一、 问题描述

线性判别式分析(Linear Discriminant Analysis, LDA)，也叫做 Fisher 线性判别(Fisher Linear Discriminant ,FLD)，是模式识别的经典算法，它是在 1996 年由 Belhumeur 引入模式识别和人工智能领域的。性鉴别分析的基本思想是将高维的模式样本投影到最佳鉴别矢量空间，以达到抽取分类信息和压缩特征空间维数的效果，投影后保证模式样本在新的子空间有最大的类间距离和最小的类内距离，即模式在该空间中有最佳的可分离性。因此，它是一种有效的特征抽取方法。使用这种方法能够使投影后模式样本的类间散布矩阵最大，并且同时类内散布矩阵最小。就是说，它能够保证投影后模式样本在新的空间中有最小的类内距离和最大的类间距离，即模式在该空间中有最佳的可分离性。

1. 有 USPS 手写体 3 和手写体 8 两组数据，每一组数据中有 1100 个 256 维的样本，用 Fisher 线性判别的方法分出手写体的 3 和 8，用十倍交叉法或者十次测试来验证正确率。

2. 现有 sonar 数据集 sonar1 和 sonar2 两组数据，sonar1 有 72 个 60 维样本，sonar2 有 111 个 60 维的样本,用 Fisher 线性判别的方法分出 sonar1 和 sonar2，并进行 10 倍交叉验证正确率。

二、Fisher 线性判别原理描述

训练样本集为 $D=\{x_1,\ldots,x_N\}$，每个样本是一个 d 维向量，其中 W1 类的样本为：$D_1=\{\mathrm{x}_1^1,\ldots,\mathrm{x}_{N_1}^1\}$，W2 的样本为 $D_2=\{\mathrm{x}_1^2,\ldots,\mathrm{x}_{N_2}^2\}$ 。寻找一个投影方向 W，（W 也是一个 d 维向量），投影以后的样本变成

$$y=w^Tx，\ i=1,2,\ldots,N \qquad \cdots\cdots\cdots（1）$$

在原样本空间中，类均值向量为

$$m_i=\frac{1}{N}\sum_{x_j\in D_i}x_j,i=1,2 \qquad \cdots\cdots\cdots（2）$$

定义各类内离散度矩阵(within-class scatter matrix)为

$$S_i=\sum_{x_j\in D_i}(x_j-m_i)(x_j-m_i)^T,i=1,2 \qquad \cdots\cdots\cdots（3）$$

总类内离散度矩阵(pooled with-class scatter matrix)为

$$S_w=S_1+S_2 \qquad \cdots\cdots\cdots（4）$$

类间离散度矩阵(between-class scatter matrix)定义为

$$S_b=(m_1-m_2)(m_1-m_2)^T \qquad \cdots\cdots\cdots（5）$$

在投影后的一维空间，两类的均值为

$$\tilde{m}_i = \frac{1}{N_i}\sum_{y_j \in \mathrm{y}_i} y_j = \frac{1}{N_i}\sum_{x_j \in D_i} w^T x_j = w^T m_i, i = 1,2 \qquad \cdots\cdots\cdots（6）$$

类内离散度不再是一个矩阵，而是一个值

$$\tilde{S}_i^2 = \sum_{y_j \in \mathrm{y}_i} (y_j - \tilde{m}_i)^2, i = 1,2 \qquad \cdots\cdots\cdots（7）$$

总的类内离散度为

$$\tilde{S}_w = \tilde{S}_1^2 + \tilde{S}_2^2 \qquad \cdots\cdots\cdots\cdots（8）$$

类间离散度为

$$\tilde{S}_b = (\tilde{m}_1 - \tilde{m}_2)^2 \qquad \cdots\cdots\cdots（9）$$

前面已经提出，希望寻找的投影方向使投影后的两类尽可能分开，而各类内部又尽可能聚集，这一目标可以表示成如下的准则

$$\max J_F(w) = \frac{\tilde{S}_b}{\tilde{S}_w} = \frac{(\tilde{m}_1 - \tilde{m}_2)^2}{\tilde{S}_1^2 + \tilde{S}_2^2} \qquad \cdots\cdots\cdots（10）$$

将公式 7 和 9 带入公式 1 中，fisher 判别准则变成

$$\max J_F(w) = \frac{w^T S_b w}{w^T S_w w} \qquad \cdots\cdots\cdots\cdots（11）$$

为了求使上述式子最大投影方向 w，把其转换为最优化问题

$$\begin{aligned} &\max w^T S_b w \\ &s.t.\ w^T S_w w = c \neq 0 \end{aligned} \qquad \cdots\cdots\cdots\cdots（12）$$

利用拉格朗日乘子转化最后得出最优投影方向为

$$w = S_w^{-1}(m_1 - m_2) \qquad \cdots\cdots\cdots\cdots（13）$$

阈值确定：本文根据最优贝叶斯分类器，如果不考虑先验概率的不同，采取的阈值为

$$w_0 = -\frac{1}{2}(\tilde{m}_1 - \tilde{m}_2) \qquad \cdots\cdots\cdots\cdots（14）$$

采取决策原则

$$g(x) = w^T x + w_0 \begin{cases} > 0, x \in w_1 \\ < 0, x \in w_2 \end{cases} \qquad \cdots\cdots\cdots\cdots（15）$$

三、源代码

Fisher 识别 usps 手写体

```
load('usps_all');
x3=data(:,:,3); %取出手写体 3 的数据集
x8=data(:,:,8); %取出手写体 8 的数据集
[a,b]=size(x3);%[256,1100]
for n=1:10
    test3=double(x3(:,(n-1)*b/10+1:n*b/10));%测试样本,b/10=110
    test8=double(x8(:,(n-1)*b/10+1:n*b/10));%每次 110 个样本,矩阵 256x110
    train3=double(x3); train8=double(x8);%此时 train3 和 trian8 为[256,1100]
    train3(:,(n-1)*b/10+1:n*b/10)=[];%训练样本
    train8(:,(n-1)*b/10+1:n*b/10)=[];%把测试样本部分设为空大小[256,990]
     %sum(x,2)表示行求和(每一行数据全部加起来)
     %m3=[256,1]/990    m3,m8 为 110x1 矩阵
    m3=sum(train3,2)*10/9/b;
    m8=sum(train8,2)*10/9/b;
    S3=zeros(a,a);S8=zeros(a,a);%[256,256]
    for i=1:9*b/10      % 1—>990
        temp3=train3(:,i)-m3;%train3(:,i)代表第 i 列的所有元素[256,1]
        temp8=train8(:,i)-m8;
        S3=S3+temp3*(temp3)';
        S8=S8+temp8*(temp8)';
    end
    %%%%%%%%%离散度矩阵
    Sw=S3+S8; %这里 S3,S8 和书上 S1,S2 对应
  %%%%%%%%%%得出最优化投影方向 w*
    w=inv(Sw)*(m3-m8);
```

```
    %%%% 阈值 w0=-1/2(m~1+m~2)
    %%%% m~i=w'mi (w'是转置矩阵)
    w0=-1/2*(w'*m3+w'*m8);
    count=0;   %count 表示识别正确的次数
    for i=1:b/10 %%j=1->110
        g3=w'*test3(:,i)+w0;
        if g3>0 count=count+1;
        end
    end
    for i=1:b/10 %%k=1->110
        g8=w'*test8(:,i)+w0;
        if g8<0 count=count+1;
        end
    end
    R(n)=count/(b/10+b/10); %10 倍交叉验证的各个正确率
  % disp(R(n))
    disp(['第',num2str(n),'次的正确率为',num2str(R(n))]);
end
Avg_r=sum(R)/10;
disp(['平均正确率为',num2str(Avg_r)]);
```

```
第 1 次的正确率为 0.98182
第 2 次的正确率为 0.97273
第 3 次的正确率为 0.98182
第 4 次的正确率为 0.99091
第 5 次的正确率为 0.96818
第 6 次的正确率为 0.97273
第 7 次的正确率为 0.95455
第 8 次的正确率为 0.98636
第 9 次的正确率为 0.98636
```

第 10 次的正确率为 0.96818

平均正确率为 0.97636

Fisher 识别 sonar 数据集

```
clear
load('sonar.mat');
A = double(sonar(1:90,:));
B = double(sonar(101:190,:));
indices = crossvalind('Kfold',90,10);%10 为交叉验证折数
for a = 1:10    %实验记进行 10 次(交叉验证折数)，求 10 次的平均值作为实验结果，
    test = (indices == a); train = ~test;   %产生测试集合训练集索引
    A_test = A (test,:);
    A_train = A (train,:);
    B_test = B (test,:);
    B_train = B (train,:);
        %计算样本均值
    m1=mean(A_train)';
    m2=mean(B_train)';
        %s1、s2 分别代表表示第一类、第二类样本的类内离散度矩阵
    s1=zeros(size(A_train,2));
    for i=1:size(A_train,1)
        s1 = s1 + (A_train(i,:)'-m1)*(A_train(i,:)'-m1)';
    end;
    s2=zeros(size(B_train,2));
    for i=1:size(B_train,1)
        s2 = s2 + (B_train(i,:)'-m2)*(B_train(i,:)'-m2)';
    end;
    %计算总类内离散度矩阵 Sw
    Sw=s1+s2;
```

```
    %计算 fisher 准则函数取极大值时的解 w
    w=inv(Sw)*(m1-m2);
    %计算阈值 w0
    ave_m1 = w'*m1;
    ave_m2 = w'*m2;
    w0 = -(ave_m1+ave_m2)/2;
    countA=0;
    countB=0;
    %计算正确率
    for i=1:9
        %判别函数
        g1 = w'*A_test(i,:)'+w0;
        if g1>0
            countA = countA+1;
        end
    end
    disp(['第',num2str(a),'次识别 d1 的正确率为',num2str(countA/9)]);
    for i=1:9
        g2 = w'*B_test(i,:)'+w0;
        if g2<0
            countB = countB+1;
        end
    end
    disp(['第',num2str(a),'次识别 d2 的正确率为',num2str(countB/9)]);
end
```

```
第 1 次识别 d1 的正确率为 0.77778
第 1 次识别 d2 的正确率为 0.88889
第 2 次识别 d1 的正确率为 0.77778
第 2 次识别 d2 的正确率为 0.66667
```

第 3 次识别 d1 的正确率为 0.44444

第 3 次识别 d2 的正确率为 0.77778

第 4 次识别 d1 的正确率为 0.88889

第 4 次识别 d2 的正确率为 0.66667

第 5 次识别 d1 的正确率为 1

第 5 次识别 d2 的正确率为 0.77778

第 6 次识别 d1 的正确率为 0.66667

第 6 次识别 d2 的正确率为 0.77778

第 7 次识别 d1 的正确率为 0.55556

第 7 次识别 d2 的正确率为 0.88889

第 8 次识别 d1 的正确率为 0.77778

第 8 次识别 d2 的正确率为 0.77778

第 9 次识别 d1 的正确率为 0.55556

第 9 次识别 d2 的正确率为 0.66667

第 10 次识别 d1 的正确率为 0.66667

第 10 次识别 d2 的正确率为 0.77778