SVM神经网络的信息粒化时序回归预测----上证指数开盘指数变化趋势和变化空间预测

该案例作者申明：

1：本人长期驻扎在此板块里，对该案例提问，做到有问必答。本套书籍官方网站为：video.ourmatlab.com

2：点此从当当预定本书：《Matlab神经网络30个案例分析》。

3：此案例有配套的教学视频，视频下载方式video.ourmatlab.com/vbuy.html。

4：此案例为原创案例，转载请注明出处（《Matlab神经网络30个案例分析》）。

5：若此案例碰巧与您的研究有关联，我们欢迎您提意见，要求等，我们考虑后可以加在案例里。

by liyang[faruto] @ faruto's Studio~ Email:faruto@163.com QQ:516667408
http://blog.sina.com.cn/faruto http://www.matlabsky.com http://www.mfun.la
http://video.ourmatlab.com

Contents

- 清空环境变量
- 原始数据的提取
- 对原始数据进行模糊信息粒化
- 利用SVM对Low进行回归预测
- 对于Low的回归预测结果分析
- 利用SVM对R进行回归预测
- 对于R的回归预测结果分析
- 利用SVM对Up进行回归预测
- 对于Up的回归预测结果分析
- 子函数 SVMcgForRegress.m

清空环境变量

```
function chapter15
```

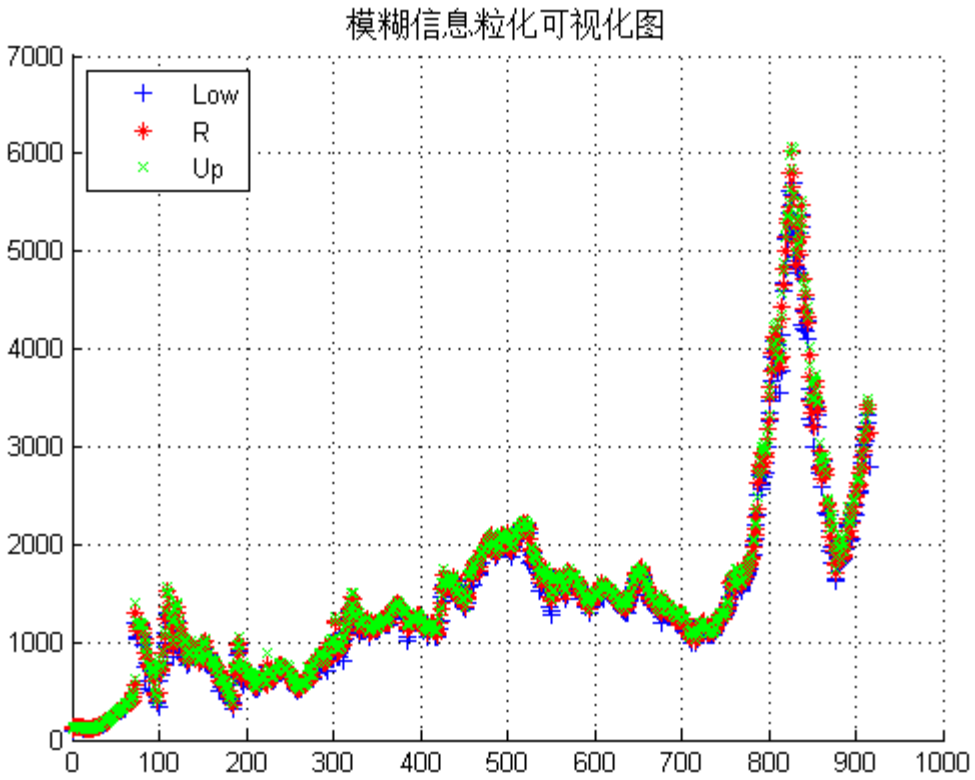
```
tic;
close all;
clear;
clc;
format compact;
```

原始数据的提取

```
% 载入测试数据上证指数(1990.12.19-2009.08.19)
% 数据是一个4579*6的double型的矩阵,每一行表示每一天的上证指数
% 6列分别表示当天上证指数的开盘指数,指数最高值,指数最低值,收盘指数,当日交易量,当日交易额.
load chapter15_sh.mat;

% 提取数据
ts = sh_open;
time = length(ts);

% 画出原始上证指数的每日开盘数
figure;
plot(ts,'LineWidth',2);
title('上证指数的每日开盘数(1990.12.20-2009.08.19)','FontSize',12);
grid on;
snapnow;
```

对原始数据进行模糊信息粒化

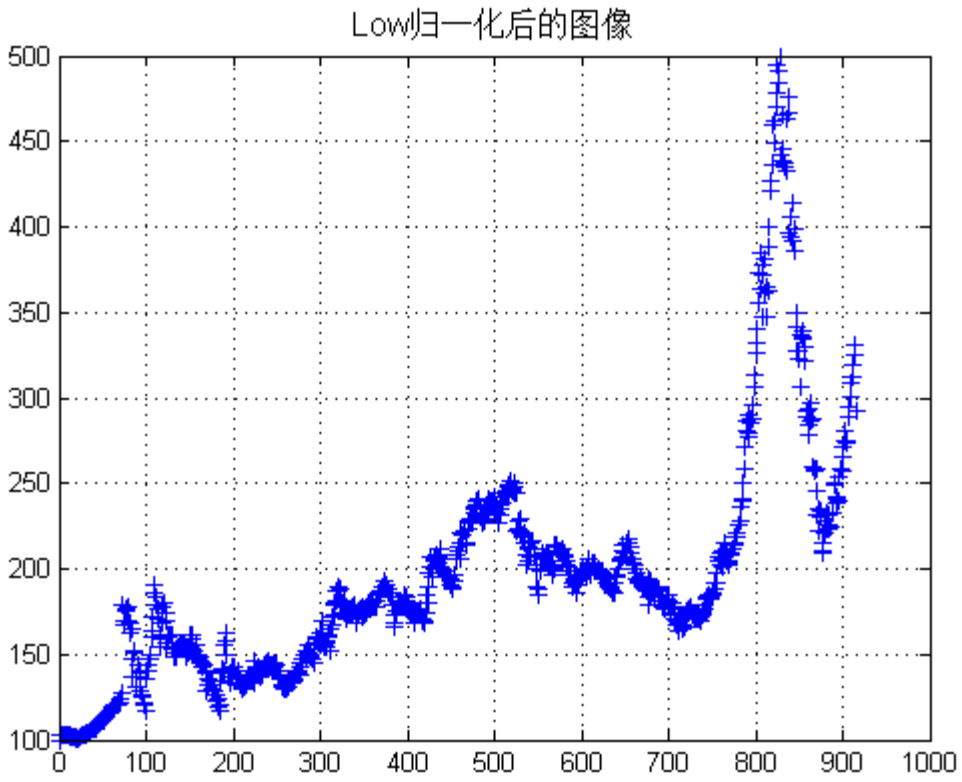
```
win_num = floor(time/5);
tsx = 1:win_num;
tsx = tsx';
[Low,R,Up]=FIG_D(ts','triangle',win_num);

% 模糊信息粒化可视化图
figure;
hold on;
plot(Low,'b+');
plot(R,'r*');
plot(Up,'gx');
hold off;
legend('Low','R','Up',2);
title('模糊信息粒化可视化图','FontSize',12);
grid on;
snapnow;
```



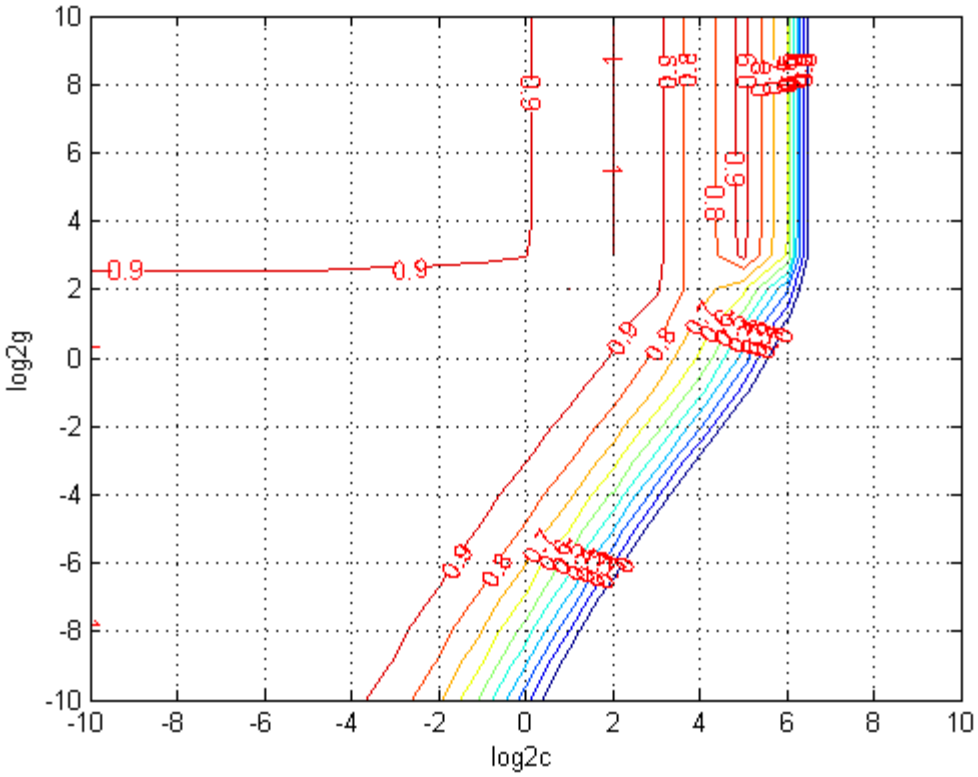
利用SVM对Low进行回归预测

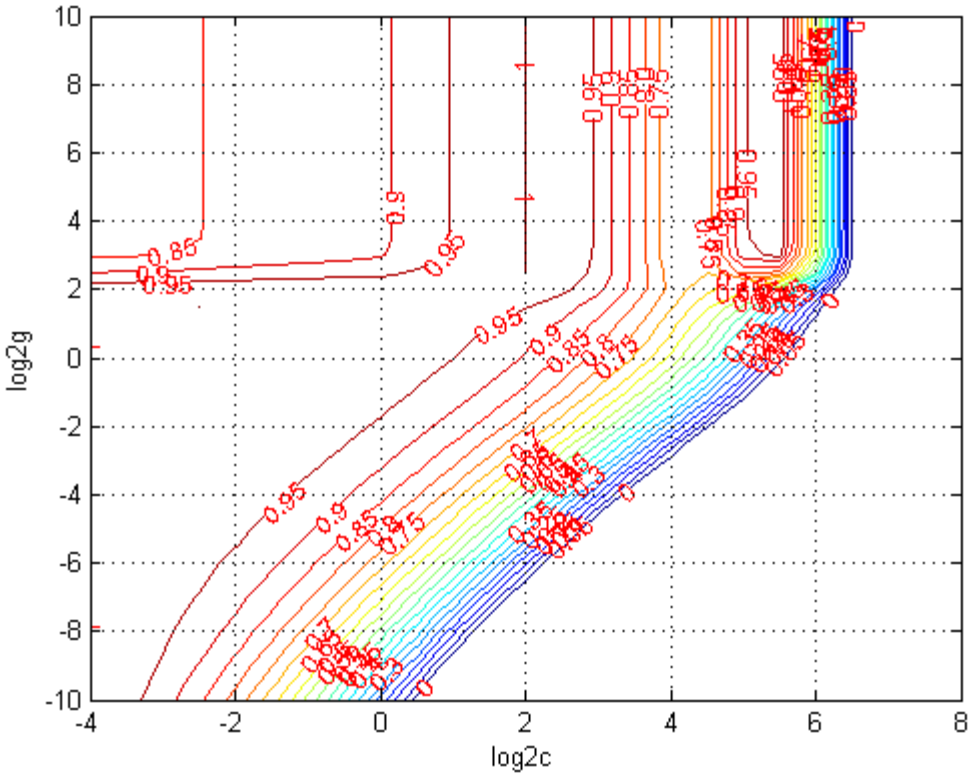
```
% 数据预处理,将Low进行归一化处理
% mapminmax为matlab自带的映射函数
[low,low_ps] = mapminmax(Low);
low_ps.ymin = 100;
low_ps.ymax = 500;
% 对Low进行归一化
[low,low_ps] = mapminmax(Low,low_ps);
% 画出Low归一化后的图像
figure;
plot(low,'b+');
title('Low归一化后的图像','FontSize',12);
grid on;
% 对low进行转置,以符合libsvm工具箱的数据格式要求
low = low';
snapnow;

% 选择回归预测分析中最佳的SVM参数c&g
% 首先进行粗略选择
[bestmse,bestc,bestg] = SVMcgForRegress(low,tsx,-10,10,-10,10,3,1,1,0.1);

% 打印粗略选择结果
disp('打印粗略选择结果');
str = sprintf( 'SVM parameters for Low:Best Cross Validation MSE = %g Best c = %g Best g =
%g',bestmse,bestc,bestg);
disp(str);

% 根据粗略选择的结果图再进行精细选择
[bestmse,bestc,bestg] = SVMcgForRegress(low,tsx,-4,8,-10,10,3,0.5,0.5,0.05);

% 打印精细选择结果
disp('打印精细选择结果');
str = sprintf( 'SVM parameters for Low:Best Cross Validation MSE = %g Best c = %g Best g =
%g',bestmse,bestc,bestg);
disp(str);

% 训练SVM
cmd = ['-c ', num2str(bestc), ' -g ', num2str(bestg) , ' -s 3 -p 0.1'];
low_model = svmtrain(low, tsx, cmd);

% 预测
[low predict,low mse] = svmpredict(low,tsx,low model);
```

```
low_predict = mapminmax('reverse',low_predict,low_ps);
predict_low = svmpredict(1,win_num+1,low_model);
predict_low = mapminmax('reverse',predict_low,low_ps);
predict_low
```




```
打印粗略选择结果
SVM parameters for Low:Best Cross Validation MSE = 35.0879 Best c = 256 Best g = 0.03125
打印精细选择结果
SVM parameters for Low:Best Cross Validation MSE = 35.0177 Best c = 256 Best g = 0.0220971
Mean squared error = 22.0054 (regression)
Squared correlation coefficient = 0.995366 (regression)
Mean squared error = 85135.8 (regression)
Squared correlation coefficient = -1.#IND (regression)
predict_low =
   2.7968e+003
```

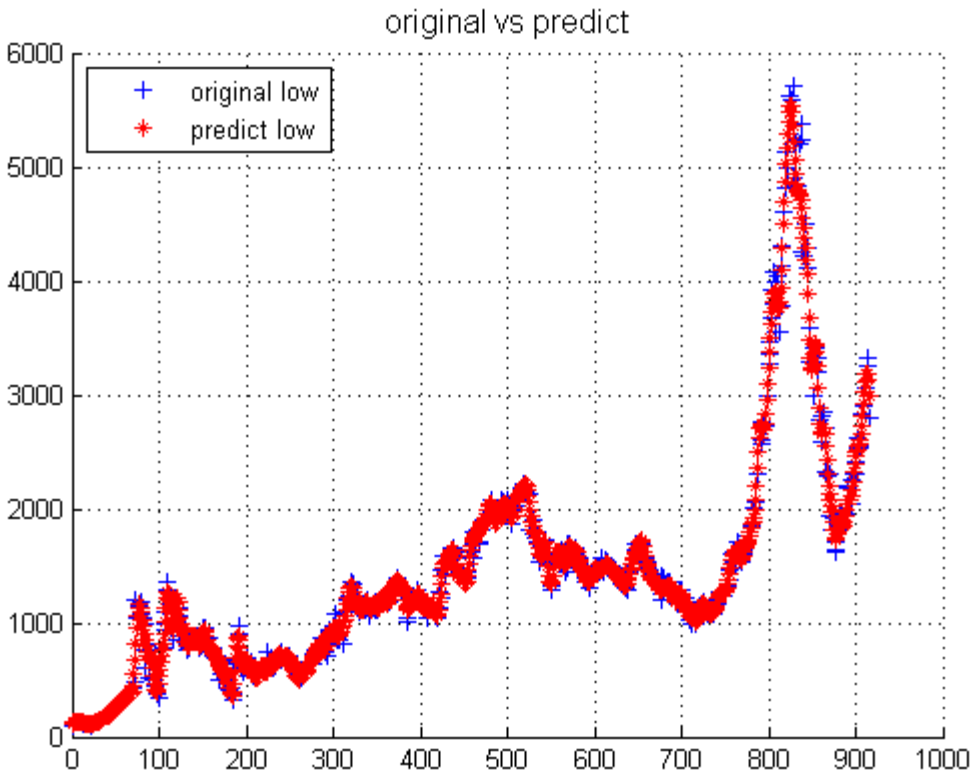


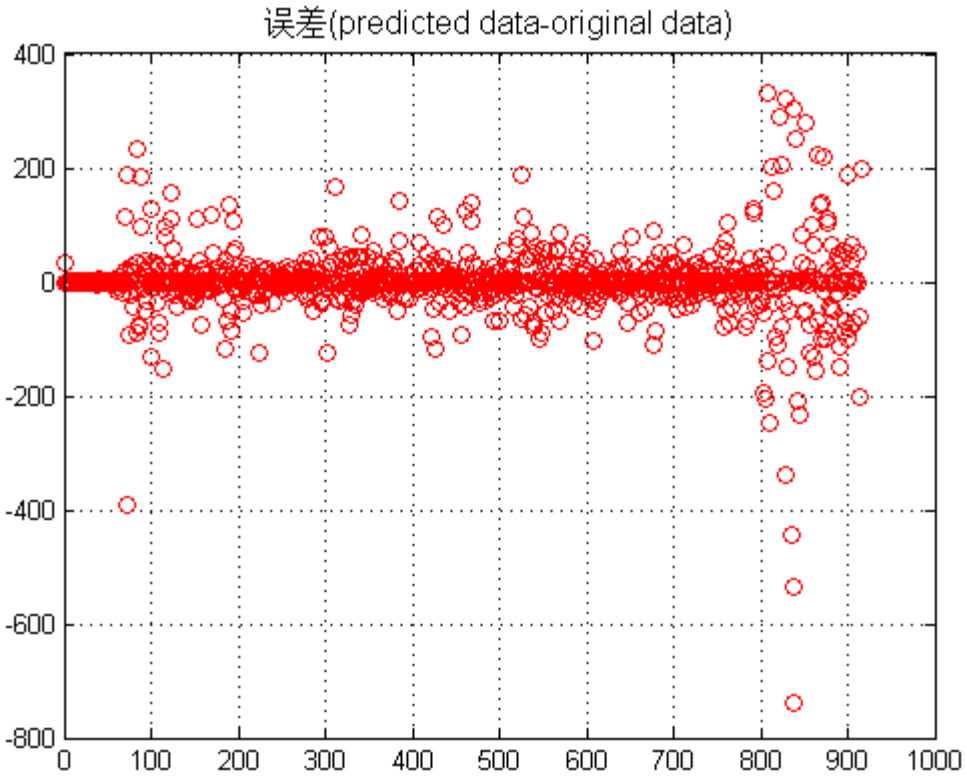


对于**Low**的回归预测结果分析

```
figure;
hold on;
plot(Low,'b+');
plot(low_predict,'r*');
legend('original low','predict low',2);
```

```
title('original vs predict','FontSize',12);
grid on;
figure;
error = low_predict - Low';
plot(error,'ro');
title('误差(predicted data-original data)','FontSize',12);
grid on;
snapnow;
```

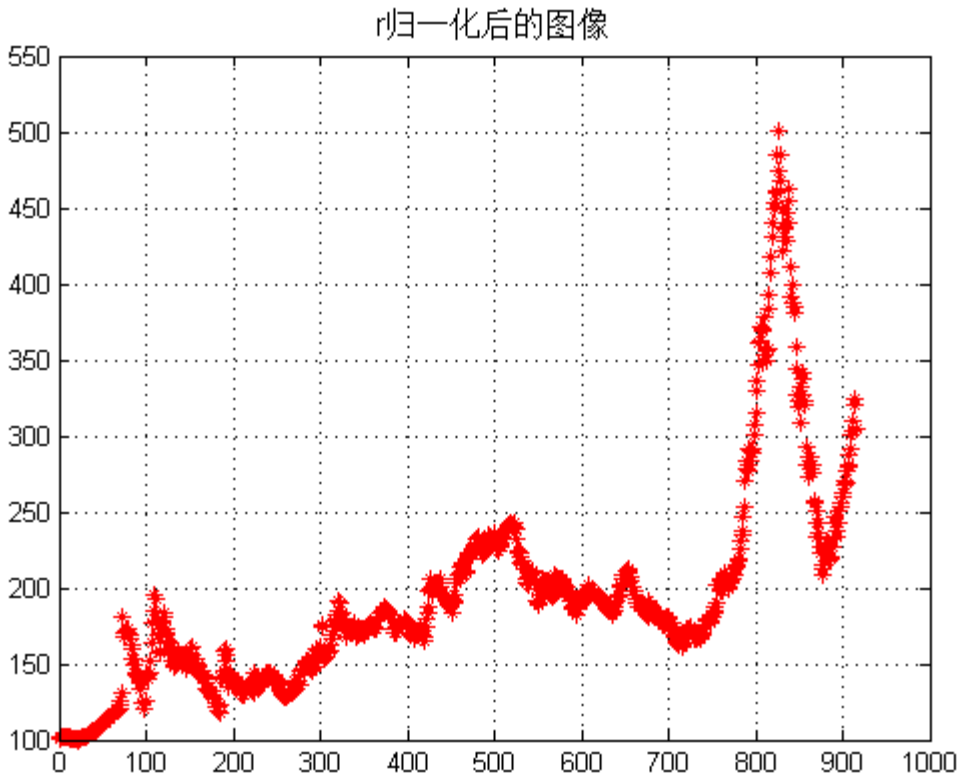




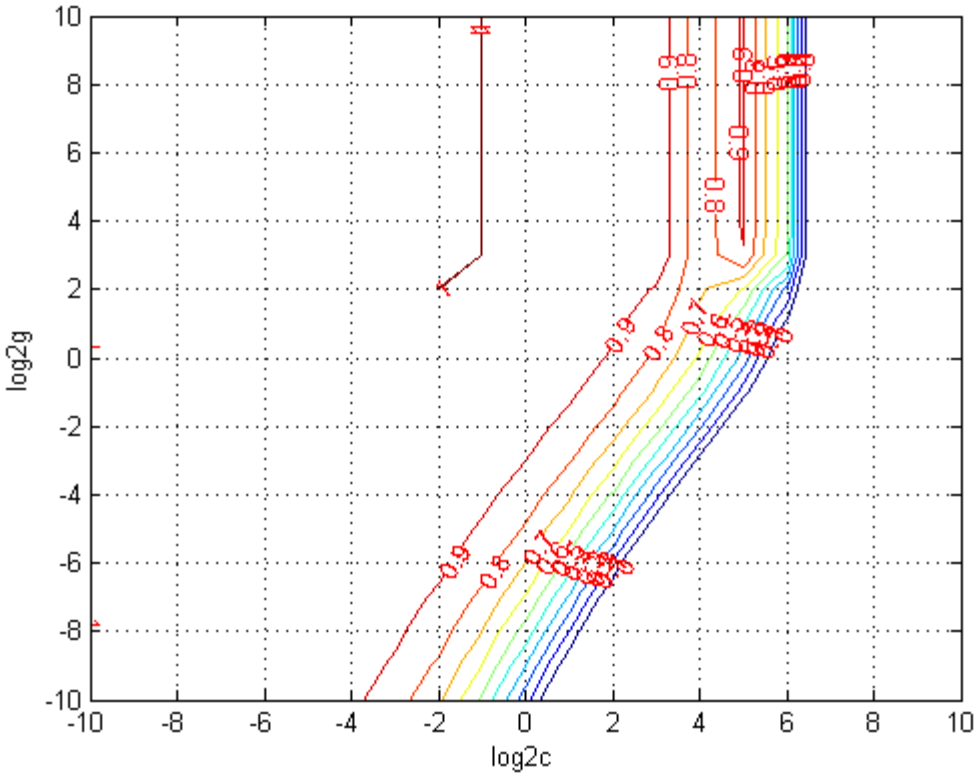
利用SVM对R进行回归预测

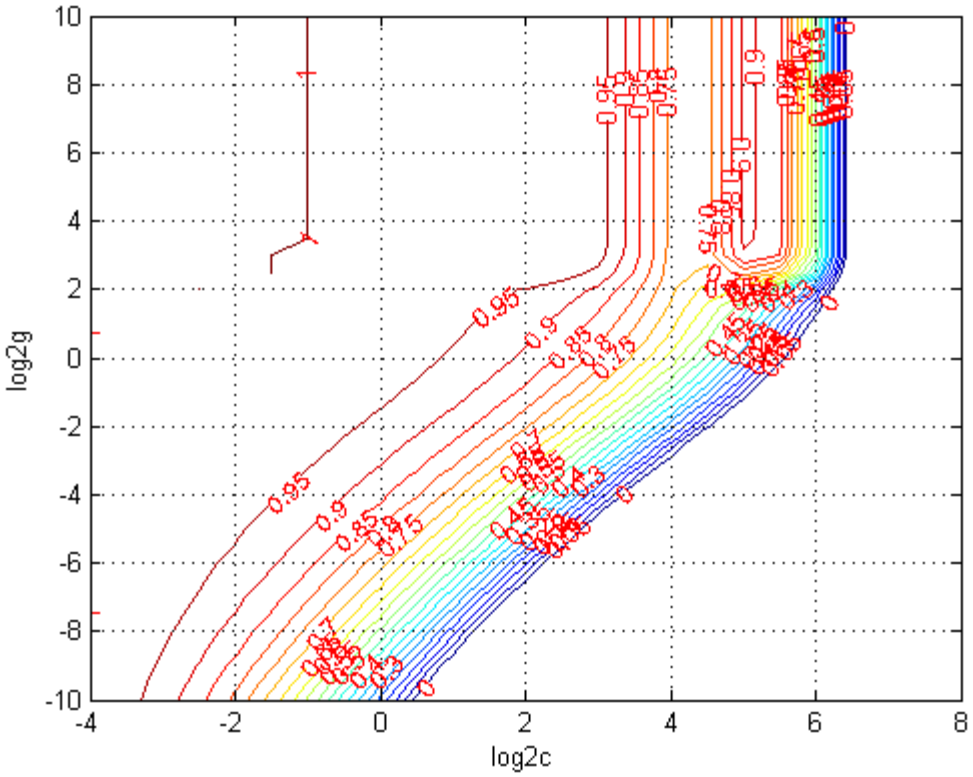
```
% 数据预处理,将R进行归一化处理
% mapminmax为matlab自带的映射函数
[r,r_ps] = mapminmax(R);
r_ps.ymin = 100;
r_ps.ymax = 500;
% 对R进行归一化
[r,r_ps] = mapminmax(R,r_ps);
% 画出R归一化后的图像
figure;
plot(r,'r*');
title('r归一化后的图像','FontSize',12);
grid on;
% 对R进行转置,以符合libsvm工具箱的数据格式要求
r = r';
snapnow;

% 选择回归预测分析中最佳的SVM参数c&g
% 首先进行粗略选择
[bestmse,bestc,bestg] = SVMcgForRegress(r,tsx,-10,10,-10,10,3,1,1,0.1);

% 打印粗略选择结果
disp('打印粗略选择结果');
str = sprintf( 'SVM parameters for R:Best Cross Validation MSE = %g Best c = %g Best g =
%g',bestmse,bestc,bestg);
disp(str);

% 根据粗略选择的结果图再进行精细选择
[bestmse,bestc,bestg] = SVMcgForRegress(r,tsx,-4,8,-10,10,3,0.5,0.5,0.05);

% 打印精细选择结果
disp('打印精细选择结果');
str = sprintf( 'SVM parameters for R:Best Cross Validation MSE = %g Best c = %g Best g =
%g',bestmse,bestc,bestg);
disp(str);

% 训练SVM
cmd = ['-c ', num2str(bestc), ' -g ', num2str(bestg) , ' -s 3 -p 0.1'];
r_model = svmtrain(r, tsx, cmd);

% 预测
[r predict,r mse] = svmpredict(r,tsx,low model);
```

```
r_predict = mapminmax('reverse',r_predict,r_ps);
predict_r = svmpredict(1,win_num+1,r_model);
predict_r = mapminmax('reverse',predict_r,r_ps);
predict_r
```




```
打印粗略选择结果
SVM parameters for R:Best Cross Validation MSE = 22.7823 Best c = 256 Best g = 0.03125
打印精细选择结果
SVM parameters for R:Best Cross Validation MSE = 22.7823 Best c = 256 Best g = 0.03125
Mean squared error = 26.2007 (regression)
Squared correlation coefficient = 0.995898 (regression)
Mean squared error = 84653.7 (regression)
Squared correlation coefficient = -1.#IND (regression)
predict_r =
   2.9500e+003
```

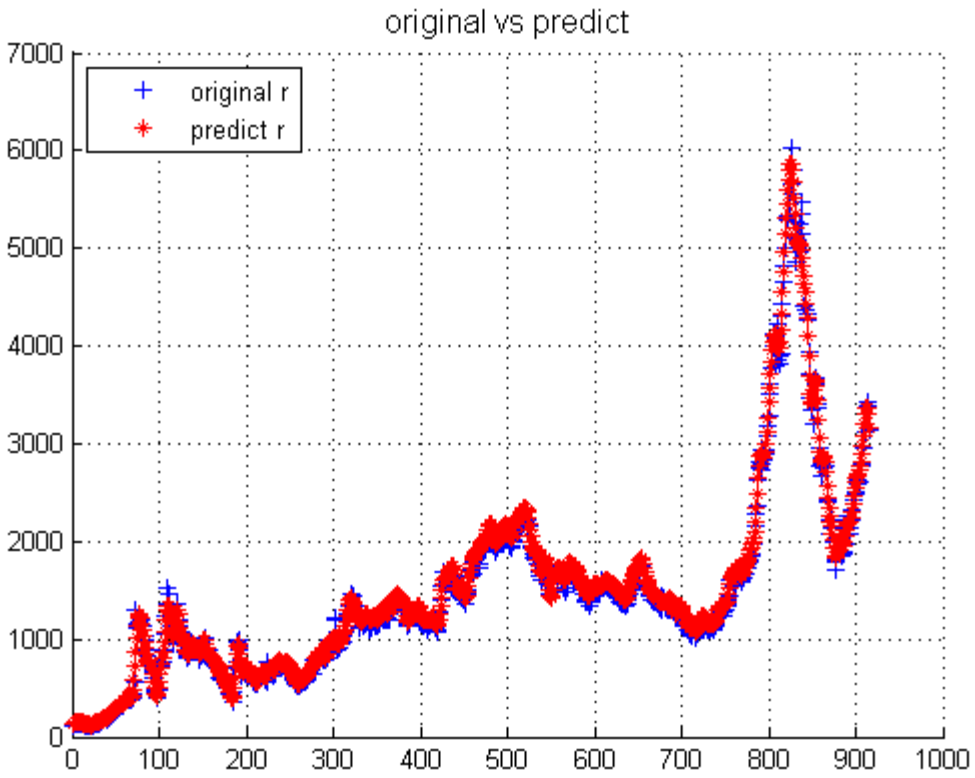


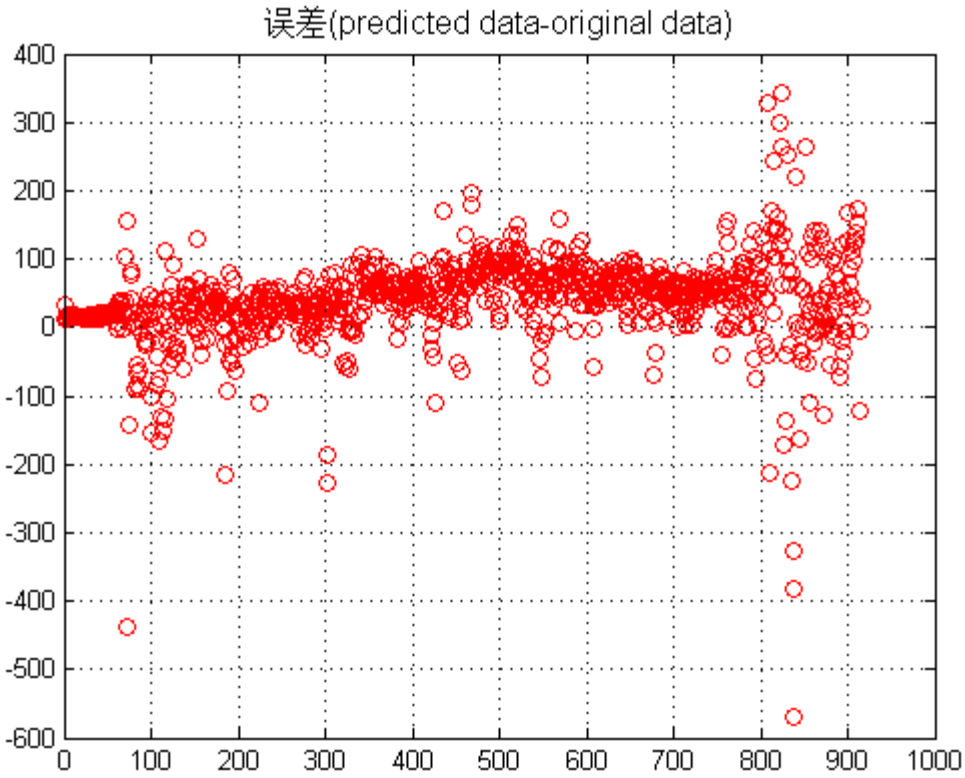


对于R的回归预测结果分析

```
figure;
hold on;
plot(R,'b+');
plot(r_predict,'r*');
legend('original r','predict r',2);
```

```
title('original vs predict','FontSize',12);
grid on;
figure;
error = r_predict - R';
plot(error,'ro');
title('误差(predicted data-original data)','FontSize',12);
grid on;
snapnow;
```

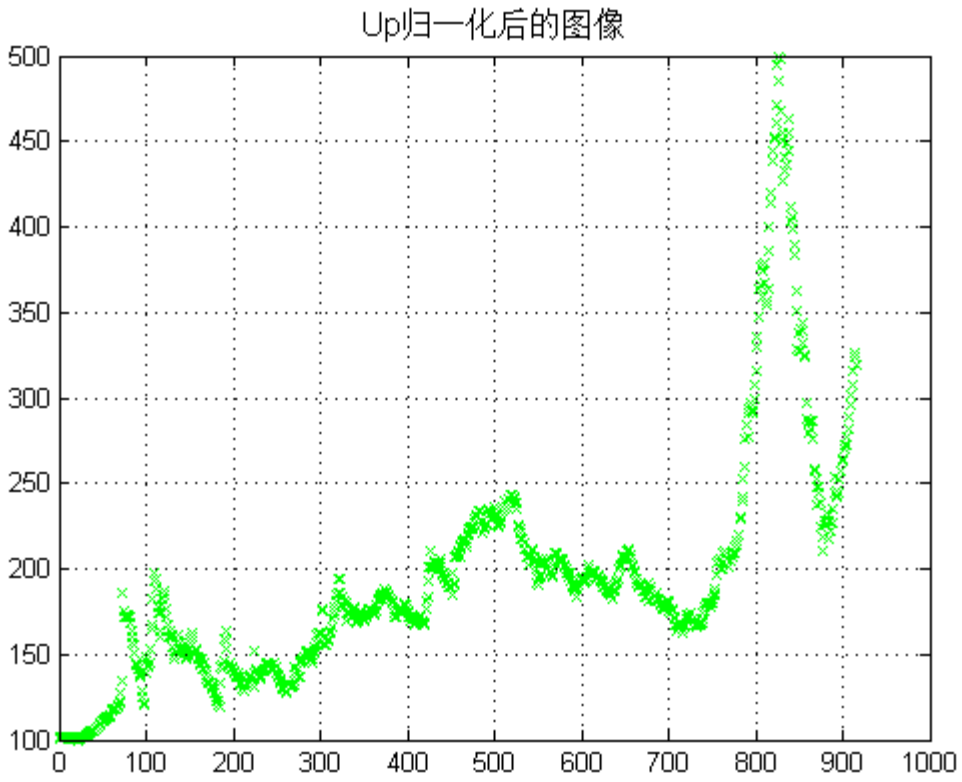




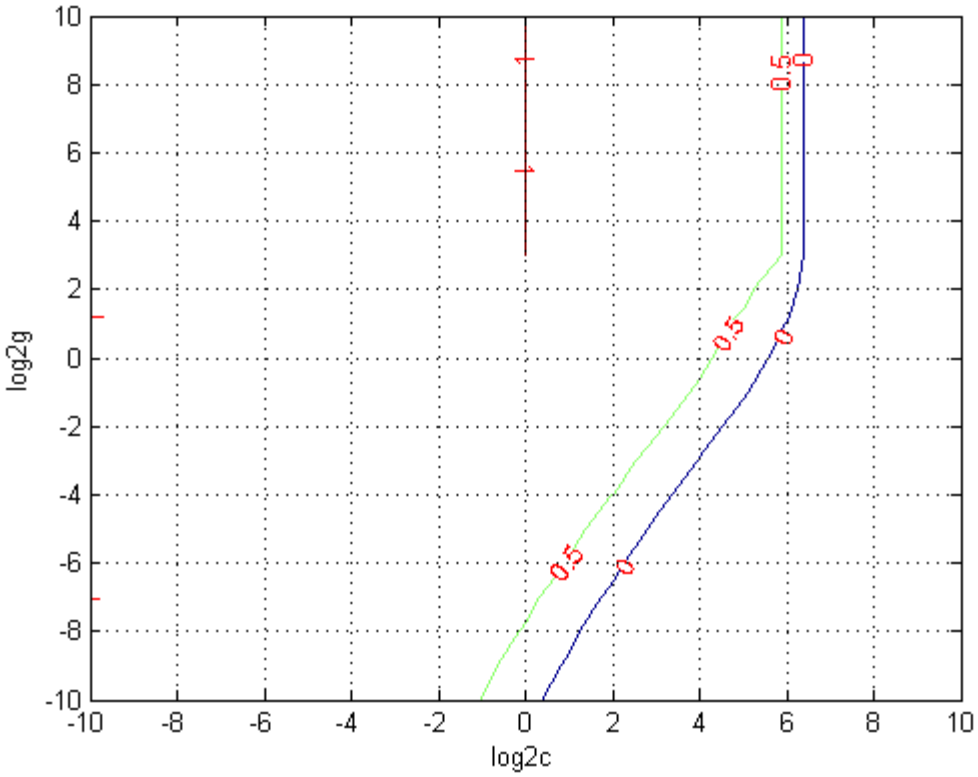
利用SVM对Up进行回归预测

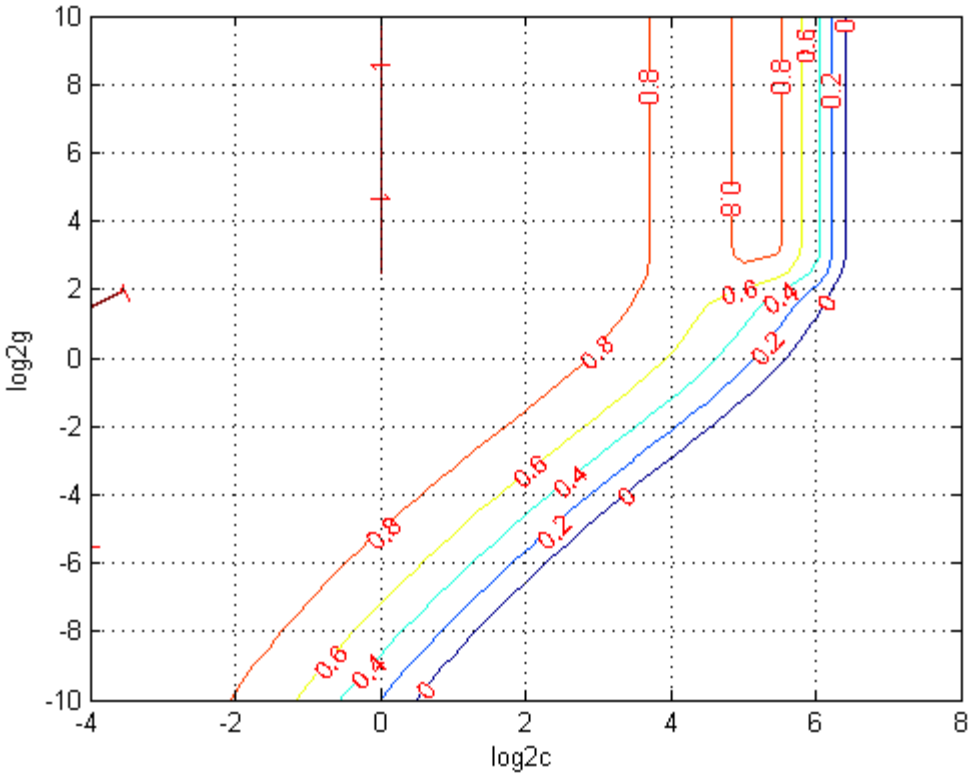
```
% 数据预处理,将up进行归一化处理
% mapminmax为matlab自带的映射函数
[up,up_ps] = mapminmax(Up);
up_ps.ymin = 100;
up_ps.ymax = 500;
% 对Up进行归一化
[up,up_ps] = mapminmax(Up,up_ps);
% 画出Up归一化后的图像
figure;
plot(up,'gx');
title('Up归一化后的图像','FontSize',12);
grid on;
% 对up进行转置,以符合libsvm工具箱的数据格式要求
up = up';
snapnow;

% 选择回归预测分析中最佳的SVM参数c&g
% 首先进行粗略选择
[bestmse,bestc,bestg] = SVMcgForRegress(up,tsx,-10,10,-10,10,3,1,1,0.5);

% 打印粗略选择结果
disp('打印粗略选择结果');
str = sprintf( 'SVM parameters for Up:Best Cross Validation MSE = %g Best c = %g Best g =
%g',bestmse,bestc,bestg);
disp(str);

% 根据粗略选择的结果图再进行精细选择
[bestmse,bestc,bestg] = SVMcgForRegress(up,tsx,-4,8,-10,10,3,0.5,0.5,0.2);

% 打印精细选择结果
disp('打印精细选择结果');
str = sprintf( 'SVM parameters for Up:Best Cross Validation MSE = %g Best c = %g Best g =
%g',bestmse,bestc,bestg);
disp(str);

% 训练SVM
cmd = ['-c ', num2str(bestc), ' -g ', num2str(bestg) , ' -s 3 -p 0.1'];
up_model = svmtrain(up, tsx, cmd);

% 预测
[up predict,up mse] = svmpredict(up,tsx,up model);
```

```
up_predict = mapminmax('reverse',up_predict,up_ps);
predict_up = svmpredict(1,win_num+1,up_model);
predict_up = mapminmax('reverse',predict_up,up_ps);
predict_up
```




```
打印粗略选择结果
SVM parameters for Up:Best Cross Validation MSE = 23.8758 Best c = 512 Best g = 0.0625
打印精细选择结果
SVM parameters for Up:Best Cross Validation MSE = 23.895 Best c = 256 Best g = 0.0220971
Mean squared error = 11.108 (regression)
Squared correlation coefficient = 0.997625 (regression)
Mean squared error = 96798.9 (regression)
Squared correlation coefficient = -1.#IND (regression)
predict_up =
   3.2673e+003
```



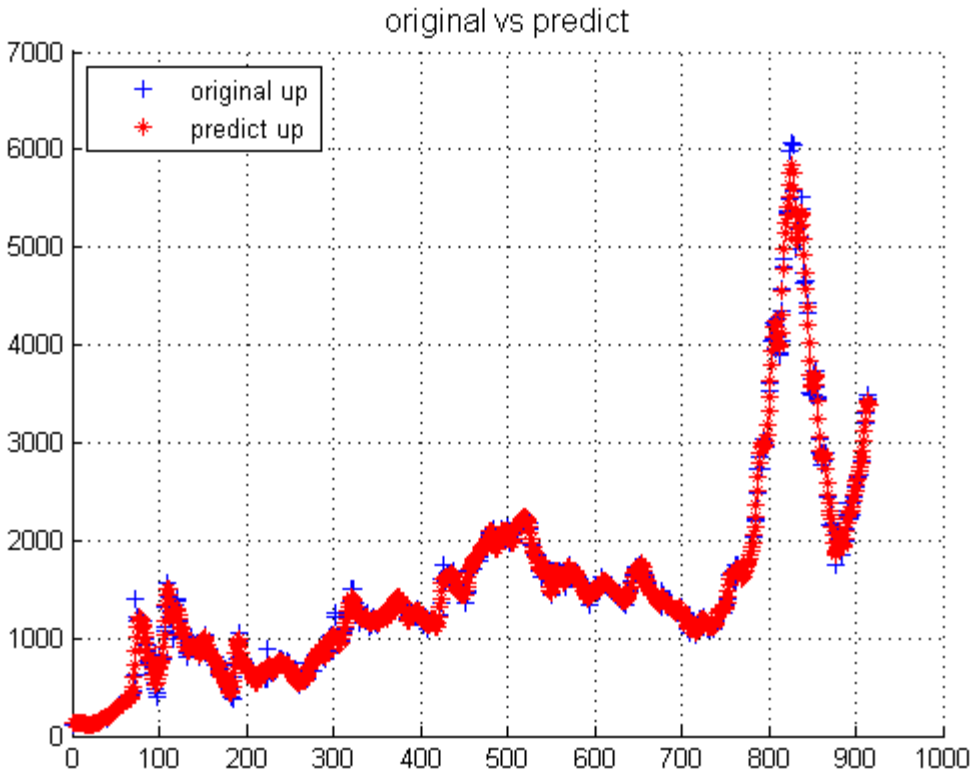


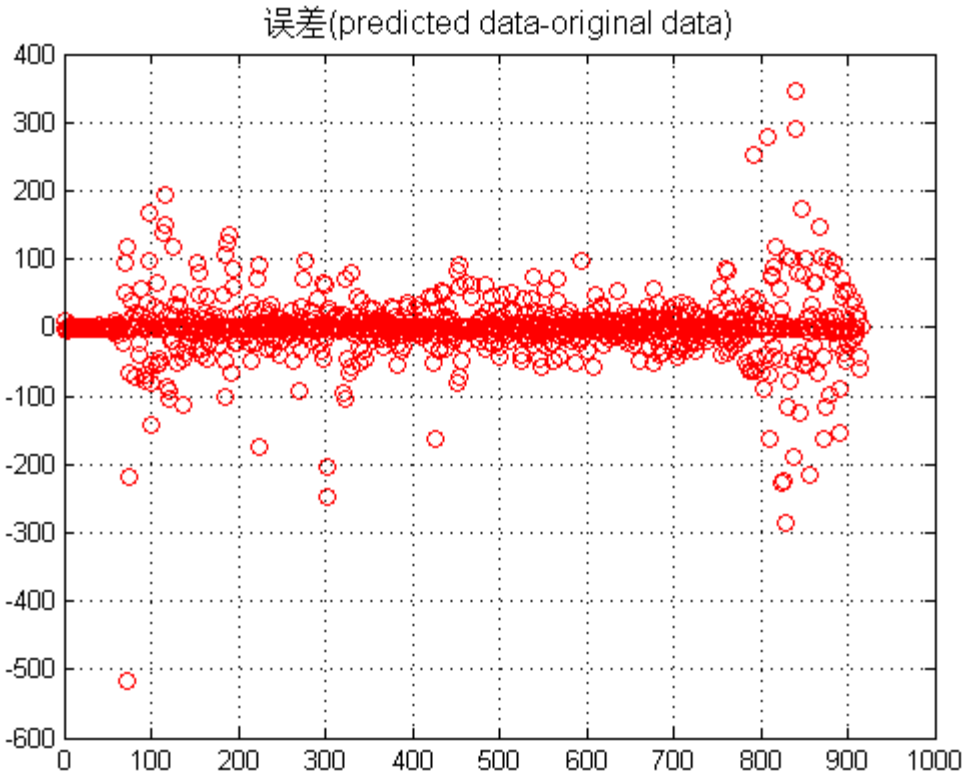
对于Up的回归预测结果分析

```
figure;
hold on;
plot(Up,'b+');
plot(up_predict,'r*');
legend('original up','predict up',2);
```

```
title('original vs predict','FontSize',12);
grid on;
figure;
error = up_predict - Up';
plot(error,'ro');
title('误差(predicted data-original data)','FontSize',12);
grid on;
toc;
snapnow;
% web http://www.matlabsky.com/forum-31-1.html
web http://www.matlabsky.com/forum-31-1.html -new;
```

```
Elapsed time is 2725.383778 seconds.
```





子函数 **SVMcgForRegress.m**

```
function [mse,bestc,bestg] =
SVMcgForRegress(train_label,train,cmin,cmax,gmin,gmax,v,cstep,gstep,msestep)
% SVMcgForClass
% 输入:
% train_label:训练集标签.要求与libsvm工具箱中要求一致.
% train:训练集.要求与libsvm工具箱中要求一致.
% cmin:惩罚参数c的变化范围的最小值(取以2为底的对数后),即 c_min = 2^(cmin).默认为 -5
% cmax:惩罚参数c的变化范围的最大值(取以2为底的对数后),即 c_max = 2^(cmax).默认为 5
% gmin:参数g的变化范围的最小值(取以2为底的对数后),即 g_min = 2^(gmin).默认为 -5
% gmax:参数g的变化范围的最小值(取以2为底的对数后),即 g_min = 2^(gmax).默认为 5
% v:cross validation的参数,即给测试集分为几部分进行cross validation.默认为 3
% cstep:参数c步进的大小.默认为 1
% gstep:参数g步进的大小.默认为 1
% msestep:最后显示MSE图时的步进大小.默认为 20
% 输出:
% bestacc:Cross Validation 过程中的最高分类准确率
% bestc:最佳的参数c
% bestg:最佳的参数g

% about the parameters of SVMcgForRegress
if nargin < 10
    msestep = 0.1;
end
if nargin < 7
    msestep = 0.1;
    v = 3;
    cstep = 1;
    gstep = 1;
end
if nargin < 6
    msestep = 0.1;
    v = 3;
    cstep = 1;
    gstep = 1;
    gmax = 5;
end
if nargin < 5
    msestep = 0.1;
    v = 3;
    cstep = 1;
    gstep = 1;
```

```
    gmax = 5;
    gmin = -5;
end
if nargin < 4
    msestep = 0.1;
    v = 3;
    cstep = 1;
    gstep = 1;
    gmax = 5;
    gmin = -5;
    cmax = 5;
end
if nargin < 3
    msestep = 0.1;
    v = 3;
    cstep = 1;
    gstep = 1;
    gmax = 5;
    gmin = -5;
    cmax = 5;
    cmin = -5;
end
% X:c Y:g cg:mse
[X,Y] = meshgrid(cmin:cstep:cmax,gmin:gstep:gmax);
[m,n] = size(X);
cg = zeros(m,n);
% record accuracy with different c & g,and find the best mse with the smallest c
bestc = 0;
bestg = 0;
mse = 10^10;
basenum = 2;
for i = 1:m
    for j = 1:n
        cmd = ['-v ',num2str(v),' -c ',num2str( basenum^X(i,j) ),' -g ',num2str(
basenum^Y(i,j) ),' -s 3'];
        cg(i,j) = svmtrain(train_label, train, cmd);

        if cg(i,j) < mse
            mse = cg(i,j);
            bestc = basenum^X(i,j);
            bestg = basenum^Y(i,j);
        end
        if ( cg(i,j) == mse && bestc > basenum^X(i,j) )
            mse = cg(i,j);
            bestc = basenum^X(i,j);
            bestg = basenum^Y(i,j);
        end

    end
end

[cg,ps] = mapminmax(cg);
% draw the accuracy with different c & g
figure;
[C,h] = contour(X,Y,cg,0:msestep:1);
clabel(C,h,'FontSize',10,'Color','r');
xlabel('log2c','FontSize',10);
ylabel('log2g','FontSize',10);
grid on;
```

Matlab神经网络30个案例分析

相关论坛：

《Matlab神经网络30个案例分析》官方网站：video.ourmatlab.com

Matlab技术论坛：www.matlabsky.com

Matlab函数百科：www.mfun.la

Matlab中文论坛：www.ilovematlab.com

Published with MATLAB® 7.11