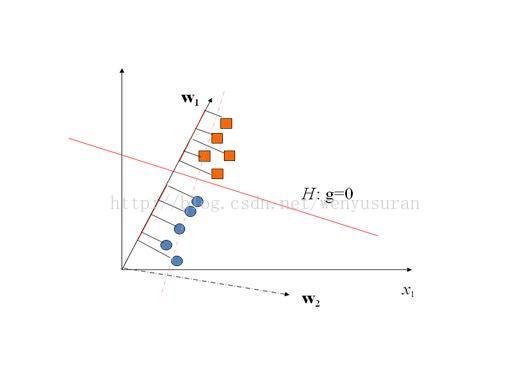
Fisher分类器

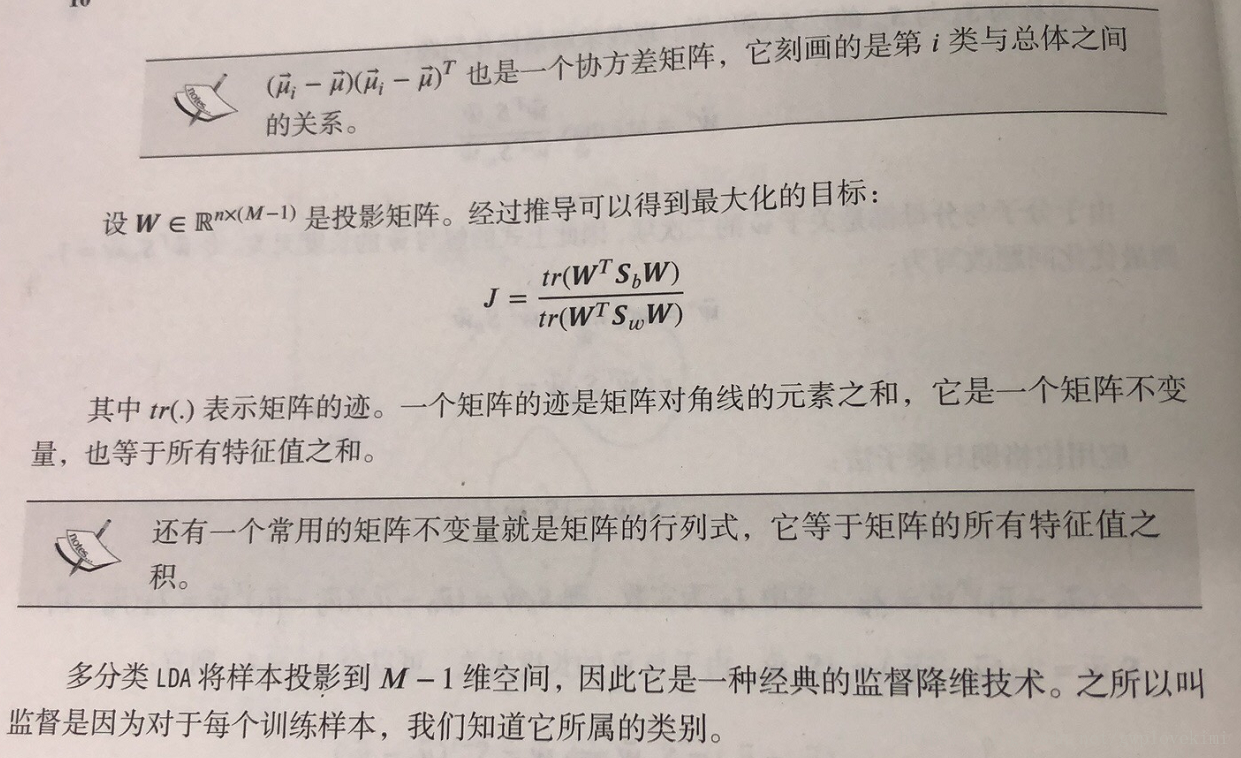
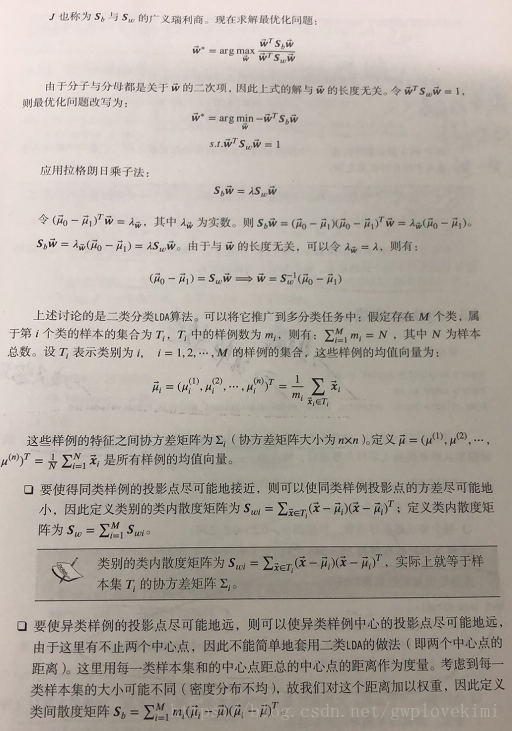
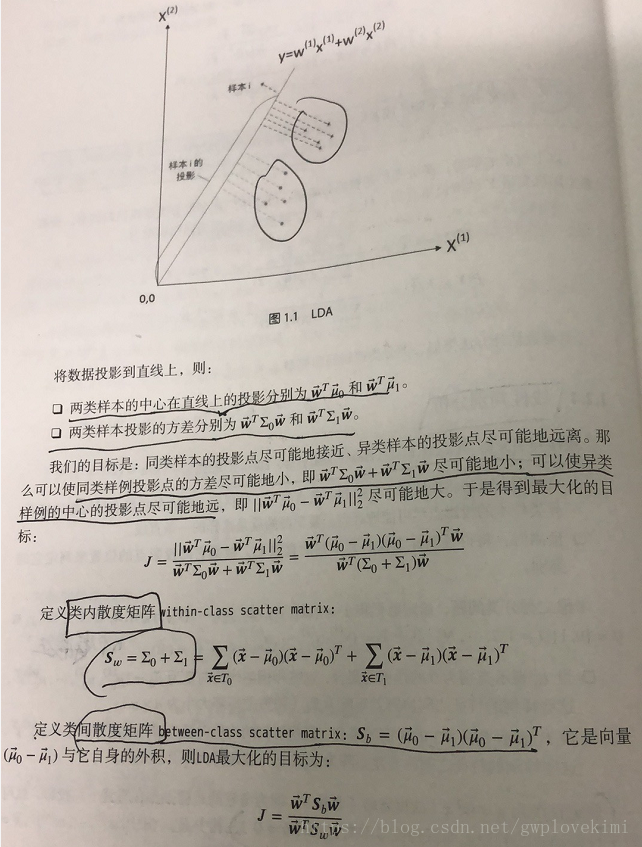
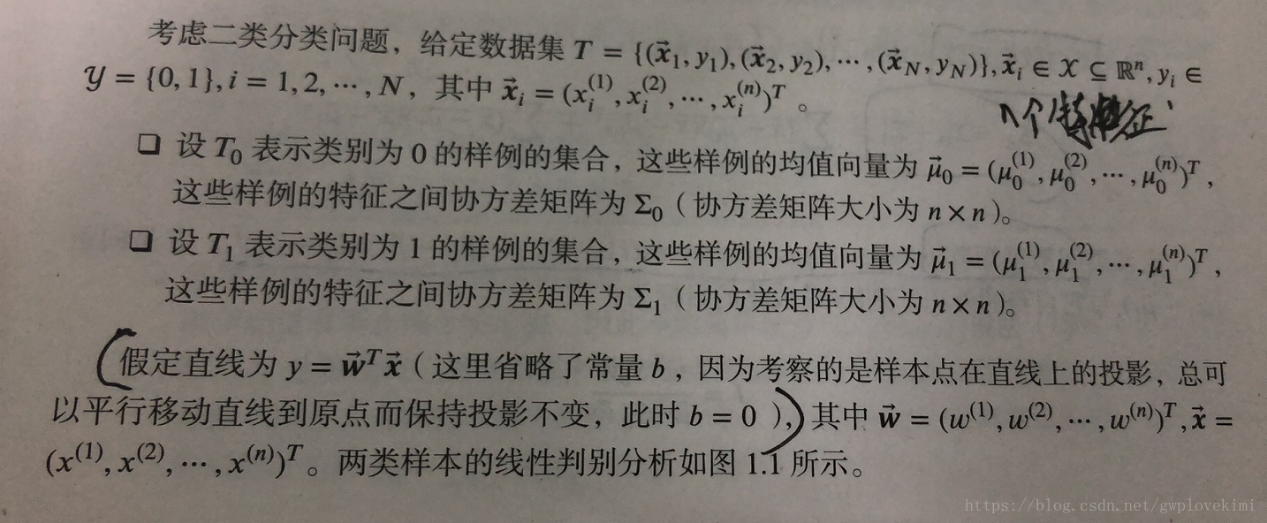
# 一、Fisher的基本思想是：

Fisher准则基本原理：找到一个最合适的投影轴，使两类样本在该轴上投影之间的距离尽可能远，而每一类样本的投影尽可能紧凑，从而使分类效果为最佳。



例如上图中：通过将方块点和圆点向w1投影，然后再在设置合适的阈值即可将方块和圆点分离。

# 二、推导过程



# 三、优缺点

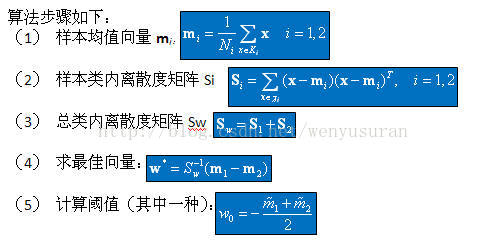
## 优点：

1. 一般对于线性可分的样本，总能找到一个投影方向；降维仍然线性可分。
2. 可以直接求解法向量
3. 适用随机模型，多类情况

## 缺点：

1. 如果M1=M2，W=0。则这样的样本线性不可分；M1！=M2，未必线性可分；SW不可逆，未必不可分。
2. 对线性不可分的情况，无法确定其分类。

# 四、算法步骤



# 五、数据集描述

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%分别产生x轴和y轴都为正态分布的随机序列

%假设x轴和y轴序列相互独立，可产生二维正态分布随机序列

%w1、w2分别用来保存两个训练集的横坐标和纵坐标

%用normrnd函数产生正态分布函数

%normrnd（mean,omega,[row,column]）

%mean:均值;omega:标准差

%row:产生随机序列的行数;column:产生随机序列的列数

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

X1 = normrnd(40,10,[200,1]);

Y1 = normrnd(40,10,[200,1]);

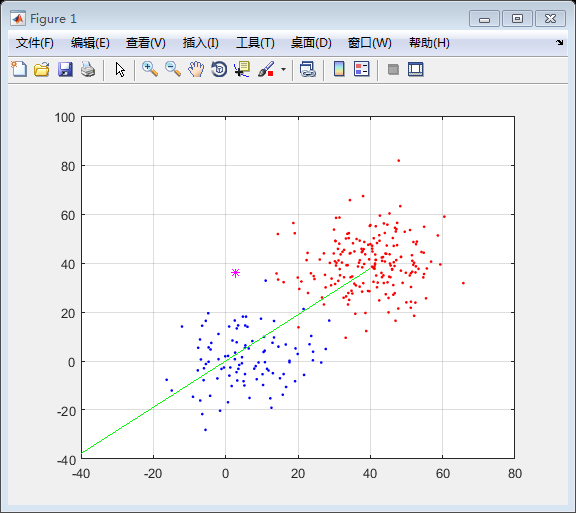
w1=[X1, Y1];

X2 = normrnd(5 ,10,[100,1]);

Y2 = normrnd(0 ,10,[100,1]);

w2=[X2, Y2];

# 六、实验结果



代码

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%本m文件实现fisher算法，并对两个二维正态分布随机序列

%进行训练，进而可在屏幕上任意取点，程序可输出属于第一类

%还是第二类

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%分别产生x轴和y轴都为正态分布的随机序列

%假设x轴和y轴序列相互独立，可产生二维正态分布随机序列

%w1、w2分别用来保存两个训练集的横坐标和纵坐标

%用normrnd函数产生正态分布函数

%normrnd（mean,omega,[row,column]）

%mean:均值;omega:标准差

%row:产生随机序列的行数;column:产生随机序列的列数

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

X1 = normrnd(40,10,[200,1]);

Y1 = normrnd(40,10,[200,1]);

w1=[X1, Y1];

X2 = normrnd(5 ,10,[100,1]);

Y2 = normrnd(0 ,10,[100,1]);

w2=[X2, Y2];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%以下部分为fisher算法的实现

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%计算样本均值

m1=mean(w1)';

m2=mean(w2)';

%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(2);

[row1,colum1]=size(w1);

for i=1:row1

s1 = s1 + (w1(i,:)'-m1)\*(w1(i,:)'-m1)';

end;

s2=zeros(2);

[row2,colum2]=size(w2);

for i=1:row2

s2 = s2 + (w2(i,:)' - m2)\*(w2(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;

%画出两类训练样本点

figure(1)

plot(X1,Y1,'.r',X2,Y2,'.b');%画出两类样本点

hold on;grid;

%画出取极大值时的解w

x = [-40:0.1:40];

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

plot(x,y,'g')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%以下为测试部分

%利用ginput随机选取屏幕上的点（可连续取10个点）

%程序可根据点的位置自动地显示出属于那个类

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for i=1:10

[x,y]=ginput(1);

plot(x,y,'m\*');

sample=[x,y];

holdall

if(sample\*w- w0>0)

disp('it belong to the first class');

else

disp('it belong to the second class');

end;

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%