

**机器学习作业**

**题 目: 感知机**

**学 号: 919106840333**

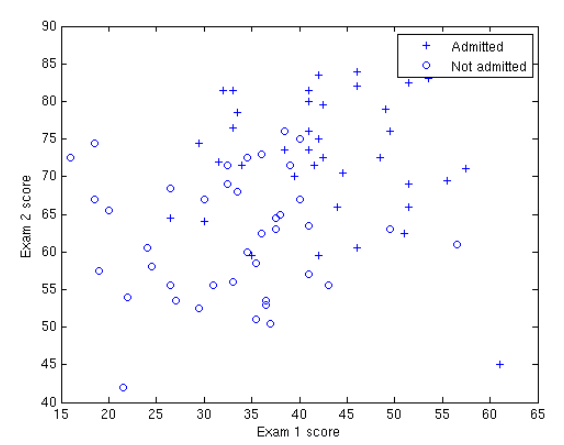
**姓 名: 孙傲歆**

**指导教师: 潘金山**

**2021年11月**

**一、题目要求**

• 给出下列训练数据:



• 实现感知机算法并将其与Logistic回归进行比较(基于SGD);

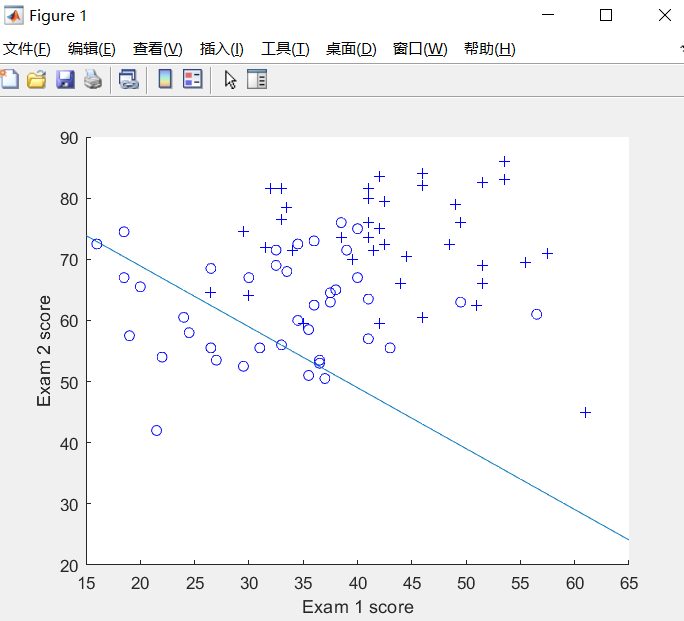
**二、算法思路**

**1.SGD求解感知机算法:**

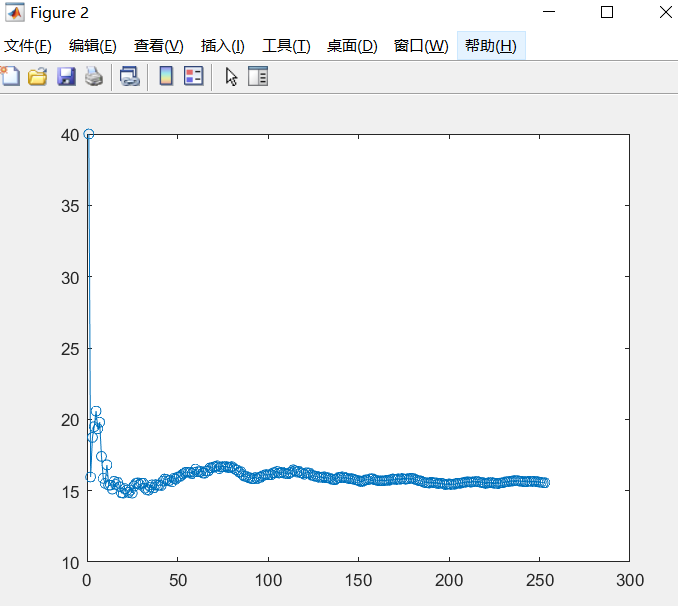
首先我们根据题目所给数据，可以确定目标函数形式为。且y只在1和-1之间取值。我们先将x数据进行标准化处理，公式为：(X-X\_mean)/X\_std。由于原数据集y的取值是0和1，所以我们将的y的数据进行了修改，1不变，0变为-1。我们取向量为[0;0]（即全0列向量），。并设置学习率a=0.5，按照感知机算法的梯度下降法的公式进行迭代。每次在错误数据集中随机取5个样本加入计算，当两次迭代结果的损失函数值之差小于0.0001时，我们认为结果收敛，停止迭代，并且在图中画出所有数据点，并将结果以直线形式显示出来，直线以上表示一类数据，直线以下表示另一类数据。同时我们对每次迭代计算结果进行记录，绘制每次迭代计算结果的折线图，观察其变化趋势。

**三、运行结果**

**1.SGD求解感知机:**



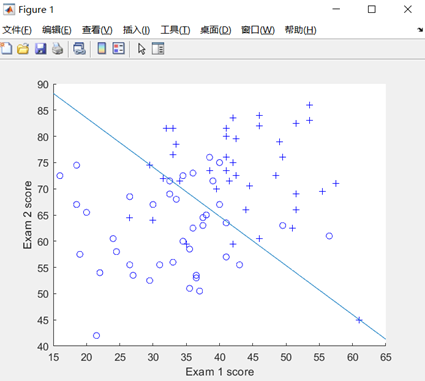
**图1.1**



**图1.2**

上图为代码运行结果。**图1.1**是数据显示，我们将所有点的信息绘制在图上并分类显示，o表示y值为1，+表示y值为-1。直线是计算结果的显示，直线之上表示y=1的一类，直线之下表示y=-1的一类。**图1.2**是每次迭代结果的折线图（不包含初始值），从图中可以看出，经过约250次迭代，结果达到了收敛。

**2.与logistics回归的比较:**



**图2.1**

**图2.1**是SGD求解logistics回归的结果，可以看出，对于题目所给的数据集，logistics回归的结果是明显要优于感知机的。因为感知机算法适用于对那些线性可分的数据集进行分类。而题目所给的数据明显是线性不可分的，所以结果有较大误差。

**四、可运行代码**

**1.get\_loss函数：**

function y = get\_loss(data\_y,h)

loss1 = data\_y'\*log(h);

loss2 = (1-data\_y)'\*log((1-h));

y=(1.0/80)\*(-loss1-loss2);

end

**2.homework\_4.v**

%SGD感知机算法

clc;

clear;

x=load("ex4Data/ex4x.dat");

y=load("ex4Data/ex4y.dat");

[m,n]=size(y);

figure(1);

xlabel('Exam 1 score');

ylabel('Exam 2 score');

for i=1:m

hold on;

if(y(i)==1)

plot(x(i,1),x(i,2),'b+');

else

plot(x(i,1),x(i,2),'bo');

end

end

mean=mean(x);

std=std(x);

for i=1:m

for z=1:2

x(i,z)=(x(i,z)-mean(z))/std(z);

end

end

a = 0.5;

w=[0;0];

b=1;

yy=zeros(80,1);

for i=1:80

if(y(i)==0)

yy(i)=-1;

else

yy(i)=1;

end

end

old\_loss = 0;

loss=get\_loss(x,yy,w,b);

temp\_loss=[];

while abs(old\_loss-loss) > 0.0001

tempnum=[];

for j=1:80

if((yy(j)\*(w\*x(j)+b))<=0)

tempnum=[tempnum,j];

end

end

k=randperm(length(tempnum));

z=k(1:5);

for i=1:5

w=w+a\*yy(z(i))\*x(z(i));

b=b+a\*yy(z(i));

end

old\_loss = loss;

loss=get\_loss(x,yy,w,b);

temp\_loss=[temp\_loss,old\_loss];

%disp(old\_loss);

end

theta=[w;b];

plot\_x2 =zeros(1,51);

plot\_x1 =15:1:65;

for i=15:65

plot\_x2(i-14) = -(theta(3)+theta(1)\*((i-mean(1))/std(1)))/theta(2);

plot\_x2(i-14) = plot\_x2(i-14)\*std(2)+mean(2);

end

hold on;

plot(plot\_x1,plot\_x2,'-');

figure(2);

plot(temp\_loss,'-o');

disp(theta);