信陽師範学院
数学与统计学院
SCHOOL OF MATHEMATICS AND STATISTICS

第12章 MATLAB多元统计分析

讲授人：牛言涛

日期：2020年4月30日

目录

CONTENTS

12.3 判别分析

信阳师范学院
数学与统计学院
SCHOOL OF MATHEMATICS AND STATISTICS

- 在一些自然科学和社会科学的研究中，研究对象用某种方法已划分为若干类型，当得到的一个新样品数据(多元)，要确定该样品属于已知类型中的哪一类，这样的问题属于判别分析。例如在地质找矿中要根据某异常点的地质结构、化探和物探的各项指标来判断该异常点属于哪一种矿化类型；医生要根据某人的各项化验指标的结果来判断该人属于什么病症；调查了某地区的土地生产率、劳动生产率、人均收入、费用水平、农村工业比重等指标，来确定该地区属于哪一种经济类型地区等等。
- 判别分析的基本思想是根据已知类别的样本所提供的信息，总结出分类的规律性，建立判别公式和判别准则，判别新的样本点所属类型。
- 判别分析适用于数据集较小的情况，因为数据量够大的话神经网络的准确率会比传统的判别分析高得多。
- 本节介绍距离判别分析、Bayes判别分析和Fisher判别分析及其MATLAB软件的实现。

12.3 判别分析

信阳师范学院
数学与统计学院
SCHOOL OF MATHEMATICS AND STATISTICS

从统计数据分析的角度，可概括为如下模型：

设有k个总体$G_1, G_2, \cdots, G_k$，它们都是p元总体，其数量指标是$X = (X_1, X_2, \cdots, X_p)^T$

1. 若总体G_i的分布函数已知，对任一新样品数据$x = (x_1, x_2, \cdots, x_p)^T$，判断它来自哪一个总体。

2. 通常各个总体G_i的分布是未知的，由从各个总体取得的样本(训练样本)来估计。一般，先估计各个总体的均值向量与协方差矩阵。

原则：从统计学的角度，要求判别准则在某种准则下是最优的，例如错判的概率最小等。根据不同的判别准则，有不同的判别方法，如距离判别、Bayes判别、Fisher判别等。

1. 距离判别分析常见距离定义

信阳师范学院
数学与统计学院
SCHOOL OF MATHEMATICS AND STATISTICS

1. 闵可夫斯基距离

设有n维向量 $x=(x_1,x_2,\cdots,x_n)^T$, $y=(y_1,y_2,\cdots,y_n)^T$

- 绝对距离 $d_1(x,y)=\sum_{i=1}^{n}|x_i-y_i|$
- 欧式距离 $d_2(x,y)=\sqrt{\sum_{i=1}^{n}(x_i-y_i)^2}$
- 闵科夫斯基距离$d_r(x,y)=\left(\sum_{i=1}^{n}|x_i-y_i|^r\right)^{\frac{1}{r}}$, $r(r>0)$为常数

当$r=2$和1时闵可夫斯基距离分别为欧氏距离和绝对距离。

1. 距离判别分析常见距离定义

2. 马氏距离：由印度统计学家马哈拉诺比斯(PC Mahalanobis)提出，由于马氏距离具有统计意义，在距离判别分析时经常应用。

(1) 同一总体的两个向量之间的马氏距离 $d(x,y)=\sqrt{(x-y)^T\Sigma^{-1}(x-y)}$

其中Σ为总体协方差矩阵，通常取Σ为实对称正定矩阵。显然，当Σ为单位矩阵时马氏距离就是欧氏距离。

(2) 一个向量到一个总体的马氏距离 $d(x,G)=\sqrt{(x-\mu)^T\Sigma^{-1}(x-\mu)}$

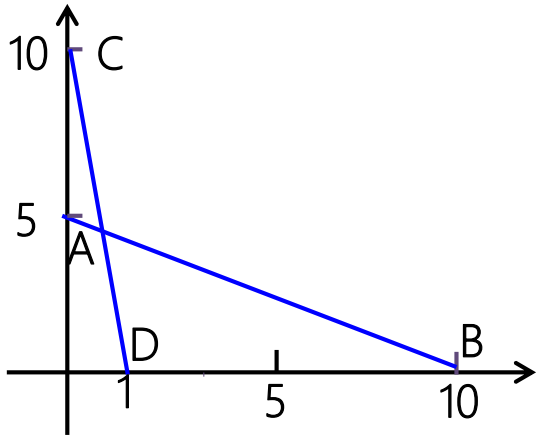
总体G的均值向量为μ，协方差矩阵为Σ，则上式称为n维向量X与总体G的马氏距离。

(3) 两个总体之间的马氏距离

设有两个总体G_1,G_2，两个总体的均值向量分别为μ_1,μ_2，协方差矩阵相等且为Σ，则两个总体之间的马氏距离为 $d(G_1,G_2)=\sqrt{(\mu_1-\mu_2)^T\Sigma^{-1}(\mu_1-\mu_2)}$ 。

1. 距离判别分析常见距离定义

通常，在判别分析时不采用欧氏距离的原因在于，该距离与量纲有关。例如平面上有A, B, C, D四个点，横坐标为代表重量(kg)，纵坐标代表长度(cm)，如图。



这时 $AB=\sqrt{5^2+10^2}=\sqrt{125}$，$CD=\sqrt{10^2+1^2}=\sqrt{101}$

显然 $AB>CD$

如果现在长度用mm为单位，重量的单位保持不变，于是A点的坐标为(0,50)，C点的坐标为(0,100)，此时计算线段的长度为

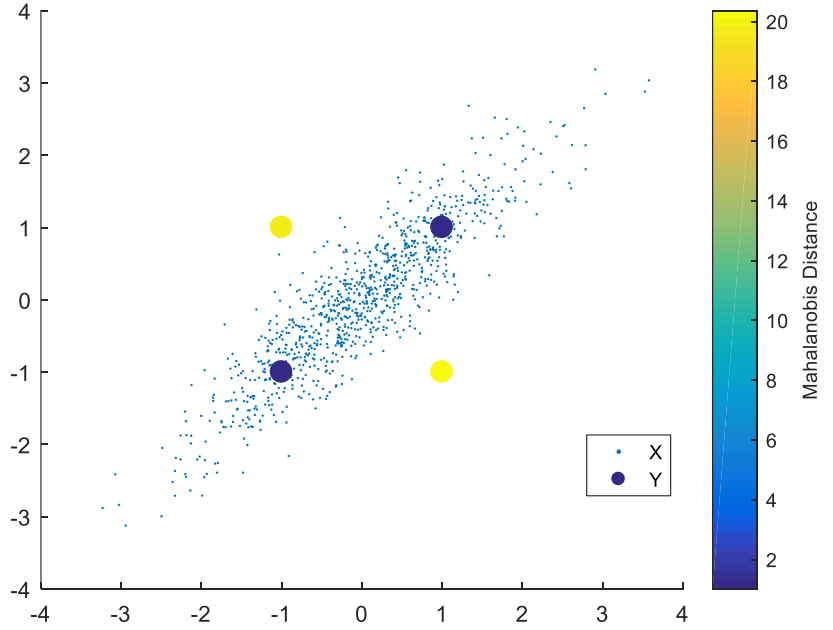
$AB=\sqrt{50^2+10^2}=\sqrt{2600}$，$CD=\sqrt{100^2+1^2}=\sqrt{10001}$，此时，$AB<CD$

这表明欧氏距离有一个缺陷，当向量的分量是不同的量纲时欧氏距离的大小竟然与指标的单位有关，而马氏距离则与量纲无关。

1. 距离判别分析常见距离定义

MATLAB中有一个命令：d=mahal(Y,X)，计算X矩阵每一个点（行）至Y矩阵中每一个点（行）的马氏距离。其中Y的列数必须等于X的列数，但它们的行数可以不同。X的行数必须大于列数。输出d是距离向量。

```
>> rng('default') % For reproducibility
>> X = mvnrnd([0;0],[1 .9;.9 1],1000);
>> Y = [1 1;1 -1;-1 1;-1 -1];
>> d2_mahal = mahal(Y,X)
d2_mahal =
    1.1095   20.3632   19.5939    1.0137
>> d2_Euclidean = sum((Y-mean(X)).^2,2)
d2_Euclidean =
    2.0931    2.0399    1.9625    1.9094
>> scatter(X(:,1),X(:,2),10,'.') % Scatter plot with points of size 10
>> hold on
>> scatter(Y(:,1),Y(:,2),100,d2_mahal,'o','filled')
>> hb = colorbar;
>> ylabel(hb,'Mahalanobis Distance')
>> legend('X','Y','Location','best')
```



2. 两总体的距离判别分析

先考虑两个总体的情况。设G_1, G_2为两个不同的p元已知总体，均值向量分别是μ_1, μ_2，协方差矩阵分别是Σ_1, Σ_2。设$x = (x_1, x_2, \cdots, x_p)^T$是一个待判样品，距离判别准则为

$$\begin{cases} x \in G_1, & d(x, G_1) \le d(x, G_2) \\ x \in G_2, & d(x, G_1) > d(x, G_2) \end{cases}$$

1. 两个总体协方差矩阵相等的情况

考虑样品x到两个总体的马氏距离平方差：$d^2(x, G_2) - d^2(x, G_1) = 2(x - \bar{\mu})^T \Sigma^{-1}(\mu_1 - \mu_2)$

其中 $\bar{\mu} = \frac{1}{2}(\mu_1 + \mu_2)$。令 $W(x) = (x - \bar{\mu})^T \Sigma^{-1}(\mu_1 - \mu_2)$，其中 $\Sigma = \sum_{i=1}^{k} \frac{(n_i - 1) S_i}{n - k}$，$k$为总体数

于是判别准则简化为：$\begin{cases} x \in G_1, & W(x) \ge 0 \\ x \in G_2, & W(x) < 0 \end{cases}$

由于总体的均值、协方差矩阵通常是未知的，数据资料来自两个总体的训练样本，于是用样本的均值、样本的协方差矩阵代替总体的均值与协方差。

案例分析

信阳师范学院
数学与统计学院
SCHOOL OF MATHEMATICS AND STATISTICS

例1： (1989年国际数学竞赛A题)蠓的分类

蠓是一种昆虫，分为很多类型，其中有一种名为Af，是能传播花粉的益虫；另一种名为Apf，是会传播疾病的害虫。这两种类型的蠓在形态上十分相似，很难区别。现测得6只Apf和9只Af蠓虫的触角长度和翅膀长度数据：

Apf：(1.14,1.78), (1.18,1.96), (1.20,1.86), (1.26,2.00), (1.28,2.00), (1.30,1.96)；

Af：(1.24,1.72), (1.36,1.74), (1.38,1.64), (1.38,1.82), (1.38,1.90), (1.40,1.70), (1.48,1.82),(1.54,1.82), (1.56,2.08).

若两类蠓虫协方差矩阵相等，试判别(1.24,1.8), (1.28,1.84), (1.4, 2.04)三个蠓虫属于哪一类?

按照协方差矩阵相等进行判别分析

```
apf = [1.14,1.78;1.18,1.96;1.20,1.86;1.26,2.;1.28,2;1.30,1.96];
af = [1.24,1.72;1.36,1.74;1.38,1.64;1.38,1.82;1.38,1.90;1.40,...
    1.70;1.48,1.82;1.54,1.82;1.56,2.08];
x = [1.24,1.8;1.28,1.84; 1.4,2.04];   % 输入原始待判别数据
m1 = mean(apf);   m2 = mean(af);   %计算样本均值
s1 = cov(apf);    s2 = cov(af);   % 协方差矩阵
%计算总体的协方差矩阵
s = ((length(apf)-1)*s1+(length(af)-1)*s2)/(length(af)+length(apf)-2);
for i=1:length(x)
    % 计算判别函数值
    W(i)=(x(i,:)-1/2*(m1+m2))*inv(s)*(m1-m2)';
end
disp(['类别函数值：',num2str(W)])
```

类别函数值：2.164　　1.3568　　1.9802

```
>> plot(apf(:,1),apf(:,2),'rh')
>> hold on
>> plot(af(:,1),af(:,2),'bp')
>> plot(x(:,1),x(:,2),'ko','MarkerFaceColor','y')
>> apfm = mean(apf);
>> plot(apfm(1),apfm(2),'r*','LineWidth',2)
>> afm = mean(af);
>> plot(afm(1),afm(2),'b*','LineWidth',2)
>> legend('apf','af','待判X','apf均值','af均值')
>> legend('boxoff')
```

由判别准则可知，三只蠓虫均属于Apf.

2. 两总体的距离判别分析

2. 两个总体协方差矩阵不相等的情况

样本x到两个总体的马氏距离平方分别为：

$$d^2(x,G_1)=(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1),\ d^2(x,G_2)=(x-\mu_2)^T\Sigma_2^{-1}(x-\mu_2)$$

令 $W(x)=d^2(x,G_2)-d^2(x,G_1)=(x-\mu_2)^T\Sigma_2^{-1}(x-\mu_2)-(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1)$

则判别准则：$\begin{cases} x\in G_1, & W(x)\geq 0 \\ x\in G_2, & W(x)<0 \end{cases}$

当两个总体的协方差矩阵不等时，可以建立MATLAB的判别法如下：

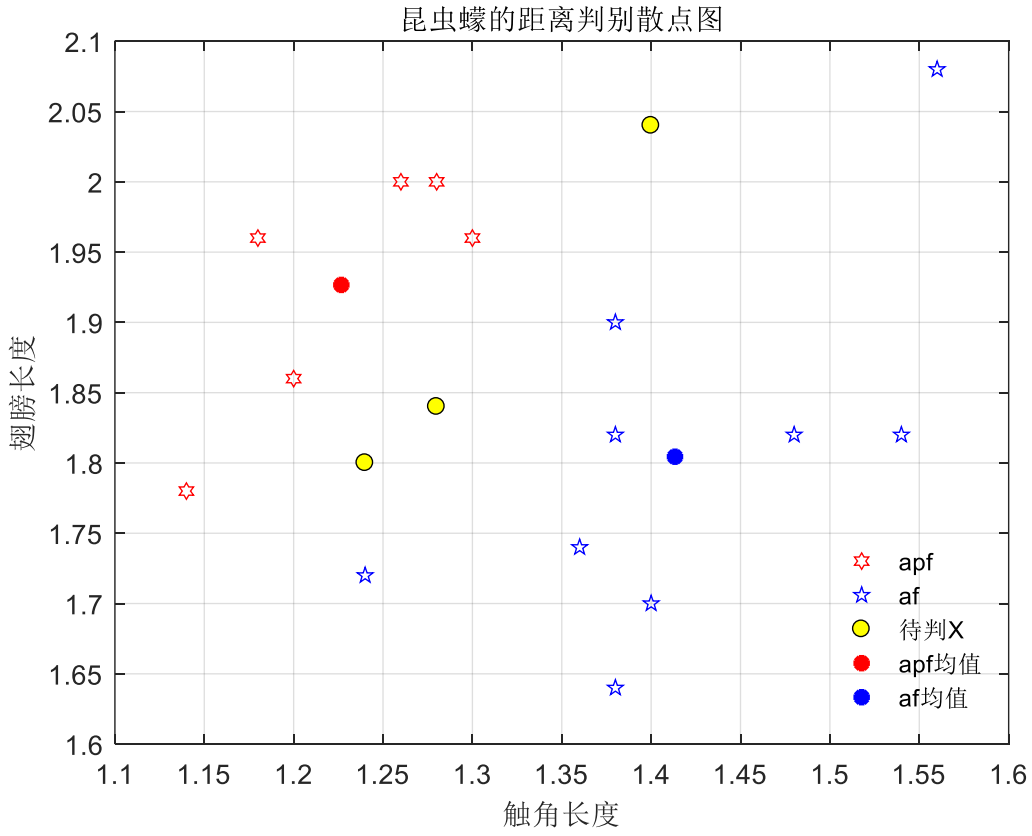
$$\begin{cases} y\in G_1,\ mahal(y,G_1)<mahal(y,G_2) \\ y\in G_2,\ mahal(y,G_1)>mahal(y,G_2) \\ \text{pending judgment},\ mahal(y,G_1)=mahal(y,G_2) \end{cases}$$

按照协方差矩阵不相等进行判别分析

例1： (1989年国际数学竞赛A题)蠓的分类

```
%假设协方差矩阵不相等
apf=[1.14,1.78; 1.18,1.96;1.20,1.86;1.26,2.;1.28,2;1.30,1.96];
af = [1.24,1.72;1.36,1.74;1.38,1.64;1.38,1.82;1.38,1.90;1.40,...
    1.70;1.48,1.82;1.54,1.82;1.56,2.08];
x=[1.24,1.8;1.28,1.84;1.4,2.04];    % 输入原始待判别数据
W = mahal(x,apf) - mahal(x,af)  % 计算判别函数
W =
    1.7611
    3.8812
    3.6468
```

由判别准则可知，三只蠓虫均属于Af.



3. 两个总体协方差矩阵相等的检验

以上两个例题的结果大相径庭，由此我们不禁要问究竟哪个结果更可靠？问题的关键在于：两类蠓虫总体的协方差矩阵是否相等？着手解决协方差矩阵的检验。

$$H_0 : S_i = S \leftrightarrow H_1 : S_i \neq S \ , (i = 1, 2)$$

检验统计量：$Q_i = (n_i - 1)[\ln |S| - \ln |S_i| - p + tr(S^{-1} S_i)] \sim \chi^2(p(p+1)/2) \quad (i = 1, 2)$

```
p=2;   alpha = 0.05;
Q01 = (n1-1)*(log(det(s))-log(det(s1))-p+trace(inv(s)*s1));
Q02 = (n2-1)*(log(det(s))-log(det(s2))-p+trace(inv(s)*s2));
lamda = chi2inv(1-alpha,3);
if Q01 < lamda && Q02 < lamda
    disp('两总体协方差矩阵相同.')
end
```

结果:Q01=2.5784，Q02=0.7418；临界值为：7.8147。

故，两总体协方差矩阵相同，判别结果应该是apf。

拓展：多个总体协方差矩阵相等的检验

多个总体协方差矩阵相等的检验：

- 原假设：H_0：$\Sigma_1 = \Sigma_2 = \cdots = \Sigma_k$
- 备择假设：H_1：$\Sigma_1, \Sigma_2, \cdots \Sigma_k$至少有一对不相等

当原假设成立时，统计量 $\xi = (1-d)M \sim \chi^2(f)$，$f$为自由度。其中

$$d = \begin{cases} \dfrac{2p^2+3p-1}{6(p+1)(k-1)}\left[\sum\limits_{i=1}^{k}\dfrac{1}{n_i-1} - \dfrac{1}{n-k}\right], & n_i\text{不全等} \\ \dfrac{(2p^2+3p-1)(k+1)}{6(p+1)(n-k)}, & n_i\text{全等} \end{cases}$$

$$M = (n-k)\ln|S| - \sum_{i=1}^{k}(n_i-1)\ln|S_i|,\quad S = \sum_{i=1}^{k}\frac{(n_i-1)S_i}{n-k},\quad f = \frac{p(p+1)(k-1)}{2},\quad n = n_1+n_2+\cdots+n_k$$

拓展：多个总体协方差矩阵相等的检验

信阳师范学院
数学与统计学院

COVEqual.m

```m
function covEqu = COVEqual1(G)
% COVEqual1函数用于判断各类别样本协方差是否相等
% 输出参数covEqu为一结构体，包括：
% (1)Class_Num为类别数，(2)Sample_Gnum每个总体样本数，(3)prior_p先验概率，
% (4)Gm每个总体均值，(5)Gcov每个总体协方差矩阵，(6)G每个总体样本集，(7)S总体协方差矩阵
% (8)equal为1是总体协方差相等，0则不全相等，(8)输出判断结果字符串
% 输入参数：G为元胞数组，每一个元胞存储G的一个总体

    %% 样本预处理
    k = size(G,2);  %总体类别数
    n = zeros(1,k); %每个总体样本数
    for i = 1:k
        n(i) = size(G{i},1);  %总体Gi的样本数，对应ni
    end

    %% 基本统计量的计算
    sn = sum(n);  %总体合并的样本数，对应公式中n
    fn = size(G{1},2);  %特征数，对应公式中p
    def = fn*(fn+1)*(k-1)/2;  %自由度
    if length(unique(n)) == 1 %每个总体样本量全等
        d = (2*fn^2+3*fn-1)*(k+1)/(6*(fn+1)*(sn-k));
    else  %每个总体样本量不全相等
        d = (2*fn^2+3*fn-1)*(sum(1./(n-1))-1/(sn-k))/(6*(fn+1)*(k-1));
    end
    prior_p = n./sn; %先验概率
    Gm = cell(k,1);   %计算每个总体均值
    Gcov = cell(k,1);   %计算每个总体协方差矩阵
    for j = 1:k
        Gm{j} = mean(G{j});%计算每个总体均值
        Gcov{j} = cov(G{j});%计算每个总体协方差阵
    end

    %% 多个总体协方差矩阵相等的检验，计算统计量
    S = 0;
    sm = 0;
    for i = 1:k
        S = S + (n(i)-1)*Gcov{i};
        sm = sm + (n(i)-1)*log(det(Gcov{i}));
    end
    S = S/(sn-k);
    M = ((sn-k)*log(det(S))-sm);
    T = (1-d)*M;  %计算统计量观测值
    critical_value = chi2inv(0.95,def); %临界值

    %% 构造输出参数结构体
    covEqu.Class_Num = k;
    covEqu.G = G;
    covEqu.Sample_Gnum = n;
    covEqu.prior_p = prior_p;
    covEqu.Gm = Gm;
    covEqu.Gcov = Gcov;
    covEqu.S = S;
    if T < critical_value
        covEqu.equal = 1;
        covEqu.covstring = '协方差相等';
    else
        covEqu.equal = 0;
        covEqu.covstring = '协方差不全相等';
    end
end
```

4. 多个总体的距离判别

信阳师范学院
数学与统计学院
SCHOOL OF MATHEMATICS AND STATISTICS

1.总体协方差矩阵相等时的判别

设有k个总体$G_1, G_2, \cdots, G_k$，$x_1^{(j)}, x_2^{(j)}, \cdots, x_{n_j}^{(j)}$是取自总体$G_j (j = 1, 2, \ldots, k)$的训练样本，记

$$\bar{x}^{(j)} = \frac{1}{n_j}\sum_{i=1}^{n_j} x_i^{(j)}, (j = 1, 2, \cdots, k) \quad S_j = \frac{1}{n_j - 1}\sum_{i=1}^{n_i}(x_i^{(j)} - \bar{x}^{(j)})(x_i^{(j)} - \bar{x}^{(j)})^T, S = \Sigma = \sum_{j=1}^{k}\frac{(n_j - 1)}{(n-k)}S_j$$

于是未知样品到各总体的判别函数为：$d^2(x, G_j) - d^2(x, G_i) = 2[(x - \frac{1}{2}(\mu_i + \mu_j)]^T \Sigma^{-1}(\mu_i - \mu_j)$

其中 $\hat{W}_{ij}(x) = [(x - \frac{1}{2}(\mu_i + \mu_j)]^T \Sigma^{-1}(\mu_i - \mu_j)$，判别准则为：对所有的$j (j \neq i)$, $\hat{W}_{ij}(x) > 0$，则$x \in G_i$

2.总体协方差矩阵不全相等时的判别

计算样品到各总体马氏距离平方：$d^2(x, G_j) = (x - \bar{x}^{(j)})^T \Sigma_j^{-1}(x - \bar{x}^{(j)}), \quad j = 1, 2, \cdots, k$

记 $d_j^2(x) = d^2(x, G_j), \quad j = 1, 2, \cdots, k$，判别准则为：若 $d_{j_0}^2(x) = \min\limits_{1 \le j \le k} d_j^2(x)$，则判别$x \in G_{j0}$

实际计算时，用训练样本对总体做估计 $d^2(x, G_j) = (x - \bar{x}^{(j)})^T S^{-1}(x - \bar{x}^{(j)}), \quad j = 1, 2, \cdots, k$

4. 多个总体的距离判别

信阳师范学院
数学与统计学院
SCHOOL OF MATHEMATICS AND STATISTICS

Distance_DA.m

```
function [covEqu,dda] = Distance_DA(X,xtest)
% 函数Distance_DA实现距离判别法，并进行总体协方差的判别
% 输入参数：X为样本元胞数组，每一个元胞存储X的一个总体
% 输出参数：dda是一个结构体，待判样本距离各总体距离矩阵；待判样本最终类别
% covEqu为判断协方差矩阵是否相等的各类信息

    %% 1、各总体协方差判断是否相等，调用函数COVEqual
    covEqu = COVEqual(X);
    sample_n = size(xtest,1); %待判定的数据样本量
    mG = zeros(covEqu.Class_Num,size(xtest,2)); %各组样本总体均值
    for i = 1:covEqu.Class_Num
        mG(i,:) = covEqu.Gm{i}; %各总体均值构成一个矩阵，一行代表一个总体
    end
    S = covEqu.S;  %所有样本总体协方差矩阵

    %% 2、根据多总体判别公式，计算距离判别函数并进行判别分析
    class = zeros(1,sample_n);
    DistanceG = cell(1,sample_n);
    % （1）各总体协方差不全相等
    if covEqu.equal == 0
        for i = 1:sample_n  %表示总体所属组
            Jw = zeros(1,covEqu.Class_Num); %存储各总体距离向量
            for j = 1:covEqu.Class_Num %类别数
                %计算待判样本到每个总体的距离
                Jw(j) = (xtest(i,:)- mG(j,:))*(S\(xtest(i,:)-mG(j,:))');
                [~,ind] = min(Jw); %取距离最小的Gj的索引
                class(i) = ind; %类别赋值最小值索引
            end
            DistanceG{i} = Jw;
        end
    % （2）各总体协方差相等
    elseif covEqu.equal == 1
        Jw = zeros(covEqu.Class_Num); %存储各总体距离矩阵
        for i = 1:sample_n  %待判观测样本数
            for Gj = 1:covEqu.Class_Num %类别数
                for Gk = 1:covEqu.Class_Num %类别数
                    %计算待判样本到每两个总体之间的距离
                    Jw(Gj,Gk) = (xtest(i,:)- (mG(Gj,:)+mG(Gk,:))/2)*...
                        (S\(mG(Gj,:)-mG(Gk,:)))';
                    % 由判别规则，不考虑小于0情况。
                    % 此外，JW矩阵是关于对角线对称，但对应元素正负相反
                    if Jw(Gj,Gk) < 0 %跳出循环并对标记赋值为0，即此Gj不考虑
                        sign = 0;  break;
                    else %考虑Gj到其他总体的距离，只要是大于0，标记为1
                        sign = 1;
                    end
                end
                %当Gk循环完毕，如果满足对所有Gj!=Gk，距离都大于0，则归类Gj
                if sign == 1 %根据标记sign，对第i个样本进行判别归类
                    class(i) = Gj;
                    break
                end
            end
            DistanceG{i} = Jw;
        end
    end

    %% 3、输出结构体组合
    dda.DistanceG = DistanceG;
    dda.class = class;
end
```

4. 多个总体的距离判别

信阳师范学院
数学与统计学院
SCHOOL OF MATHEMATICS AND STATISTICS

例1： (1989年国际数学竞赛A题)蠓的分类

```
>> apf=[1.14,1.78; 1.18,1.96;1.20,1.86;1.26,2.;1.28,2;1.30,1.96];
>> af = [1.24,1.72;1.36,1.74;1.38,1.64;1.38,1.82;1.38,1.90;1.40,...
1.70;1.48,1.82;1.54,1.82;1.56,2.08];
>> xtest=[1.24,1.8;1.28,1.84;1.4,2.04];
>> X{1} = apf;
>> X{2} = af;
>> [covEqu,dda] = Distance_DA(X,xtest)
```

两类总体协方差矩阵相等！

类别函数值：1 1 1，即属于apf

```
covEqu =
  包含以下字段的 struct:

      Class_Num: 2
              G: {[6×2 double]  [9×2 double]}
     Sample_Gnum: [6 9]
         prior_p: [0.4000 0.6000]
              Gm: {2×1 cell}
            Gcov: {2×1 cell}
               S: [2×2 double]
           equal: 1
       covstring: '协方差相等'
dda =
  包含以下字段的 struct:

    DistanceG: {[2×2 double]  [2×2 double]  [2×2 double]}
        class: [1 1 1]
```

多个总体的距离判别

例2：为了，研究某种疾病，对一批60人分成3组：G_1, G_2, G_3，同时进行4项指标的检验，β脂蛋白(X1)、甘油三酯(X2)、α脂蛋白(X3)、前β脂蛋白(X4)，检测的结果如表所示。现将3组检验数据为一个总体。对三个待判数(190,67,30,17)，(315,100,35,19)，(240,60,37,18)进行判别归类。

序号	G1				G2				G3			
	X1	X2	X3	X4	X1	X2	X3	X4	X1	X2	X3	X4
1	260	75	40	18	310	122	30	21	320	64	39	17
2	200	72	34	17	310	60	35	18	260	59	37	11
3	240	87	45	18	190	40	27	15	360	88	28	26
4	170	65	39	17	225	65	34	16	295	100	36	12
5	270	110	39	24	170	65	37	16	270	65	32	21
…	…	…	…	…	…	…	…	…	…	…	…	…

多个总体的距离判别

```
>> disease = xlsread('disease.xlsx');
>> X{1} = disease(:,1:4);
>> X{2} = disease(:,5:8);
>> X{3} = disease(:,9:12);
%待判定的数据
>> xtest=[190 67 30 17;315 100 35 19;240 60 37 18];
>> [covEqu,dda] = Distance_DA(X,xtest)
```

三类总体协方差矩阵相等!

类别函数值: 1 3 2

由以上判别准则可知，三个待判数据

(190,67,30,17), (315,100,35,19),

(240,60,37,18）分别属于G1、G3和G2。

```
covEqu =
  包含以下字段的 struct:

      Class_Num: 3
              G: {[20×4 double]  [20×4 double]  [20×4 double]}
     Sample_Gnum: [20 20 20]
         prior_p: [0.3333 0.3333 0.3333]
              Gm: {3×1 cell}
            Gcov: {3×1 cell}
               S: [4×4 double]
          equal: 1
       covstring: '协方差相等'
dda =
  包含以下字段的 struct:

    DistanceG: {[3×3 double]  [3×3 double]  [3×3 double]}
        class: [1 3 2]
```

5. 判别准则的评价

- 当一个判别准则提出以后，还要研究它的优良性，即考察它的误判概率。
- 若属于G_1的样品被误判为属于G_2的个数为N_1个，属于G_2的样品被误判为属于G_1的个数为N_2个，两类总体的样品总数为N，则误判概率p的估计为：$\hat{p}=\dfrac{N_1+N_2}{N}$.
- 针对具体情况，通常采用回代法和交叉法进行误判概率的估计。

（1）回代误判率

– 设G_1,G_2为两个总体，$X_1,X_2,\cdots,X_m$和$Y_1,Y_2,\cdots,Y_n$是分别来自G_1,G_2的训练样本，以全体训练样本作为$m+n$个新样品，逐个代入已建立的判别准则中判别其归属，这个过程称为回判。

– 若属于G_1的样品被误判为属于G_2的个数为N_1个，属于G_2的样品被误判为属于G_1的个数为N_2个，则误判率估计为：$\hat{p}=\dfrac{N_1+N_2}{m+n}$.

5. 判别准则的评价

（2）交叉误判率估计

交叉误判率估计是每次剔除一个样品，利用其余的$m+n-1$个训练样本建立判别准则再用所建立的准则对删除的样品进行判别。对训练样本中每个样品都做如上分析，以其误判的比例作为误判率。步骤：

① 从总体为G_1的训练样本开始，剔除其中一个样品，剩余的$m-1$个样品与G_2中的全部样品建立判别函数；

② 用建立的判别函数对剔除的样品进行判别；

③ 重复步骤①，②，直到G_1中的全部样品依次被删除，又进行判别，其误判个数记为N_1^*；

④ 对G_2的样品重复步骤①，②，③直到G_2中的全部样品依次被删除又进行判别，其误判的样品个数记为N_2^*；

于是交叉误判率估计为：$\hat{p}^* = \dfrac{N_1^* + N_2^*}{m+n}$ 。

5. 判别准则的评价

信阳师范学院
数学与统计学院

Distance_DRule.m

```
function rule = Distance_DRule(covEqu)
% Distance_DRule函数用于计算回代误判率和交叉误判率
% 返回rule结构体，包含判别类别和误判率

    n = sum(covEqu.Sample_Gnum); %所有总体的样本数和
    Gc = covEqu.Class_Num;  %类别数，及总体数

    %% 1、回代误判
    k = 1;
    ClassReturn = zeros(n,1);
    for i = 1:Gc
        Gi = covEqu.G{i}; % 每次取一个总体
        Gin = size(Gi,1); % 每个总体的样本数
        for j = 1:Gin
            [~,dda] = Distance_DA(covEqu.G,Gi(j,:)); %每个样本回代判别
            ClassReturn(k) = dda.class; %存储所属总体类别
            k = k + 1;
        end
    end

    %% 2、交叉误判
    classCorss = zeros(n,1);
    k = 1;
    for i = 1:Gc  %每次取一个总体
        Gi = covEqu.G{i};
        Gin = size(Gi,1);
        for j = 1:Gin %取该总体Gi的每一个样本
            covEqu.G{i} = Gi([1:i-1,i+1:Gin],:); %每次删除一个样本
            %每删除一个样本，进行判别分析该样本所属总体
            [~,dda] = Distance_DA(covEqu.G,Gi(j,:));
            classCorss(k) = dda.class;
            k = k + 1;
        end
    end

    %% 3、分别计算回代误判率和交叉误判率
    Gn = [0,covEqu.Sample_Gnum]; %每个总体的样本两向量
    resR = zeros(Gc,1);
    resC = zeros(Gc,1);
    for k = 1:Gc
        sind = sum(Gn(1:k)) + 1; %每个总体的起始行索引
        eind = sum(Gn(2:k+1)); %每个总体的最终行索引
        resR(k) = sum(ClassReturn(sind:eind) == k);
        resC(k) = sum(classCorss(sind:eind) == k);
    end
    ErrorRateOfReturn = 1 - sum(resR)/n; %回代误判率
    ErrorRateOfCross = 1 - sum(resC)/n; %交叉误判率

    %% 4、输出变量结构体组合
    rule.ClassReturn = ClassReturn;
    rule.ErrorRateOfReturn = ErrorRateOfReturn;
    rule.classCorss = classCorss;
    rule.ErrorRateOfCross = ErrorRateOfCross;
end
```

案例分析

例3：根据表的数据，判别两类总体的协方差是否相等，然后用马氏距离判别未知地区的类别，并计算回代误判率与交叉误判率。

类别	农	林	牧	渔	类别	农	林	牧	渔	类别	农	林	牧	渔
1	503.1	21.8	332.3	188.5	1	439.9	39.4	292.3	101.2	2	254	8.6	80.9	1.1
1	405.9	11.3	236.4	5.8	1	769.9	50.9	605	41	2	28.9	1.8	32.5	0.1
1	450.6	15.7	224.6	20.1	2	89.7	9.5	105.2	9.6	2	49.4	3.5	30.3	2.1
1	529.5	73.7	195.9	308.8	2	86.7	1.5	60.8	20.6	2	348.8	10.1	134	3.9
1	688	66.2	371.6	132.3	2	95.5	3.5	88.4	40.1	2	899.4	34	685.9	61.2
1	433.2	82.3	215.5	330.5	2	191.3	12.3	96.3	1.7	2	1142.7	30.8	448.5	334.2
1	405.9	54	226.1	104.3	2	307.6	26.1	216.2	6	x	431.3	47.2	210.6	14.4
1	658.3	27.1	352.6	134.8	2	141.3	43.3	58.2	82.3	x	1401.3	47.2	654.7	350.7
1	665.7	51.9	480.3	85.2	2	250.4	11.2	154.4	15.2	x	1331.6	57	693.8	20.4
1	817.9	56.8	423.2	390.1	2	337.4	23.6	114.1	3.8	x	279.9	15.1	118.5	5.1

判别准则的评价

```
nlmy = xlsread('nlmydata.xlsx');
X{1}= nlmy(1:12,2:end);
X{2}= nlmy(13:26,2:end);
xtest = nlmy(27:30,2:end);
[covEqu,dda] = Distance_DA(X,xtest)
rule = Distance_DRule(covEqu)
```

两总体的协方差矩阵相同！

待判样本判别结果分别是：G1,G1,G1,G2

回代误判率：0.1923

交叉误判率：0.2308

```
covEqu =
  包含以下字段的 struct:

      Class_Num: 2
              G: {[12×4 double]  [14×4 double]}
    Sample_Gnum: [12 14]
        prior_p: [0.4615 0.5385]
             Gm: {2×1 cell}
           Gcov: {2×1 cell}
              S: [4×4 double]
         equal: 1
      covstring: '协方差相等'
dda =
  包含以下字段的 struct:

    DistanceG: {[2×2 double]  [2×2 double]  [2×2 double]  [2×2 double]}
        class: [1 1 1 2]
rule =
  包含以下字段的 struct:

         ClassReturn: [26×1 double]
    ErrorRateOfReturn: 0.1923
          classCorss: [26×1 double]
     ErrorRateOfCross: 0.2308
```

判别准则的评价

例4：为了研究某地区人口死亡状况，已按某种方法将15个已知样品分为三类，指标及原始数据见下表，试建立判别函数并判定另外4个待判样品分别属于哪类。其中X1：0岁死亡率，X2：1岁死亡率，X3：10岁死亡率，X4：55岁死亡率，X5：80岁死亡率，X6：平均预期寿命。

组别	序号	X1	X2	X3	X4	X5	X6	组别	序号	X1	X2	X3	X4	X5	X6
第一组	1	34.16	7.44	1.12	7.87	95.19	69.30	第三组	1	34.03	5.41	0.07	5.20	90.10	69.50
	2	33.06	6.34	1.08	6.77	94.08	69.70		2	32.11	3.02	0.09	3.14	85.15	70.80
	3	32.26	9.24	1.04	8.97	97.30	68.80		3	44.12	15.12	1.08	15.15	103.12	64.80
	4	40.17	13.45	1.43	13.88	101.20	66.20		4	54.17	25.03	2.11	25.15	110.14	63.70
	5	50.06	23.03	2.83	23.74	112.52	63.30		5	28.07	2.01	0.07	3.02	81.22	68.30
第二组	1	33.24	6.24	1.18	22.90	160.01	65.40	待判样本	1	50.22	6.66	1.08	22.54	170.60	65.20
	2	32.22	4.22	1.06	20.70	124.70	68.70		2	34.64	7.33	1.11	7.78	95.16	69.30
	3	41.15	10.08	2.32	32.84	172.06	65.85		3	33.42	6.22	1.12	22.95	160.31	68.30
	4	53.04	25.74	4.06	34.87	152.03	63.50		4	44.02	15.36	1.07	16.45	105.30	64.20
	5	38.03	11.20	6.07	27.84	146.32	66.80								

判别准则的评价

```
>> dead = xlsread('deaddata.xlsx');
>> X{1} = dead(1:5,2:end);
>> X{2} = dead(6:10,2:end);
>> X{3} = dead(11:15,2:end);
>> xtest = dead(16:end,2:end);
>> [covEqu,dda] = Distance_DA(X,xtest)
>> rule = Distance_DRule(covEqu)
```

从结果得出：

各总体协方差不全相等；

待判样本分别输入总体3、1、2、3

回代误判率为0；

交叉误判率为0.0667.

```
covEqu =
  包含以下字段的 struct:

      Class_Num: 3
              G: {[5×6 double]  [5×6 double]  [5×6 double]}
    Sample_Gnum: [5 5 5]
        prior_p: [0.3333 0.3333 0.3333]
             Gm: {3×1 cell}
           Gcov: {3×1 cell}
              S: [6×6 double]
          equal: 0
      covstring: '协方差不全相等'
dda =
  包含以下字段的 struct:

    DistanceG: {[180.9387 244.9086 140.4527]  [2.2539 71.4917 3.8808]  [88
        class: [3 1 2 3]
rule =
  包含以下字段的 struct:

         ClassReturn: [15×1 double]
    ErrorRateOfReturn: 0
          classCorss: [15×1 double]
     ErrorRateOfCross: 0.0667
```

6. MATLAB自带函数classify

- MATLAB统计工具箱中提供了classify函数，用来对未知类别的样本进行判别，可以进行距离判别和先验分布为正态分布的贝叶斯判别。
- class = classify(sample, training, group)：
  - 输入参数sample是待判别的样本数据矩阵，training是用于构造判别函数的训练样本数据矩阵，它们的每一行对应一个观测，每一列对应一个变量，sample和training具有相同的列数。
  - 参数group是与training相应的分组变量，group和training具有相同的行数，group中每一个元素指定了training中相应观测所在的组。group可以是一个分类变量（categorical variable，即用水平表示分组）、数值向量、字符串数组或字符串元胞数组。输出参数class是一个行向量，用来指定sample中各观测所在的组，class与group具有相同的数据类型。
  - classif函数把group中的NAN或空字符作为缺失数据，从而忽略training中相应的观测。

6. MATLAB自带函数classify

- class = classify(sample, training, group,type)：允许用户通过type参数指定判别函数的类型，type的可能取值如下所示。
  - linear —线性判别函数（默认情况）。假定$G_i \sim N_p(\mu_i, \Sigma), i = 1, 2, \cdots, k$，即各组的先验分布均为协方差矩阵相同的$p$元正态分布，此时由样本得出协方差矩阵的联合估计$\hat{\Sigma}$；
  - diaglinear — 与'linear'类似，此时用一个对角矩阵作为协方差矩阵的估计；
  - quadratic — 二次判别函数，假定各组的先验分布均为p元正态分布，但是协方差矩阵并不完全相同，此时分别给出各个协方差矩阵的估计$\hat{\Sigma}_i$ ，$i = 1, 2, \cdots, k$；
  - diagquadratic —与quadratic类似，此时用一个对角矩阵作为协方差矩阵的估计；
  - mahalanobis — 各组的协方差矩阵不全相等并未知时的距离判别，此时分别得出各组的协方差矩阵的估计。

6. MATLAB自带函数classify

- class = classify(sample, training, group,type,prior)：允许用户通过参prior数指定各组的先验概率，默认各组先验概率相等。prior由三种类型数据：
  - 一个元素全为正数的数值向量，向量的长度等于group中所包含的组的个数，prior中元素的顺序应与group中各组出现的顺序一致。prior中各元素chuyu5其所有元素之和即为各组的先验概率。
  - 一个结构体变量，包括两个字段：prob和group，其中prob是元素全为正数的数值向量，group为分组变量（不含重复行，即不含多余的分组信息），prob用来指定group中各组的先验概率。
  - 字符串empirical，根据training和group计算各组出现的频率，作为各组先验概率的估计。
- [class,err] = classify(sample, training, group,type,prior)：返回基于training数据的误判概率的估计值err。

6. MATLAB自带函数classify


```
% 例2 疾病判别分析，协方差矩阵相等
>> disease = xlsread('disease.xlsx');
>> sample = [190 67 30 17;315 100 35 19;240 60 37 18];
>> training = [disease(:,1:4);disease(:,5:8);disease(:,9:12)];
>> group = [ones(20,1);2*ones(20,1);3*ones(20,1)];
>> [class,err] = classify(sample,training,group)
class =
     1
     3
     2
err =
    0.4833
```

```
% 例3 农林牧副渔判别分析，协方差矩阵相等
>> nlmy = xlsread('nlmydata.xlsx');
>> sample = nlmy(27:end,2:end);
>> training = nlmy(1:26,2:end);
>> group = nlmy(1:26,1);
>> [class,err] = classify(sample,training,group)
class =
     1
     1
     1
     2
err =
    0.1905
```

6. MATLAB自带函数classify

信阳师范学院
数学与统计学院
SCHOOL OF MATHEMATICS AND STATISTICS

```
% 例4 死亡状况地区判别分析，协方差矩阵不全相等
>> dead = xlsread('deaddata.xlsx');
>> sample = dead(16:end,2:end);
>> training =[dead(1:5,2:end);dead(6:10,2:end);dead(11:15,2:end)];
>> group = [ones(5,1);2*ones(5,1);3*ones(5,1)];
>> [class,err] = classify(sample,training,group)
class =
     3
     1
     2
     3
err =
     0
```

通过三个案例可知，自编函数与MATLAB自带函数classify判别结果一致，且回代误判率基本一致。

```
covEqu =
  包含以下字段的 struct:

      Class_Num: 3
              G: {[5×6 double]  [5×6 double]  [5×6 double]}
    Sample_Gnum: [5 5 5]
        prior_p: [0.3333 0.3333 0.3333]
             Gm: {3×1 cell}
           Gcov: {3×1 cell}
              S: [6×6 double]
          equal: 0
      covstring: '协方差不全相等'
dda =
  包含以下字段的 struct:

    DistanceG: {[180.9387 244.9086 140.4527]  [2.2539 71.4917 3.8808]  [88
        class: [3 1 2 3]
rule =
  包含以下字段的 struct:

         ClassReturn: [15×1 double]
    ErrorRateOfReturn: 0
           classCorss: [15×1 double]
     ErrorRateOfCross: 0.0667
```

二. 贝叶斯判别法

- 距离判别只要求知道总体的数字特征，不涉及总体的分布函数，当参数和协方差未知时，就用样本的均值和协方差矩阵来估计。距离判别方法简单实用，但没有考虑到每个总体出现的机会大小，即先验概率，没有考虑到错判的损失。贝叶斯判别法正是为了解决这两个问题提出的判别分析方法。
- Bayes判别：许多时候用户对各类别的比例分布情况有一定的先验信息，也就是用样本所属分类的先验概率进行分析。比如客户对投递广告的反应绝大多数都是无回音，如果进行判别，自然也应当是无回音的居多。此时，Bayes判别恰好适用。
- Bayes判别就是根据总体的先验概率，使误判的平均损失达到最小而进行的判别。其最大优势是可以用于多组判别问题。但是适用此方法必须满足三个假设条件，即各种变量必须服从多元正态分布、各组协方差矩阵必须相等、各组变量均值均有显著性差异。
- 贝叶斯公式 $P(B_i \mid A) = \dfrac{P(A \mid B_i)P(B_i)}{\sum_j P(A \mid B_j)P(B_j)}$

1. 两个正态总体的Bayes判别

一般讨论，考虑两个p元总体G_1, G_2分别具有概率密度函数$f_1(x), f_2(x)$，设出现的先验概率为：

$$p_1 = P(G_1), \quad p_2 = P(G_2) \text{ ，且 } p_1 + p_2 = 1$$

当取得新样品 $x = (x_1, x_2, \cdots, x_p)^T$后，根据Bayes公式的后验概率分别为

$$P(G_1 \mid x) = \frac{p_1 f_1(x)}{p_1 f_1(x) + p_2 f_2(x)}, \quad P(G_2 \mid x) = \frac{p_2 f_2(x)}{p_1 f_1(x) + p_2 f_2(x)}$$

两个总体的Bayes判别准则为

$$\begin{cases} x \in G_1, \ P(G_1 \mid x) \geq P(G_2 \mid x) \Leftrightarrow p_1 f_1(x) \geq p_2 f_2(x) \\ x \in G_2, \ P(G_1 \mid x) < P(G_2 \mid x) \Leftrightarrow p_1 f_1(x) < p_2 f_2(x) \end{cases}$$

1. 两个正态总体的Bayes判别

信阳师范学院
数学与统计学院
SCHOOL OF MATHEMATICS AND STATISTICS

（1）两个总体协方差矩阵相等的情形

设总体G_1, G_2的协方差矩阵相等且为Σ，概率密度函数为：

$$f_j(x)=\frac{1}{(2\pi)^{\frac{p}{2}}|\Sigma|^{\frac{1}{2}}}\exp\left\{-\frac{1}{2}(x-\mu_j)^{\mathrm{T}}\Sigma^{-1}(x-\mu_j)\right\},\ j=1,2$$

马氏平方距离

$$p_j f_j(x)=\exp\left\{\ln p_j+\ln f_j(x)\right\}=\exp\left\{-\frac{1}{2}(x-\mu_j)^T\Sigma_j^{-1}(x-\mu_j)+\ln|\Sigma_j|-2\ln p_j\right\}\Big/(2\pi)^{\frac{p}{2}}=\exp\left\{-\frac{1}{2}d_j^2(x)\right\}\Big/(2\pi)^{\frac{p}{2}}$$

看大小　　大小相反

后验概率最大原则

基于两正态总体后验概率的Bayes判别准则为

$$\begin{cases}x\in G_1,\ d_1^2(x)\le d_2^2(x)\Leftrightarrow P(G_1\mid x)\ge P(G_2\mid x)\\ x\in G_2,\ d_1^2(x)>d_2^2(x)\Leftrightarrow P(G_1\mid x)<P(G_2\mid x)\end{cases}$$

其中$d_j^2(x)=(x-\mu_j)^T\Sigma^{-1}(x-\mu_j)+\ln|\Sigma|-2\ln p_j,(j=1,2)$为广义平方距离。

1. 两个正态总体的Bayes判别

（1）两个总体协方差矩阵相等的情形

记 $d_2^2(x)-d_1^2(x)=2\left(w_1(x)-w_2(x)\right)$

$C(1|2)=C(2|1)$

损失相等的Bayes判别准则 $\begin{cases} x\in G_1, & w_1(x)\geq w_2(x) \\ x\in G_2, & w_1(x)<w_2(x) \end{cases}$，$w_j(x)=\mu_j^{T}\Sigma^{-1}x-\frac{1}{2}\mu_j^{T}\Sigma^{-1}\mu_j+\ln p_j,\ j=1,2$

在实际问题中，关于先验概率p_1,p_2，通常用下列两种方式选取:

1）采用等概率选取，即 $p_1=p_2=0.5$

2）按训练样本的容量$n_1,n_2\cdots,n_k$的比例选取，即 $p_1=\frac{n_1}{n_1+n_2}$，　$p_2=\frac{n_2}{n_1+n_2}$

平均损失率 $L(R)=p_1l(1,R)+p_2l(2,R)=C(2|1)p_1P(2|1,R)+C(1|2)p_2P(1|2,R)$

1. 两个正态总体的Bayes判别

（2）两个总体协方差矩阵不相等的情形

设总体的协方差矩阵不相等分别为Σ_1, Σ_2概率密度函数为：

$$f_j(x)=\frac{1}{(2\pi)^{\frac{p}{2}}|\Sigma_j|^{\frac{1}{2}}}\exp\left\{-\frac{1}{2}(x-\mu_j)^{\mathrm{T}}\Sigma_j^{-1}(x-\mu_j)\right\}, j=1,2$$

则基于两正态总体误判损失相等的Bayes判别准则$\begin{cases} x\in G_1,\ d_1^2(x)\le d_2^2(x) \\ x\in G_2,\ d_1^2(x)>d_2^2(x) \end{cases}$

其中 $d_j^2(x)=(x-\mu_j)^T\Sigma_j^{-1}(x-\mu_j)-\ln|\Sigma_j|-2\ln p_j, \quad j=1,2$

2. 多个总体的Bayes判别

一般讨论，设有k个总体$G_1, G_2, \cdots, G_k$的概率密度为$f_j(x)$各总体出现的先验概率为

$$P_j = P\left(G_j\right), j = 1, 2, \ldots, k, \quad s.t. \sum_{j=1}^{k} p_j = 1$$

当出现样品x时，总体G_i的后验概率 $P(G_i \mid x) = \dfrac{p_i f_i(x)}{\sum_{j=1}^{k} p_j f_j(x)}$

Bayes判别准则为：若 $P(G_i \mid x) = \max\limits_{1 \le j \le k}\{P(G_j \mid x)\}(i = 1, 2, \cdots, k)$，则判样本 $x \in G_i$。

注：当达到最大后验概率的G_i不止一个时，可判为达到最大后验概率的总体的任何一个.

2. 多个总体的Bayes判别

（1）当$\Sigma_1 = \Sigma_2 = \cdots = \Sigma_k = \Sigma$时，设$G_j \sim N_p(\mu_j, \Sigma),\ j = 1,2,\cdots,k$

线性判别函数为$W_j(x) = a_j^T x + b_j$，其中$a_j^T = \mu_j{}^T \Sigma^{-1}, b_j = -\frac{1}{2}\mu_j{}^T \Sigma^{-1}\mu_j + \ln p_j, j = 1,2,\cdots,k$

基于误判损失相等的Bayes判别准则为$x \in G_i$，若$W_i(x) = \max\limits_{1\le j\le k}\{W_j(x)\}$

基于后验概率的Bayes判别准则为$x \in G_i$，若$d_i^2(x) = \min\limits_{1\le j\le k}\{d_j^2(x)\}$

其中$d_j^2(x) = (x-\mu_j)^T \Sigma^{-1}(x-\mu_j) - 2\ln p_j, j = 1,2,\cdots,k$

在实际问题中，由于$\mu_1, \mu_2, \cdots, \mu_k$及$\Sigma$未知，各总体的训练样本均值$\bar{x}^{(1)}, \bar{x}^{(2)}, \cdots, \bar{x}^{(k)}$及$S$估计。

（2）当$\Sigma_1, \Sigma_2, \cdots, \Sigma_k$不全相等时，设$G_j \sim N_p(\mu_j, \Sigma_j), j = 1,2,\cdots,k,$

则基于后验概率的Bayes判别准则为$x \in G_i$，若$d_i^2(x) = \min\limits_{1\le j\le k}\{d_j^2(x)\}$

其中$d_j^2(x) = (x-\mu_j)^T \Sigma_j^{-1}(x-\mu_j) + \ln\left|\Sigma_j\right| - 2\ln p_j$

多个总体的Bayes判别

信阳师范学院
数学与统计学院
SCHOOL OF MATHEMATICS AND STATISTICS

Bayes_DA.m

```
function [covEqu,bda] = Bayes_DA(X,xtest)
% 函数Bayes_DA实现多个总体的贝叶斯判别法，并进行总体协方差的判别
% 输入参数：X为样本元胞数组，每一个元胞存储X的一个总体
% 输出参数：bda是一个结构体，待判样本距离各总体距离矩阵；待判样本最终类别
% covEqu为判断协方差矩阵是否相等的各类信息

    %% 1、各总体协方差判断是否相等，调用函数COVEqual
    covEqu = COVEqual(X);
    sample_n = size(xtest,1); %待判定的数据样本量
    mG = zeros(covEqu.Class_Num,size(xtest,2)); %各组样本总体均值
    for i = 1:covEqu.Class_Num
        mG(i,:) = covEqu.Gm{i}; %各总体均值构成一个矩阵，一行代表一个总体
    end
    S = covEqu.S;  %所有样本总体协方差矩阵
    p = covEqu.prior_p;  %先验概率

    %% 2、根据多总体判别公式，计算贝叶斯判别函数并进行判别分析
    class = zeros(1,sample_n);
    DistanceG = cell(1,sample_n);
    % （1）各总体协方差不全相等
    if covEqu.equal == 0
        for i = 1:sample_n  %表示总体所属组
            Jw = zeros(1,covEqu.Class_Num); %存储各总体距离向量
            for j = 1:covEqu.Class_Num %类别数
                %计算待判样本到每个总体的距离
                Sj = covEqu.Gcov{j}; %每个总体的协方差矩阵
                Jw(j) = (xtest(i,:)- mG(j,:))*(Sj\(xtest(i,:)-mG(j,:))') - ...
                    log(det(Sj)) - 2*log(p(j));
                [~,ind] = min(Jw); %取距离最小的Gj的索引
                class(i) = ind; %类别赋值最小值索引
            end
            DistanceG{i} = Jw;
        end

    % （2）各总体协方差相等
    elseif covEqu.equal == 1
        Jw = zeros(1,covEqu.Class_Num); %存储各总体距离矩阵
        for i = 1:sample_n  %待判观测样本数
            for j = 1:covEqu.Class_Num %类别数
                Jw(j)=mG(j,:)*(S\xtest(i,:)') - ...
                    1/2*mG(j,:)*(S\mG(j,:)')+log(p(j));
            end
            [~,ind] = max(Jw); %取最大值索引
            class(i) = ind; %归属类别
            DistanceG{i} = Jw;
        end
    end

    %% 3、输出结构体组合
    bda.DistanceG = DistanceG;
    bda.class = class;
end
```

案例分析

例4：对破产的企业收集它们在破产前两年的年度财务数据，对财务良好的企业也收集同一时间的数据.数据涉及四个变量：X1现金流量/总债务，X2净收益/总资产，X3流动资产/流动债务，以及X4流动资产/净销售额，数据如表所示．假定两总体G1,G2均服从四元正态分布，在误判损失相等且先验概率按比例分配的条件下，对待判样本进行bayes判别.

G1(破产企业)				G2(非破产企业)				待判数据			
-0.45	-0.41	1.09	0.45	0.51	0.1	2.49	0.54	-0.23	-0.3	0.33	0.18
-0.56	-0.31	1.51	0.16	0.08	0.02	2.01	0.53	0.15	0.05	2.17	0.55
0.06	0.02	1.01	0.4	0.38	0.11	3.27	0.35	-0.28	-0.23	1.19	0.66
-0.07	-0.09	1.45	0.26	0.19	0.05	2.25	0.33	0.48	0.09	1.24	0.18
-0.1	-0.09	1.56	0.67	0.32	0.07	4.24	0.63				
-0.14	-0.07	0.71	0.28	0.12	0.05	2.52	0.69				
0.04	0.01	1.5	0.71	-0.02	0.02	2.05	0.35				
-0.06	-0.06	1.37	0.4	0.22	0.08	2.35	0.4				
-0.13	-0.14	1.42	0.44	0.17	0.07	1.8	0.52				

案例分析

```
fin = xlsread('Finance.xlsx');
X{1} = fin(:,1:4);
X{2} = fin(:,5:8);
xtest = fin(1:4,9:12);
[covEqu,bda] = Bayes_DA(X,xtest) %协方差不全相等
```

```
covEqu =
  包含以下字段的 struct:

      Class_Num: 2
              G: {[9×4 double]  [9×4 double]}
    Sample_Gnum: [9 9]
        prior_p: [0.5000 0.5000]
             Gm: {2×1 cell}
           Gcov: {2×1 cell}
              S: [4×4 double]
         equal: 0
      covstring: '协方差不全相等'
bda =
  包含以下字段的 struct:

    DistanceG: {[36.6485 496.6586]  [31.4092 19.3846]  [22.2618 246.0833]  [73.5471 37.8813]}
        class: [1 2 1 2]
```

判别结果：

第1个属于破产企业

第2个属于非破产企业

第3个属于破产企业

第4个属于非破产企业

%回代误判率和交叉误判率仍采用距离判别分析中自定义函数，只是其中调用函数从[~,dda] = Distance_DA(covEqu.G,Gi(j,:))修改成[~,dda] = Bayes_DA(covEqu.G,Gi(j,:))即可。

```
>> rule = Bayes_DARule(covEqu)
```

```
>> rule = Bayes_DARule(covEqu)
rule =
  包含以下字段的 struct:

        ClassReturn: [18×1 double]
  ErrorRateOfReturn: 0
          classCorss: [18×1 double]
   ErrorRateOfCross: 0
```

案例分析

例：某医院利用心电图检测来对人群进行划分，数据见表. “g=1”表示健康人，“g=2”表示主动脉硬化患者，“g=3”表示冠心病患者， $X1, X2$ 表示测得的心电图中表明心脏功能的两项不相关的指标。某受试者心电图该两项指标的数据分别为380.20，9.08. 设先验概率按比例分配，进行bayes判别，判定其归属.

编号	X_1	X_2	类别	编号	X_1	X_2	类别	编号	X_1	X_2	类别
1	261.01	7.36	1	9	273.84	8.79	1	17	274.57	9.67	2
2	185.39	5.99	1	10	303.59	8.53	1	18	409.42	10.49	2
3	249.58	6.11	1	11	231.03	6.15	1	19	330.34	9.61	3
4	137.13	4.35	1	12	308.90	8.49	2	20	331.47	13.72	3
5	231.34	8.79	1	13	258.69	7.16	2	21	352.50	11.00	3
6	231.38	8.53	1	14	355.54	9.43	2	22	347.31	11.19	3
7	260.25	10.02	1	15	476.69	11.32	2	23	189.59	5.46	3
8	259.51	9.79	1	16	316.12	8.17	2	24	380.20	9.08	待判

多个总体的Bayes判别

```
heart = xlsread('heart.xlsx');
heart = [heart(:,1:4);heart(:,5:8)];
X{1} = heart(1:11,2:3);
X{2} = heart(12:18,2:3);
X{3} = heart(19:23,2:3);
xtest = heart(24,2:3);
[covEqu,bda] = Bayes_DA(X,xtest)
rule = Bayes_DARule(covEqu)
rule =
  包含以下字段的 struct:
          ClassReturn: [23×1 double]
    ErrorRateOfReturn: 0.2174
           classCorss: [23×1 double]
     ErrorRateOfCross: 0.2609
```

```
covEqu =
  包含以下字段的 struct:

      Class_Num: 3
              G: {[11×2 double]  [7×2 double]  [5×2 double]}
    Sample_Gnum: [11 7 5]
        prior_p: [0.4783 0.3043 0.2174]
             Gm: {3×1 cell}
           Gcov: {3×1 cell}
              S: [2×2 double]
         equal: 1
      covstring: '协方差相等'
bda =
  包含以下字段的 struct:

    DistanceG: {[14.6537 17.6026 14.8508]}
        class: 2
```

案例分析

信阳师范学院
数学与统计学院

例6：2008年全国部分地区城镇居民人均年家收入情况见表。按四种指标分为二类，用bayes判别判定青海、广东两省区属于哪一类，并用回代法和交叉法对误判率进行估计(假定误判损失相等).

地区	工薪收入	经营净收入	财产性收入	转移性收入	类别	地区	工薪收入	经营净收入	财产性收入	转移性收入	类别
北京	18738.96	778.36	452.75	7707.87	1	河南	9043.52	1161.96	156.46	3545.86	3
上海	21791.11	1399.14	369.12	6199.77	1	湖北	9474.81	1114.68	244.13	3340.65	3
天津	12849.73	863.52	256.87	7203.93	2	湖南	9070.97	1575.08	316.48	3614.74	3
江苏	12319.86	1999.61	307.31	5548.78	2	重庆	10957.62	788.26	205.94	3265.92	3
浙江	15538.83	3161.87	1324.94	4955.14	2	宁夏	8793.54	1856.94	182.67	3285.49	3
福建	12668.82	2185.13	952.91	3879.29	2	广西	10321.2	1314.4	441.15	3316.44	3
山东	12940.62	1194.4	346.9	3067.05	2	四川	9117	1040.14	262.9	3265.06	3
西藏	12314.69	303.34	138.08	891.42	2	贵州	7811.16	770.86	110.9	3492.7	3
河北	8891.5	1078.67	224.86	3946.39	3	云南	8596.88	1165.96	849.45	3505.74	3
山西	9019.35	983.21	202.31	3654.11	3	陕西	9794.82	544	151.46	3356.85	3
内蒙古	10284.43	1555.31	324.64	3031.05	3	甘肃	8354.63	638.76	65.33	2610.61	3
辽宁	9494.59	1483.3	248.04	4610.32	3	新疆	9422.22	938.15	141.75	1976.49	3
黑龙江	7393.39	1241.37	122.83	3506.48	3	青海	8595.48	763.07	50.17	3458.63	x
安徽	9302.38	959.43	293.92	3603.72	3	广东	15188.39	2405.92	701.25	3382.95	x
江西	9105.96	1106.31	265.35	2985.96	3						

案例分析

```
ffd = xlsread('familyfin.xlsx');
ffd = [ffd(:,1:4);ffd(:,7:10)];
X{1} = ffd(1:2,:);
X{2} = ffd(3:8,:);
X{3} = ffd(9:27,:);
xtest = ffd(28:29,:);
[covEqu,bda] = Bayes_DA(X,xtest) %协方差相等
rule = Bayes_DARule(covEqu)
```

```
covEqu =
  包含以下字段的 struct:

      Class_Num: 3
              G: {[2×4 double]  [6×4 double]  [19×4 double]}
    Sample_Gnum: [2 6 19]
        prior_p: [0.0741 0.2222 0.7037]
             Gm: {3×1 cell}
           Gcov: {3×1 cell}
              S: [4×4 double]
          equal: 1
      covstring: '协方差相等'
bda =
  包含以下字段的 struct:

    DistanceG: {[-40.7889 38.2650 49.4098]  [86.4216 121.1960 107.7104]}
        class: [3 2]
rule =
  包含以下字段的 struct:

         ClassReturn: [27×1 double]
    ErrorRateOfReturn: 0
          classCorss: [27×1 double]
     ErrorRateOfCross: 0.0741
```

3. Bayes判别平均误判率

Byaes判别的有效性可以通过平均误判率来确定。这里仅对两个正态总体G_1, G_2且协方差矩阵相等的情况下研究平均误判率的计算。

设总体 $G_i \sim N_p(\mu_j, \Sigma)(i=1,2)$ ，其先验概率$p_1 = P(G_1), p_2 = P(G_2)$，两个总体$G_1, G_2$的马氏平方距离记为 $\delta = (\mu_1 - \mu_2)^T \Sigma^{-1} (\mu_1 - \mu_2)$

则基于误判损失相等时的平均误判率为

$$p^* = P(2|1)p_1 + P(1|2)p_2 = p_1 \Phi\left(\frac{d - \frac{\delta}{2}}{\sqrt{\delta}}\right) + p_2\left[1 - \Phi\left(\frac{d + \frac{\delta}{2}}{\sqrt{\delta}}\right)\right]$$ ，其中 $d = \ln p_1 - \ln p_2$

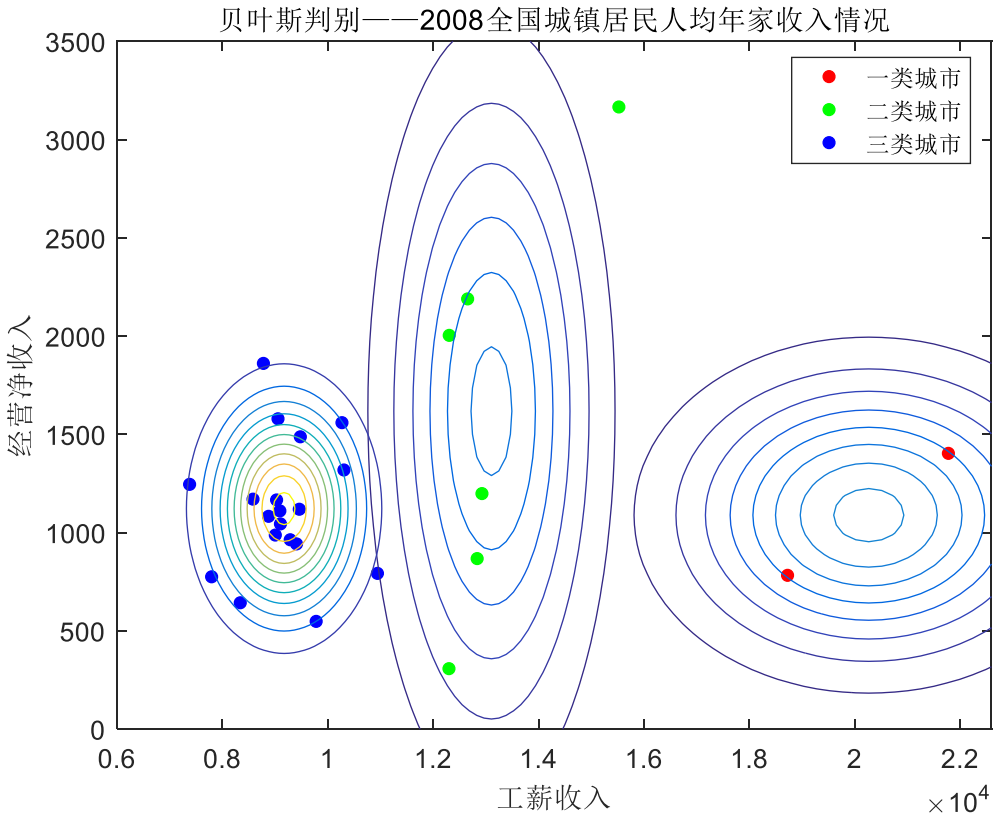
从上式知，当总体G_1, G_2的马氏平方距离δ越大，即两总体的分离程度越大时，平均误判概率最小。推广到一般情况也成立。

4. 贝叶斯判别分析MATLAB工具箱

1. Mdl = fitcnb(Tbl,ResponseVarName) returns a multiclass naive Bayes model (Mdl), trained by the predictors in table Tbl and class labels in the variable Tbl.ResponseVarName.
2. Mdl = fitcnb(Tbl,formula) returns a Mdl, trained by the predictors in table Tbl. formula is an explanatory model of the response and a subset of predictor variables in Tbl used to fit Mdl.
3. Mdl = fitcnb(Tbl,Y) returns a Mdl, trained by the predictors in the table Tbl and class labels in the array Y.
4. Mdl = fitcnb(X,Y) returns a Mdl, trained by predictors X and class labels Y.
5. Mdl = fitcnb(___,Name,Value) returns a naive Bayes classifier with additional options specified by one or more Name,Value pair arguments, using any of the previous syntaxes. For example, you can specify a distribution to model the data, prior probabilities for the classes, or the kernel smoothing window bandwidth.

案例分析

```
ffd = xlsread('familyfin.xlsx');
ffd = [ffd(1:15,1:5);ffd(1:12,7:11)];
X = ffd(:,1:2);  %取工薪与经营两列数据
Y = ffd(:,5);  %取分类数据
Mdl = fitcnb(X,Y,'ClassNames',{'1','2','3'})
plotcnb(Mdl)  %自定义函数
title('贝叶斯判别——2008全国城镇居民人均年家收入情况')
xlabel('工薪收入')
ylabel('经营净收入')
legend('一类城市','二类城市','三类城市')
```



绘制分类数据高斯等高线

信阳师范学院
数学与统计学院
SCHOOL OF MATHEMATICS AND STATISTICS

该函数仅适用于训练样本变量（指标）个数为2的总体。

plotcnb.m

```
function plotcnb(model)
    n = length(model.ClassNames);  %类别数
    X = model.X; %提取模型的原始特征数据
    Y = model.Y; %提取模型的原始分类数据
    gscatter(X(:,1),X(:,2),Y);    %按分类或分组来画离散点,适用于画多个类别的离散样本分布图
    h = gca; cxlim = h.XLim; cylim = h.YLim;
    hold on
    Params = cell2mat(model.DistributionParameters);
    Mu = Params(2*(1:n)-1,1:2); % Extract the means
    Sigma = zeros(2,2,n);
    for j = 1:n
        Sigma(:,:,j) = diag(Params(2*j,:)).^2;       % Create diagonal covariance matrix
        xlim = Mu(j,1) + 4*[-1 1]*sqrt(Sigma(1,1,j));
        ylim = Mu(j,2) + 4*[-1 1]*sqrt(Sigma(2,2,j));
        f = @(x1,x2)reshape(mvnpdf([x1(:),x2(:)],Mu(j,:),Sigma(:,:,j)),size(x1));
        fcontour(f,[xlim ylim]) % Draw contours for the multivariate normal distributions
    end
    h.XLim = cxlim;
    h.YLim = cylim;
end
```

案例分析

```
ffd = xlsread('familyfin.xlsx');
ffd = [ffd(1:15,1:5);ffd(1:14,7:11)];
X = ffd(1:27,1:4);
Y = ffd(1:27,5);  %取分类数据
testdata = ffd(28:29,1:4);%待判样品
Mdl = fitcnb(X,Y,'DistributionNames','norm')
predata = Mdl.predict(testdata)    %预测待判样本
predata =
     3   2
preclass = Mdl.predict(X);  %回代预测  sLabels1 = resubPredict(Mdl);
cfm = confusionmat(Y,preclass)  %构造混淆矩阵
cfm =
     2     0     0
     0     6     0
     0     0    19
```

```
>> cvm = crossval(Mdl);  %交叉验证
>> loss = kfoldLoss(cvm)  %交叉验证误判率
loss =
    0.0476
```

结果分析：青海属于三类城市，广东属于二类城市；混淆矩阵看出判别全部正确；泛化误差为0.0476.

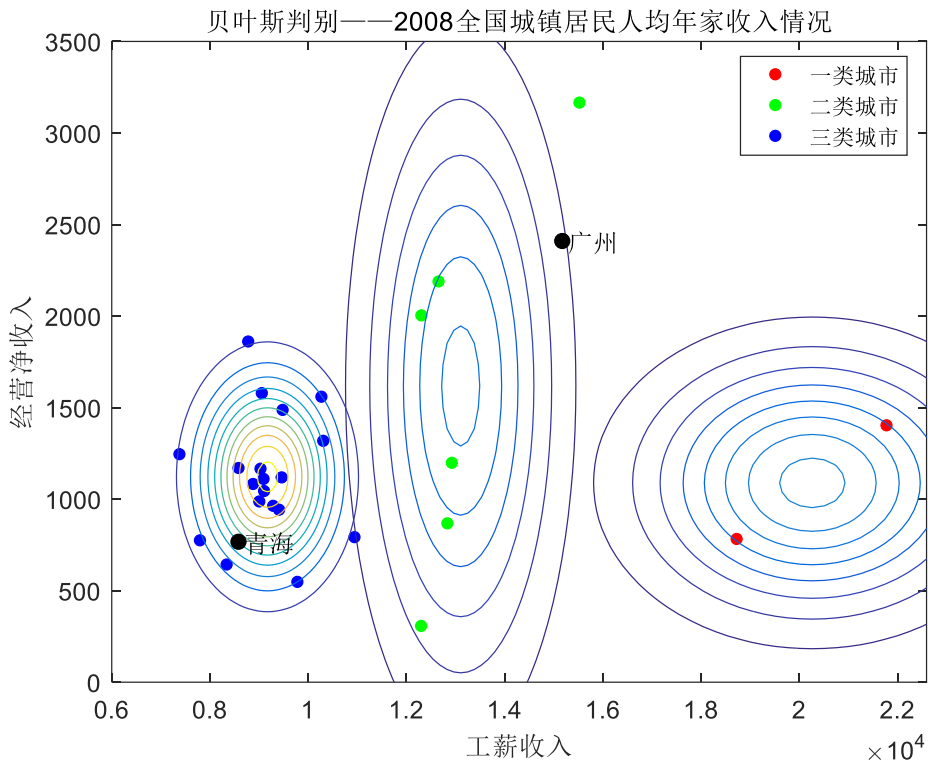
最新版本的matlab可以使用如下函数绘制混淆矩阵图

```
isLabels = resubPredict(Mdl);
ConfusionMat = confusionchart(Y,isLabels);
```

案例分析

```
>> plot(testdata(:,1),testdata(:,2),'ko','MarkerFaceColor','k')
>> text(testdata(1,1)+100,testdata(1,2),'青海')
>> text(testdata(2,1)+100,testdata(2,2),'广州')
```

从图中也可以判别青海与广州分别属于三类城市和二类城市。



训练一个朴素的贝叶斯分类器

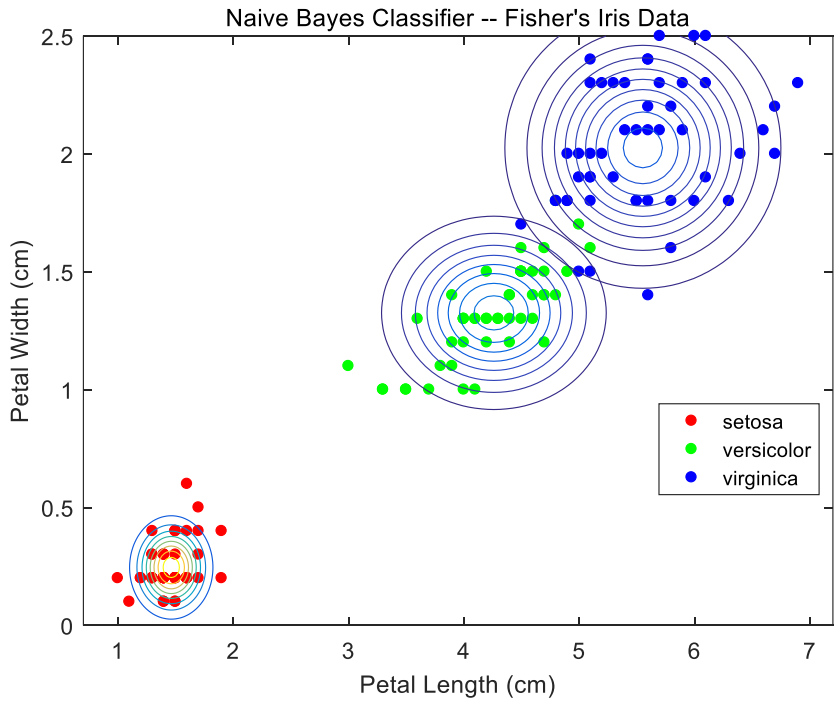
信阳师范学院
数学与统计学院
SCHOOL OF MATHEMATICS AND STATISTICS

```
load fisheriris    %加载matlab自带数据集
X = meas(:,3:4);  %取后两列数据作为数据特征，绘制分类离散图
Y = species; %类别，三类：'setosa','versicolor','virginica'
tabulate(Y) %类别频数表
%%训练Bayes分类模型. 指定分类顺序.
Mdl = fitcnb(X,Y,'ClassNames',{'setosa','versicolor','virginica'})
```

%默认情况下, 软件使用具有一定平均值和标准偏差的高斯分布对每个类中的预测器分布进行建模。使用点表示法显示特定高斯拟合的参数。例如, 显示setosa中第一个特征的拟合.

```
setosaIndex = strcmp(Mdl.ClassNames,'setosa');
estimates = Mdl.DistributionParameters{setosaIndex,1};  %平均值为1.4620, 标准偏差为0.1737.
```

```
plotcnb(Mdl)
title('Naive Bayes Classifier -- Fisher''s Iris Data')
xlabel('Petal Length (cm)');    ylabel('Petal Width (cm)')
legend('setosa','versicolor','virginica')
hold off
```



指定先验概率

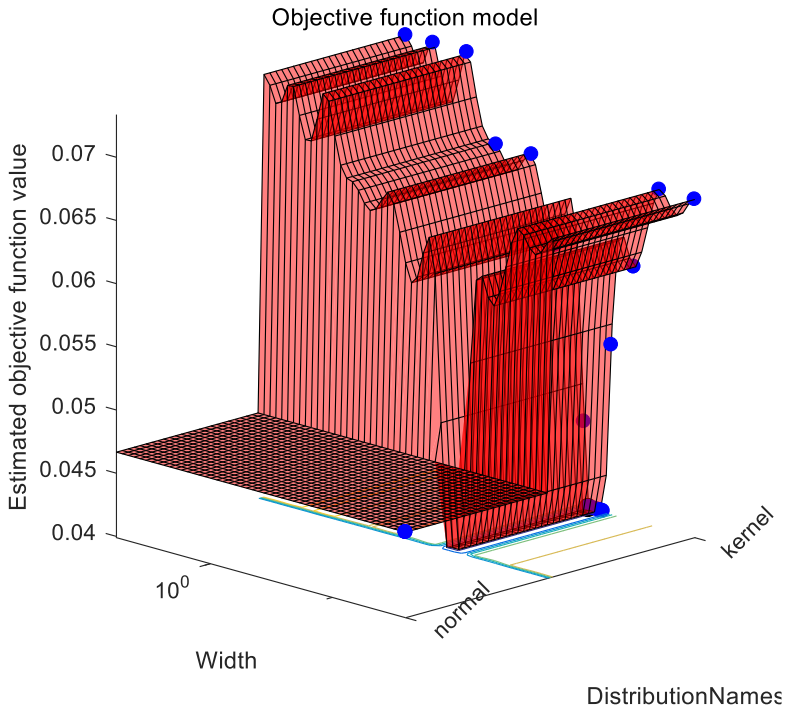
```
load fisheriris
X = meas;    Y = species;
classNames = {'setosa','versicolor','virginica'};    % Class order
prior = [0.5 0.2 0.3];   %指定类顺序和先验概率分布
Mdl = fitcnb(X,Y,'ClassNames',classNames,'Prior',prior)

defaultPriorMdl = Mdl;
FreqDist = cell2table(tabulate(Y));
defaultPriorMdl.Prior = FreqDist{:,3};  %每类都是33.33%
defaultCVMdl = crossval(defaultPriorMdl);
defaultLoss = kfoldLoss(defaultCVMdl)   %默认先验概率泛化误差
CVMdl = crossval(Mdl);
Loss = kfoldLoss(CVMdl)   %指定先验概率泛化误差

isLabels1 = resubPredict(Mdl);
ConfusionMat1 = confusionmat(Y,isLabels1)   %混淆矩阵
```

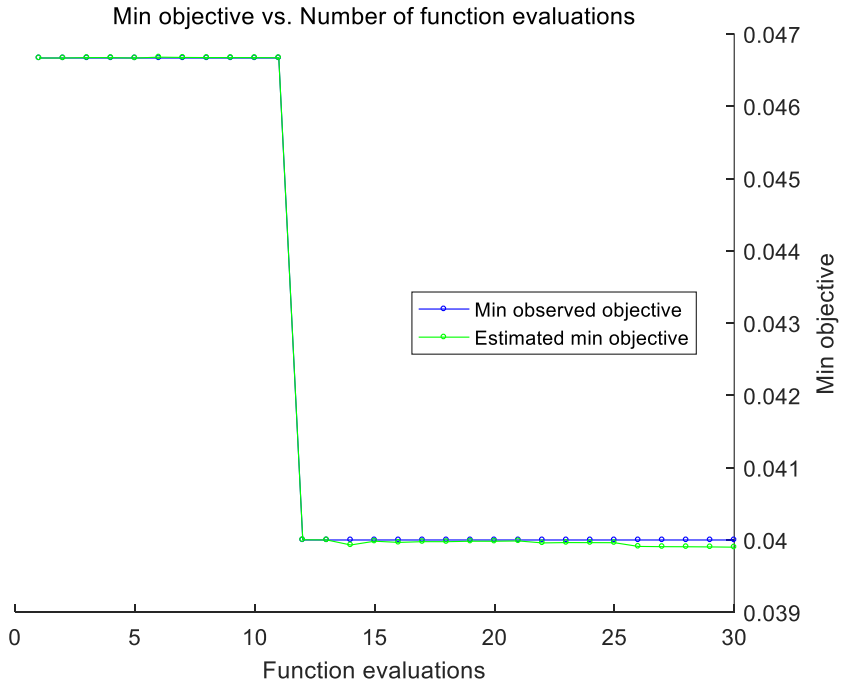
```
defaultLoss =
    0.0533
Loss =
    0.0340
% 指定先验概率泛化误差更低
ConfusionMat1 =
    50     0     0
     0    47     3
     0     3    47
```

优化贝叶斯判别模型

```
load fisheriris
X = meas;  Y = species;
classNames = {'setosa','versicolor','virginica'};
Mdl = fitcnb(X,Y,'ClassNames',classNames,'OptimizeHyperparameters','auto',...
    'HyperparameterOptimizationOptions',struct('AcquisitionFunctionName',...
    'expected-improvement-plus'))
```

使用Optimize Hyper parameters（超参数）名称-值对来最小化朴素Bayes分类器中的交叉验证损失。





利用交叉验证对比分类器

信阳师范学院
数学与统计学院
SCHOOL OF MATHEMATICS AND STATISTICS

```
load fisheriris
X = meas;
Y = species;
rng(1); % For reproducibility
%使用默认选项和k-折叠交叉验证训练和交叉验证一个朴素的贝叶斯分类器
CVMdl1 = fitcnb(X,Y,'ClassNames',{'setosa','versicolor','virginica'},'CrossVal','on');

t = templateNaiveBayes(); %创建一个默认的NaiveBayes二进制分类器模板
%cvmdl2是一个分类分区ECOC（纠错输出码）模型。可以使用与fitcnb相同的名称-值对。
CVMdl2 = fitcecoc(X,Y,'CrossVal','on','Learners',t);  %支持向量机多分类模型

classErr1 = kfoldLoss(CVMdl1,'LossFun','ClassifErr')
classErr2 = kfoldLoss(CVMdl2,'LossFun','ClassifErr')
```

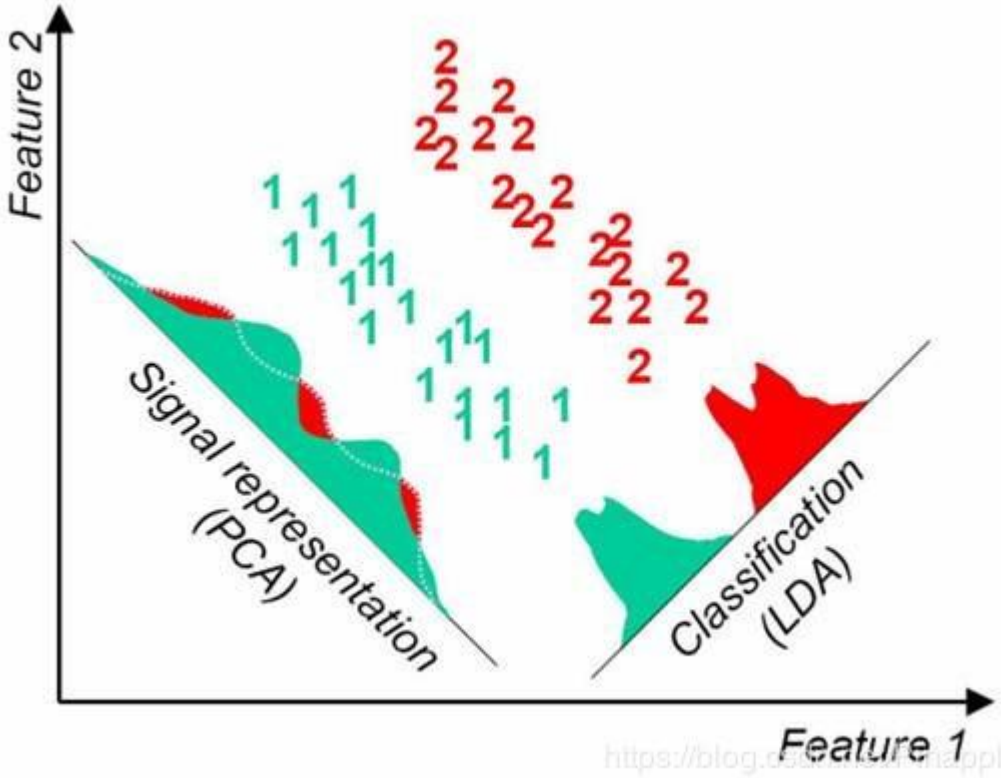
```
classErr1 =

    0.0533

classErr2 =

    0.0467
```

三. Fisher判别分析

将高维度空间的样本投影到低维空间上，使得投影后的样本数据在新的子空间上有最小的类内距离以及最大的类间距离，使得在该子空间上有最佳的可分离性。

Fisher判别分析和PCA差别：两个方法是从不同的角度来降维。

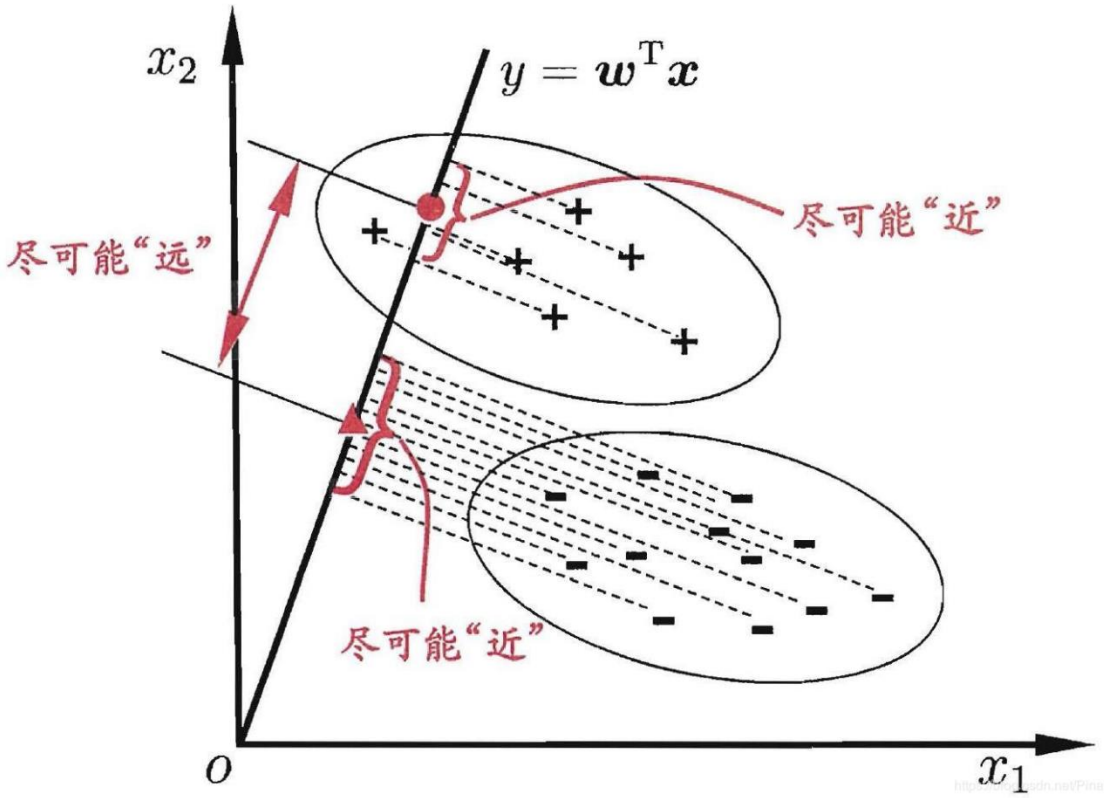
- PCA是找到方差尽可能大的维度，使得信息尽可能都保存，不考虑样本的可分离性，不具备预测功能。
- LAD(线性判别分析) 是找到一个低维的空间，投影后，使得可分离性最佳，投影后可进行判别以及对新的样本进行预测。



1. Fisher判别分析思想简介

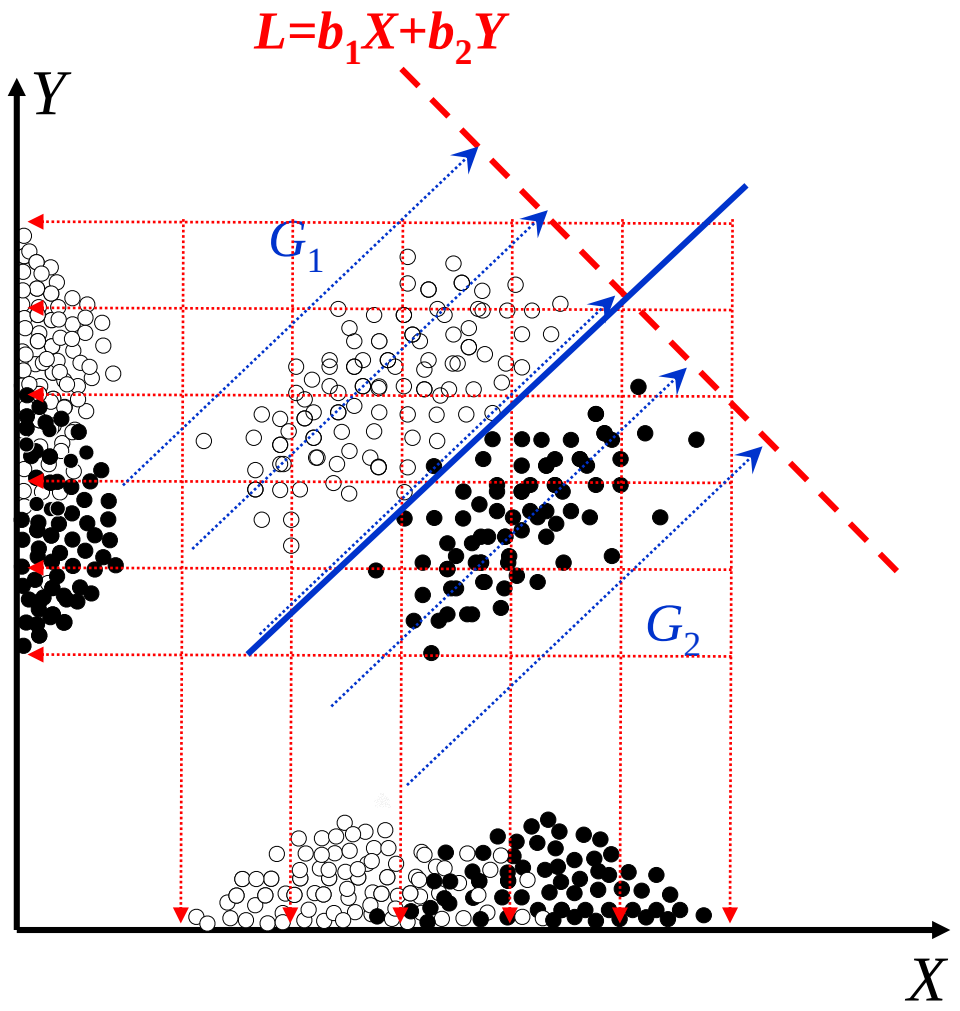
Fisher判别分析是要实现有最大的类间距离，以及最小的类内距离。

- Fisher判别分析的思想非常朴素：投影降维，给定训练样例集，设法将样例投影到一条直线上，使得同类样例的投影点尽可能接近、不同类样例的投影点尽可能远离。在对新样本进行分类时，将其投影到同样的这条直线上，再根据新样本投影点的位置来确定它的类别。
- 二维示意图中的"+"、"-"分别代表正例和反例，椭圆表示数据簇的外轮廓，虚线表示投影，红色实心圆和红色实心三角形分别表示两类样本投影后的中心点。



1. Fisher判别分析思想简介

- 所谓Fisher判别法，就是一种先投影的方法。Fisher准则的基本原理，就是要找到一个最合适的投影轴，使两类样本在该轴上投影的交迭部分最少，从而使分类效果为最佳。
- 考虑只有两个（预测）变量的判别分析问题。假定只有两种已知类型的训练样本，数据中的每个观测值是二维空间的一个点。其中一类有38个点（用“o”表示），另一类有44个点（用“*”表示）。按照原来的变量（横坐标和纵坐标），很难将这两种点分开。
- 于是就寻找一个方向，也就是图上的虚线方向，沿着这个方向朝和这个虚线垂直的一条直线进行投影会使得这两类分得最清楚。可以看出，如果向其他方向投影，判别效果不会比这个好。
- 有了投影之后，再用距离远近的方法来得到判别准则。



2. Fisher判别分析原理分析

对x_n的分量做线性组合可得标量 $y_n = w^T x_n$，$n = 1, 2, \cdots, N_i$，这样便得到N个一维样本y_n组成的集合。从而将多维转换到了一维。考虑把d维空间中的数据点投影到一条直线上去的问题，需要解决的两个问题：(1)怎样找到最好的投影直线方向；(2)怎样向这个方向实现投影，这个投影变换就是要寻求的解向量w^*。这两个问题就是Fisher方法要解决的基本问题。

1. 样本在d维X空间

(1) 各类样本均值向量m_i：$m_i = \frac{1}{n_i} \sum_{x_k \in X_i} x_k$，$i = 1, 2$

(2)样本类内离散度矩阵S_i与总类内离散度矩阵S_w：$S_i = \sum_{x \in \Gamma_i} (x - m_i)(x - m_i)^T$ $i = 1, 2$，$S_w = S_1 + S_2$

(3) 样本类间离散度矩阵S_b：$S_b = (m_1 - m_2)(m_1 - m_2)^T$

其中，S_w是对称半正定矩阵，而且当$N > d$时通常是非奇异的。S_b也是对称半正定矩阵，在两类情况下，它的秩最大等于1。

2. Fisher判别分析原理分析

信阳师范学院
数学与统计学院
SCHOOL OF MATHEMATICS AND STATISTICS

2.样本在一维Y空间

(1) 各类样本均值 $\tilde{m}_i = \frac{1}{N_i}\sum_{y\in Y_i} y,\ \ i=1,2$

(2)样本类内离散度、总类内离散度和类间离散度

$$\tilde{S}_i = \sum_{y\in Y_i}\left(y-\tilde{m}_i\right)^2 \quad i=1,2;\ \ \tilde{S}_w = \tilde{S}_1+\tilde{S}_2;\ \ \tilde{S}_b = \left(\tilde{m}_1-\tilde{m}_2\right)^2$$

定义Fisher准则函数：根据Fisher选择投影方向w的原则，使原样本向量在该方向上的投影能兼顾：

类间分布尽可能分开，类内样本投影尽可能密集。

用以评价投影方向w的函数为：$J_F(w) = \frac{\tilde{S}_b}{\tilde{S}_1+\tilde{S}_2} = \frac{\left(\tilde{m}_1-\tilde{m}_2\right)^2}{\tilde{S}_1+\tilde{S}_2}$

这个函数称为Fisher准则函数。应该寻找使分子尽可能大，分母尽可能小的w作为投影向量。

2. Fisher判别分析原理分析

将 $J_F(w)$ 变成w的显函数：

$$\tilde{m}_i = \frac{1}{N_i}\sum_{y\in Y_i} y = \frac{1}{N_i}\sum_{y\in Y_i} w^T x = w^T m_i, \quad i = 1,2$$

$$\tilde{S}_b = \left(\tilde{m}_1 - \tilde{m}_2\right)^2 = \left(w^T m_1 - w^T m_2\right)^2 = w^T\left(m_1 - m_2\right)\left(m_1 - m_2\right)^T w = w^T S_b w$$

$$\tilde{S}_i = \sum_{y\in Y_i}\left(y - \tilde{m}_i\right)^2 = \sum_{x\in X_i}\left(w^T x - w^T m_i\right)^2 = w^T\left[\sum_{x\in X_i}\left(x - m_i\right)\left(x - m_i\right)^T\right]w = w^T S_i w \quad i = 1,2;$$

$$\tilde{S}_1 + \tilde{S}_2 = w^T\left(S_1 + S_2\right)w = w^T S_w w$$

得出最终表达式（即广义Rayleigh）：$J_F(w) = \dfrac{\tilde{S}_b}{\tilde{S}_1 + \tilde{S}_2} = \dfrac{w^T S_b w}{w^T S_w w}$

广义瑞利熵Rayleigh的性质：$\min_w = -\dfrac{1}{2} w^T S_b w, \quad s.t.\ w^T S_w w = C$

2. Fisher判别分析原理分析

信阳师范学院
数学与统计学院
SCHOOL OF MATHEMATICS AND STATISTICS

最佳W值的确定：

- 最佳w值的确定实际上就是对Fisher准则函数求取其达极大值时的w^*。
- 对于这个问题可以采用拉格朗日乘子算法解决。由于分子分母是关于w的二次型，因此$J_F(w)$解w与的长度无关。保持分母为一非零常数c的条件下，求其分子项的极大值。

$$J_F(w)=\frac{w^T S_b w}{w^T S_w w} \qquad 令 \quad w^T S_w w = c \neq 0$$

定义Lagrange函数：$L(w,\lambda)=w^T S_b w-\lambda\left(w^T S_w w-c\right)$

对拉格朗日函数分别对w求偏导并置为0来求w的解。令

$$\frac{\partial L(w,\lambda)}{\partial w}=S_b w-\lambda S_w w=0 \Rightarrow S_b w^*=\lambda S_w w^* \Rightarrow S_w^{-1} S_b w^*=\lambda w^*$$

这是一个求矩阵$S_w^{-1}S_b$的特征值问题，即对应最大特征向量。

2. Fisher判别分析原理分析

信阳师范学院
数学与统计学院
SCHOOL OF MATHEMATICS AND STATISTICS

最佳W值的确定：

$$S_w^{-1}S_b w^* = \lambda w^* \qquad S_b = (m_1 - m_2)(m_1 - m_2)^T$$

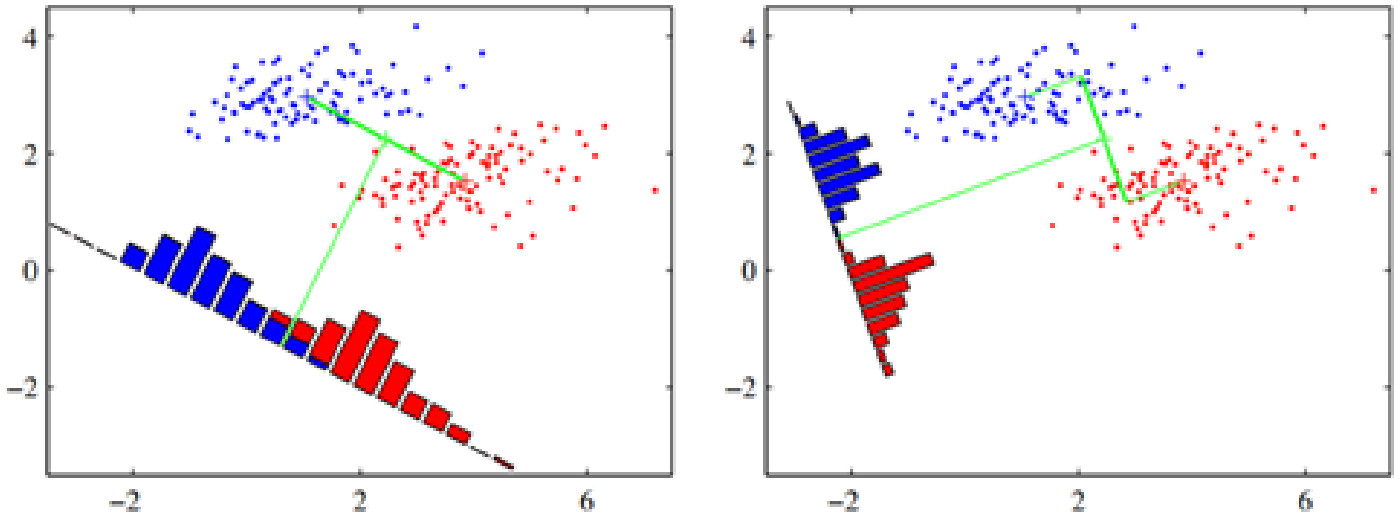
$$\lambda w^* = S_w^{-1}S_b w^* = S_w^{-1}(m_1 - m_2)\underbrace{(m_1 - m_2)^T w^*}_{\text{数值R}} = S_w^{-1}(m_1 - m_2)R \Rightarrow w^* = \frac{R}{\lambda}S_w^{-1}(m_1 - m_2)$$

实际上我们关心的只是向量w^*的方向，其数值大小对分类器没有影响。因此在忽略了数值因子R/λ后，可得：$w^* = S_w^{-1}(m_1 - m_2)$，该式就是使用Fisher准则求最佳法线向量的解。

向量w^*就是使Fisher准则函数$J_F(w)$达极大值的解，也就是按Fisher准则将d维X空间投影到一维Y空间的最佳投影方向，该向量w^*的各分量值是对原d维特征向量求加权和的权值。

2. Fisher判别分析原理分析

最佳投影方向的理解：$w^* = S_w^{-1}(m_1 - m_2)$



这幅图左边就是单纯的最大化类间距离，所以左图中w和 (m1−m2)是平行的。右图是Fisher线性判别式得到的w。

- $(m_1 - m_2)$是一向量，显然从两类均值在变换后距离最远这一点看，对与$(m_1 - m_2)$平行的向量投影可使两均值点的距离最远。
- 但是如从使类间分得较开，同时又使类内密集程度较高这样一个综合指标来看，则需根据两类样本的分布离散程度对投影方向作相应的调整，这就体现在对 $(m_1 - m_2)$ 向量按S_w^{-1}作一线性变换，从而使Fisher准则函数达到极值点。

2. Fisher判别分析原理分析

信阳师范学院
数学与统计学院
SCHOOL OF MATHEMATICS AND STATISTICS

判别函数的确定：

（1）当维数d与样本数N都很大时，可采用贝叶斯决策规则，获得一种在一维空间的“最优”分类器。

（2）当上述条件不满足时，一般可采用以下几种方法确定分界阈值点y_0：

$$(1)\quad y_0=-\frac{\tilde{m}_1+\tilde{m}_2}{2};\qquad (2)\quad y_0=-\frac{N_1\tilde{m}_1+N_2\tilde{m}_2}{N_1+N_2}=\tilde{m}$$

当$P(w_1)$与$P(w_2)$已知时可用 $(3)\quad y_0=\frac{\tilde{m}_1+\tilde{m}_2}{2}-\frac{\ln\left(P(w_1)/P(w_2)\right)}{N_1+N_2-2}$

2. Fisher判别分析原理分析

决策规则：(1)式中只考虑采用均值连线中点作为阈值点，相当于贝叶斯决策中先验概率相等的情况。其中而(2)与(3)则是以不同方式考虑$P(w_1)$与$P(w_2)$不等的影响，以减小先验概率不等时的错误率。

当y_0确定之后，则可按以下规则分类：$\begin{cases} y=\left(w^*\right)^T x > y_0 \rightarrow x \in w_1 \\ y=\left(w^*\right)^T x < y_0 \rightarrow x \in w_2 \end{cases}$

使用Fisher准则方法确定最佳线性分界面的方法是一个著名的方法，尽管提出该方法的时间比较早，仍见有人使用，如人脸识别中用于特征提取。

3. Fisher判别分析MATLAB实现——二分类

信阳师范学院
数学与统计学院
SCHOOL OF MATHEMATICS AND STATISTICS

FisherLDA_C2.m

```
function ldam = FisherLDA_C2(data)
% FisherLDA函数实现Fisher线性判别，仅针对类别数为2
% data为输入训练样本数据，最后一列为类别标志1、2
% 输出参数ldam为一个结构体：包含了fisher判别中的理论矩阵和判别结果

    %% 1、数据处理
    sn = size(data,1); %总样本数
    data = data(randperm(sn),:);  %打乱顺序，随机抽样
    n8 = round(sn*0.8); %抽取80%的样本作为训练样本
    xtrain = data(1:n8,1:end-1); %训练样本
    trainLabel = data(1:n8,end); %训练样本标签
    xtest = data(n8+1:end,1:end-1); %20%的测试样本
    testLabel = data(n8+1:end,end); %测试样本标签

    %% 2、对类别数的判断
    cn = length(unique(trainLabel));  %类别数
    if cn ~= 2
        warning('该函数FisherLDA()只针对类别数为2。');
        ldam = [];
        return
    end

    %% 3、按类别分类样本，并计算类均值向量和类内离散度矩阵
    m = zeros(cn,size(xtrain,2));
    Sinner = cell(1,cn);
    train = cell(1,cn);
    Ni = zeros(1,cn);
    for i = 1:cn
        train{i} = xtrain(trainLabel == i,:);
        m(i,:) = mean(train{i});  %类均值向量
        Ni(i) = size(train{i},1); %每个总体的样本量
        r = size(train{i},1);
        %类内离散度矩阵
        Sinner{i} = (train{i} - ones(r,1)*m(i,:))'*...
            (train{i} - ones(r,1)*m(i,:));
    end

    %% 4、计算总类内离散度矩阵SinnerSum，投影方向W和判别函数以及阈值T
    SinnerSum = Sinner{1} + Sinner{2};
    W = (SinnerSum\(m(1,:) - m(2,:))')'; %投影方向
    %T = -0.5*W*(m(1,:)+m(2,:))';%阈值T
    T = -(Ni(1)*W*m(1,:)' + Ni(2)*W*m(2,:)')/sum(Ni); %阈值T

    %% 5、20%的测试样本
    nt = size(xtest,1); %样本量
    newlabel = zeros(1,nt); %预测正确分类标签
    LabelVal = zeros(1,cn); %y+y0后的判别值
    for i = 1:nt
        %变量初始化时，W和T根据类别数初始化，故取1
        g = W*xtest(i,:)' + T;
        if g > 0
            newlabel(i) = 1;
        elseif g < 0
            newlabel(i) = 2;
        end
        LabelVal(i) = g;
    end
```

3. Fisher判别分析MATLAB实现——二分类

```
59        %% 6、构造混淆矩阵
60 —      fuzzA = zeros(cn,cn);
61 —      for i = 1:nt
62 —          for j = 1:cn %共两类，故简单化处理
63 —              if newlabel(i) == 1 && testLabel(i) == j
64 —                  fuzzA(1,j) = fuzzA(1,j) + 1;
65 —              elseif newlabel(i) == 2 && testLabel(i) == j
66 —                  fuzzA(2,j) = fuzzA(2,j) + 1;
67 —              end
68 —          end
69 —      end
70
71        %% 7、输出参数赋值组合
72 —      acc = sum(diag(fuzzA))/nt*100;
73 —      ldam.M = m;
74 —      ldam.Sinner = Sinner;
75 —      ldam.SinnerSum = SinnerSum;
76 —      ldam.W = W;
77 —      ldam.T = T;
78 —      ldam.testlabel_Num = nt;
79 —      ldam.correct_Num = sum(diag(fuzzA));
80 —      ldam.TrueAndTest_Label = [testLabel,newlabel'];
81 —      ldam.LabelVal = LabelVal;
82 —      ldam.confusion_Matrix = fuzzA;
83 —      ldam.predict_accuracy = acc;
84 —  end
```

```
>> load wine.data   %三分类数据
>> wine = [wine(:,2:end),wine(:,1)];
>> wine = wine(1:130,:);  %前130为二分类
>> ldam = FisherLDA_C2(wine)
```

```
>> ldam = FisherLDA_C2(wine)
ldam =
  包含以下字段的 struct:

                M: [2×13 double]
           Sinner: {[13×13 double]  [13×13 double]}
        SinnerSum: [13×13 double]
                W: [1×13 double]
                T: -0.8055
     testlabel_Num: 26
       correct_Num: 26
 TrueAndTest_Label: [26×2 double]
          LabelVal: [1×26 double]
  confusion_Matrix: [2×2 double]
  predict_accuracy: 100
>> cm = ldam.confusion_Matrix
cm =
    13     0
     0    13
```

3. Fisher判别分析MATLAB实现——二分类

```
>> load BreastTissuedata.mat
>> data(:,1) = [];   %删除第一列病人ID
>> btd = data(:,2:end);
>> btd = zscore(btd);  %标准化
>> btd = [btd,data(:,1)]; %最后一列为类别标签
>> ldam = FisherLDA_C2(btd)
```

```
ldam =
  包含以下字段的 struct:

                  M: [2×30 double]
             Sinner: {[30×30 double]  [30×30 double]}
          SinnerSum: [30×30 double]
                  W: [1×30 double]
                  T: 5.4332e-04
       testlabel_Num: 114
         correct_Num: 112
   TrueAndTest_Label: [114×2 double]
            LabelVal: [1×114 double]
    confusion_Matrix: [2×2 double]
    predict_accuracy: 98.2456
>> cm = ldam.confusion_Matrix
cm =
    73     1
     1    39
```

```
acc = zeros(100,1);
for i = 1:100
    ldam = FisherLDA_C2(btd);
    acc(i) = ldam.predict_accuracy;
end
```

```
macc = mean(acc)  %取平均
macc =
   96.4035
```

从100次随机抽样进行平均正确率来看，使用fisher判别所获得的线性判别模型具有一定的泛化能力。

4. Fisher判别分析MATLAB实现——多分类

信阳师范学院
数学与统计学院
SCHOOL OF MATHEMATICS AND STATISTICS

FisherLDA_MultClass.m

```
function ldam = FisherLDA_MultClass(data)

    %% 1、数据处理
    sn = size(data,1); %总样本数
    data = data(randperm(sn),:);  %打乱顺序，随机抽样
    %data(:,1:end-1) = zscore(data(:,1:end-1)); %标准化数据
    n8 = round(sn*0.8); %抽取80%的样本作为训练样本
    xtrain = data(1:n8,1:end-1); %训练样本
    trainLabel = data(1:n8,end); %训练样本标签
    xtest = data(n8+1:end,1:end-1); %20%的测试样本
    testLabel = data(n8+1:end,end); %测试样本标签

    %% 2、按各总体计算各种统计量值
    cn = length(unique(trainLabel));  %类别数
    Ni = zeros(cn,1); %存储每个总体的样本数
    vars = size(xtrain,2);  %变量数，即总体指标数
    Sw = zeros(vars,vars); %存储总类内离散度
    train = cell(cn,1); %存储每个总体样本
    Sinner = cell(vars,vars); %存储类内离散度矩阵
    m = zeros(cn,vars); %存储类内均值，一行代表一个总体
    for i = 1:cn
        train{i} = xtrain(trainLabel == i,:); %取每个总体
        m(i,:) = mean(train{i});  %类均值向量
        %类内离散度矩阵
        Sinner{i} = (train{i} - m(i,:))'*(train{i} - m(i,:));
        Ni(i) = size(train{i},1); %每个总体的样本量
        Sw = Sw + Ni(i)*Sinner{i};
    end
    M = mean(m); %计算总体均值
    Sw = Sw/sn; %总类内离散度
    Sb = zeros(vars,vars); %计算类间离散度矩阵
    for i = 1:cn
        Sb = Sb + Ni(i)*(M - m(i,:))'*(M - m(i,:));
    end
    Sb = Sb/sn;  %类间离散度矩阵


    %% 3、求最大特征值和特征向量
    A = repmat(0.1,[1,size(Sw,1)]);
    B = diag(A); %避免Sw奇异，加上一个以0.1为对角线的矩阵
    [V,D] = eig((Sw + B)\Sb);
    [~,ind] = max(diag(D));
    W = V(:,ind)'; %最大特征值所对应的特征向量

    %% 4、训练样本投影后中心值
    mY = zeros(1,cn); %存储训练样本投影后每个总体类内均值
    for i = 1:cn
        Y = W*train{i}'; %训练样本投影值
        mY(i) = mean(Y); %每个总体投影后类内均值
    end

    %% 5、20%的测试样本
    nt = size(xtest,1); %样本量
    newlabel = zeros(nt,1); %预测正确分类标签
    for i = 1:nt
        ytest = W*xtest(i,:)'; %测试样本投影
        minT = mean(M); %设置阈值，一个较大的数，任意
        % 实现测试样本与哪个总体距离近，归类哪个总体
        for j = 1:cn
            if abs(ytest - mY(j)) < minT
                newlabel(i) = j;
                minT = abs(ytest - mY(j));
```

4. Fisher判别分析MATLAB实现——多分类

```
63 —                end
64 —            end
65 —        end
66
67          %% 6、构造混淆矩阵
68 —        fuzzA = zeros(cn); %存储混淆矩阵
69 —        for i = 1:nt %对每个测试样本进行类别分析
70 —            c = testLabel1(i);
71 —            n1 = newlabel1(i);
72 —            if n1 == c
73 —                fuzzA(c,c) = fuzzA(c,c) + 1; %对角线元素+1，即正确分类
74 —            else
75 —                fuzzA(c,n1) = fuzzA(c,n1) + 1; %对应正确行错误列 + 1
76 —            end
77 —        end
78
79          %% 7、输出参数赋值组合
80 —        acc = sum(diag(fuzzA))/nt*100; %正确率
81 —        ldam.M = M; %总体均值
82 —        ldam.MY = mY; %总体投影均值
83 —        ldam.Sinner = Sinner; %类内离散度矩阵
84 —        ldam.SinnerSum = Sw; %类内总离散度矩阵
85 —        ldam.SoutSum = Sb; %类间总离散度矩阵
86 —        ldam.W = W; %最终投影方向W
87 —        ldam.testlabel_Num = nt; %测试样本量
88 —        ldam.correct_Num = sum(diag(fuzzA)); %归类正确样本量
89 —        ldam.TrueAndTest_Label = [testLabel,newlabel]; %标准与预测标签值矩阵
90 —        ldam.confusion_Matrix = fuzzA; %混淆矩阵
91 —        ldam.predict_accuracy = acc; %正确率
92 —    end
```

```
>> iris = xlsread('iris.xlsx');
>> label = [ones(50,1);2*ones(50,1);3*ones(50,1)];
>> iris = [iris,label];
>> ldam = FisherLDA_MultClass(iris)
```

```
ldam =
  包含以下字段的 struct:

                    M: [5.8413 3.0566 3.7592 1.2058]
               Sinner: {4×4 cell}
            SinnerSum: [4×4 double]
              SoutSum: [4×4 double]
                    W: [-0.2725 -0.3013 0.6312 0.6607]
        testlabel_Num: 30
          correct_Num: 30
    TrueAndTest_Label: [30×2 double]
     confusion_Matrix: [3×3 double]
     predict_accuracy: 100
>> ldam.confusion_Matrix
ans =
     7     0     0
     0    15     0
     0     0     8
```

案例分析

```
load wine.data   %三分类数据
wine = [wine(:,2:end),wine(:,1)];
ldam = FisherLDA_MultClass(wine)
```

```
ldam =
  包含以下字段的 struct:

                M: [1×13 double]
           Sinner: {13×13 cell}
        SinnerSum: [13×13 double]
          SoutSum: [13×13 double]
                W: [1×13 double]
      testlabel_Num: 36
        correct_Num: 35
  TrueAndTest_Label: [36×2 double]
   confusion_Matrix: [3×3 double]
   predict_accuracy: 97.2222
>> cm = ldam.confusion_Matrix
cm =
    16     0     0
     0    13     1
     0     0     6
```

```
load wine.data   %三分类数据
wine = [wine(:,2:end),wine(:,1)];
k = 1;
for i = 1:100
    ldam = FisherLDA_MultClass(wine);
    if ldam.predict_accuracy == 0
        continue
    end
    acc(k) = ldam.predict_accuracy;
    k = k + 1;
end
macc = mean(acc)
macc =
   92.2222
```

进行100次实验，每次取20%的随机样本进行判别，取最终正确率的平均值为92.222%，说明模型具有一定的泛化能力。有些正确率为0的，可能存在阈值设置问题。

4. Fisher判别分析MATLAB实现——多分类

```
>> ffd = xlsread('familyfin.xlsx');
>> fin = [ffd(:,1:5);ffd(1:12,7:11)];
>> ldam = FisherLDA_MultClass(fin)
```

```
ldam =
  包含以下字段的 struct:

                M: [1.4207e+04 1.3131e+03 428.2323 4.6535e+03]
           Sinner: {4×4 cell}
        SinnerSum: [4×4 double]
          SoutSum: [4×4 double]
                W: [-0.5167 0.7041 0.1899 -0.4486]
      testlabel_Num: 5
        correct_Num: 5
  TrueAndTest_Label: [5×2 double]
   confusion_Matrix: [3×3 double]
   predict_accuracy: 100
>> cm = ldam.confusion_Matrix
cm =
     0     0     0
     0     1     0
     0     0     4
```

5. Fisher待判样本预测MATLAB实现

FisherLDA_predict.m

```
function newlabel = FisherLDA_predict(model,xtest)
    gn = size(model.confusion_Matrix,1);  %类别数
    newlabel = zeros(1,size(xtest,1)); %预测正确分类标签
    for i = 1:size(xtest,1)
        if gn == 2  % 二分类
            g = model.W*xtest(i,:)' + model.T;
            if g > 0
                newlabel(i) = 1;
            elseif g < 0
                newlabel(i) = 2;
            end
        else % 多分类
            ytest = model.W*xtest(i,:)'; %测试样本投影
            minT = mean(model.M); %设置阈值，一个较大的数，任意
            for j = 1:gn  %cn类别数
                if abs(ytest - model.MY(j)) < minT
                    newlabel(i) = j;
                    minT = abs(ytest - model.MY(j));
                end
            end
        end
    end
end
```

```
ffd = xlsread('familyfin.xlsx');
fin = [ffd(:,1:5);ffd(1:12,7:11)];
ldam = FisherLDA_MultClass(fin)
>> newlabel = FisherLDA_predict(ldam,xtest)
newlabel =
     3     2
```

%最终判断青海是第3类，广东是第2类。

四. 判别分析函数fitcdiscr

`Mdl = fitcdiscr(Tbl,ResponseVarName)` returns a fitted discriminant analysis model based on the input variables (also known as predictors, features, or attributes) contained in the table `Tbl` and output (response or labels) contained in `ResponseVarName`.

`Mdl = fitcdiscr(Tbl,formula)` returns a fitted discriminant analysis model based on the input variables contained in the table `Tbl`. `formula` is an explanatory model of the response and a subset of predictor variables in `Tbl` used to fit `Mdl`.

`Mdl = fitcdiscr(Tbl,Y)` returns a fitted discriminant analysis model based on the input variables contained in the table `Tbl` and response `Y`.

`Mdl = fitcdiscr(X,Y)` returns a discriminant analysis classifier based on the input variables `X` and response `Y`.

`Mdl = fitcdiscr( ___ ,Name,Value)` fits a classifier with additional options specified by one or more name-value pair arguments, using any of the previous syntaxes. For example, you can optimize hyperparameters to minimize the model’ s cross-validation loss, or specify the cost of misclassification, the prior probabilities for each class, or the observation weights.

四. 判别分析函数fitcdiscr

```
load fisheriris
X = meas;
Y = species;
d = fitcdiscr(X,Y)
confusionmat(Y,predict(d,X))
```

```
ans =
    50     0     0
     0    48     2
     0     1    49
```

```
d =
  ClassificationDiscriminant
             PredictorNames: {'SepalLength'  'SepalWidth'  'PetalLength'  'PetalWidth'}
               ResponseName: 'Species'
      CategoricalPredictors: []
                 ClassNames: [setosa    versicolor    virginica]
             ScoreTransform: 'none'
            NumObservations: 150
                DiscrimType: 'linear'
                         Mu: [3×4 double]
                     Coeffs: [3×3 struct]

  Properties, Methods

ans =
    50     0     0
     0    48     2
     0     1    49
```

四. 判别分析函数fitcdiscr

```
>> ffd = xlsread('familyfin.xlsx');
>> A1 = [ffd(1:15,1:5);ffd(1:12,7:11)];
>> d = fitcdiscr(A1(:,1:4),A1(:,5))
d =
  ClassificationDiscriminant
             ResponseName: 'Y'
    CategoricalPredictors: []
               ClassNames: [1 2 3]
          ScoreTransform: 'none'
         NumObservations: 27
             DiscrimType: 'linear'
                      Mu: [3×4 double]
                  Coeffs: [3×3 struct]
```

```
>> confusionmat(A1(:,5),predict(d,A1(:,1:4)))
ans =
     2     0     0
     0     6     0
     0     0    19
```

分类函数——有监督学习

类名	方法名	函数名	说明
线性回归	多元线性回归	fitlm	具有多个预测变量的线性回归
	逐步回归	stepwise	交互式逐步回归
	多目标的多元线性回归	mvregress	使用多变量输出的线性回归
	有正则化的多元线性回归	lasso	使用弹性网正则化的多元线性回归
		ridge	Ridge回归
非线性回归		fitnlm	拟合非线性回归模型
广义线性模型	正态分布拟合	fitglm	'Distribution' 设置为 'normal'
	二项分布拟合	fitglm	'Distribution' 设置为 'binomial'
	泊松分布拟合	fitglm	'Distribution' 设置为 'poisson'
	gamma分布拟合	fitglm	'Distribution' 设置为 'gamma'
	反高斯分布拟合	fitglm	'Distribution' 设置为 'inverse gaussian'
	进行变量选择的逐步回归	stepwiseglm	交互式逐步回归
	带有正则化的广义线性回归	lassoglm	使用弹性网正则化的广义线性回归
回归分类 决策树、（CART）	分类树	fitctree	训练分类二叉决策树
	回归树	fitrtree	训练回归二叉决策树
支持向量机	二分类支持向量机	fitcsvm	训练二分类支持向量机分类
	多分类支持向量机	fitcecoc	适用SVM或其他分类器的多类模型
判别分析		fitcdiscr	拟合判别分析分类器
朴素贝叶斯分类器		fitcnb	训练朴素贝叶斯分类
最近邻	k-近邻	fitcknn	拟合k-近邻分类器

分类函数——无监督学习

信阳师范学院
数学与统计学院
SCHOOL OF MATHEMATICS AND STATISTICS

类名	方法名	函数名	说明
分层聚类	通过聚类树进行聚类	cluster	返回聚类后各样本类别
	通过数据进行聚类	clusterdata	返回聚类后各样本类别
	分成聚类树	linkage	训练分层聚类树
通过距离聚类	K-means聚类	kmeans	
	K-medoids聚类	kmedoids	
最近邻	全局最近邻搜索	ExhaustiveSearcher	准备全局最近邻居搜索
	KD树搜索	KDTreeSearcher	生成KD树
		createns	使用KD树搜索
	KNN搜索	knnsearch	使用Kd-tree或全局k-最近邻搜索
	范围搜索	rangesearch	使用全局与Kd-tree查找指定范围的近邻
高斯混合模型	高斯混合模型	fitgmdist	拟合高斯混合模型
	基于高斯混合模型的聚类	cluster	生成基于高斯混合模型的聚类
隐马尔可夫模型	估计隐马尔可夫模型	hmmtrain	通过观测估计隐马尔科夫模型参数
		hmmestimate	通过状态和观测估计参数
	生成观测序列	hmmgenerate	生成隐马尔可夫模型状态和观测
	最可能状态路径	hmmviterbi	计算最可能的状态路径
	后验状态概率	hmmdecode	计算隐马尔可夫模型后验状态概率

分类函数——集成学习

信阳师范学院
数学与统计学院
SCHOOL OF MATHEMATICS AND STATISTICS

类名	方法名	函数名	说明
Boosting	二分类：AdaBoostM1	fitensemble	'Method' 配置为 'AdaBoostM1'
	二分类：LogitBoost	fitensemble	'Method' 配置为 ' LogitBoost'
	二分类：GentleBoost	fitensemble	'Method' 配置为 ' GentleBoost'
	二分类：RobustBoost	fitensemble	'Method' 配置为 ' RobustBoost'
	多分类： AdaBoostM2	fitensemble	'Method' 配置为 ' AdaBoostM2'
	多分类： LPBoosts	fitensemble	'Method' 配置为 ' LPBoosts'
	多分类：TotalBoost	fitensemble	'Method' 配置为 ' TotalBoost'
	多分类：RUSBoost	fitensemble	'Method' 配置为 ' RUSBoost'
	回归：LSBoost	fitensemble	'Method' 配置为 'LPBoost'
	提升二分类为多分类模型	fitcecoc	基于二分类模型训练多分类模型
Bagging（多分类或回归）		fitensemble	'Method' 配置为 'Bag'
随机子空间（多分类或回归）		fitensemble	'Method' 配置为 'Subspace'


01
IDEA
02
LEARNING
03
CREATIVE
INSPIRATION
EDUCATION
BUSINESS
PHOTO BOOK

感谢聆听