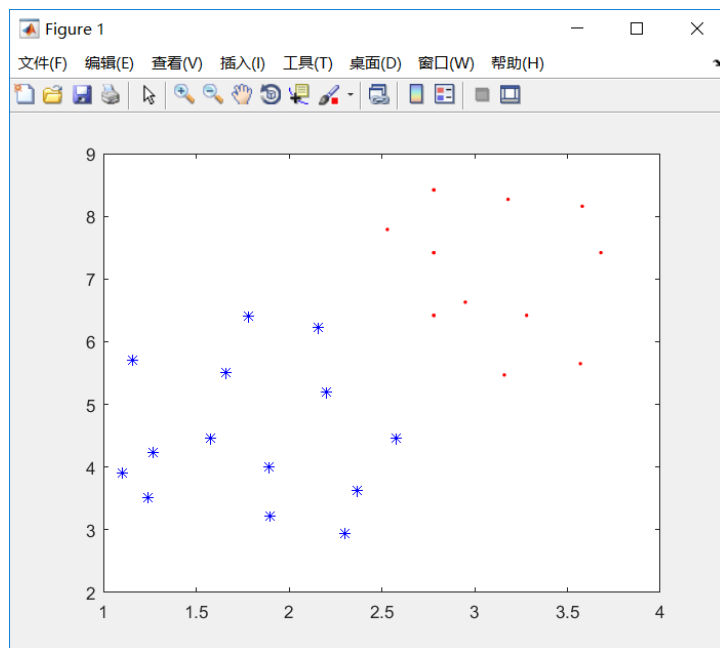


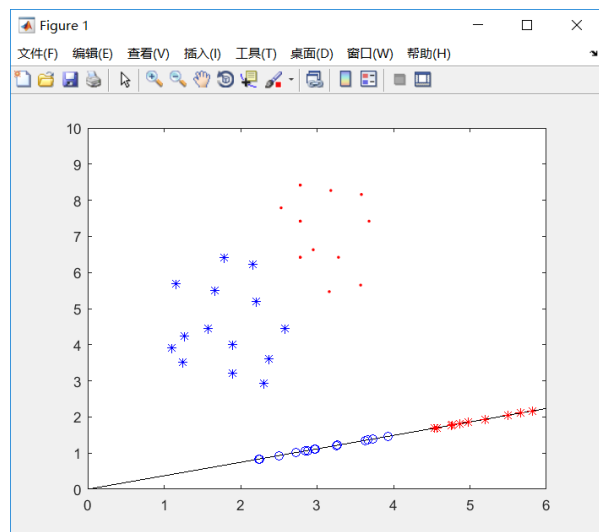
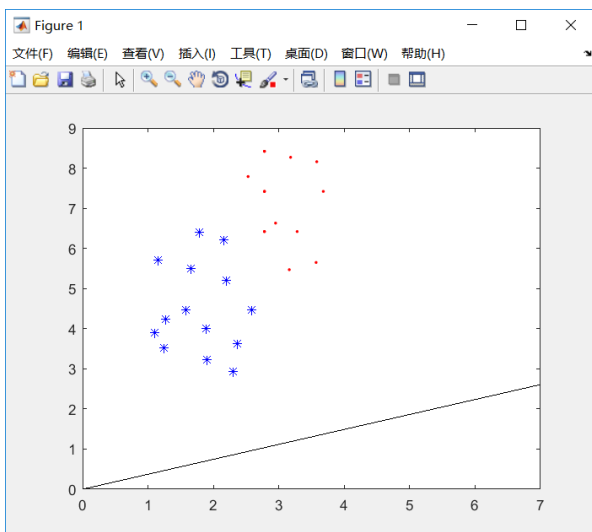
计算机科学与技术 学院

机器学习 课程实验报告

学号：	姓名：	班级：
实验题目：Linear Discriminant Analysis		
实验学时：2	实验日期：2018/10/15	
实验目的： In this exercise, you need to implement Linear Discriminant Analysis(LDA). 熟悉使用 LDA。		
硬件环境： 操作系统 Windows 10 家庭中文版 64-bit CPU Intel Core i5 7200U @ 2.50GHz 41 ° C Kaby Lake-U/Y 14nm 工艺 RAM 8.00GB 单个的-通道 未知 (15-15-15-35) 主板 HP 81D1 (U3E1) 图像 Generic PnP Monitor (1920x1080@60Hz) Intel HD Graphics 620 (HP) 存储器 476GB NVMe THNSN5512GPAK T0 (未知) 40GB Microsoft 虚拟磁盘 (File-backed Virtual) 光盘驱动器 没有检测到光纤磁盘驱动 音频 Conexant ISST Audio		
软件环境： Win10 + matlabR2016a		
实验步骤与内容： 1. 对于 LDA 两类的情况如下，首先加载数据集并进行可视化如下：		



2. 而后编写 MyLDA 的函数算法实现对两类数据的 LDA 分类（具体代码见文档尾的附录）得到所需的特征向量后进行可视化并将其映射在 1da 形成的子空间上，如下：



3. 其中 matlab 的执行代码如下：

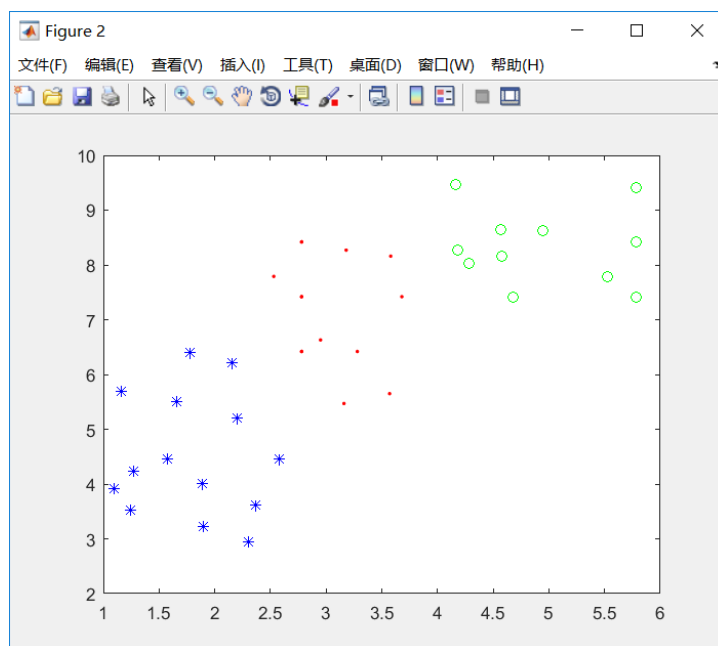
```
d_c1=load('ex3red.dat');
d_c2=load('ex3blue.dat');
figure;
plot(d_c1(:,1),d_c1(:,2),'r');
hold on;
plot(d_c2(:,1),d_c2(:,2),'*b');
hold on;
W=MyLDA(d_c1,d_c2);
n1=length(d_c1);
n2=length(d_c2);
k=W(2)/W(1);
plot([0,7],[0,7*k],'-k');
axis([0 6 0 10]);
hold on;
```

```

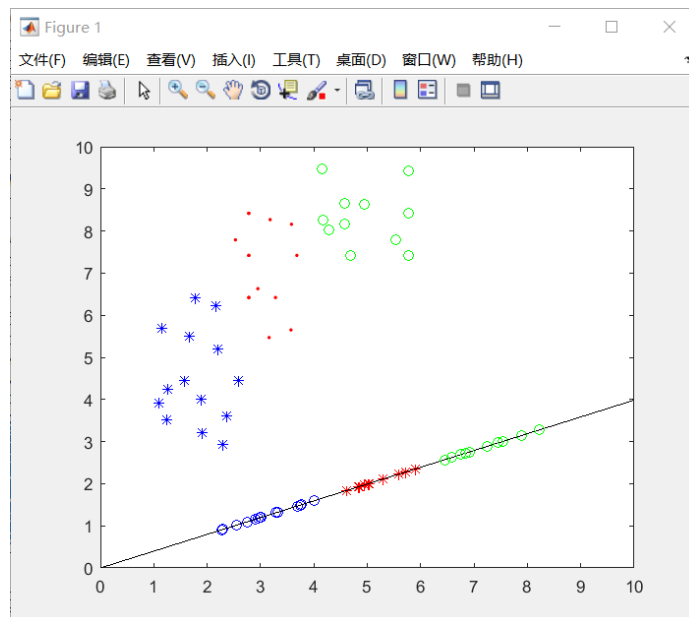
for i=1:n1
temp=d_c1(i,:);
newx=(temp(1)+k*temp(2))/(k*k+1);
newy=k*newx;
plot(newx,newy,'*r');
end;
for i=1:n2
temp=d_c2(i,:);
newx=(temp(1)+k*temp(2))/(k*k+1);
newy=k*newx;
plot(newx,newy,'ob');
end;
hold off;

```

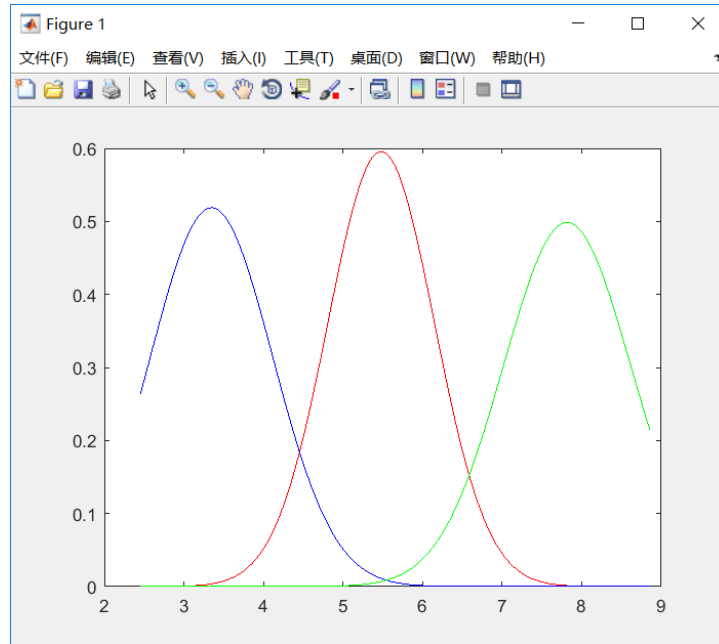
4. 对于 N 类的 LDA，以 3 类数据为例，同样先载入数据集进行显示如下：



5. 编写函数方法实现对 3 及 N 类数据集的 LDA，调用后进行显示如下：



将其转换为高斯分布观察最终分类效果如下（具体 LDA 算法代码见文档尾的附录）：



6. 其中对于第二部分实验内容的 matlab 的执行代码如下：

%加载数据集

```
d_c1=load('ex3red.dat');  
d_c2=load('ex3blue.dat');  
d_c3=load('ex3green.dat');
```

%显示原始数据集

```
figure;  
plot(d_c1(:,1),d_c1(:,2),'r');  
hold on;  
plot(d_c2(:,1),d_c2(:,2),'*b');  
hold on;  
plot(d_c3(:,1),d_c3(:,2),'og');
```

%调用 LDA

```
W3=MyLDA_3(d_c1,d_c2,d_c3);  
n1=length(d_c1);  
n2=length(d_c2);  
n3=length(d_c3);  
k3=W3(2)/W3(1);
```

%显示结果

```
close figure 2;  
axis([0 8 0 10]);  
hold on;  
plot([0,10],[0,10*k3],'-k');
```

```
for i=1:n1  
temp=d_c1(i,:);  
newx=(temp(1)+k3*temp(2))/(k3*k3+1);
```

```

newy=k3*newx;
plot(newx,newy,'*r');
end;

for i=1:n2
temp=d_c2(i,:);
newx=(temp(1)+k3*temp(2))/(k3*k3+1);
newy=k3*newx;
plot(newx,newy,'ob');
end;

for i=1:n3
temp=d_c3(i,:);
newx=(temp(1)+k3*temp(2))/(k3*k3+1);
newy=k3*newx;
plot(newx,newy,'og');
end;
hold on;

axis([0 10 0 10]);
hold on;

hold off;
close;
clc

y1_w1=d_c1*W3;
y1_w2=d_c2*W3;
y1_w3=d_c3*W3;
figure;
%显示高斯分布的结果
miny=min([min(y1_w1),min(y2_w2),min(y3_w3)]);
miny=min([min(y1_w1),min(y1_w2),min(y1_w3)]);
maxy=max([max(y1_w1),max(y1_w2),max(y1_w3)]);
y_w1=miny:0.05:maxy;

y1w1mu=mean(y1_w1);
y1w1sig=std(y1_w1);
y1w1pdf=mvnpdf(y_w1',y1w1mu,y1w1sig);
plot(y_w1,y1w1pdf,'r');
hold on;

y2w1mu=mean(y1_w2);
y2w1sig=std(y1_w2);
y2w1pdf=mvnpdf(y_w1',y2w1mu,y2w1sig);
plot(y_w1,y2w1pdf,'b');

```

```

        hold on;

        y3w1mu=mean(y1_w3);
        y3w1sig=std(y1_w3);
        y3w1pdf=mvnpdf(y_w1',y3w1mu,y3w1sig);
        plot(y_w1,y3w1pdf,'g');
        hold on;
        hold off;

```

结论分析与体会：

本次实验学习了使用 LDA 在简单的小数据集上进行验证与实际运用，不过对于小数据集的分类感觉 KNN 和决策树的算法效果会更好一些，不过通过实验深入掌握了 LDA 的算法原理与其具体实现，matlab 的画图也用的比能比较熟练了。

附录：程序源代码

一、二分类 LDA：

```

function [W] = MyLDA(w1,w2)

m1=mean(w1);%第一类样本均值
m2=mean(w2);%第二类样本均值
m=mean([w1;w2]);%总样本均值

%第二步：计算类内离散度矩阵 Sw
n1=size(w1,1);%第一类样本数
n2=size(w2,1);%第二类样本数
%求第一类样本的散列矩阵 s1
s1=0;
for i=1:n1
    s1=s1+(w1(i,:)-m1)'*(w1(i,:)-m1);
end
%求第二类样本的散列矩阵 s2
s2=0;
for i=1:n2
    s2=s2+(w2(i,:)-m2)'*(w2(i,:)-m2);
end
Sw=(n1*s1+n2*s2)/(n1+n2);
%第三步：计算类间离散度矩阵 Sb
Sb=(n1*(m-m1)'*(m-m1)+n2*(m-m2)'*(m-m2))/(n1+n2);
%第四步：求最大特征值和特征向量
%[V,D]=eig(inv(Sw)*Sb);%特征向量 V，特征值 D
A = repmat(0.1,[1,size(Sw,1)]);
B = diag(A);
[V,D]=eig(inv(Sw + B)*Sb);
[a,b]=max(max(D));
W=V(:,b);%最大特征值对应的特征向量
end

```

二、多分类 3 及以上类别的 LDA:

```
function [w] = MyLDA_3(x1,x2,x3)
%UNTITLED2 此处显示有关此函数的摘要
% 此处显示详细说明
m1=mean(x1);
m2=mean(x2);
m3=mean(x3);
m=mean([x1;x2;x3]);

n1=size(x1,1);%第一类样本数
n2=size(x2,1);%第二类样本数
n3=size(x3,1);%第三类样本数

s1=0;
for i=1:n1
    s1=s1+(x1(i,:)-m1)'*(x1(i,:)-m1);
end
s2=0;
for i=1:n2
    s2=s2+(x2(i,:)-m2)'*(x2(i,:)-m2);
end
s3=0;
for i=1:n3
    s3=s3+(x3(i,:)-m3)'*(x3(i,:)-m3);
end
Sw=(n1*s1+n2*s2+n3*s3)/(n1+n2+n3);

Sb=(n1*(m-m1)'*(m-m1)+n2*(m-m2)'*(m-m2)+n3*(m-m3)'*(m-m3))/(n1+n2+n3);
A = repmat(0.1,[1,size(Sw,1)]);
B = diag(A);
[V,D]=eig((Sw + B)\Sb);
[a,b]=max(max(D));
w=V(:,b);%最大特征值对应的特征向量

end
```

% (2) 任意类别输入数据的 LDA, 借鉴资料链接:

<https://www.mathworks.com/matlabcentral/fileexchange/29673-lda-linear-discriminant-analysis> by Will Dwinnell and Deniz Sevis:

```
function W = LDA(Input,Target,Priors)
[n m] = size(Input);

ClassLabel = unique(Target);
k = length(ClassLabel);
```

```

% 初始化
nGroup      = NaN(k,1);
GroupMean    = NaN(k,m);
PooledCov    = zeros(m,m);
W            = NaN(k,m+1);

if (nargin >= 3) PriorProb = Priors; end

for i = 1:k,
    Group      = (Target == ClassLabel(i));
    nGroup(i)  = sum(double(Group));

    GroupMean(i,:) = mean(Input(Group,:));

    PooledCov = PooledCov + ((nGroup(i) - 1) / (n - k)) .*
cov(Input(Group,:));
end

if (nargin >= 3)
    PriorProb = Priors;
else
    PriorProb = nGroup / n;
end

for i = 1:k,

    Temp = GroupMean(i,:) / (PooledCov + 0 * eye(size(PooledCov)));

    W(i,1) = -0.5 * Temp * GroupMean(i,:)' + log(PriorProb(i));

    W(i,2:end) = Temp;
end
clear Temp
end

```