**Source code of SVM.**

%%SVM

%% clear

close all;

clear;

clc;

format compact;

load('C:\Users\Administrator\Desktop\SVM.mat')

%Two matrices synthesize a matrix

data(1:104,:)=unnamed(1:104,:);

%input data

wine=data(:,1:2);

wine\_labels=data(:,3);

% training set and test set

% training set

train\_wine = [wine(1:30,:);wine(55:85,:)];

% Label of training set

train\_wine\_labels = [wine\_labels(1:30);wine\_labels(55:85)];

% test set

test\_wine = [wine(31:54,:);wine(86:104,:)];

% Label of test set

test\_wine\_labels = [wine\_labels(31:54);wine\_labels(86:104)];

%% Data preprocessing

% normalization

[mtrain,ntrain] = size(train\_wine);

[mtest,ntest] = size(test\_wine);

dataset = [train\_wine;test\_wine];

[dataset\_scale,ps] = mapminmax(dataset',0,1);

dataset\_scale = dataset\_scale';

train\_wine = dataset\_scale(1:mtrain,:);

test\_wine = dataset\_scale( (mtrain+1):(mtrain+mtest),: );

{

ga\_option.maxgen = 200;

ga\_option.sizepop = 20;

ga\_option.cbound = [0,100];

ga\_option.gbound = [0,100];

ga\_option.v = 5;

ga\_option.ggap = 0.9;

[bestacc,bestc,bestg] = gaSVMcgForClass(train\_wine\_labels,train\_wine,ga\_option);

disp('Print selection results');

str = sprintf( 'Best Cross Validation Accuracy = %g%% Best c = %g Best g = %g',bestacc,bestc,bestg);

disp(str);

cmd = ['-c ',num2str(bestc),' -g ',num2str(bestg),'-t',1];

model = svmtrain(train\_wine\_labels,train\_wine,cmd);

%}

%{

[bestacc,bestc,bestg] = psoSVMcgForClass(train\_wine\_labels,train\_wine);

disp('Print selection results');

str = sprintf( 'Best Cross Validation Accuracy = %g%% Best c = %g Best g = %g',bestacc,bestc,bestg);

disp(str);

cmd = ['-c ',num2str(bestc),' -g ',num2str(bestg),'-t',1];

model = svmtrain(train\_wine\_labels,train\_wine,cmd);

%}

model = svmtrain(train\_wine\_labels, train\_wine, '-c 22.2201 -g 0.82378 -t 1');

[predict\_label, accuracy\_test, decision\_values] = svmpredict(test\_wine\_labels, test\_wine, model);

[predict\_label\_train, accuracy\_train, decision\_values\_train] = svmpredict(train\_wine\_labels, train\_wine, model);

total\_0 = length(find(data(:,3) == 0));

total\_1 = length(find(data(:,3) == 1));

count\_0 = length(find(train\_wine\_labels == 0));

count\_1 = length(find(train\_wine\_labels == 1));

number\_0 = length(find(test\_wine\_labels == 0));

number\_1 = length(find(test\_wine\_labels == 1));

number\_0\_0\_train = length(find(predict\_label\_train == 0 & train\_wine\_labels == 0));

number\_1\_1\_train = length(find(predict\_label\_train == 1 & train\_wine\_labels == 1));

number\_0\_1\_train = length(find(predict\_label\_train == 1 & train\_wine\_labels == 0));

number\_1\_0\_train = length(find(predict\_label\_train == 0 & train\_wine\_labels == 1));

number\_0\_0\_sim = length(find(predict\_label == 0 & test\_wine\_labels == 0));

number\_1\_1\_sim = length(find(predict\_label == 1 & test\_wine\_labels == 1));

number\_0\_1\_sim = length(find(predict\_label == 1 & test\_wine\_labels == 0));

number\_1\_0\_sim = length(find(predict\_label == 0 & test\_wine\_labels == 1));

disp(['total：' num2str(104)...

' 1：' num2str(total\_0)...

' 2：' num2str(total\_1)]);

disp(['total training set：' num2str(61)...

' 1：' num2str(count\_0)...

' 2：' num2str(count\_1)]);

disp(['total test set：' num2str(43)...

' 1：' num2str(number\_0)...

' 2：' num2str(number\_1)]);

disp(['test set：']);

disp(['1：' num2str(number\_0\_0\_sim)...

' error：' num2str(number\_0 - number\_0\_0\_sim)...

' correct p1=' num2str(number\_0\_0\_sim/number\_0\*100) '%']);

disp(['2：' num2str(number\_1\_1\_sim)...

' error：' num2str(number\_1 - number\_1\_1\_sim)...

' correct p3=' num2str(number\_1\_1\_sim/number\_1\*100) '%']);

disp(['total test set：' num2str(43)...

' correct p=' num2str((number\_0\_0\_sim+number\_1\_1\_sim)/(number\_0+number\_1)\*100) '%']);

disp([' 0 to 1：' num2str(number\_0\_1\_sim)]);

disp([' 1 to 0：' num2str(number\_1\_0\_sim)]);

disp(['training set：']);

disp(['1：' num2str(number\_0\_0\_train)...

' error：' num2str(count\_0 - number\_0\_0\_train)...

' correct p1=' num2str(number\_0\_0\_train/count\_0\*100) '%']);

disp(['2：' num2str(number\_1\_1\_train)...

' error：' num2str(count\_1 - number\_1\_1\_train)...

' correct p3=' num2str(number\_1\_1\_train/count\_1\*100) '%']);

disp(['total training set：' num2str(61)...

' correct p=' num2str((number\_0\_0\_train+number\_1\_1\_train)/(count\_0+count\_1)\*100) '%']);

disp([' 0 to 1：' num2str(number\_0\_1\_train)]);

disp([' 1 to 0：' num2str(number\_1\_0\_train)]);

%% results analysis

figure;

hold on;

plot(test\_wine\_labels,'ko');

plot(predict\_label,'r\*');

set(gca,'ytick',1:1:3);

grid on;

xlabel('Test set samples','fontweight','bold','fontsize',12);

ylabel('Category label','fontweight','bold','fontsize',12);

legend('The actual test set classification','Predicted test set classification');

title('Test set of the actual classification and prediction classification figure','fontweight','bold','FontSize',14);

set(gcf,'color','white');

% gtext('Accuracy = 94.7%(378/399)','fontweight','bold','fontsize',10);

% I=getframe(gcf);

% imwrite(I.cdata,'SVM classification result.tif','tiff','Resolution',600);

%%

% leave-one-out cross validation

disp([' leave-one-out cross validation:']);

dataNew=data(:,1:end-1);

target=data(:,3);

indices=(1:1:104)';

total\_index=zeros(104,1);

result\_classificition=zeros(104,1);

for k=1:104

test=(indices==k);

train=~test;

train\_data=dataNew(train,:);

train\_target=target(train,:);

test\_data=dataNew(test,:);

test\_target=target(test,:);

[mtrain,ntrain] = size(train\_data);

[mtest,ntest] = size(test\_data);

dataset = [train\_data;test\_data];

[dataset\_scale,ps] = mapminmax(dataset',0,1);

dataset\_scale = dataset\_scale';

train\_data = dataset\_scale(1:mtrain,:);

test\_data = dataset\_scale( (mtrain+1):(mtrain+mtest),: );

model = svmtrain(train\_target, train\_data, '-c 22.2201 -g 0.82378 -t 1');

[predict\_label, accuracy, decision\_values\_coo] = svmpredict(test\_target, test\_data, model);

total\_index(k)=accuracy(1,1);

result\_classificition(k)=predict\_label;

end

Average\_Cross\_Validation\_Accuracy=sum(total\_index(:))/104

total\_0 = length(find(data(:,3) == 0));

total\_1 = length(find(data(:,3) == 1));

number\_0 = length(find(wine\_labels == 0));

number\_1 = length(find(wine\_labels == 1));

number\_0\_0\_cross = length(find(result\_classificition == 0 & wine\_labels == 0));

number\_1\_1\_cross = length(find(result\_classificition == 1 & wine\_labels == 1));

number\_0\_1\_cross = length(find(result\_classificition == 1 & wine\_labels == 0));

number\_1\_0\_cross = length(find(result\_classificition == 0 & wine\_labels == 1));

disp(['total：' num2str(104)...

' 1：' num2str(total\_0)...

' 2：' num2str(total\_1)]);

disp(['1：' num2str(number\_0\_0\_cross)...

' error：' num2str(number\_0 - number\_0\_0\_cross)...

' correct p1=' num2str(number\_0\_0\_cross/number\_0\*100) '%']);

disp(['2：' num2str(number\_1\_1\_cross)...

' error：' num2str(number\_1 - number\_1\_1\_cross)...

' correct p3=' num2str(number\_1\_1\_cross/number\_1\*100) '%']);

disp(['total：' num2str(104)...

' correct p=' num2str((number\_0\_0\_cross+number\_1\_1\_cross)/(number\_0+number\_1)\*100) '%']);

disp([' 0 to 1：' num2str(number\_0\_1\_cross)]);

disp([' 1 to 0：' num2str(number\_1\_0\_cross)]);