模式识别大作业

——Fisher 线性判别分析

姓名：李怡桐

学号：**16020510073**

学院：人工智能学院

任课教师：张向荣

一、问题重述

编程实现采用 Fisher 线性判别分析法对Sonar数据集、Iris数据集前两类进行降维，并利用阈值法对其进行分类，选用10倍交叉验证法或者随机抽样法（10次平均）实现测试。

- **Sonar，Iris 数据集的说明**

IRIS数据集以鸢尾花的特征作为数据来源，包含150个数据集，分为3 类，每类50个数据，每个数据包含4个属性，可通过花萼长度，花萼宽度，花瓣长度，花瓣宽度4个属性预测鸢尾花卉属于（Setosa，Versicolour，Virginica）三个种类中的哪一类。本题选择其数据的前两类进行降维，即前100个数据。

Sonar数据集包含208个数据集，有60维，分为2类，第一类为98个数据，第二类为110个数据，每个数据包含60个属性，是在数据挖掘、数据分类中非常常用的测试集、训练集。

二、Fisher线性判别分析法

Fisher线性判别分析的基本思想：选择一个投影方向（线性变换，线性组合），将高维问题降低到一维问题来解决，同时变换后的一维数据满足每一类内部的样本尽可能聚集在一起，不同类的样本相隔尽可能地远。

Fisher线性判别分析，就是通过给定的训练数据，确定投影方向W和阈值w0，即确定线性判别函数，然后根据这个线性判别函数，对测试数据进行测试，得到测试数据的类别。

线性判别函数的一般形式可表示成为

$$g(X) = W^T X + w_0 \tag{1}$$

其中,X为训练样本集，每个样本为一个d维向量，W为投影方向，也是一个D维向量。

Fisher线性判别分析法选择投影方向W的原则，即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的要求。如下为具体步骤：

- **投影方向W的确定**

各类样本均值向量m_i

$$m_i = \frac{1}{N_I} \sum_{x_j \in X_i} x_j, i = 1, 2 \tag{2}$$

样本类内离散度矩阵S_i和总类内离散度矩阵S_w

$$S_i = \sum_{x_j \in X_i} (x - m_i)(x - m_i)^T, i = 1, 2 \tag{3}$$

$$S_w = S_1 + S_2 \tag{4}$$

样本类间离散度矩阵 S_b

$$S_b = (m_2 - m_2)(x - m_i)^T \tag{5}$$

在投影后的一维空间中，各类样本均值

$$\tilde{m_i} = w^T m_i \tag{6}$$

样本类内离散度和总类内离散度

$$\tilde{S_w} = w^T S_w w \tag{7}$$

样本类间离散度

$$\tilde{S_b} = w^T S_b w \tag{8}$$

Fisher准则函数为

$$max J_F(w) = \frac{\tilde{m_1} - \tilde{m_2}^2}{\tilde{S_1}^2 + \tilde{S_2}^2} \tag{9}$$

这是一个等式约束下的极值问题，通过引入拉格朗日乘子转化成拉格朗日函数的无约束极值问题。通过化简，可以得到投影方向为：

$$w^* = S_w^{-1}(m_1 - m_2) \tag{10}$$

- **阈值w_0的确定**

Fisher判别函数最优的解给出了一个投影方向。要得到分类面，需要在投影后的方向上确定一个分类阈值w_0。Fisher线性判别所得的方向实际上就是最优贝叶斯决策的方向，可以得到：

$$w_0 = -\frac{1}{2}(m_1 + m_2)^T S_w^{-1}(m_1 - m_2) - ln\frac{P(w_2)}{P(w_1)} \tag{11}$$

- **Fisher线性判别的决策规则**

确定了投影方向和阈值，可以得到决策规则：

$$g(x) = w^T(x - \frac{1}{2}(m_1 + m_2)) - ln\frac{P(w_2)}{P(w_1)} \tag{12}$$

如果g(x)大于0，x属于w_1；如果g(x)小于0，则x属于w_1；如果g(x)=0,则可将x任意分到某一类或拒绝。

三、算法描述

- **Iris数据集**

选取Iris数据集前100组数据，即前两类的数据保存在矩阵Iris1和Iris2中，将两组50行4列的数据按行随机重新排列，抽取前45行数据作为训练集，后五行数据作为测试集。即完成了随机抽样的过程。之后，依据线性判别分析法求类均值向量、类间离散度矩阵，得到投影方向w和阈值w_0的值。

将测试集内数据代入决策规则进行判定，并与类标进行比较。分类正确kind值加一，最终计算总正确率。

将上述过程进行十次，每次均为随机抽取数据，最终求综合准确率。

- **Sonar数据集**

选取Sonar数据集,前98组数据保存在矩阵Sonar1中，后110组数据保存在Sonaar2中，将第98行60列的数据和110行60列的数据按行随机重新排列，分别抽取前88行数据和99行数据作为训练集，后其他数据作为测试集。即完成了随机抽样的过程。之后，依据线性判别分析法求类均值向量、类间离散度矩阵，得到投影方向w和阈值w_0的值。

将测试集内数据代入决策规则进行判定，并与类标进行比较。分类正确kind值加一，最终计算总正确率。

将上述过程进行十次，每次均为随机抽取数据，最终求综合准确率。

四、结果及分析

- **Iris数据集**

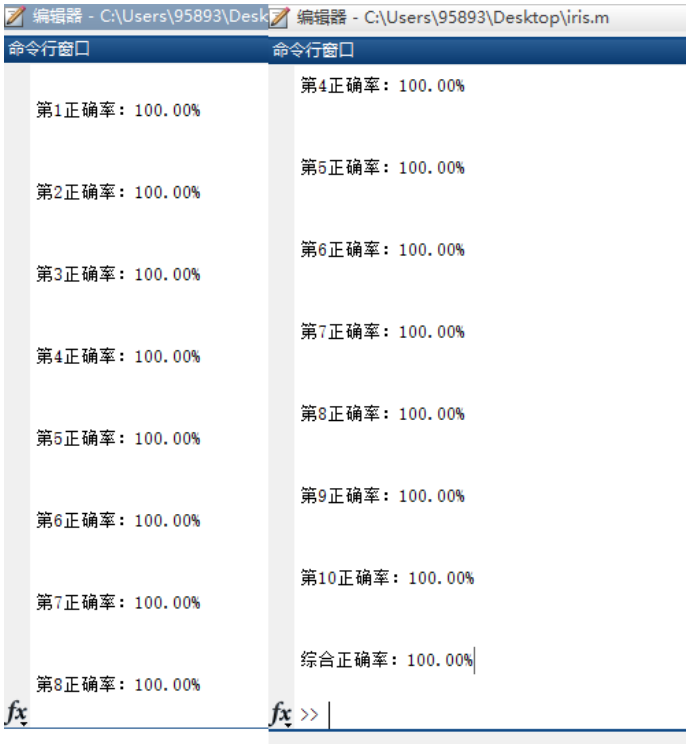

图 1: Iris数据集结果图

- **Sonar数据集**

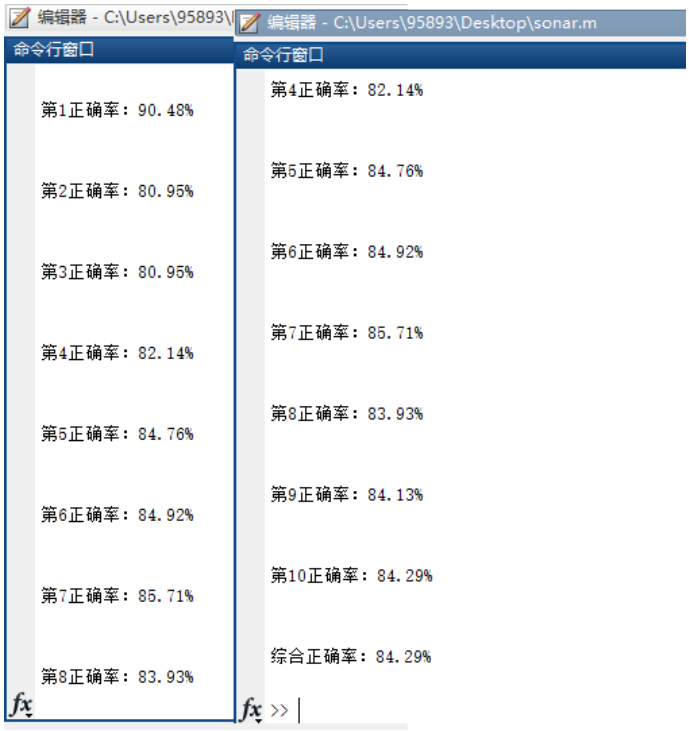

图 2: Sonar数据集结果图

从结果可以看出，对Iris数据集的测试结果综合正确率达到百分之百，对Sonar数据集的综合正确率为84.29%。由于即使随机抽取十次求结果的平均值，其结果仍然具有偶然性，经过多次运行发现，Iris数据集的测试结果综合正确率基本稳定在百分之百，Sonar数据集的综合正确率范围在80%左右波动。

附录

- **源代码一、Iris数据集**

```
clc
clear
data=xlsread('iris.xlsx');
Iris1=data(1:50,2:5);
Iris2=data(51:100,2:5);
%类均值向量
m1 = mean(Iris1);
m2 = mean(Iris2);
kind1 = 0;
kind2 = 0;

for p=1:1:10
vector1= randperm(50);
re1 = zeros(size(Iris1));
for i=1:50
re1 (i,:)= Iris1 (vector1(i),:);
end
rtriris1= re1(1:45,:);
rteiris1= re1 (46:50,:);
vector2= randperm(50);
re2 = zeros(size(Iris2));
for i=1:50
re2 (i,:)= Iris2 (vector2(i),:);
end
rtriris2= re2(1:45,:);
rteiris2= re2(46:50,:);
%各类内离散度矩阵
s1 = zeros(4);
s2 = zeros(4);
for i=1:1:45
    s1 = s1 + (rtriris1 (i,:) - m1)'*( rtriris1 (i,:) - m1);
end
for i=1:1:45
    s2 = s2 + (rtriris2 (i,:) - m2)'*( rtriris2 (i,:) - m2);
end
%总类内离散矩阵
sw12 = s1 + s2;
%投影方向
w12 = ((sw12^-1)*(m1 - m2)')';
%判别函数以及阈值（即）Tw0
T12 = -0.5 * (m1 + m2)*inv(sw12)*(m1 - m2)';
```

```
newiris1=[];
newiris2=[];
for i=1:5
    x = rteiris1 (i,:);
    g12 = w12 * x' + T12;
    if (g12 > 0)
       newiris1=[newiris1;x];
        kind1=kind1+1;
    elseif (g12 < 0)
         newiris2=[newiris2;x];
    end
end
for i=1:5
    x = rteiris2 (i,:);
    g12 = w12 * x' + T12;
    if (g12 > 0)
      newiris1=[newiris1;x];
    elseif (g12 < 0)
            kind2=kind2+1;
           newiris2=[newiris2;x];
    end
end
scorrect=(kind1+kind2)/(10*p);
fprintf('\第n%正确率：d%.2f%%\n\n',p,scorrect* 100);
end
correct=(kind1+kind2)/100;
fprintf('\综合正确率：n%.2f%%\n\n',correct* 100);
```

- **源代码二、sonar数据集**

```
clc
clear
data= xlsread('sonar.xlsx');
Sonar1=data(1:98,1:60);
Sonar2=data(99:208,1:60);
%类均值向量
m1 = mean(Sonar1);
m2 = mean(Sonar2);
kind1 = 0;
kind2 = 0;

for p=1:1:10
vector1= randperm(98);
re1 = zeros(size(Sonar1));
for i=1:98
```

```
re1 (i,:)= Sonar1 (vector1(i),:);
end
rtrsonar1= re1(1:88,:);
rtesonar1= re1 (89:98,:);
vector2= randperm(110);
re2 = zeros(size(Sonar2));
for i=1:110
re2 (i,:)= Sonar2 (vector2(i),:);
end
rtrsonar2= re2(1:99,:);
rtesonar2= re2(100:110,:);
%各类内离散度矩阵
s1 = zeros(60);
s2 = zeros(60);
for i=1:1:88
    s1 = s1 + (rtrsonar1 (i,:) - m1)'*( rtrsonar1 (i,:) - m1);
end
for i=1:1:99
    s2 = s2 + (rtrsonar2 (i,:) - m2)'*( rtrsonar2 (i,:) - m2);
end
%总类内离散矩阵
sw12 = s1 + s2;
%投影方向
w12 = ((sw12^-1)*(m1 - m2)')';
%判别函数以及阈值（即）Tw0
T12 = -0.5 * (m1 + m2)*inv(sw12)*(m1 - m2)';
newsonar1=[];
newsonar2=[];
for i=1:10
    x = rtrsonar1 (i,:);
    g12 = w12 * x' + T12;
    if (g12 > 0)
       newsonar1=[newsonar1;x];
        kind1=kind1+1;
    elseif (g12 < 0)
         newsonar2=[newsonar2;x];
    end
end
for i=1:11
    x = rtesonar2 (i,:);
    g12 = w12 * x' + T12;
    if (g12 > 0)
      newsonar1=[newsonar1;x];
    elseif (g12 < 0)
            kind2=kind2+1;
```

```
            newsonar2=[newsonar2;x];
    end
end
scorrect=(kind1+kind2)/(21*p);
fprintf('\第n%正确率：d%.2f%%\n\n',p,scorrect* 100);
end
correct=(kind1+kind2)/210;
fprintf('\综合正确率：n%.2f%%\n\n',correct* 100);
```