**模式识别——Bayes编程实习报告**

**姚毅2014302590137**

**武汉大学-模式识别课程**

本程序使用matlab编写，程序实现了基于最小错误率的Bayes算法，采用马氏距离进行分类，可以自由选取ROI区域并且读取样本进行训练。程序总用时：5小时，其中2小时编写代码，2小时调试。1小时撰写编程实习报告。

经过测试程序可以运行，并且程序得到的分类结果与用人眼分辨处理的相应结果相符，程序可靠性很高。另外，程序的亮点有：

**1.数据自由无限制**，读取txt文件中的数据，自动识别特征数目以及样本数目。使用者可自由更换数据，支持多特征多类别，可以对txt样本进行训练，例如书上的例题。

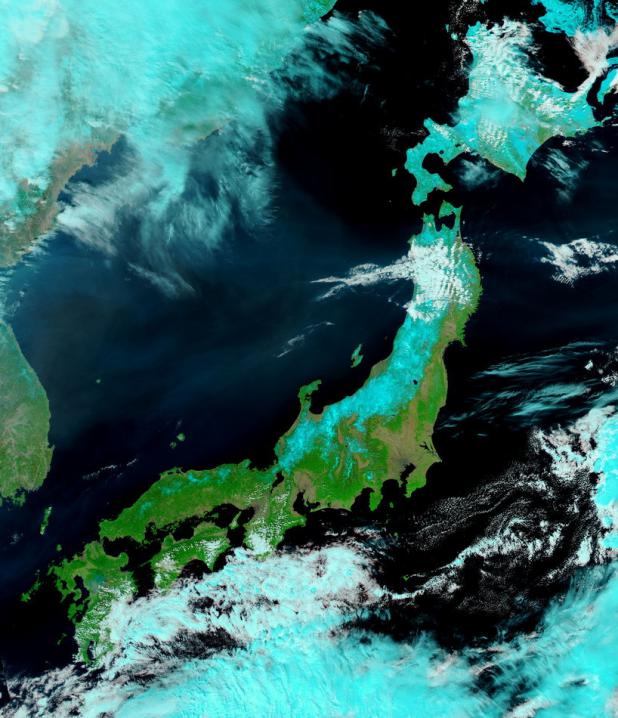
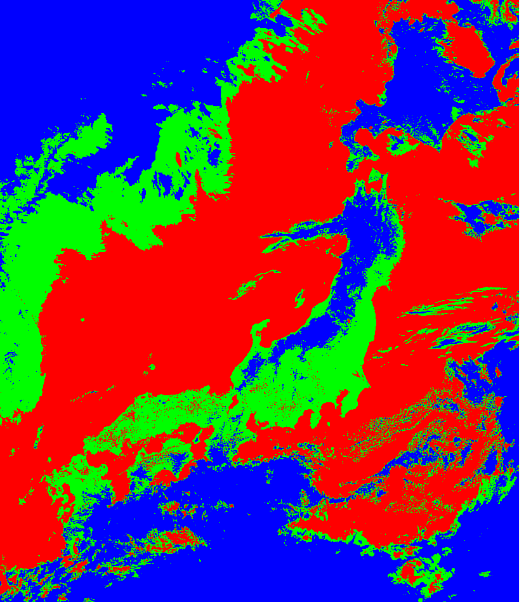
**2、ROI区域可以自己在程序内进行选择并读取ROI文件。**

**3、实现了对遥感图像的处理。**采用命令输入的形式，可以选择数据来源，可以是txt的数据文件，也可以是遥感图像

本实习报告主要从以下4方面讲解本次实习中遇到的问题以及算法相关：

1. **Bayes算法基本原理**
2. **算法代码化过程，程序中遇到的问题以及解决方法**
3. **程序可行性测试以及结果输出**
4. **附录：matlab源代码；**

程序结果：



**一、Bayes算法基本原理**

由于客观事物的复杂性，分类器作出各种判决时的风险是不一样的。例如将癌细胞误判为正常细胞的风险就比将正常细胞误判为癌细胞的风险大。因此，在贝叶斯分类器中引入了风险的概念。在实际应用中根据具体情况决定各种风险的大小，通常用一组系数Cij来表示。Cij表示分类器将被识别样本分类为ωi，而该样本的真正类别为ωj时的风险。设计最小风险分类器的基本思想是用后验概率计算将 x分类为ωi的条件风险比较各Ri(x)的大小,与最小值对应的类别是分类的结果。评价这种分类器的标准是平均风险，它的平均风险最小。在实际应用时，后验概率是难以获得的，根据模式类别的多少和Cij的取值方式,可设计出各种分类器，例如模式为两类时，判别函数为：



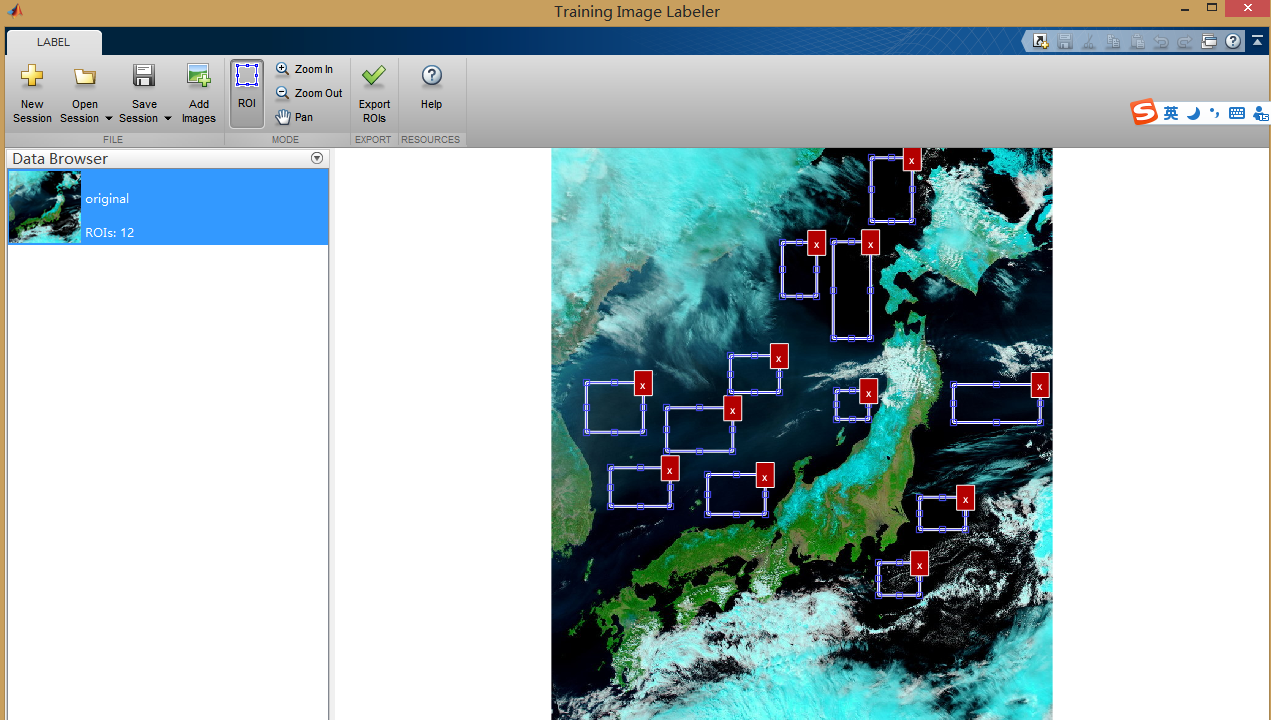
如果选择C11和C22为零，C12和C21为1，它就是两类最小错误概率分类器。实际上，最小错误概率分类器是最小风险分类器的一种特殊情况。

设计贝叶斯分类器的关键是要知道样本特征 x的各种概率密度函数。条件概率密度函数为多元正态分布是研究得最多的分布。这是由于它的数学表达式易于分析，在实际应用中也是一种常见的分布形式。经常使用参数方法来设计正态分布的判别函数。

参考资料：福永圭之介著,陶笃纯译：《统计图形识别导论》,科学出版社，北京，1978。

**二、Bayes算法代码化过程以及相关问题**

1）弹出对话框，进行ROI区域的选取。



2）读取ROI区域数据，保存在train三维空间内（三个维度分别对应：样本、特征、类别），等待接下来的训练。

3）训练ROI样本，利用协方差和平均值，求出分类器对应的先验概率、概率密度等参数，对应的分类器为：

g1=-1/2\*R'\*inv(sigma1)\*R+class\_f(1,:)\*inv(sigma1)\*R-1/2\*class\_f(1,:)'\*inv(sigma1)\*class\_f(1,:)'-1/2\*log(abs(det(sigma1)))+log(pw1);

1. 读取遥感图像，进行每个样本的类别判断，这里和kmeans中使用了同一种转换数据格式的方法，都将三维的数据转换成了二维数据，然后行进一个个地判断。对应转换：

for i=1:rows

for j=1:lines

for k=1:deep

C\_data((i-1)\*lines+j,k)=C\_data\_temp(i,j,k);

end

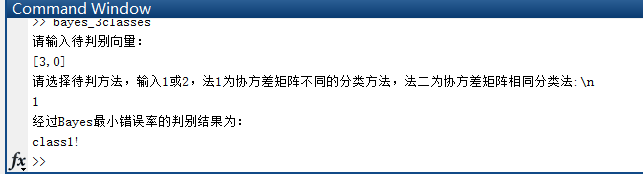
end

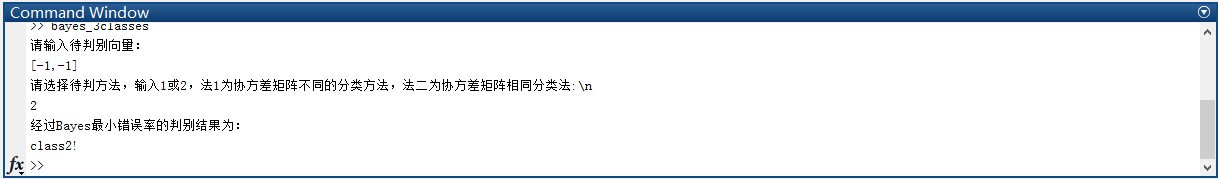
end

1. 根据分类结果赋值，给分为的三类分别赋予RGB三种颜色。之后显示图像

**三、程序运行结果**

对日本海域的遥感图像分类结果如第一页所示，不再展示，下面展示对书上例子的结果展示：（读取TXT格式数据，然后输入待分类样本进行分类）





**四、附录：matlab源代码**

function bayes

M=3;

trainingImageLabeler; %弹出对话框进行ROI数据的选取

choice=input('Choose your data file type: \n1:txt \n2:picture \n');

%选择自己使用TXT数据还是对遥感图像进行处理

%C\_text=textscan(file,'%s',n,'Delimiter','')

if choice==1

%file=fopen('data.txt','r');

C\_data=importdata('data.txt');

%C\_data=cell2mat(C\_data);

[m,n]=size(C\_data);

%fclose(file);

end

if choice==2

% img=input('input your picture name');

img='original.jpg'; %打开原始图像

C\_data\_temp=imread(img);

clouds=importdata('clouds.mat');% 读入选择的ROI样本区域

land=importdata('land.mat');

ocean=importdata('ocean.mat');

temp\_samples=1;

for i=1:M

if i==1

[num\_roi\_temp,useless]=size(clouds.objectBoundingBoxes);

end

if i==2

[num\_roi\_temp,useless]=size(land.objectBoundingBoxes);

end

if i==3

[num\_roi\_temp,useless]=size(ocean.objectBoundingBoxes);

end

for j=1:num\_roi\_temp

x=coulds.objectBoundingBoxes(j,1);

y=coulds.objectBoundingBoxes(j,2);

width=coulds.objectBoundingBoxes(j,3);

length=coulds.objectBoundingBoxes(j,4);

for ii=1:width

for jj=1:length

train(temp\_samples,:,i)=C\_data\_temp(x+ii,y+jj,:);

%对ROI样本区域内的每个样本写入train空间，等待训练

temp\_samples=temp\_samples+1;

end

end

end

end

%将三维的数组格式转化一下，成为我写的数据格式。

[rows,lines,deep]=size(C\_data\_temp);

new\_img=zeros(rows,lines);

m=rows\*lines;

n=deep;

for i=1:rows

for j=1:lines

for k=1:deep

C\_data((i-1)\*lines+j,k)=C\_data\_temp(i,j,k);

end

end

end

C\_data=double(C\_data);

end

if choice~=1&&choice~=2

disp('your input number is wrong, please reinput it!\n');

return;

end

%注意，这里C\_data是要求处理的遥感图像

for i=1:M

for j=1:n

class\_f(i,j)=mean(train(:,n,i));

end

end

sigma1=cov(train(:,1,1),train(:,2,1),train(:,3,1));

sigma2=cov(train(:,1,2),train(:,2,2),train(:,3,2));

sigma3=cov(train(:,1,3),train(:,2,3),train(:,3,3));

if choice==1

disp('请输入待判别向量：')

R=input('');

R=R';

end

pw1=1/3;

pw2=1/3;

pw3=1/3;

disp('请选择待判方法，输入1或2，法1为协方差矩阵不同的分类方法，法二为协方差矩阵相同分类法:\n')

Choice\_2=1;

%Choice=input('');

%注意，这里的class\_f(1,:) 是个行向量，所以跟公式的转置情况是相反的

for i=1:rows

for j=1:lines

R=C\_data((i-1)\*lines+j,:)';

if Choice\_2==1

g1=-1/2\*R'\*inv(sigma1)\*R+class\_f(1,:)\*inv(sigma1)\*R-1/2\*class\_f(1,:)'\*inv(sigma1)\*class\_f(1,:)'-1/2\*log(abs(det(sigma1)))+log(pw1);

g2=-1/2\*R'\*inv(sigma2)\*R+class\_f(2,:)\*inv(sigma2)\*R-1/2\*class\_f(1,:)'\*inv(sigma2)\*class\_f(2,:)'-1/2\*log(abs(det(sigma2)))+log(pw2);

g3=-1/2\*R'\*inv(sigma3)\*R+class\_f(3,:)\*inv(sigma3)\*R-1/2\*class\_f(1,:)'\*inv(sigma3)\*class\_f(3,:)'-1/2\*log(abs(det(sigma3)))+log(pw3);

else if Choice\_2==2

g1=class1\_f'\*inv(sigma1+sigma2+sigma3)\*R-1/2\*class\_f(1,:)\*inv(sigma1+sigma2+sigma3)\*class\_f(1,:)'+log(pw1);

g2=class2\_f'\*inv(sigma1+sigma2+sigma3)\*R-1/2\*class\_f(2,:)\*inv(sigma1+sigma2+sigma3)\*class\_f(2,:)'+log(pw2);

g3=class3\_f'\*inv(sigma1+sigma2+sigma3)\*R-1/2\*class\_f(3,:)\*inv(sigma1+sigma2+sigma3)\*class\_f(3,:)'+log(pw3);

else

disp('选择有误，请重新输入');

end

end

Gi=[g1,g2,g3];

G=max(Gi);

if G==g1

New\_Img(i,j,1)=255;

New\_Img(i,j,2)=0;

New\_Img(i,j,3)=0;

else if G==g2

New\_Img(i,j,1)=0;

New\_Img(i,j,2)=255;

New\_Img(i,j,3)=0;

else

New\_Img(i,j,1)=0;

New\_Img(i,j,2)=0;

New\_Img(i,j,3)=255;

end

end

end

end

imshow(New\_Img);

imwrite(New\_Img,'New','bmp');

end