
    s1 = rho1(k);
    s2 = 1;
    for j = i:k-1
        s1 = s1 - rho1(k-j)*f(k-1,j);
```

```
        s2 = s2 - rho1(j)*f(k-1,j);
    end
    f(k,k) = s1 / s2;
    for j = 1:k-1
        f(k,j) = f(k-1,j)-f(k,k)*f(k-1,k-j);
    end
end
```

所有代码如下:

```
clear, clc, close all
% xlsdata = xlsread('data.xlsx',1);
% data = xlsdata(:,2);
data = load('data.mat');
data = data.y4;
data_length = length(data);

% 平均函数
avg = mean(data);
avg_my = sum(data)/data_length; %matlab 验证

% 零均值化
data = data - avg;

% 方差
gama0 = var(data);

% k 取值
L=50;

% 自协方差函数
for i = 1:L
    gama1(i) = data(i+1:end)*data(1:end-i)'/data_length; % 有偏估计
    gama2(i) = data(i+1:end)*data(1:end-i)'/(data_length-L); % 无偏估计
end

% 自相关系数函数
rho1 = gama1 / gama0;
rho2 = gama2 / gama0;
autocorr(data, L); %matlab 验证

figure;
plot([1 rho1],'o')
```

```
% 偏相关函数
f(1,1) = rho1(1);
for k = 2:L
    s1 = rho1(k);
    s2 = 1;
    for j = i:k-1
        s1 = s1 - rho1(k-j)*f(k-1,j);
        s2 = s2 - rho1(j)*f(k-1,j);
    end
    f(k,k) = s1 / s2;
    for j = 1:k-1
        f(k,j) = f(k-1,j)-f(k,k)*f(k-1,k-j);
    end
end

pcorr = diag(f)'; % 偏相关系数
parcorr(data, L);%   matlab 验证
figure;
plot(pcorr);
```