Machine Learning Homework 4.1

专业：软件工程　　姓名：沈金龙　　学号：18214806

1．作业题目：A classification problem with two classes

There is a classification problem having two classes, with equal prior probabilities, and is shown in Figure 2.1.

1) Generate a figure same like Fig.2.1. The blue class is generated from a single Gaussian while the red class comes from a mixture of two Gaussians.

2) Because we know the class priors and the class-conditional densities, it is straight forward to evaluate and plot the true posterior probabilities as well as the minimum misclassification-rate decision boundary, as shown in Figure 2.1.

3) Evaluate the optimal decision boundary for minimizing the misclassification rate (which corresponds to the contour along which the posterior probabilities for each class equal 0.5) and show is by the green curve.

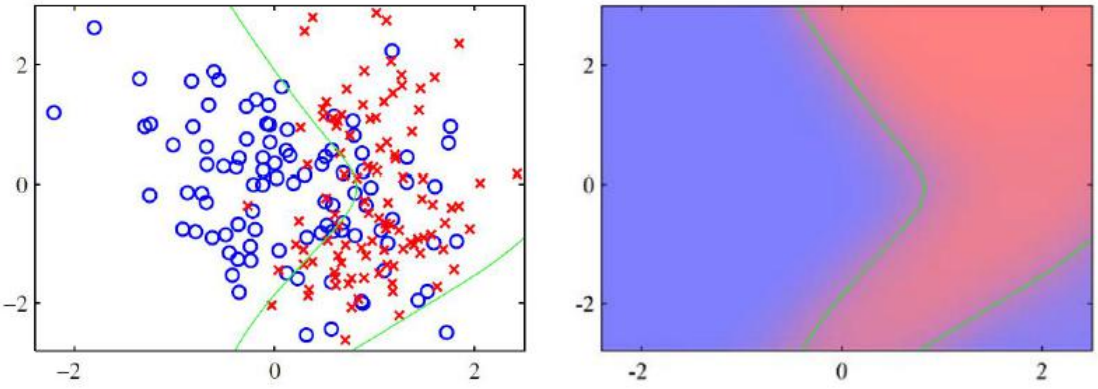


Figure 2.1 The left plot shows the synthetic classification data set with data from the two classes shown in red and blue. On the right is a plot of the true posterior probabilities, shown on a colour scale going from pure red denoting probability of the red class is 1 to pure blue denoting probability of the red class is 0. Because these probabilities are known, the optimal decision boundary for minimizing the misclassification rate (which corresponds to the contour along which the posterior probabilities for each class equal 0.5) can be evaluated and is shown by the green curve. This decision boundary is also plotted on the left-hand figure.

2.实验过程及代码：

1）本实验采用 MATLAB R2016a 以及 Python 3.7 完成，首先使用 matlab 生成样本点，使其满足高斯分布。

```
% 生成training sample
MU1 = [1 3];
MU2 = [3 1];
SIGMA1 = [1.5,0;0,1];
SIGMA2 = [1,0.5;0.5,2];

M1 = mvnrnd(MU1,SIGMA1,100);
M2 = mvnrnd(MU2,SIGMA2,100);

plot(M1(:,1),M1(:,2),'bO',M2(:,1),M2(:,2),'r*')
```

2）接着采用最大似然估计出高斯的参数，然后用最小错误率贝叶斯分类器进行分类操作。

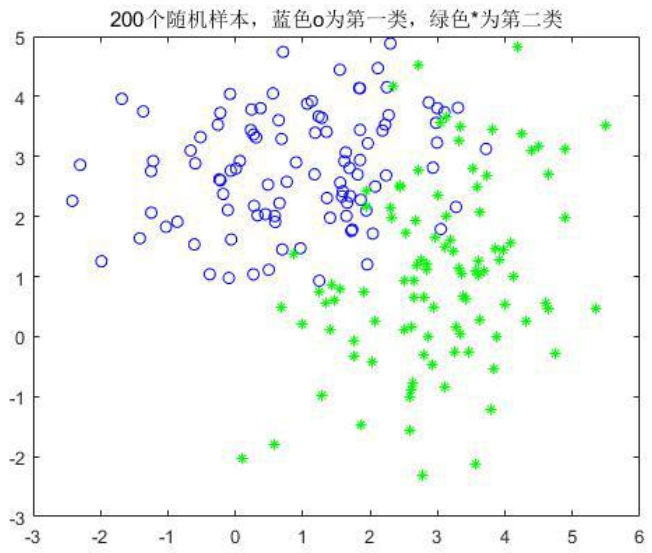
```
for i=1:(varargin{3}+varargin{7})
    x=w(i,1);
    y=w(i,2);
    g1=mvnpdf([x,y],varargin{1},varargin{2})*varargin{4};
    g2=mvnpdf([x,y],varargin{5},varargin{6})*varargin{8};
    if g1>g2
        if 1<=i&&i<=varargin{3}
            n1=n1+1;%第一类正确个数
            plot(x,y,'bo');%蓝色o表示正确分为第一类的样本
            hold on;
        else
            plot(x,y,'r^');% 红色的上三角形表示第二类错误分为第一类 selonsy
            hold on;
        end
        R1(m,1)=x;R1(m,2)=y;m=m+1;
        R(i)=1;
    else
        if varargin{3}<=i&&i<=(varargin{3}+varargin{7})
            n2=n2+1;%第二类正确个数
            plot(x,y,'g*');%绿色*表示正确分为第二类的样本
            hold on;
        else
            plot(x,y,'rv');% 红色的下三角形表示第一类错误分为第二类 selonsy
            hold on;
        end
        R2(n,1)=x;R2(n,2)=y;n=n+1;
        R(i)=0;
    end
end
```

3）分类的结果导入到 python 中 ，并使用 contour 画等高线的方式画出分类后的决策边界。

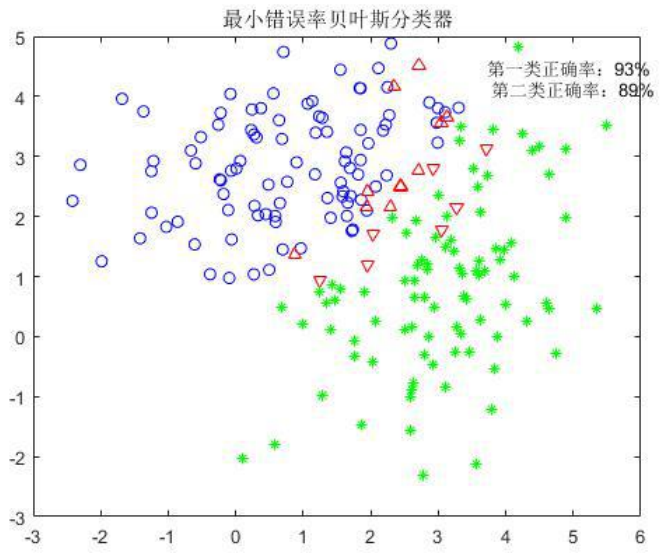
```python
# 画决策边界
def plot_decision_boundary(pred_func):

    # 设定最大最小值，附加一点点边缘填充
    x_min, x_max = X[:, 0].min() - .5, X[:, 0].max() + .5
    y_min, y_max = X[:, 1].min() - .5, X[:, 1].max() + .5
    h = 0.01

    xx, yy = np.meshgrid(np.arange(x_min, x_max, h), np.arange(y_min, y_max, h))

    # 用预测函数预测一下
    Z = pred_func(np.c_[xx.ravel(), yy.ravel()])
    Z = Z.reshape(xx.shape)

    # 然后画出图
    plt.contour(xx, yy, Z, cmap=plt.cm.Spectral)
    plt.scatter(X[:, 0], X[:, 1], c=R, cmap=plt.cm.Spectral)

clf = LogisticRegressionCV()
clf.fit(X, R)

plot_decision_boundary(lambda x: clf.predict(x))
plt.title("Logistic Regression")
plt.show()
```

3.实验结果与分析：

1）生成的样本点图：

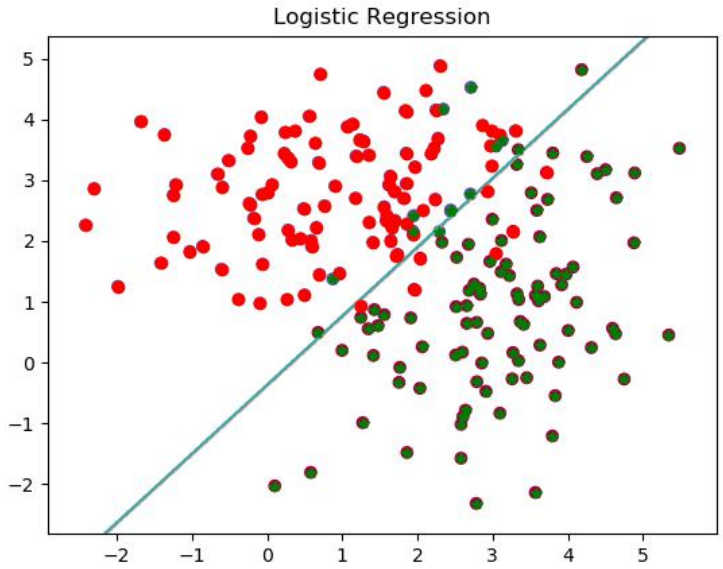


2）使用最小错误率贝叶斯分类器进行样本点的分类：



由上图可知，第一类和第二类分类的正确率分别为：93%、89%。

3）使用 contour 函数画出分类的决策边界，如下图所示：



4.总结

本次实验学习了高斯分布、先验概率、后验概率、最大似然估计、最小错误率贝叶斯分类方法等相关的知识，以及使用 contour 来画决策边界的方法。同时学会了利用 sklearn 来提高自己的动手能力，加深了对理论知识的理解和把握。