模式识别作业 4

分类问题

姓名：罗雁天

院系：清华大学电子系

学号：2018310742

日期：April 26, 2019

目录

第 1 章 迭代修正求权向量法

1.1 问题描述

给定两组数据：

$$\begin{aligned}\omega_1 &= \{(1,1);(2,0);(2,1);(0,2);(1,3);\}\\ \omega_2 &= \{(-1,2);(0,0);(-1,0);(-1,-1);(0,-2);\}\end{aligned} \tag{1.1}$$

求出其识别函数、识别界面以及绘制出识别界面将该训练样本的区分结果。

1.2 算法描述

使用迭代修正求权向量法对此问题进行二分类，我们有如下算法 Algorithm 1

Algorithm 1 迭代修正求权向量法二分类

输入： ω_1, ω_2, c
输出： 最终的权向量 W

将 ω_1, ω_2 用扩展特征向量 X, Y 表示，即在每一个坐标点的第三维都添加 1；
计算 $Z = [X; -Y]$
初始化权向量 W_0，并且置 $k = 0$
while True **do**
    if $\forall z \in Z, W_k^T \cdot z > 0$ **then**
        break
    else
        选出 $W_k^T \cdot z \leq 0$ 的分量组成 Z'
        计算 $Z'' = mean(Z')$
        迭代修正权向量 $W_{k+1} = W_k + c \cdot Z''$
    end if
end while
输出 $W = W_k$

1.3 实验结果

在此，我们尝试了两种初始化 W 的方法，并且对实验结果进行了对比。

1.3.1 随机初始化 W

在本小节，我们使用 Matlab 中的 $rand()$ 函数对 W 进行初始化，经过 $k=8$ 次迭代完全将两类分开。最终得到的权向量为 $[3.9857, 0.9991, -0.0660]^T$，因此，分类界面的方程为：

$$W^T \cdot z = 0 \Rightarrow 3.9857x + 0.9991y - 0.0660 = 0 \tag{1.2}$$

对任意的数据 $\mathbf{z} = [x, y]^T$，线性识别函数为：

$$f(\mathbf{z}) = \begin{cases} \mathbf{z} \in \omega_1 & if \quad 3.9857x + 0.9991y - 0.0660 > 0 \\ \mathbf{z} \in \omega_2 & if \quad 3.9857x + 0.9991y - 0.0660 \leq 0 \end{cases} \tag{1.3}$$

分类界面绘制如图1.1所示。

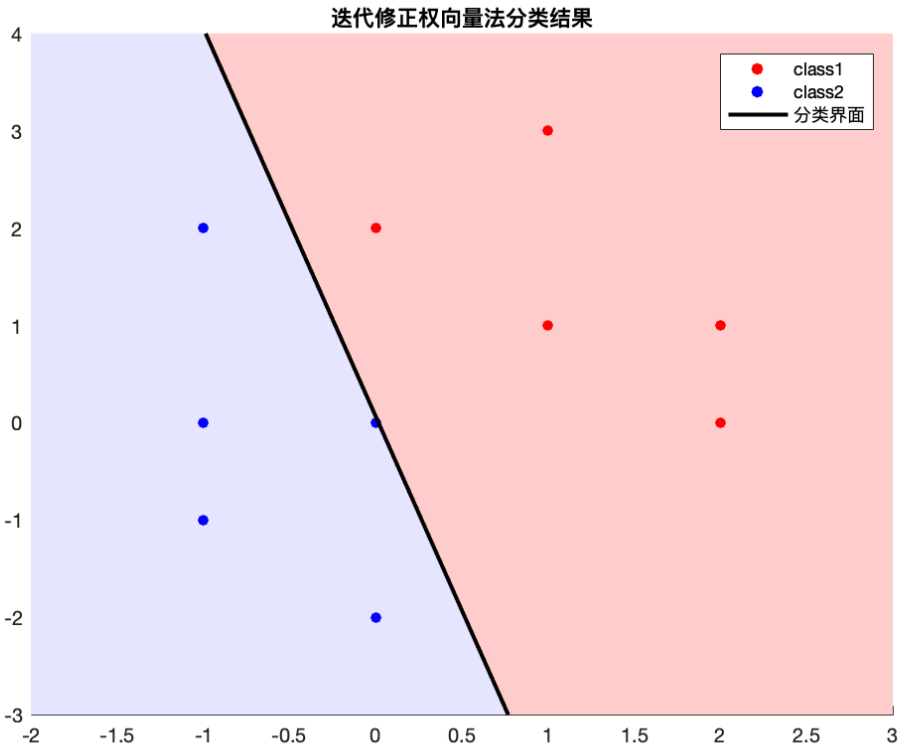

图 1.1: 使用随机初始化 W 的方式进行迭代修正求权向量法二分类结果示意图

1.3.2 全 0 初始化 W

在本小节，我们使用 Matlab 中的 $zeros()$ 函数对 W 进行初始化，经过 $k=4$ 次迭代完全将两类分开。最终得到的权向量为 $[2.4, 1.3, -1]^T$，因此，分类界面的

方程为：

$$W^T \cdot z = 0 \Rightarrow 2.4x + 1.3y - 1 = 0 \tag{1.4}$$

对任意的数据 $\mathbf{z} = [x, y]^T$，线性识别函数为：

$$f(\mathbf{z}) = \begin{cases} \mathbf{z} \in \omega_1 & if 2.4x + 1.3y - 1 > 0 \\ \mathbf{z} \in \omega_2 & if 2.4x + 1.3y - 1 \leq 0 \end{cases} \tag{1.5}$$

分类界面绘制如图1.2所示。

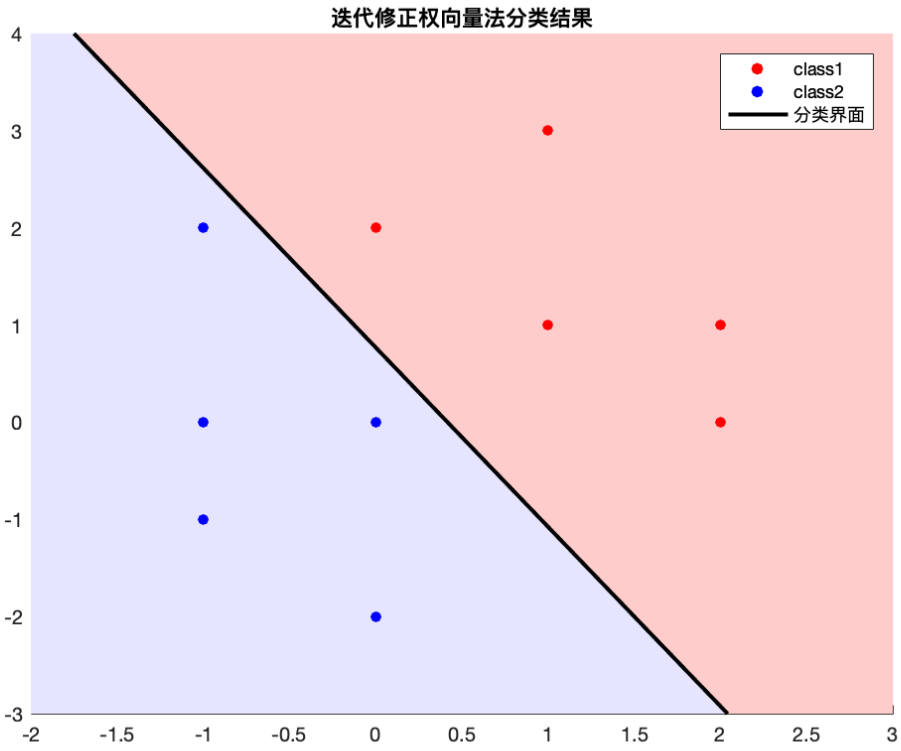

图 1.2: 使用全 0 初始化 W 的方式进行迭代修正求权向量法二分类结果示意图

第 2 章 KNN

2.1 问题描述

有两组二维数据，数据以 txt 文档的形式提供。对一个新的样本点，请尝试用 KNN 算法判断它的所属组别。

2.2 算法描述

使用 KNN 算法对数据进行二分类的算法如 Algorithm 2所示。

Algorithm 2 KNN 二分类

输入： 训练集 (X,Y)，测试数据 (z)
输出： 测试数据的类别

初始化 k 近邻的集合 *knnlist* 为前 k 个点，与 k 近邻距离的集合 *distlist* 为到前 k 个点的距离
for $x \in X$ **do**
    计算距离最大的近邻为 *maxk*，最大距离为 *maxdist*
    计算 x,z 之间的距离 *dist*
    if *dist* < *maxdist* **then**
        将 x 加入 *knnlist* 并将 *maxk* 在 *knnlist* 里删掉；
        将 *dist* 加入 *distlist* 并将 *maxdist* 在 *knnlist* 里删掉；
    end if
end for
统计 k 个最近邻样本中每个类别出现的次数
选择出现频率最大的类别作为未知样本的类别

2.3 实验结果

在实验中，我们选用不同的 k 来绘制不同的分类界面，并且对比其结果。

首先我们绘制出在 $k = 1,3,5,7$ 时的分类界面如图2.1所示，从图中可以看出，当 $k = 1$ 时，算法转换为最近邻算法，训练集上的分类正确率为 100%，但是可以看出，分类界面非常不平滑，有点过拟合；当 $k = 3$ 时，可以看出相对于 $k = 1$ 的情况，分类结果依然正确率为 100%，但是分类界面比 $k = 1$ 的情况平滑了很多；当 $k = 5$ 时，可以看出，分类界面在进一步的平滑，但是分类的正确率已经

不是 100% 了；当 $k = 7$ 时，分类界面进一步平滑，但是分类的准确率也在进一步的降低。

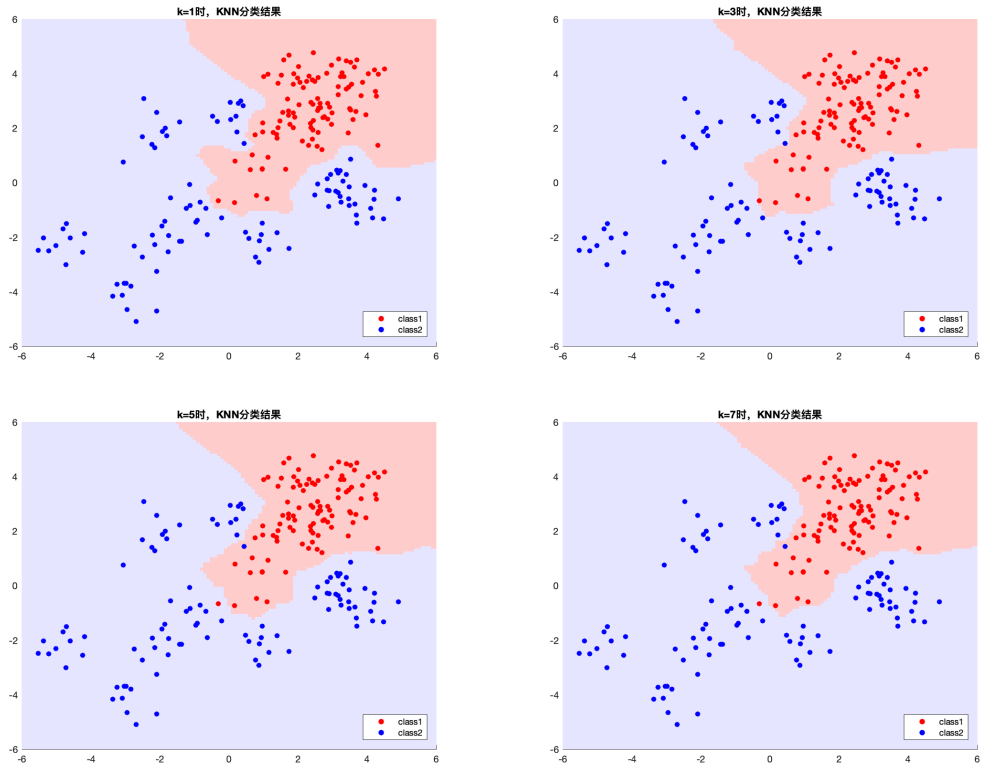

图 2.1: $k = 1,3,5,7$ 时，分类界面示意图

由此，我们可以推断出，随着 k 的增加，分类界面的平滑度会越来越高，但是训练集上分类的准确率也会越来越低，因此，我们取 $k = 1,3,5,\cdots,199$ 并计算每个 k 对应的准确率，绘制出正确率曲线如图2.2所示，从曲线中我们可以清楚地看出正确率是随着 k 的增大而逐渐变小的。同时，我们绘制出 $k = 69$ 时的分类界面如图2.3所示，从中可以看出，分类界面已经非常光滑了，但是训练集中分错的点也已经非常显而易见了。

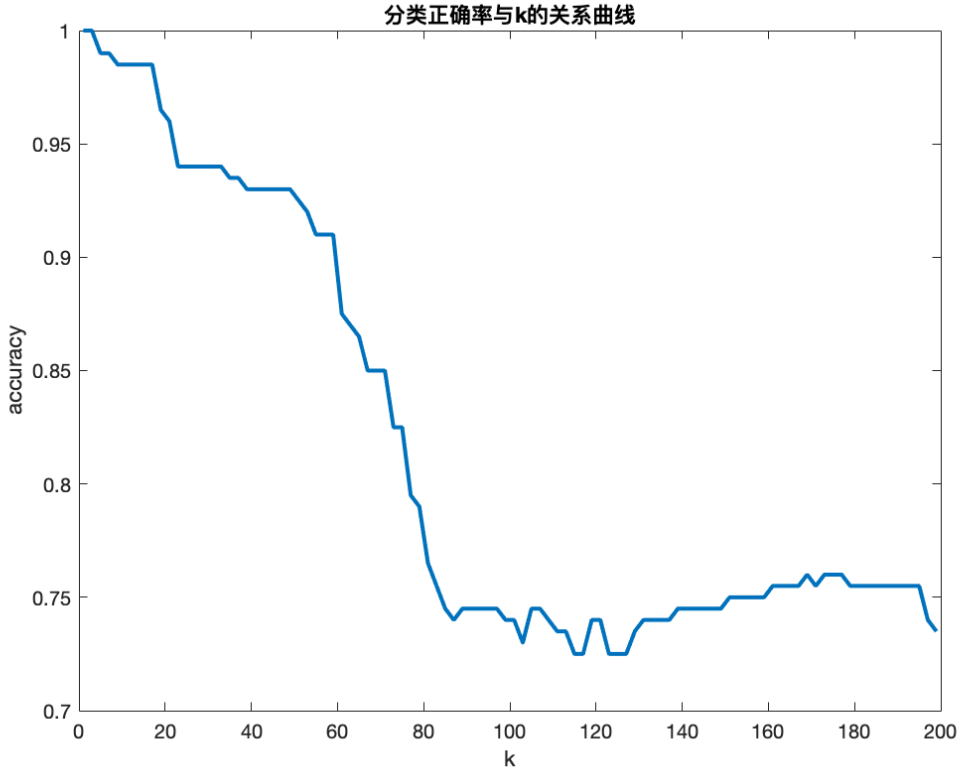

图 2.2: KNN 分类正确率与近邻数 k 的关系曲线

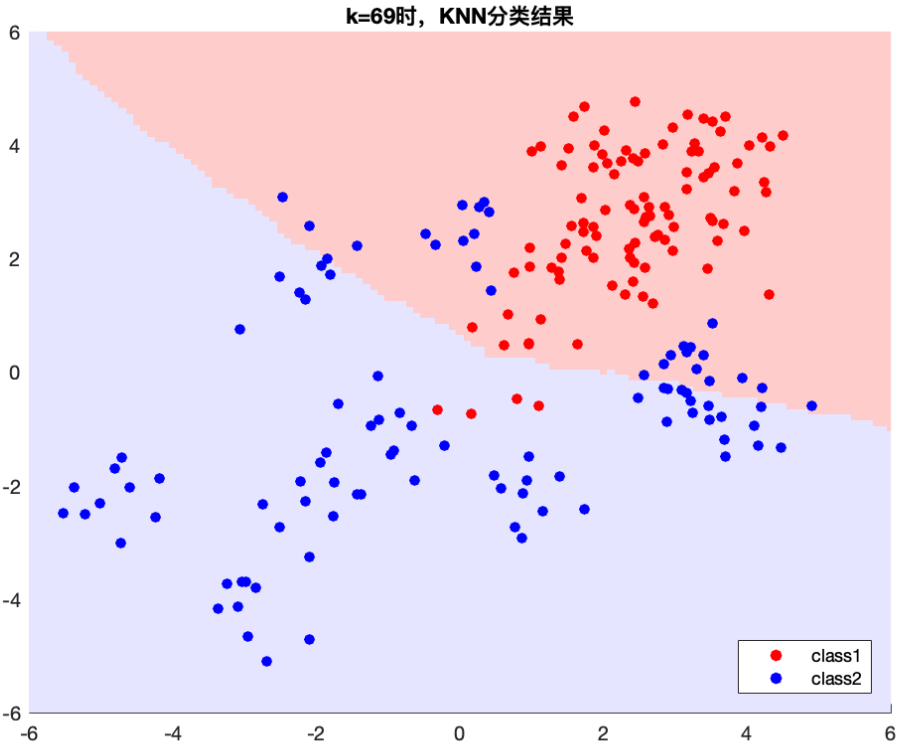

图 2.3: $k = 69$ 时，分类界面示意图

第 3 章 Fisher 判别准则

3.1 问题描述

对下列两种情况，求采用 Fisher 判别准则时的投影向量 W 和分类界面，并作图。

- 情况 1：

$$\begin{aligned}\omega_1 &= \{(2,0);(2,2);(2,4);(3,3)\}\\ \omega_2 &= \{(0,3);(-2,2);(-1,-1);(1,-2);(3,-1)\}\end{aligned} \tag{3.1}$$

- 情况 2：

$$\begin{aligned}\omega_1 &= \{(1,1);(2,0);(2,1);(0,2);(1,3)\}\\ \omega_2 &= \{(-1,2);(0,0);(-1,0);(-1,-1);(0,-2)\}\end{aligned} \tag{3.2}$$

3.2 算法描述

Fisher 判别准则本质上是找到一个投影平面，使得投影之后类内距离小、类间距离大。课件上已经给出了详细的公式推导，在此只给出相应的算法描述：

Algorithm 3 Fisher 判别准则进行二分类

输入： 训练集 (ω_1,ω_2)，测试数据 x
输出： 测试数据分类类别

1: 计算两个类的类中心：$M_i = \frac{1}{N_i}\sum_{X\in\omega_i} X, \quad i = 1,2$
2: 计算各类类内离散度矩阵：$S_i = \sum_{X\in\omega_i}(X - M_i)(X - M_i)^T$
3: 计算总类内离散度矩阵：$S_w = S_1 + S_2$
4: 计算类间离散度矩阵：$S_B = (M_1 - M_2)(M_1 - M_2)^T$
5: 计算投影向量：$W_0 = \frac{S_w^{-1}(M_1-M_2)}{\left\|S_w^{-1}(M_1-M_2)\right\|}$
6: 确定分类阈值点 y_0
7: 计算 $y = W_0^T x$
8: 如果 $y > y_0$，则 $x \in \omega_1$，否则，$x \in \omega_2$

3.3 实验结果

实验中，我们按照如下两种方式确定阈值点 y_0：

$$\begin{aligned} y_0^{(1)} &= \frac{\tilde{m}_1 + \tilde{m}_2}{2} = \frac{W_0^T(M_1 + M_2)}{2} \\ y_0^{(2)} &= \frac{N_1\tilde{m}_1 + N_2\tilde{m}_2}{N_1 + N_2} = \frac{W_0^T(N_1M_1 + N_1M_2)}{N_1 + N_2} \end{aligned} \tag{3.3}$$

3.3.1 情况 1

此种情况下，使用 $y_0^{(1)} = 1.6965$ 作为阈值点时，我们可以得到最终的权向量为 $W_0 = [0.8357, 0.5492]^T$，因此，分类界面的方程为：

$$W^T \cdot z = y_0 \Rightarrow 0.8357x + 0.5492y - 1.6965 = 0 \tag{3.4}$$

对任意的数据 $\mathbf{z} = [x, y]^T$，线性识别函数为：

$$f(\mathbf{z}) = \begin{cases} \mathbf{z} \in \omega_1 & if \quad 0.8357x + 0.5492y - 1.6965 > 0 \\ \mathbf{z} \in \omega_2 & if \quad 0.8357x + 0.5492y - 1.6965 \leq 0 \end{cases} \tag{3.5}$$

绘制出分类界面示意图如图3.1所示。

当采用 $y_0^{(2)} = 1.5388$ 作为阈值点时，我们可以得到最终的权向量为 $W_0 = [0.8357, 0.5492]^T$，因此，分类界面的方程为：

$$W^T \cdot z = y_0 \Rightarrow 0.8357x + 0.5492y - 1.5388 = 0 \tag{3.6}$$

对任意的数据 $\mathbf{z} = [x, y]^T$，线性识别函数为：

$$f(\mathbf{z}) = \begin{cases} \mathbf{z} \in \omega_1 & if \quad 0.8357x + 0.5492y - 1.5388 > 0 \\ \mathbf{z} \in \omega_2 & if \quad 0.8357x + 0.5492y - 1.5388 \leq 0 \end{cases} \tag{3.7}$$

绘制出分类界面示意图如图3.2所示。

对比两种情况可以发现，对训练集数据分类的正确率并没有达到 100%，这是由于此问题在二维空间是线性不可分的，使用 Fisher 判别准则仅仅是使得类内距离小，类间距离大，并没有保证完全将其分开。并且，使用两种判别阈值的结果差别不大。

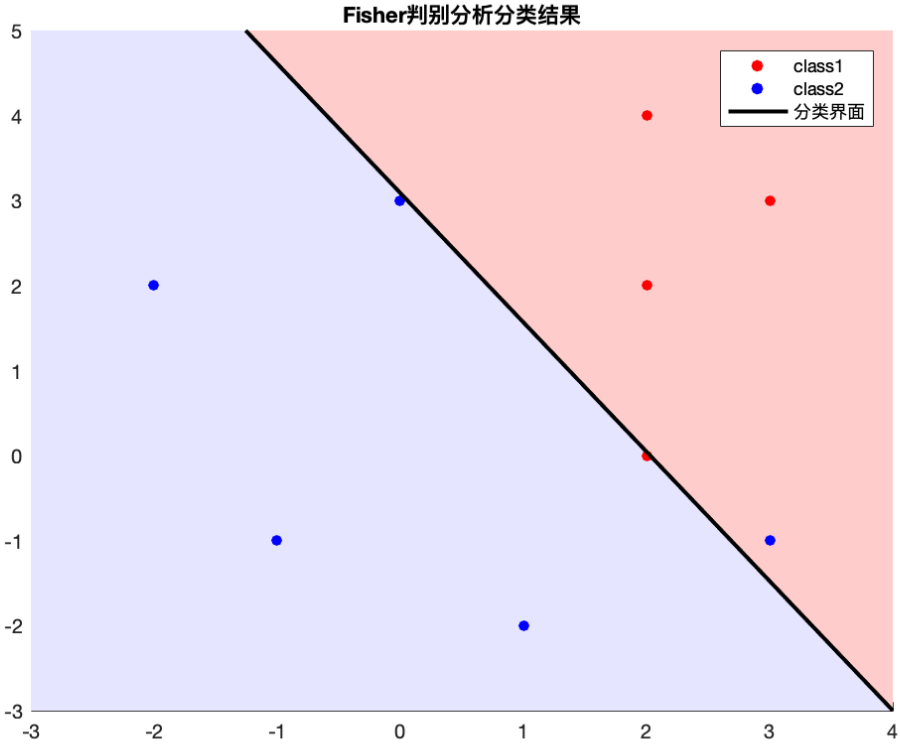

图 3.1: 使用 Fisher 判别分析对情况 1 中数据进行分类的结果 (阈值采用 $y_0^{(1)}$)

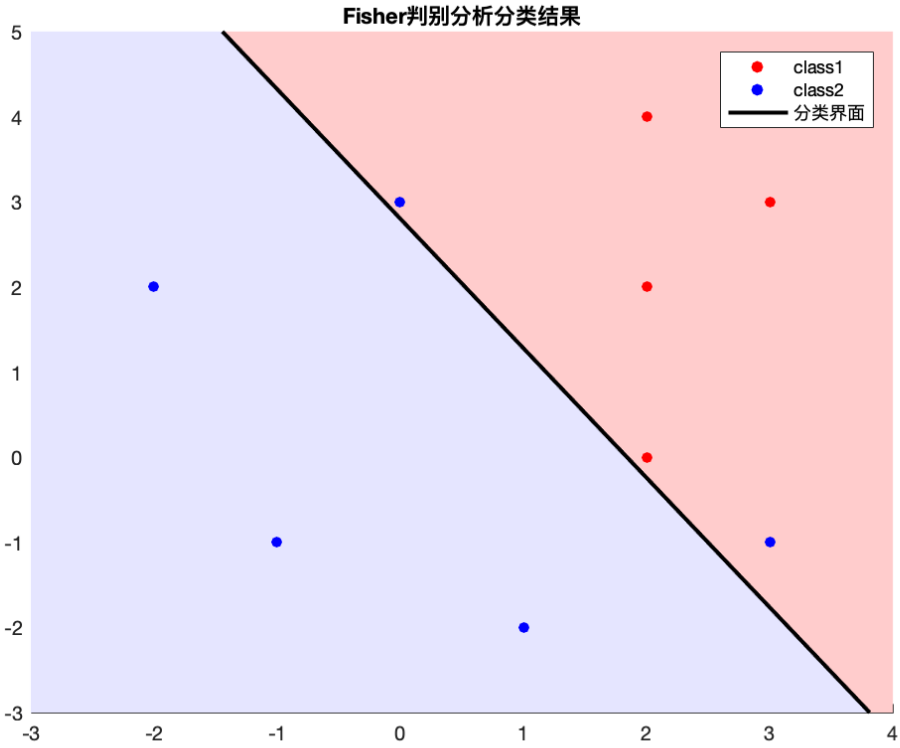

图 3.2: 使用 Fisher 判别分析对情况 1 中数据进行分类的结果 (阈值采用 $y_0^{(2)}$)

3.3.2 情况 2

由于情况 2 中正负样本数是一致的，因此 $y_0^{(1)} = y_0^{(2)}$，所以在此只进行了一次实验。得到最终的权向量为 $W_0 = [0.9185, 0.3953]^T$，阈值 $y_0 = 0.5128$ 因此，分类界面的方程为：

$$W^T \cdot z = y_0 \Rightarrow 0.9185x + 0.3953y - 0.5128 = 0 \tag{3.8}$$

对任意的数据 $\mathbf{z} = [x, y]^T$，线性识别函数为：

$$f(\mathbf{z}) = \begin{cases} \mathbf{z} \in \omega_1 & if \quad 0.9185x + 0.3953y - 0.5128 > 0 \\ \mathbf{z} \in \omega_2 & if \quad 0.9185x + 0.3953y - 0.5128 \leq 0 \end{cases} \tag{3.9}$$

绘制出分类界面示意图如图3.3所示。

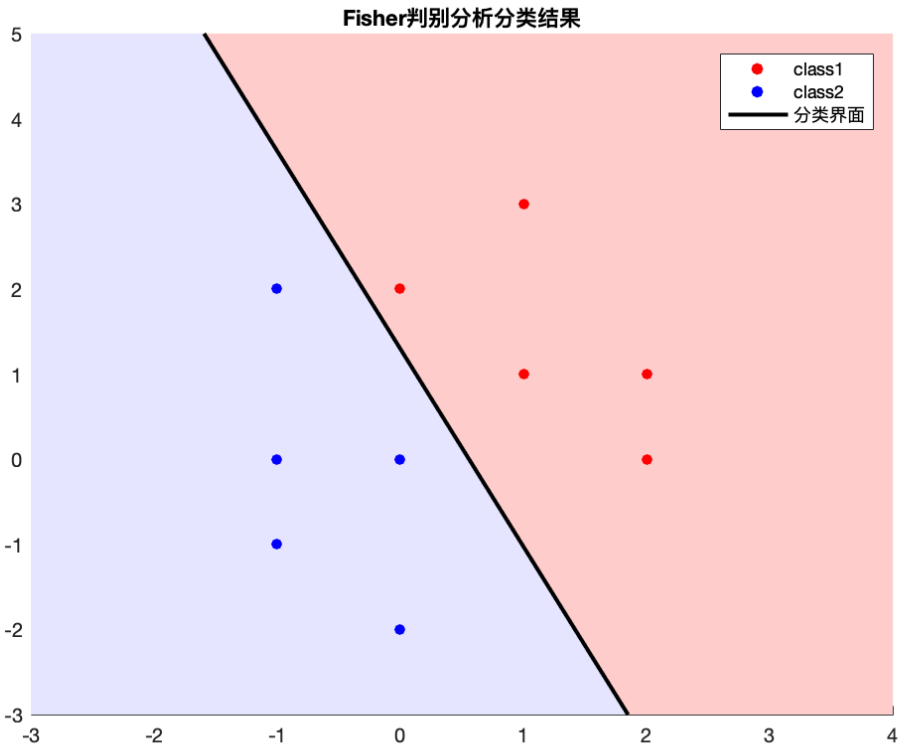

图 3.3: 使用 Fisher 判别分析对情况 2 中数据进行分类的结果

由于此问题是线性可分的问题，因此可以得到较好的分类界面，训练数据的分类正确率也达到了 100%，与第一章中迭代修正求权向量的方法相比，此种方法对权向量的初始值不敏感，分类界面直观上也比随机初始化权向量的迭代修正求权向量的方法好。

第 4 章 反思与总结

至此，我们已经学过的二分类算法有最小欧式距离分类、迭代修正权向量法分类、Fisher 判别准则分类，在此，我们使用如下数据：

$$\begin{aligned}\omega_1 &= \{(1,1);(2,0);(2,1);(0,2);(1,3)\}\\ \omega_2 &= \{(-1,2);(0,0);(-1,0);(-1,-1);(0,-2)\}\end{aligned} \tag{4.1}$$

将三种算法的结果与 SVM 分类的结果进行对比如图4.1所示，从直观上来看，对于此数据集来说，使用 0 初始化的迭代修正权向量法、Fisher 判别分析法和 SVM 分类的效果均比最小欧式距离分类的效果好。

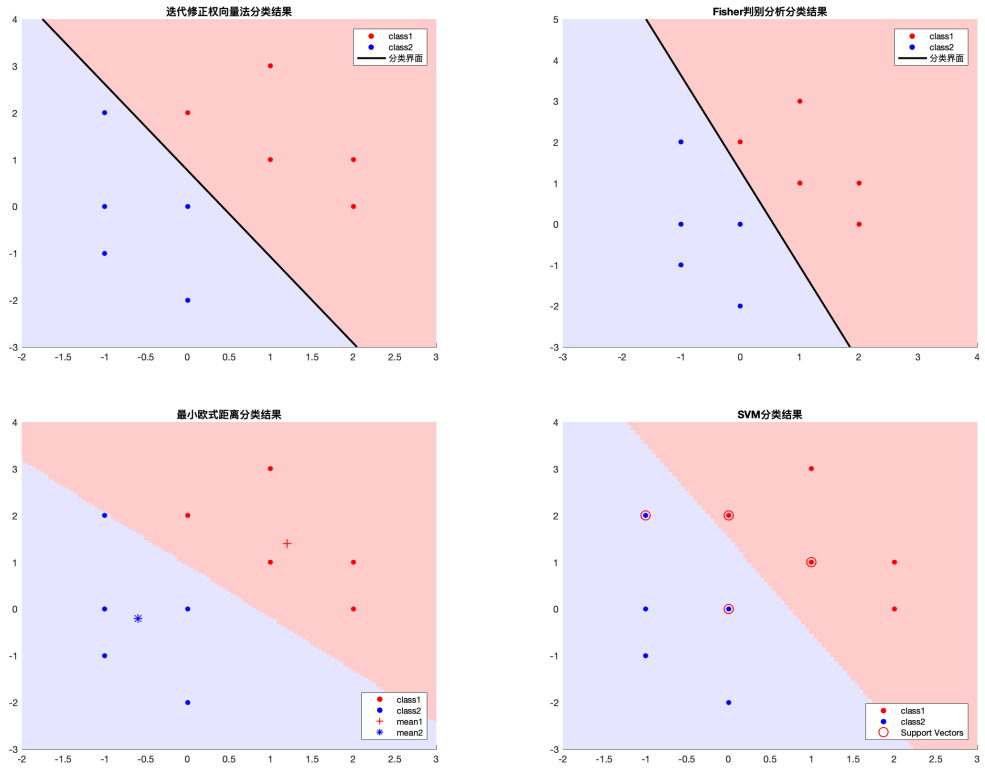

图 4.1: 四种二分类结果对比图

第 5 章 代码说明

本次实验使用 Matlab 语言编写，所有代码放置在“code/”文件夹下:

- iterative.m: 迭代修正求权向量法分类的主程序，直接执行便可以得到分类界面；
- knnclassify.m: KNN 算法分类的主程序，直接执行便可以得到不同 k 下的分类界面以及正确率曲线；
- fishermain.m: Fisher 判别分析算法分类的主程序，直接执行便可以得到情况 1 和情况 2 的分类界面图像；
- fisherclassify.m: 使用 Fisher 判别分析分类的函数，输入 ω_1, ω_2，输出权向量 W_0 和分类阈值 y_0，并且绘制分类界面图像。
- compare.m: 使用最小欧式距离分类和 SVM 分类的代码，直接执行绘制分类界面的图像。