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：201700301042	姓名： 陈佳睿	班级： 17 人工智能
实验题目：线性判别分析（LDA）		
实验目的：利用给定的一个三分类数据，实现线性判别的二分类方法和多分类方法		
硬件环境： **Macbook pro 8GB** 内存		
软件环境： Macos majove ，octave		

实验步骤与内容：

LDA 在模式识别领域中有广泛的应用，它是一种监督学习下的降维技术

其核心思想为”投影后类内方差最小，类间方差最大”

我们要将数据在低维度上进行投影，投影后希望每个类别数据的投影点尽可能的接近，不同类别的数据的类别中心之间的距离尽可能的大

一．数学基础知识

LDA原理与流程

给定数据集 $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{m}$

第 i 类示例的集合 X_i

第 i 类示例的均值向量 $\boldsymbol{\mu}_i$

第 i 类示例的协方差矩阵 $\boldsymbol{\Sigma}_i$

两类样本的中心在直线上的投影：$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_0$ 和 $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_1$

两类样本的协方差：$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\Sigma}_0\boldsymbol{w}$ 和 $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\Sigma}_1\boldsymbol{w}$

同类样例的投影点尽可能接近 → $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\Sigma}_0\boldsymbol{w}+\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\Sigma}_1\boldsymbol{w}$ 尽可能小

异类样例的投影点尽可能远离 → $\|\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_0-\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_1\|_2^2$ 尽可能大

于是，最大化

$$J=\frac{\|\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_0-\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_1\|_2^2}{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\Sigma}_0\boldsymbol{w}+\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\Sigma}_1\boldsymbol{w}}=\frac{\boldsymbol{w}^{\mathrm{T}}(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)^{\mathrm{T}}\boldsymbol{w}}{\boldsymbol{w}^{\mathrm{T}}(\boldsymbol{\Sigma}_0+\boldsymbol{\Sigma}_1)\boldsymbol{w}}$$

类内散度矩阵 (within-class scatter matrix)

$$\begin{aligned}\mathbf{S}_w&=\boldsymbol{\Sigma}_0+\boldsymbol{\Sigma}_1\\&=\sum_{\boldsymbol{x}\in X_0}(\boldsymbol{x}-\boldsymbol{\mu}_0)(\boldsymbol{x}-\boldsymbol{\mu}_0)^{\mathrm{T}}+\sum_{\boldsymbol{x}\in X_1}(\boldsymbol{x}-\boldsymbol{\mu}_1)(\boldsymbol{x}-\boldsymbol{\mu}_1)^{\mathrm{T}}\end{aligned}$$

类间散度矩阵 (between-class scatter matrix)

$$\mathbf{S}_b=(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)^{\mathrm{T}}$$

LDA的目标：最大化广义瑞利商 (generalized Rayleigh quotient)

$$\boxed{J=\frac{\boldsymbol{w}^{\mathrm{T}}\mathbf{S}_b\boldsymbol{w}}{\boldsymbol{w}^{\mathrm{T}}\mathbf{S}_w\boldsymbol{w}}}$$

$\boldsymbol{w}$ 成倍缩放不影响 J 值
仅考虑方向

令 $\boldsymbol{w}^{\mathrm{T}}\mathbf{S}_w\boldsymbol{w}=1$，最大化广义瑞利商等价形式为

$$\begin{aligned}\min_{\boldsymbol{w}}\quad&-\boldsymbol{w}^{\mathrm{T}}\mathbf{S}_b\boldsymbol{w}\\\text{s.t.}\quad&\boldsymbol{w}^{\mathrm{T}}\mathbf{S}_w\boldsymbol{w}=1\end{aligned}$$

运用拉格朗日乘子法，有 $\mathbf{S}_b\boldsymbol{w}=\lambda\mathbf{S}_w\boldsymbol{w}$

$\mathbf{S}_b\boldsymbol{w}$ 的方向恒为 $\boldsymbol{\mu}_0-\boldsymbol{\mu}_1$，不妨令 $\mathbf{S}_b\boldsymbol{w}=\lambda(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)$

于是 $\boldsymbol{w}=\mathbf{S}_w^{-1}(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)$

实践中通常是进行奇异值分解 $\mathbf{S}_w=\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\mathrm{T}}$ ⟶ β

然后 $\mathbf{S}_w^{-1}=\mathbf{V}\boldsymbol{\Sigma}^{-1}\mathbf{U}^{\mathrm{T}}$

假定有 N 个类

- 全局散度矩阵 $\mathbf{S}_t = \mathbf{S}_b + \mathbf{S}_w = \sum_{i=1}^{m} (\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^T$
- 类内散度矩阵 $\mathbf{S}_w = \sum_{i=1}^{N} \mathbf{S}_{w_i} \quad \mathbf{S}_{w_i} = \sum_{\boldsymbol{x} \in X_i} (\boldsymbol{x} - \boldsymbol{\mu}_i)(\boldsymbol{x} - \boldsymbol{\mu}_i)^T$
- 类间散度矩阵 $\mathbf{S}_b = \mathbf{S}_t - \mathbf{S}_w = \sum_{i=1}^{N} m_i (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T$

多分类LDA有多种实现方法：采用 $\mathbf{S}_b$, $\mathbf{S}_w$, $\mathbf{S}_t$ 中的任何两个

例如，$\max_{\mathbf{W}} \frac{\operatorname{tr}(\mathbf{W}^T\mathbf{S}_b\mathbf{W})}{\operatorname{tr}(\mathbf{W}^T\mathbf{S}_w\mathbf{W})} \Longrightarrow \mathbf{S}_b\mathbf{W} = \lambda\mathbf{S}_w\mathbf{W}$

$\mathbf{W} \in \mathbb{R}^{d\times(N-1)}$

$\mathbf{W}$ 的闭式解是 $\mathbf{S}_w^{-1}\mathbf{S}_b$ 的 N-1 个最大广义特征值所对应的特征向量组成的矩阵

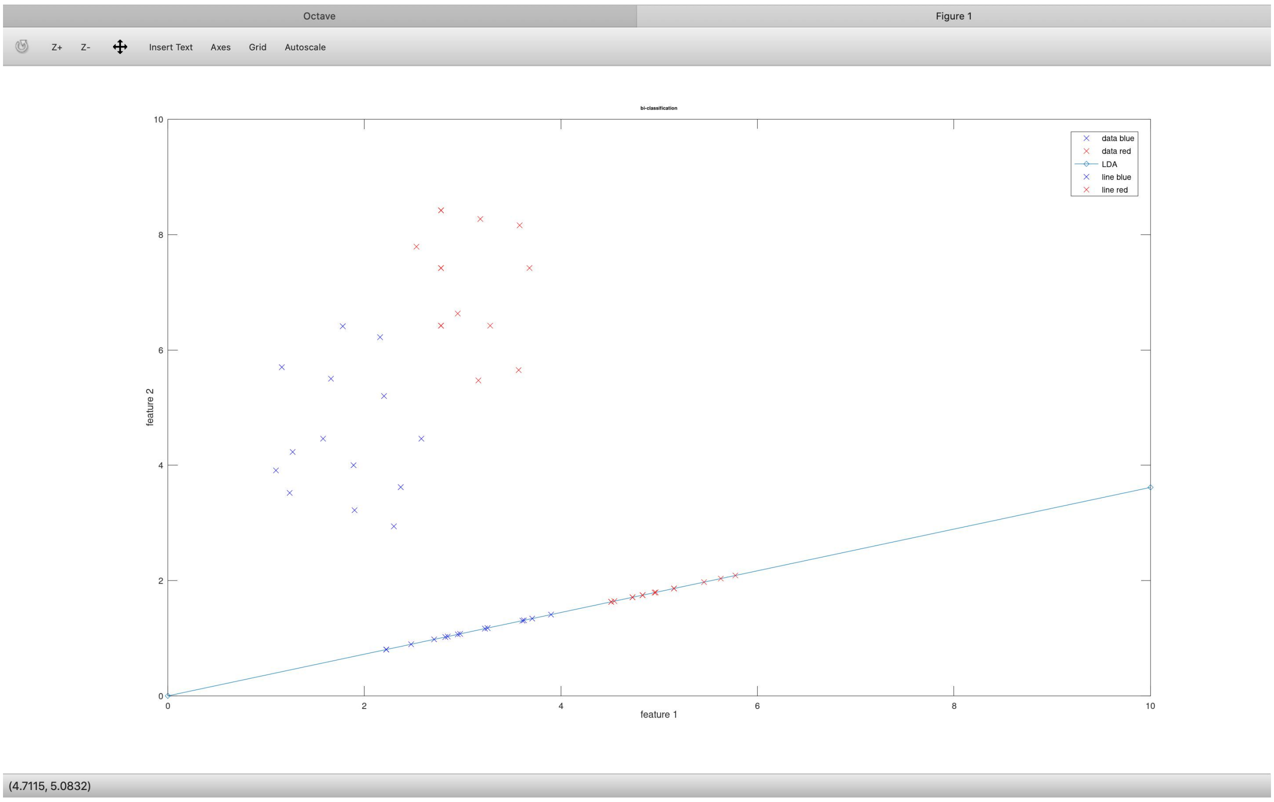
二．octave 对于数据进行二分类

1.数据读取

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

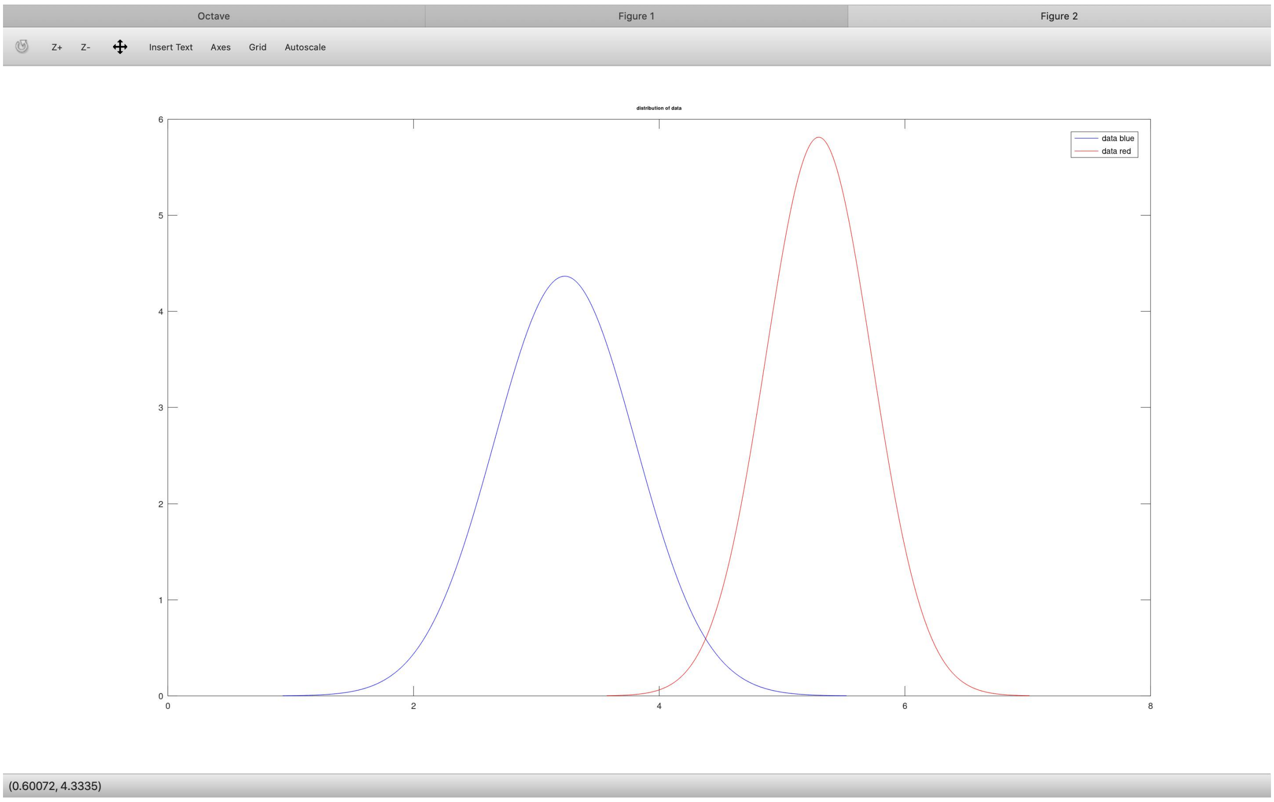
2.投影向量 w 的计算

```
>> 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);  //这里我将 w 进行了单位化
>> x =[0:0.01,10];
>> y = (w(2)/w(1))*x;
>> zb = xb*w*w';
>> zr = xr*w*w';
```



最终得到的二分类图像结果如上图

接下来我根据高斯公式，将我降维数据的分布曲线绘制出来



二分类的数据分布的交叉部分比较少，所以可以认为使用 lda 进行降维的效果比较好

二．多分类（本实验为 3 分类）

1.定义全局散度函数

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

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，类间散度矩阵定义如下：

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

2.多分类的优化即为实现以下函数：

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

3.多分类实现

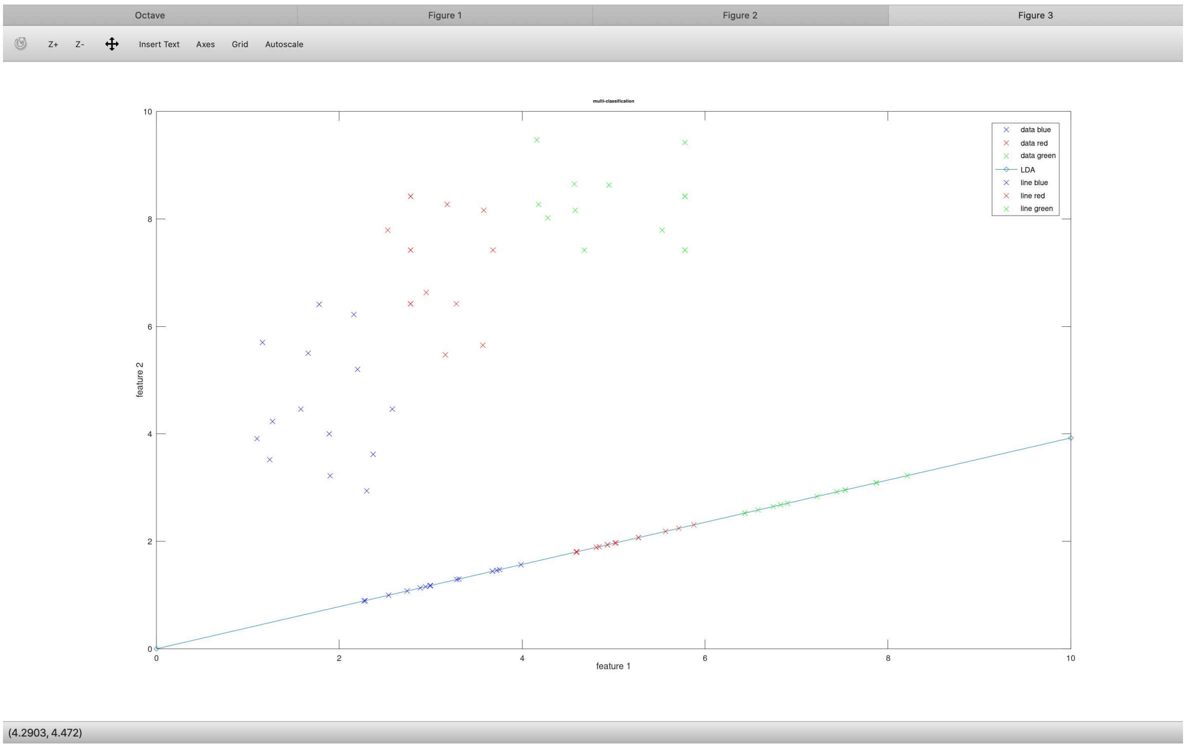
1)求解投影矩阵 w

有了以上实验进行二分类 lda 的经验，根据上述列出的计算公式我们可以很快通过 octave 计算出三个类的全局散度和三个类的类内散度值之和，并由此计算出类间的散度矩阵

```
>> 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);
```

使用 octave 提供的 eig 函数

计算出 Sw 的逆矩阵点乘 Sb 所得到的矩阵的特征向量和特征值

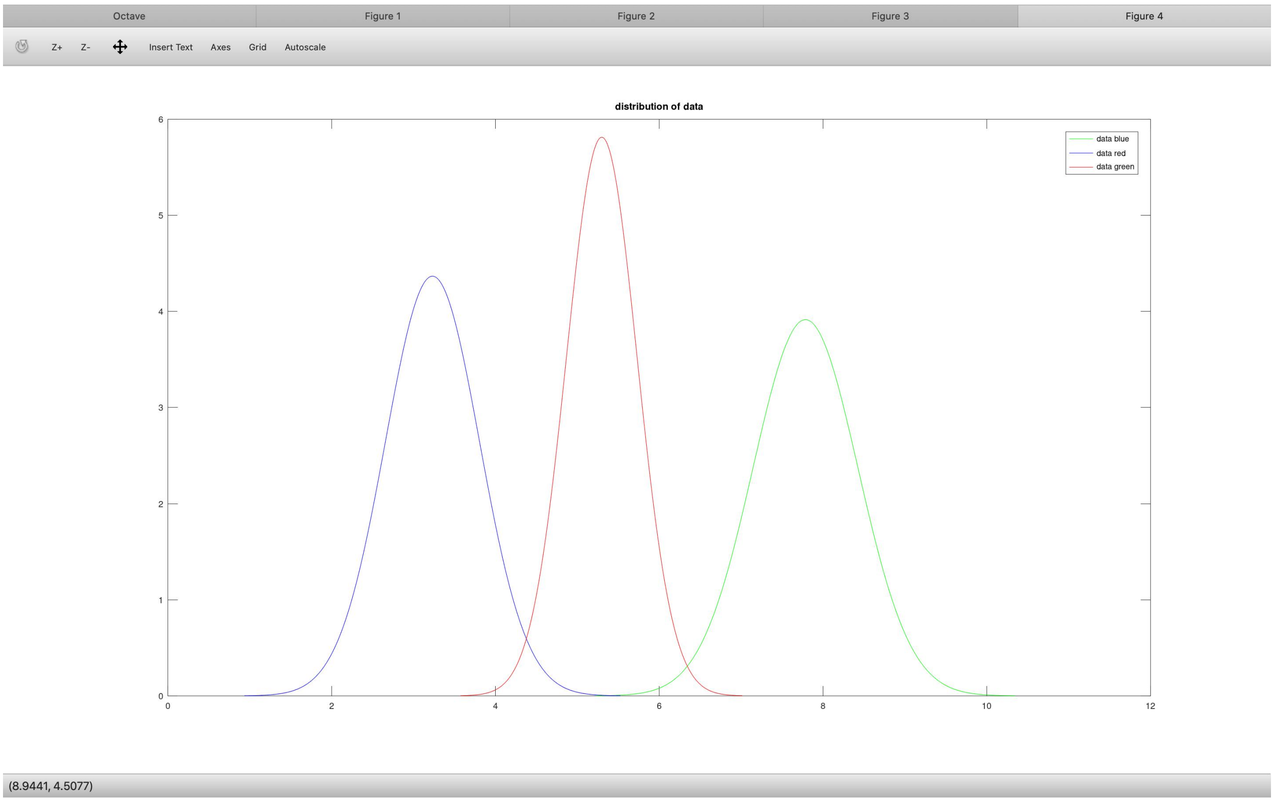


最终分类的结果如上图所示

依旧使用高斯函数来进行统计和分布绘制

得到下面的数据分布图

三者的交叉区域比较少，三分类的任务通过 lda 算法也完成地比较好



结论分析与体会：

LDA与PCA

LDA用于降维，和PCA有很多相同，也有很多不同的地方，因此值得好好的比较一下两者的降维异同点。

相同点

1）两者均可以对数据进行降维。

2）两者在降维时均使用了矩阵特征分解的思想。

3）两者都假设数据符合高斯分布。

不同点

1）LDA是有监督的降维方法，而PCA是无监督的降维方法

2）LDA降维最多降到类别数k-1的维数，而PCA没有这个限制。

3）LDA除了可以用于降维，还可以用于分类。

4）LDA选择分类性能最好的投影方向，而PCA选择样本点投影具有最大方差的方向。这点可以从下图形象的看出，在某些数据分布下LDA比PCA降维较优。

LDA小结

LDA算法既可以用来降维，又可以用来分类，但是目前来说，主要还是用于降维。在进行图像识别相关的数据分析时，LDA是一个有力的工具。下面总结下LDA算法的优缺点。

优点

1）在降维过程中可以使用类别的先验知识经验，而像PCA这样的无监督学习则无法使用类别先验知识。

2）LDA在样本分类信息依赖均值而不是方差的时候，比PCA之类的算法较优。

缺点

1）LDA不适合对非高斯分布样本进行降维，PCA也有这个问题。

2）LDA降维最多降到类别数k-1的维数，如果我们降维的维度大于k-1，则不能使用LDA。当然目前有一些LDA的进化版算法可以绕过这个问题。

3）LDA在样本分类信息依赖方差而不是均值的时候，降维效果不好。

4）LDA可能过度拟合数据。

代码：

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

图表绘制：

```
>> yb = xb*w;
>> yr = xr*w;
>> std_blue = std(yb);
>> std_red = std(yr);
>> tb = [mean_blue-4*std_blue:0.01:mean_blue+4*std_blue];
>> tr = [mean_red-4*std_red:0.01:mean_red+4*std_red];
>> figure
>>plot(tb,exp(-(tb-mean_blue).*(tb-mean_blue)/(2*std_blue^2))/std_blue*sqrt(2*pi),
'b-')
>> hold on
>> plot(tr,exp(-(tr-mean_red).*(tr-mean_red)/(2*std_red^2))/std_red*sqrt(2*pi),'r-')
>> title("distribution of data")
>> legend('data blue','data red');
```

多分类任务：

```
>> 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);
>> 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);
error: index (-0.165): subscripts must be either integers 1 to (2^63)-1 or logicals
>> Sw = (xb-mb)'*(xb-mb)+(xr-mr)'*(xr-mr)+(xg-mg)'*(xg-mg);
>> Sb    = St - Sw;
>> [vec , val] = eig(inv(Sw)*Sb);
>> [vec,val]
ans =

    0.93089   -0.74596   11.22001    0.00000
    0.36531    0.66600    0.00000    0.26049

>> 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';
>> zf = xg*W*W';
>> zg = xg*W*W';
>> 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');
>> yg = xg*W;
>> std_green =std(yg);
std_green =    0.64029
>> mean_green = mean(yg);
mean_green =    7.7865
>> tg = [mean_green-4*std_green:0.01:mean_green+4*std_green];
>> figure
>>plot(tb,exp(-(tb-mean_blue).*(tb-mean_blue)/(2*std_blue^2))/std_blue*sqrt(2*pi),
'b-')
>> plot(tr,exp(-(tr-mean_red).*(tr-mean_red)/(2*std_red^2))/std_red*sqrt(2*pi),'r-')
>>plot(tg,exp(-(tg-mean_green).*(tg-mean_green)/(2*std_green^2))/std_green*sq
rt(2*pi),'g-')
>> hold on
>>plot(tb,exp(-(tb-mean_blue).*(tb-mean_blue)/(2*std_blue^2))/std_blue*sqrt(2*pi),
'b-')
>> hold on
>> plot(tr,exp(-(tr-mean_red).*(tr-mean_red)/(2*std_red^2))/std_red*sqrt(2*pi),'r-')
>> hold on
>> title('distribution of data');
>> legend('data blue','data red','data green');
```