# 实验报告四

**学号**：20201060287

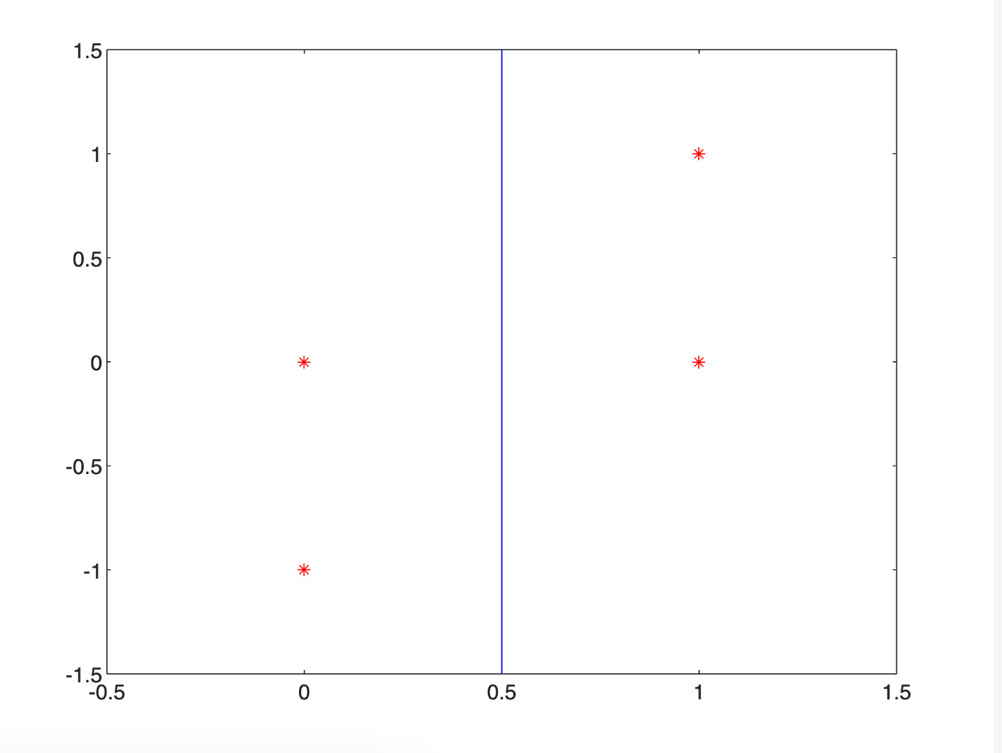
**姓名：**李昂

**实验名称：**感知器算法

**实验内容：**使用MATLAB编程环境，实现感知器算法并进行实验

**实验要求及结果：**

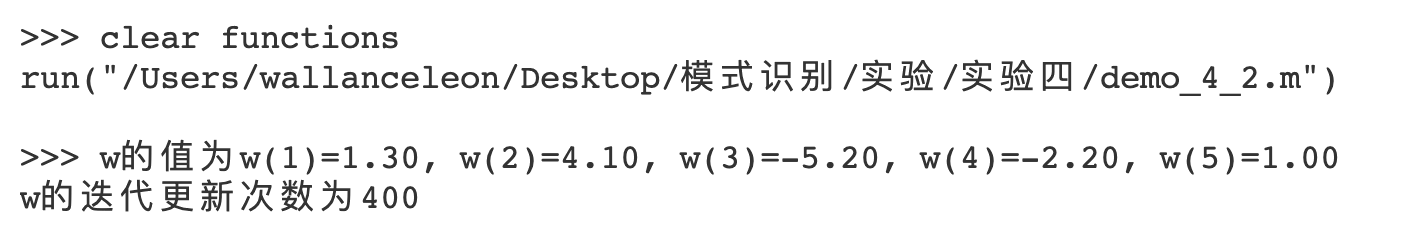
1.运行demo\_4.m文件，画出该示例程序中四个二维训练样本的分布图以及决策面图（将分布图和决策面图画在一起即可），并给出增广权向量w的值和w的迭代更新次数。  
  
w(1)=-2.00, w(2)=0.00, w(3)=1.00



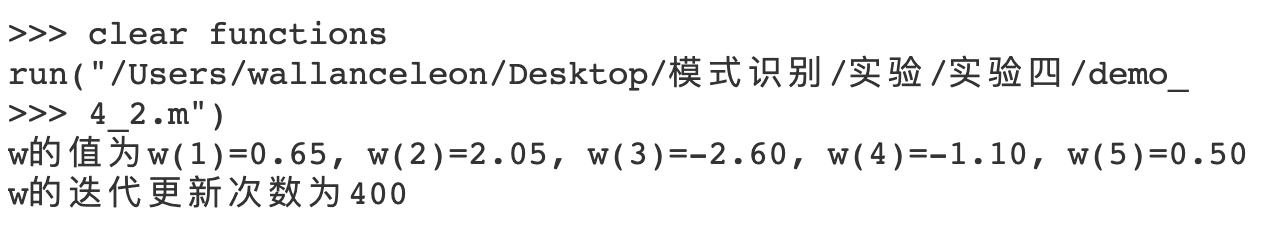
|  |
| --- |
| function [W, k] = PA(X, W, c, classes)  % X为训练样本形成的矩阵，训练样本的个数为N；W为权向量；c为校正增量  % classes为各训练样本的类别且为一个N维向量，ω1类用1表示，ω2类用-1表示  [N, n] = size(X); % 训练样本的大小N\*n，N即训练样本的个数，n即每个训练样本的维数  A = ones(N, 1);  X1 = [X A]; % 将训练样本写成增广向量形式  % 对训练样本规范化  for i = 1:N  X1(i, :) = classes(i) \* X1(i, :);  end  k = 0; % 迭代次数  a = 0; % 每一轮迭代中判别函数小于或等于0的个数，即每轮中错判的次数  b = 0; % 迭代轮数的总数  b = b + 1;  for j = 1:N  if dot(W, X1(j, :), 2) > 0  k = k + 1;  W = W;  else  a = a + 1;  W = W + c \* X1(j, :);  k = k + 1;  end  end  while (a >= 1)  a = 0;  b = b + 1;  for j = 1:N  if dot(W, X1(j, :), 2) > 0  k = k + 1;  W = W;  else  a = a + 1;  W = W + c \* X1(j, :);  k = k + 1;  end  end  end  end |

2.改写上述示例程序，采用iris数据集中的iris-setosa和iris-versicolor这两类数据进行感知器实验，分别令修正系数c=1和c=0.5，给出相应的w的值和w的迭代更新次数。

c=1.0



c=0.5



**问题分析：**

**请分析在第2部分实验中，分别取c=1和c=0.5对增广权向量w的值的影响。**

当c=1时，修正系数较大，算法的收敛速度比较快，因此w的值也相对较大。从结果可以看出，每个属性对应的权重值都比较接近，说明它们对分类的贡献相对均衡。

当c=0.5时，修正系数较小，算法的收敛速度比较慢，因此w的值也相对较小。从结果可以看出，每个属性对应的权重值都比较小，但是它们之间的差异比较大，说明某些属性对分类的贡献更加显著。

**附采用iris数据集进行感知器实验的代码（无需给出子程序PA的代码，只给出主程序代码即可）：**

|  |
| --- |
| close all;  clear all;  clc;  % 读取原始数据  [attrib1, attrib2, attrib3, attrib4, class] = textread('iris.data', '%f%f%f%f%s', 'delimiter', ',');  X = [attrib1, attrib2, attrib3, attrib4];  label\_set = char('Iris-setosa','Iris-versicolor','Iris-virginica');  label = zeros(150, 1);  label(strcmp(class, 'Iris-setosa')) = 1;  label(strcmp(class, 'Iris-versicolor')) = 2;  label(strcmp(class, 'Iris-virginica')) = 3;  % 只保留setosa和versicolor两类数据，令setosa类为正类，versicolor类为负类  X(label==3,:)=[];  label(label==3)=[];  label(label==2)=-1;  classes=label;  w0 = [0, 0, 0, 0, 0];  c = 0.5;  [w, k] = PA(X, w0, c, label);  % 输出w的值和w的迭代更新次数  fprintf('w的值为w(1)=%4.2f, w(2)=%4.2f, w(3)=%4.2f, w(4)=%4.2f, w(5)=%4.2f\n', w(1), w(2), w(3), w(4), w(5));  fprintf('w的迭代更新次数为%d\n', k); |