MATLAB统计分析与应用

——数据的预处理

主讲人：谢中华

主要内容

- 数据的平滑处理
- 数据的标准化变换
- 数据的极差归一化变换

第一节 数据的平滑处理

一、 smooth函数

调用格式：

```
yy = smooth(y)
yy = smooth(y,span)
yy = smooth(y,method)
yy = smooth(y,span,method)
yy = smooth(y,'sgolay',degree)
yy = smooth(y,span,'sgolay',degree)
yy = smooth(x,y,...)
```

【例6.1-1】产生一列正弦波信号，加入噪声信号，然后调用smooth函数对加入噪声的正弦波进行滤波（平滑处理）

```
>> t = linspace(0,2*pi,500)';
>> y = 100*sin(t);
>> noise = normrnd(0,15,500,1);
>> y = y + noise;
>> figure;
>> plot(t,y);
>> xlabel('t');
>> ylabel('y = sin(t) + 噪声');
```

移动平均法：

```
>> yy1 = smooth(y,30);
>> figure;
>> plot(t,y,'k:');
>> hold on;
>> plot(t,yy1,'k','linewidth',3);
>> xlabel('t');
>> ylabel('moving');
>> legend('加噪波形','平滑后波形');
```

lowess方法：

```
>> yy2 = smooth(y,30,'lowess');
>> figure;
>> plot(t,y,'k:');
>> hold on;
>> plot(t,yy2,'k','linewidth',3);
>> xlabel('t');
>> ylabel('lowess');
>> legend('加噪波形','平滑后波形');
```

rlowess方法：

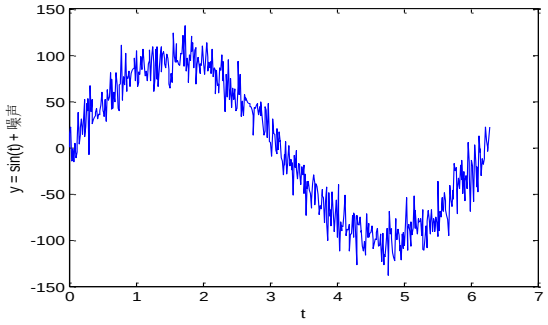
```
>> yy3 = smooth(y,30,'rlowess');
>> figure;
>> plot(t,y,'k:');
>> hold on;
>> plot(t,yy3,'k','linewidth',3);
>> xlabel('t');
>> ylabel('rlowess');
>> legend('加噪波形','平滑后波形');
```

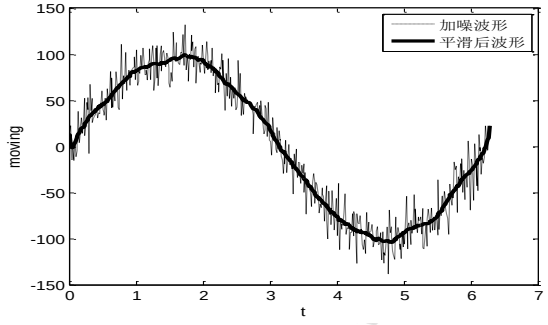
loess方法：

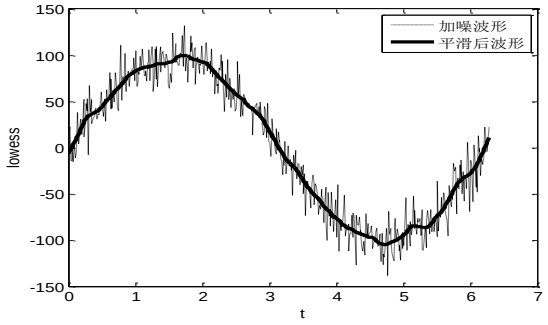
```
>> yy4 = smooth(y,30,'loess');
>> figure;
>> plot(t,y,'k:');
>> hold on;
>> plot(t,yy4,'k','linewidth',3);
>> xlabel('t');
>> ylabel('loess');
>> legend('加噪波形','平滑后波形');
```

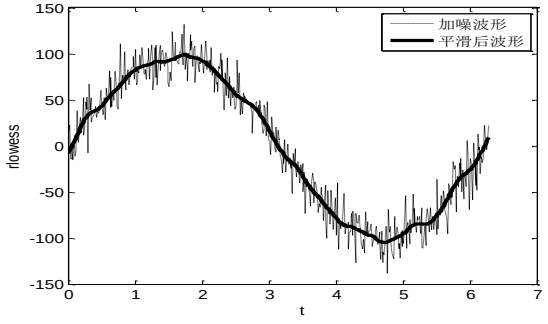
sgolay方法：

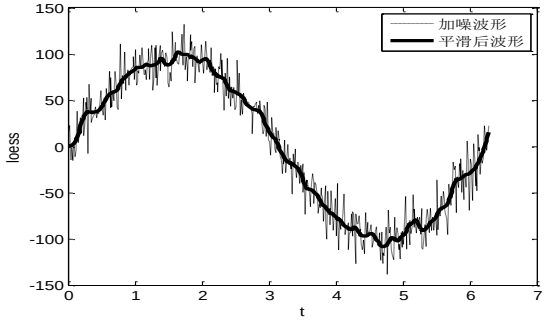
```
>> yy5 = smooth(y,30,'sgolay',3);
>> figure;
>> plot(t,y,'k:');
>> hold on;
>> plot(t,yy5,'k','linewidth',3);
>> xlabel('t');
>> ylabel('sgolay');
>> legend('加噪波形','平滑后波形');
```

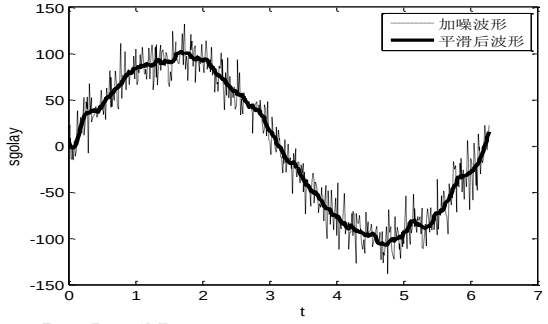

y = sin(t) + 噪声
t


加噪波形
平滑后波形
moving
t


加噪波形
平滑后波形
lowess
t


加噪波形
平滑后波形
rlowess
t


加噪波形
平滑后波形
loess
t


加噪波形
平滑后波形
sgolay
t

二、smoothts函数

调用格式：

output = smoothts(input)

output = smoothts(input, ‘b’, wsize)

output = smoothts(input, ‘g’, wsize, stdev)

output = smoothts(input, ‘e’, n)

【例6.1-2】现有上海股市日开盘价、最高价、最低价、收盘价、收益率等数据，时间跨度为2005年1月4日至2007年4月3日，共510组数据。完整数据保存在文件examp6_1_2.xls中，其中部分数据如下图所示。试调用smoothts函数对日收盘价数据进行平滑处理

	A	B	C	D	E	F	G
1	日期	开盘价	最高价	最低价	收盘价	收益率	
2	2005-1-4	1260.78	1260.78	1238.18	1242.77	-0.01428	
3	2005-1-5	1241.68	1258.58	1235.75	1251.94	0.008263	
4	2005-1-6	1252.49	1252.74	1234.24	1239.43	-0.01043	
5	2005-1-7	1239.32	1256.31	1235.51	1244.75	0.004381	
6	2005-1-10	1243.58	1252.72	1236.09	1252.4	0.007092	
7	2005-1-11	1252.71	1260.87	1247.84	1257.46	0.003792	
8	2005-1-12	1257.17	1257.19	1246.42	1256.92	-0.0002	
9	2005-1-13	1255.72	1259.5	1251.02	1256.31	0.00047	
10	2005-1-14	1255.87	1268.86	1243.87	1245.62	-0.00816	
11	2005-1-17	1235.57	1236.4	1214.07	1216.65	-0.01531	
12	2005-1-18	1215.78	1226.04	1207.05	1225.45	0.007954	
13	2005-1-19	1225.08	1225.08	1214.64	1218.11	-0.00569	
14	2005-1-20	1213.37	1213.96	1199.17	1204.39	-0.0074	

绘制日收盘价曲线图：

```
>> x = xlsread('examp6_1_2.xls');
>> price = x(:,4)';
>> figure;
>> plot(price,'k','LineWidth',2);
>> xlabel('观测序号');
>> ylabel('上海股市日收盘价');
```

盒子法：

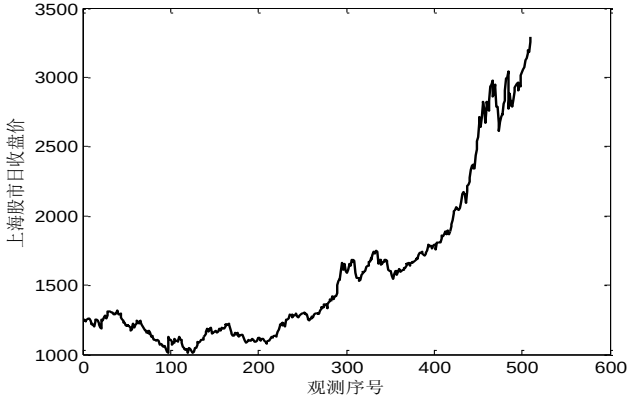
```
>> output1 = smoothts(price,'b',30);
>> output2 = smoothts(price,'b',100);
>> figure;
>> plot(price,'.');
>> hold on
>> plot(output1,'k','LineWidth',2);
>> plot(output2,'k-.','LineWidth',2);
>> xlabel('观测序号'); ylabel('Box method');
>> legend('原始散点','平滑曲线(窗宽30)',...
'平滑曲线(窗宽100)','location','northwest');
```

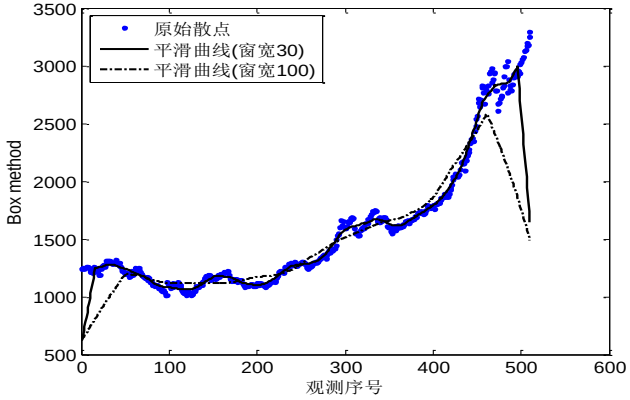
用高斯窗方法平滑数据：

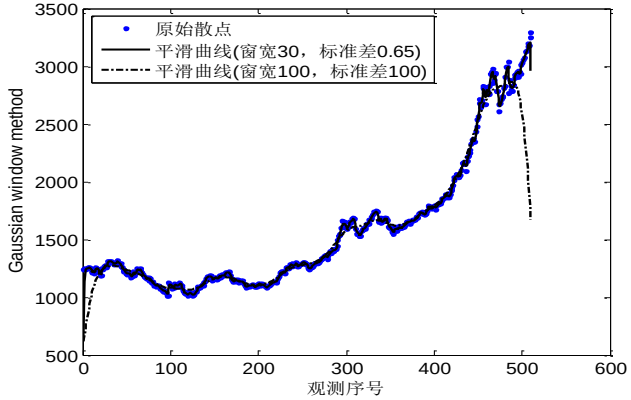
```
>> output3 = smoothts(price,'g',30);
>> output4 = smoothts(price,'g',100,100);
>> figure;
>> plot(price,'.');
>> hold on
>> plot(output3,'k','LineWidth',2);
>> plot(output4,'k-.','LineWidth',2);
>> xlabel('观测序号'); ylabel('Gaussian window method');
>> legend('原始散点','平滑曲线(窗宽30，标准差0.65)',...
        '平滑曲线(窗宽100，标准差100)','location','northwest');
```

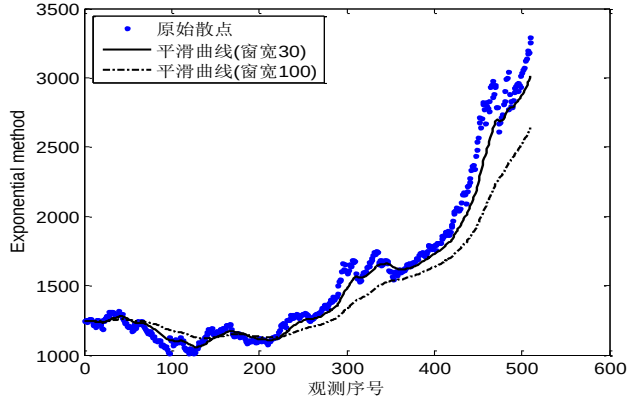
用指数法平滑数据：

```
>> output5 = smoothts(price,'e',30);
>> output6 = smoothts(price,'e',100);
>> figure;
>> plot(price,'.');
>> hold on
>> plot(output5,'k','LineWidth',2);
>> plot(output6,'k-.','LineWidth',2);
>> xlabel('观测序号'); ylabel('Exponential
method');
>> legend('原始散点','平滑曲线(窗宽30)','平滑曲
线(窗宽100)','location','northwest');
```


上海股市日收盘价
观测序号


原始散点
平滑曲线(窗宽30)
平滑曲线(窗宽100)
Box method
观测序号


原始散点
平滑曲线(窗宽30，标准差0.65)
平滑曲线(窗宽100，标准差100)
Gaussian window method
观测序号


原始散点
平滑曲线(窗宽30)
平滑曲线(窗宽100)
Exponential method
观测序号

三、medfilt1函数（一维中值滤波）

调用格式：

y = medfilt1(x,n)

y = medfilt1(x,n,blksz)

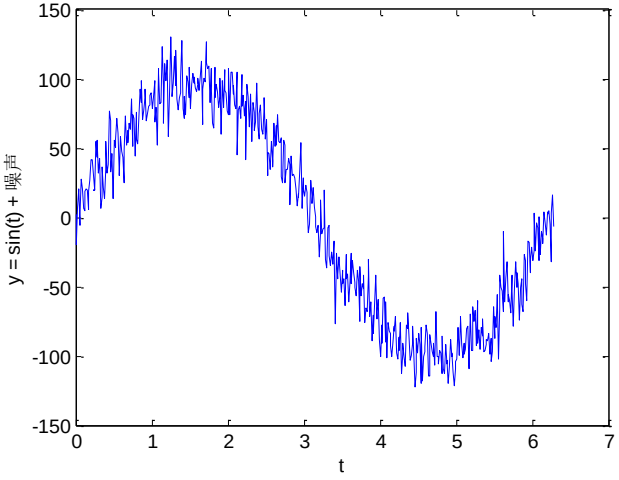
y = medfilt1(x,n,blksz,dim)

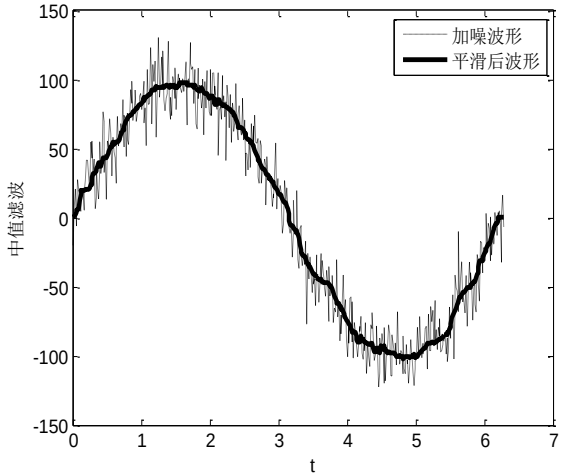
【例6.1-3】产生一列正弦波信号，加入噪声信号，然后调用medfilt1函数对加入噪声的正弦波进行滤波（平滑处理）

```
>> t = linspace(0,2*pi,500)';
>> y = 100*sin(t);
>> noise = normrnd(0,15,500,1);
>> y = y + noise;
>> figure;
>> plot(t,y);
>> xlabel('t');
>> ylabel('y = sin(t) + 噪声');
```

中值滤波：

```
>> yy = medfilt1(y,30);
>> figure;
>> plot(t,y,'k:');
>> hold on
>> plot(t,yy,'k','LineWidth',3);
>> xlabel('t');
>> ylabel('中值滤波');
>> legend('加噪波形','平滑后波形');
```


y = sin(t) + 噪声
t
150
100
50
0
-50
-100
-150
0
1
2
3
4
5
6
7


中值滤波
t
加噪波形
平滑后波形
150
100
50
0
-50
-100
-150
0
1
2
3
4
5
6
7

第二节 数据的标准化变换

一、 标准化变换公式

$$\mathbf{X}=\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \cdots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix}, \qquad \mathbf{X}^{*}=\begin{pmatrix} x_{11}^{*} & x_{12}^{*} & \cdots & x_{1p}^{*} \\ x_{21}^{*} & x_{22}^{*} & \cdots & x_{2p}^{*} \\ \vdots & \vdots & \cdots & \vdots \\ x_{n1}^{*} & x_{n2}^{*} & \cdots & x_{np}^{*} \end{pmatrix},$$

$$x_{ij}^{*}=\frac{x_{ij}-\overline{x}_{j}}{\sqrt{s_{jj}}}, \quad i=1,2,\cdots,n, \quad j=1,2,\cdots,p,$$

$$\overline{x}_{j}=\frac{1}{n}\sum_{i=1}^{n}x_{ij}, \quad \sqrt{s_{jj}}=\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_{ij}-\overline{x}_{j})^{2}}, \quad j=1,2,3,\cdots,p\ .$$

二、标准化变换的MATLAB实现

1. zscore函数

调用格式：

Z = zscore(X)

[Z,mu,sigma] = zscore(X)

[…] = zscore(X,1)

[…] = zscore(X,flag,dim)

【例6.2-1】调用rand函数产生一个随机矩

阵，然后调用zscore函数将其按列标准化

```
>> x = [rand(10,1), 5*rand(10,1), ...
10*rand(10,1), 500*rand(10,1)]
>> [xz,mu,sigma] = zscore(x)
>> std(xz)
```

第三节 数据的极差归一化变换

一、 极差归一化变换公式

$$\mathbf{X}=\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \cdots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix}, \qquad \mathbf{X}^{R}=\begin{pmatrix} x_{11}^{R} & x_{12}^{R} & \cdots & x_{1p}^{R} \\ x_{21}^{R} & x_{22}^{R} & \cdots & x_{2p}^{R} \\ \vdots & \vdots & \cdots & \vdots \\ x_{n1}^{R} & x_{n2}^{R} & \cdots & x_{np}^{R} \end{pmatrix},$$

$$x_{ij}^{R}=\frac{x_{ij}-\min\limits_{1\le k\le n} x_{kj}}{\max\limits_{1\le k\le n} x_{kj}-\min\limits_{1\le k\le n} x_{kj}}, \quad i=1,2,\cdots,n, \quad j=1,2,\cdots,p,$$

二、极差归一化变换的MATLAB实现

1. 自编的rscore函数

调用格式：

R = rscore(X)

[R,xmin,xrange] = rscore(X)

[…] = rscore(X,dim)

【例6.3-1】调用rand函数产生一个随机矩阵，然后调用rscore函数对其按列进行极差归一化变换

```
>> x = [rand(10,1), 5*rand(10,1), ...
10*rand(10,1), 500*rand(10,1)]
>> [R,xmin,xrange] = rscore(x)
```

2. 自带的mapminmax函数

调用格式：

[Y,PS] = mapminmax(X,YMIN,YMAX)

[Y,PS] = mapminmax(X,FP)

Y = mapminmax('apply',X,PS)

X = mapminmax('reverse',Y,PS)

dy_dx = mapminmax('dy_dx',X,Y,PS)

dx_dy = mapminmax('dx_dy',X,Y,PS)

【例6.3-2】调用rand函数产生一个随机矩阵，然后调用mapminmax函数对其按列进行极差归一化变换.

```
>> x = [rand(10,1), 5*rand(10,1), ...
10*rand(10,1), 500*rand(10,1)]
>> FP.ymin = 0;
>> FP.ymax = 1;
>> y = mapminmax(x',FP)'
```

Thank You