山东大学______计算机科学与技术______学院

机器学习与模式识别 课程实验报告

学号：201600301304	姓名：贾乘兴	班级： 人工智能 16
实验题目：线性判别分析		
实验学时：2 小时	实验日期： 2018/10/15	
实验目的：利用给定数据（红蓝绿数据集），实现线性判别分析的二分类方法和多分类方法，并绘制图像查看效果		
硬件环境： 16 GB 内存		
软件环境：mac os，matlab 2017b		

实验步骤与内容：

一．二分类模型

1.二分类方法是最经典的线性判别模型，在本实验的两种特征的数据集下，其思想为将数据投影到一条直线上，在直线上进行分类，所求直线为分类的最佳直线

2.判定该直线分类效果的好坏的方式为：投影到该直线后，类间距离之和较大，同时类内数据点较为聚集。类间距离我们用均值期望之间 l2 距离的平方表示，可表示为：

$$d_b = \left\|\omega^T\mu_0 - \omega^T\mu_1\right\|_2^2 = \omega^T(\mu_0-\mu_1)(\mu_0-\mu_1)^T\omega$$

类间距离要尽可能大，其中，定义类间散度矩阵：

$$S_b = (\mu_0-\mu_1)(\mu_0-\mu_1)^T$$

类内距离我们用各个类下的方差表示：

$$d_w = \omega^T(\Sigma_0+\Sigma_1)\omega = \omega^T\sum_{x\in X_0}(x-\mu_0)(x-\mu_0)^T\omega + \omega^T\sum_{x\in X_1}(x-\mu_1)(x-\mu_1)^T\omega$$

类内距离要尽量小，其中，定义类内散度矩阵：

$$S_w = \Sigma_0+\Sigma_1 = \sum_{x\in X_0}(x-\mu_0)(x-\mu_0)^T + \sum_{x\in X_1}(x-\mu_1)(x-\mu_1)^T$$

这样我们得到要最大化的目标函数为

$$J = \frac{\omega^T S_b\omega}{\omega^T S_w\omega}$$

该目标函数又被称为类间散度矩阵和类内散度矩阵的“广义瑞利商”

在求 w 的优化上，我们可以令分母为 1，将该问题变为一般性的标准优化问题

$$\arg\min_{\omega} \quad -\omega^T S_b\omega$$
$$s.t. \quad \omega^T S_w\omega = 1$$

由拉格朗日乘子法，该问题等价于

$$S_b\omega = \lambda S_w\omega$$

经过变形可得：

$$S_w^{-1} S_b \omega = \lambda \omega$$

所求 w 即为左侧矩阵最大特征值对应的特征向量
而在该二分类问题中，考虑到类间散度矩阵的特点：

$$S_b \omega = (\mu_0 - \mu_1)(\mu_0 - \mu_1)^T \omega = (\mu_0 - \mu_1)\lambda_k = \lambda_k(\mu_0 - \mu_1)$$

可得：

$$\omega = S_w^{-1}(\mu_0 - \mu_1)$$

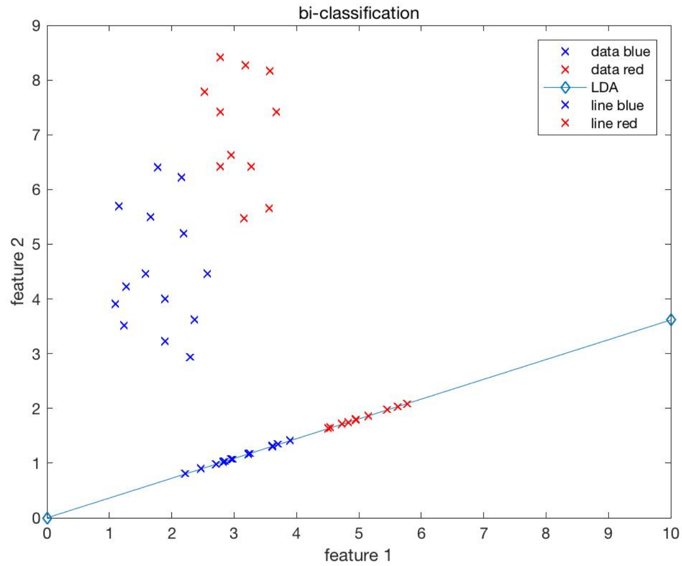
最终将 w 单位化即可，该方法较为简单
3. 实现二分类
a. 数据读取

```
%% load data
xb=load('ex3blue.dat');
xg=load('ex3green.dat');
xr=load('ex3red.dat');
```

b. 投影向量的计算

```
mb=mean(xb);
mr=mean(xr);
Sw=(xb-mb)'*(xb-mb)+(xr-mr)'*(xr-mr);
w=inv(Sw)*(mb-mr)';
w=abs(w)/sqrt(w'*w);
x=[0:0.01,10];
y=(w(2)/w(1))*x;
zb=xb*w*w';
zr=xr*w*w';
```

最终得到的二分类结果与数据展示如下

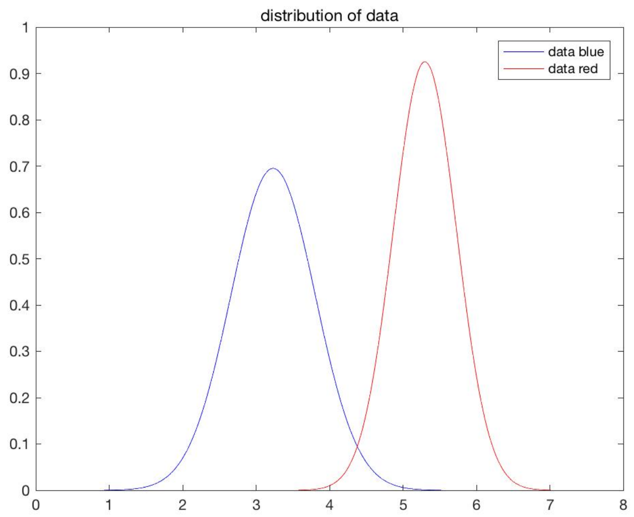


绘制图像代码如下

```
figure
```

```
plot(xb(:,1),xb(:,2),'bx');
hold on
plot(xr(:,1),xr(:,2),'rx');
hold on
plot(x,y,'-d');
plot(zb(:,1),zb(:,2),'bx');
hold on
plot(zr(:,1),zr(:,2),'rx');
xlabel('feature 1');
ylabel('feature 2');
legend('data blue','data red','LDA','line blue','line red');
title('bi-classification');
```

在得到了投影到直线上的数据上后，我们假设数据分布满足给定数据投影的均值和方差下的正态分布，对数据分布进行了可视化，可见分类效果较好



代码如下：

```
%% distribution
yb=xb*w;
yr=xr*w;
sb=std(yb);
sr=std(yr);
ub=mean(yb);
ur=mean(yr);
tb=[ub-4*sb:0.01:ub+4*sb];
tr=[ur-4*sr:0.01:ur+4*sr];
figure
plot(tb,exp(-(tb-ub).*(tb-ub)/(2*sb^2))/(sb*sqrt(2*pi)),'b-');
hold on
plot(tr,exp(-(tr-ur).*(tr-ur)/(2*sr^2))/(sr*sqrt(2*pi)),'r-');
title('distribution of data');
legend('data blue','data red');
```

二. 多分类

1.LDA 在多分类问题上较二分类复杂，可以将该问题视为降维问题
2.在二分类问题的基础上，我们定义全局散度矩阵：

$$S_t = S_b + S_w = \sum_{i=1}^{m}(x_i - \mu)(x_i - \mu)^T$$

其中，u 为全部数据的均值：

$$\mu = \frac{1}{N}\sum_{i=1}^{C} n_i \mu_i$$

在多分类问题下，类间散度矩阵可表示为：

$$S_b = S_t - S_w = \sum_{i=1}^{N} m_i(\mu_i - \mu)(\mu_i - \mu)^T$$

多分类的优化问题可表示为：

$$\max \frac{tr(W^T S_b W)}{tr(W^T S_w W)}$$

该问题仍然可以转化为求解最大特征值的特征向量组成的矩阵问题
3.多分类实现
a. 数据导入
b. 投影矩阵求解

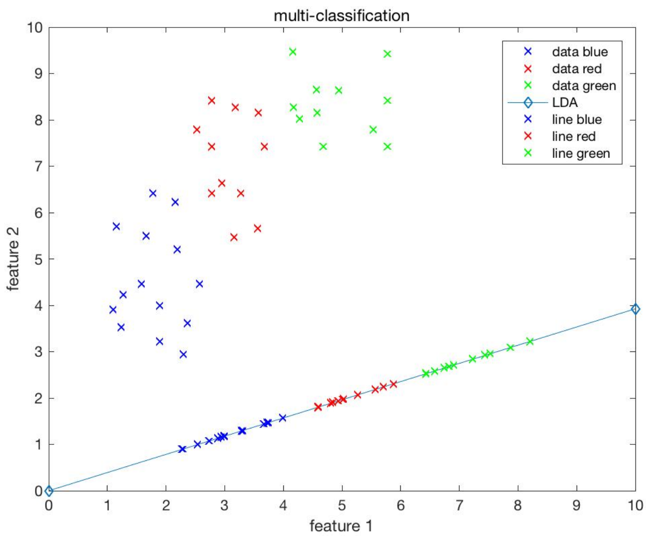
```
mb=mean(xb);
mr=mean(xr);
mg=mean(xg);
nb=length(xb);
nr=length(xr);
ng=length(xg);
na=nb+nr+ng;
ma=(nb*mb+nr*mr+ng*mg)/na;
St=(xb-ma)'*(xb-ma)+(xr-ma)'*(xr-ma)+(xg-ma)'*(xg-ma);
Sw=(xb-mb)'*(xb-mb)+(xr-mr)'*(xr-mr)+(xg-mg)'*(xg-mg);
Sb=St-Sw;
[vec,val]=eig(inv(Sw)*Sb);
lamda=max(diag(val));
W=vec(:,find(diag(val==lamda)));
x=[0:0.01,10];
y=(W(2)/W(1))*x;
zb=xb*W*W';
zr=xr*W*W';
zg=xg*W*W';
```

c. 绘制图像代码

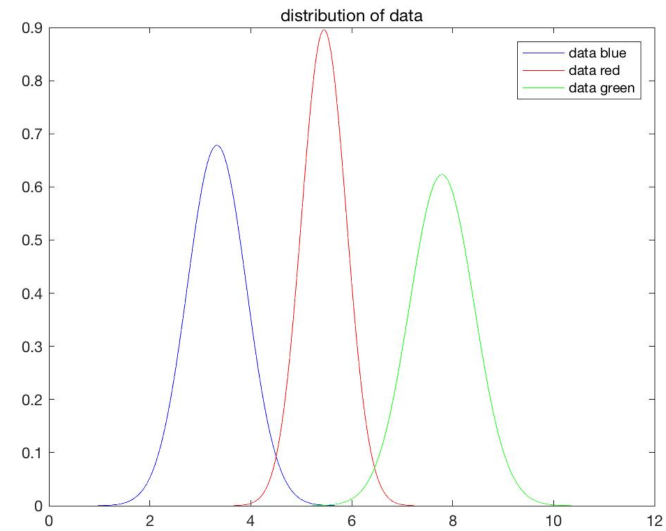
```
figure
plot(xb(:,1),xb(:,2),'bx');
hold on
plot(xr(:,1),xr(:,2),'rx');
hold on
```

```
plot(xg(:,1),xg(:,2),'gx');
hold on
plot(x,y,'-d');
plot(zb(:,1),zb(:,2),'bx');
hold on
plot(zr(:,1),zr(:,2),'rx');
hold on
plot(zg(:,1),zg(:,2),'gx');
xlabel('feature 1');
ylabel('feature 2');
legend('data blue','data red','data green','LDA','line blue','line red','line
green');
title('multi-classification');
```

可以得到多分类的结果如下：



可见该问题的分类效果较好，绘制在直线上的分布如下：



代码如下：

```
yb=xb*W;
yr=xr*W;
yg=xg*W;
sb=std(yb);
sr=std(yr);
sg=std(yg);
ub=mean(yb);
ur=mean(yr);
ug=mean(yg);
tb=[ub-4*sb:0.01:ub+4*sb];
tr=[ur-4*sr:0.01:ur+4*sr];
tg=[ug-4*sg:0.01:ug+4*sg];
figure
plot(tb,exp(-(tb-ub).*(tb-ub)/(2*sb^2))/(sb*sqrt(2*pi)),'b-');
hold on
plot(tr,exp(-(tr-ur).*(tr-ur)/(2*sr^2))/(sr*sqrt(2*pi)),'r-');
hold on
plot(tg,exp(-(tg-ug).*(tg-ug)/(2*sg^2))/(sg*sqrt(2*pi)),'g-');
title('distribution of data');
legend('data blue','data red','data green');
```

三. 分类器使用

对于相邻两类的边界计算如下：

$$bor_{i,j}=\frac{n_i\omega^T\mu_i+n_j\omega^T\mu_j}{n_i+n_j}$$

结论分析与体会： 本次实验了解了线性分类器的原理，并进行了实现，很多数学方法可以由简单的低维拓展到高维，并且在一定的问题下可以实现较好的效果

附录：程序源代码

ex3.m

```
clear,clc;
%% load data
xb=load('ex3blue.dat');
xg=load('ex3green.dat');
xr=load('ex3red.dat');
```

```matlab
%% bi-classification
figure
plot(xb(:,1),xb(:,2),'bx');
hold on
plot(xr(:,1),xr(:,2),'rx');

mb=mean(xb);
mr=mean(xr);
Sw=(xb-mb)'*(xb-mb)+(xr-mr)'*(xr-mr);
w=inv(Sw)*(mb-mr)';
w=abs(w)/sqrt(w'*w);

x=[0:0.01,10];
y=(w(2)/w(1))*x;
hold on
plot(x,y,'-d');

zb=xb*w*w';
zr=xr*w*w';
plot(zb(:,1),zb(:,2),'bx');
hold on
plot(zr(:,1),zr(:,2),'rx');
xlabel('feature 1');
ylabel('feature 2');
legend('data blue','data red','LDA','line blue','line red');
title('bi-classification');

%% distribution
yb=xb*w;
yr=xr*w;
sb=std(yb);
sr=std(yr);
ub=mean(yb);
ur=mean(yr);
tb=[ub-4*sb:0.01:ub+4*sb];
tr=[ur-4*sr:0.01:ur+4*sr];
figure
plot(tb,exp(-(tb-ub).*(tb-ub)/(2*sb^2))/(sb*sqrt(2*pi)),'b-');
hold on
plot(tr,exp(-(tr-ur).*(tr-ur)/(2*sr^2))/(sr*sqrt(2*pi)),'r-');
title('distribution of data');
legend('data blue','data red');

%% multi-classification
figure
plot(xb(:,1),xb(:,2),'bx');
```

```
hold on
plot(xr(:,1),xr(:,2),'rx');
hold on
plot(xg(:,1),xg(:,2),'gx');

mb=mean(xb);
mr=mean(xr);
mg=mean(xg);
nb=length(xb);
nr=length(xr);
ng=length(xg);
na=nb+nr+ng;
ma=(nb*mb+nr*mr+ng*mg)/na;

St=(xb-ma)'*(xb-ma)+(xr-ma)'*(xr-ma)+(xg-ma)'*(xg-ma);
Sw=(xb-mb)'*(xb-mb)+(xr-mr)'*(xr-mr)+(xg-mg)'*(xg-mg);
Sb=St-Sw;

[vec,val]=eig(inv(Sw)*Sb);
lamda=max(diag(val));
W=vec(:,find(diag(val==lamda)));

x=[0:0.01,10];
y=(W(2)/W(1))*x;
hold on
plot(x,y,'-d');

zb=xb*W*W';
zr=xr*W*W';
zg=xg*W*W';
plot(zb(:,1),zb(:,2),'bx');
hold on
plot(zr(:,1),zr(:,2),'rx');
hold on
plot(zg(:,1),zg(:,2),'gx');
xlabel('feature 1');
ylabel('feature 2');
legend('data blue','data red','data green','LDA','line blue','line red','line
green');
title('multi-classification');

%% distribution
yb=xb*W;
yr=xr*W;
yg=xg*W;
sb=std(yb);
```

```
sr=std(yr);
sg=std(yg);
ub=mean(yb);
ur=mean(yr);
ug=mean(yg);
tb=[ub-4*sb:0.01:ub+4*sb];
tr=[ur-4*sr:0.01:ur+4*sr];
tg=[ug-4*sg:0.01:ug+4*sg];
figure
plot(tb,exp(-(tb-ub).*(tb-ub)/(2*sb^2))/(sb*sqrt(2*pi)),'b-');
hold on
plot(tr,exp(-(tr-ur).*(tr-ur)/(2*sr^2))/(sr*sqrt(2*pi)),'r-');
hold on
plot(tg,exp(-(tg-ug).*(tg-ug)/(2*sg^2))/(sg*sqrt(2*pi)),'g-');
title('distribution of data');
legend('data blue','data red','data green');
```