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：201600301304 | 姓名：贾乘兴 | | 班级： 人工智能16 |
| 实验题目：线性判别分析 | | | |
| 实验学时：2小时 | | 实验日期： 2018/10/15 | |
| 实验目的：利用给定数据（红蓝绿数据集），实现线性判别分析的二分类方法和多分类方法，并绘制图像查看效果 | | | |
| 硬件环境： 16 GB 内存 | | | |
| 软件环境：mac os，matlab 2017b | | | |
| 实验步骤与内容：  一．二分类模型  1.二分类方法是最经典的线性判别模型，在本实验的两种特征的数据集下，其思想为将数据投影到一条直线上，在直线上进行分类，所求直线为分类的最佳直线  2.判定该直线分类效果的好坏的方式为：投影到该直线后，类间距离之和较大，同时类内数据点较为聚集。类间距离我们用均值期望之间l2距离的平方表示，可表示为：    类间距离要尽可能大，其中，定义类间散度矩阵：    类内距离我们用各个类下的方差表示：    类内距离要尽量小，其中，定义类内散度矩阵：    这样我们得到要最大化的目标函数为    该目标函数又被称为类间散度矩阵和类内散度矩阵的“广义瑞利商”  在求w的优化上，我们可以令分母为1，将该问题变为一般性的标准优化问题    由拉格朗日乘子法，该问题等价于    经过变形可得：    所求w即为左侧矩阵最大特征值对应的特征向量  而在该二分类问题中，考虑到类间散度矩阵的特点：    可得：    最终将w单位化即可，该方法较为简单  3.实现二分类  a．数据读取  %% load data  xb=load('ex3blue.dat');  xg=load('ex3green.dat');  xr=load('ex3red.dat');  b．投影向量的计算  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);  x=[0:0.01,10];  y=(w(2)/w(1))\*x;  zb=xb\*w\*w';  zr=xr\*w\*w';    最终得到的二分类结果与数据展示如下  ../../f1.jpg  绘制图像代码如下  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');  在得到了投影到直线上的数据上后，我们假设数据分布满足给定数据投影的均值和方差下的正态分布，对数据分布进行了可视化，可见分类效果较好  ../../f2.jpg  代码如下：  %% distribution  yb=xb\*w;  yr=xr\*w;  sb=std(yb);  sr=std(yr);  ub=mean(yb);  ur=mean(yr);  tb=[ub-4\*sb:0.01:ub+4\*sb];  tr=[ur-4\*sr:0.01:ur+4\*sr];  figure  plot(tb,exp(-(tb-ub).\*(tb-ub)/(2\*sb^2))/(sb\*sqrt(2\*pi)),'b-');  hold on  plot(tr,exp(-(tr-ur).\*(tr-ur)/(2\*sr^2))/(sr\*sqrt(2\*pi)),'r-');  title('distribution of data');  legend('data blue','data red');  二．多分类  1.LDA 在多分类问题上较二分类复杂，可以将该问题视为降维问题  2.在二分类问题的基础上，我们定义全局散度矩阵：    其中，u为全部数据的均值：    在多分类问题下，类间散度矩阵可表示为：    多分类的优化问题可表示为：    该问题仍然可以转化为求解最大特征值的特征向量组成的矩阵问题  3.多分类实现  a．数据导入  b．投影矩阵求解  mb=mean(xb);  mr=mean(xr);  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);  Sw=(xb-mb)'\*(xb-mb)+(xr-mr)'\*(xr-mr)+(xg-mg)'\*(xg-mg);  Sb=St-Sw;  [vec,val]=eig(inv(Sw)\*Sb);  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';  zg=xg\*W\*W';  c．绘制图像代码  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');  可以得到多分类的结果如下：  ../../f3.jpg  可见该问题的分类效果较好，绘制在直线上的分布如下：  ../../f4.jpg  代码如下：  yb=xb\*W;  yr=xr\*W;  yg=xg\*W;  sb=std(yb);  sr=std(yr);  sg=std(yg);  ub=mean(yb);  ur=mean(yr);  ug=mean(yg);  tb=[ub-4\*sb:0.01:ub+4\*sb];  tr=[ur-4\*sr:0.01:ur+4\*sr];  tg=[ug-4\*sg:0.01:ug+4\*sg];  figure  plot(tb,exp(-(tb-ub).\*(tb-ub)/(2\*sb^2))/(sb\*sqrt(2\*pi)),'b-');  hold on  plot(tr,exp(-(tr-ur).\*(tr-ur)/(2\*sr^2))/(sr\*sqrt(2\*pi)),'r-');  hold on  plot(tg,exp(-(tg-ug).\*(tg-ug)/(2\*sg^2))/(sg\*sqrt(2\*pi)),'g-');  title('distribution of data');  legend('data blue','data red','data green');  三．分类器使用  对于相邻两类的边界计算如下： | | | |
| 结论分析与体会： 本次实验了解了线性分类器的原理，并进行了实现，很多数学方法可以由简单的低维拓展到高维，并且在一定的问题下可以实现较好的效果 | | | |

附录：程序源代码

ex3.m

clear,clc;

%% load data

xb=load('ex3blue.dat');

xg=load('ex3green.dat');

xr=load('ex3red.dat');

%% bi-classification

figure

plot(xb(:,1),xb(:,2),'bx');

hold on

plot(xr(:,1),xr(:,2),'rx');

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;

hold on

plot(x,y,'-d');

zb=xb\*w\*w';

zr=xr\*w\*w';

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');

%% distribution

yb=xb\*w;

yr=xr\*w;

sb=std(yb);

sr=std(yr);

ub=mean(yb);

ur=mean(yr);

tb=[ub-4\*sb:0.01:ub+4\*sb];

tr=[ur-4\*sr:0.01:ur+4\*sr];

figure

plot(tb,exp(-(tb-ub).\*(tb-ub)/(2\*sb^2))/(sb\*sqrt(2\*pi)),'b-');

hold on

plot(tr,exp(-(tr-ur).\*(tr-ur)/(2\*sr^2))/(sr\*sqrt(2\*pi)),'r-');

title('distribution of data');

legend('data blue','data red');

%% multi-classification

figure

plot(xb(:,1),xb(:,2),'bx');

hold on

plot(xr(:,1),xr(:,2),'rx');

hold on

plot(xg(:,1),xg(:,2),'gx');

mb=mean(xb);

mr=mean(xr);

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);

Sw=(xb-mb)'\*(xb-mb)+(xr-mr)'\*(xr-mr)+(xg-mg)'\*(xg-mg);

Sb=St-Sw;

[vec,val]=eig(inv(Sw)\*Sb);

lamda=max(diag(val));

W=vec(:,find(diag(val==lamda)));

x=[0:0.01,10];

y=(W(2)/W(1))\*x;

hold on

plot(x,y,'-d');

zb=xb\*W\*W';

zr=xr\*W\*W';

zg=xg\*W\*W';

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');

%% distribution

yb=xb\*W;

yr=xr\*W;

yg=xg\*W;

sb=std(yb);

sr=std(yr);

sg=std(yg);

ub=mean(yb);

ur=mean(yr);

ug=mean(yg);

tb=[ub-4\*sb:0.01:ub+4\*sb];

tr=[ur-4\*sr:0.01:ur+4\*sr];

tg=[ug-4\*sg:0.01:ug+4\*sg];

figure

plot(tb,exp(-(tb-ub).\*(tb-ub)/(2\*sb^2))/(sb\*sqrt(2\*pi)),'b-');

hold on

plot(tr,exp(-(tr-ur).\*(tr-ur)/(2\*sr^2))/(sr\*sqrt(2\*pi)),'r-');

hold on

plot(tg,exp(-(tg-ug).\*(tg-ug)/(2\*sg^2))/(sg\*sqrt(2\*pi)),'g-');

title('distribution of data');

legend('data blue','data red','data green');