



模式识别作业 3

分类问题

姓名：罗雁天

院系：清华大学电子系

学号：2018310742

日期：April 19, 2019



目录

1	线性分类问题	1
1.1	问题描述	1
1.2	最小欧式距离分类	1
1.2.1	算法描述	1
1.2.2	实验结果	1
1.3	支持向量机 SVM 分类	3
1.3.1	算法描述	3
1.3.2	实验结果	3
2	非线性分类问题	5
2.1	问题描述	5
2.2	非线性映射法	5
2.2.1	算法设计	5
2.2.2	实验结果	6
2.3	SVM+ 核函数方法	6
2.4	总结	8
3	代码说明	9

第 1 章 线性分类问题

1.1 问题描述

给定两组数据：

$$\begin{aligned}\omega_1 &= \{(2, 3); (2, 2); (2, 4); (3, 3); (3, 4); (2.5, 3); (1.5, 2); (3.5, 2.5); (4, 4); (0.5, 0.5)\} \\ \omega_2 &= \{(0, 2.5); (-2, 2); (-1, -1); (1, -2); (3, 0); (-2, -2); (-3, -4); (-5, -2); (4, -1)\}\end{aligned}\quad (1.1)$$

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 最小欧式距离分类

1.2.1 算法描述

使用最小欧式距离对此问题进行二分类，我们有如下算法 Algorithm 1

Algorithm 1 最小欧式距离二分类

输入： ω_1, ω_2

输出： 识别函数，分类界面

- 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 $d_i(x) = x^T \cdot M_i - \frac{1}{2} M_i^T \cdot M_i, \quad i = 1, 2$
 - 3: 所以识别函数为： $f(x) = \begin{cases} x \in class1 & \text{如果 } d_1 \geq d_2 \\ x \in class2 & \text{如果 } d_1 < d_2 \end{cases}$
 - 4: 设分类界面上任意一点为 $p = (p_1, p_2)^T$
 - 5: 则分类界面为： $\left(p^T - \frac{M_1^T + M_2^T}{2}\right) \cdot (M_1 - M_2) = 0$
-

1.2.2 实验结果

使用 Algorithm 1，我们计算出：

$$\begin{aligned}M_1 &= [2.4, 2.8]^T \\ M_2 &= [-0.5556, -0.8333]^T\end{aligned}\quad (1.2)$$

那么对于任意的数据点 $\mathbf{x} = [x_1, x_2]^T$ ，我们有：

$$\begin{aligned} d_1 &= \mathbf{x}^T \cdot \mathbf{M}_1 - \frac{1}{2} \mathbf{M}_1^T \cdot \mathbf{M}_1 = 2.4x_1 + 2.8x_2 - 6.8 \\ d_2 &= \mathbf{x}^T \cdot \mathbf{M}_2 - \frac{1}{2} \mathbf{M}_2^T \cdot \mathbf{M}_2 = -\frac{5}{9}x_1 - \frac{5}{6}x_2 - \frac{325}{648} \end{aligned} \quad (1.3)$$

所以我们的识别函数为：

$$f(\mathbf{x}) = \begin{cases} x \in \text{class1} & \text{如果 } d_1 \geq d_2 \\ x \in \text{class2} & \text{如果 } d_1 < d_2 \end{cases} \quad (1.4)$$

设分类界面上任意一点为 $p = (p_1, p_2)^T$ ，那么我们可以得到分类界面为：

$$\begin{aligned} \left(p^T - \frac{\mathbf{M}_1^T + \mathbf{M}_2^T}{2} \right) \cdot (\mathbf{M}_1 - \mathbf{M}_2) &= 0 \\ \Rightarrow 2.9556p_1 + 3.6333p_2 - 6.2985 &= 0 \end{aligned} \quad (1.5)$$

绘制出分类界面如图1.1所示

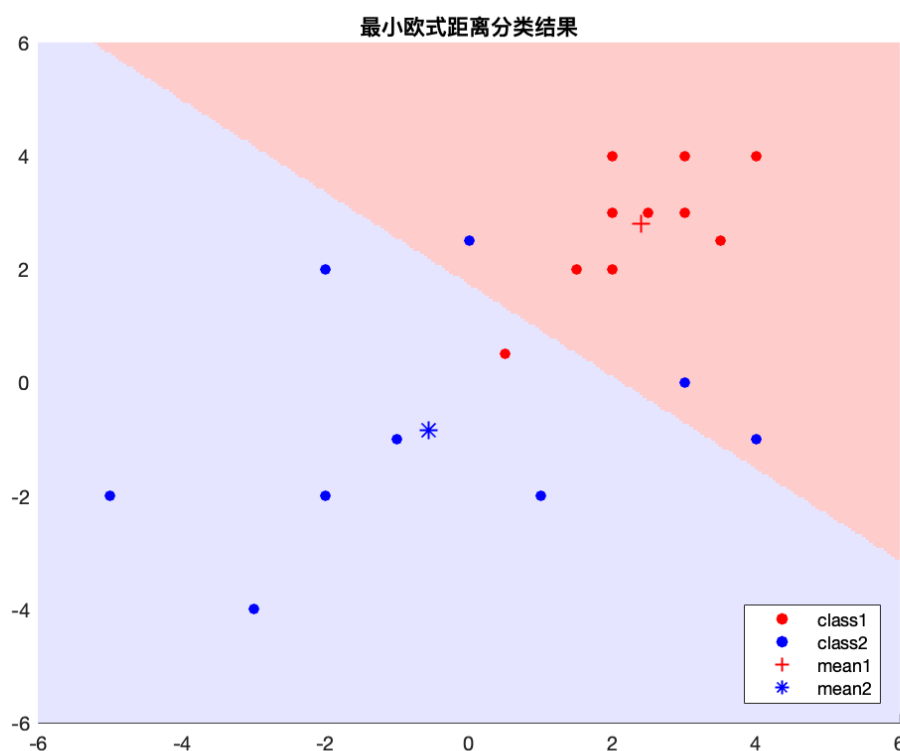


图 1.1: 使用最小欧式距离分类所得的分类界面示意图

我们可以统计出 class1 和 class2 的召回率和虚警率如表1.1所示，从中可以看出，最小距离分类会带来一定的虚警率，class2 误判为 class1 的较多。

表 1.1: 最小欧式距离分类的召回率和虚警率

类别	召回率	虚警率
class1	9/10	3/9
class2	6/9	1/10

1.3 支持向量机 SVM 分类

1.3.1 算法描述

SVM 的目标是寻找一个超平面，使得它不仅能够正确地分类每一个样本，并且要使得每一类样本中距离超平面最近的样本到超平面的距离尽可能远。可以将其转换为求解如下优化问题：

$$\begin{aligned} \min \quad & \frac{1}{2} w^T w \\ \text{s.t.} \quad & y_i(w^T x_i + b) \geq 1 \end{aligned} \quad (1.6)$$

其中， (x_i, y_i) 是训练样本， $y_i \in \{1, -1\}$ ， w, b 为超平面的权重。对于此优化问题，我们可以使用拉格朗日对偶转换为对偶问题，对偶问题可以化简为：

$$\begin{aligned} \min \quad & -\frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j x_i^T x_j + \sum_{i=1}^l \alpha_i \\ \text{s.t.} \quad & \alpha_i \geq 0, \quad i = 1, 2, \dots, l \\ & \sum_{i=1}^l \alpha_i y_i = 0 \end{aligned} \quad (1.7)$$

然后求解此二次规划问题便可以得到分类界面。

1.3.2 实验结果

使用 SVM 分类的结果如图 1.2 所示：

同样，我们也统计出此种方法下 class1 和 class2 的召回率和虚警率如表 1.2 所示，从中可以看出，使用 SVM 分类的效果要比最小欧式距离分类的效果好。

表 1.2: SVM 分类的召回率和虚警率

类别	召回率	虚警率
class1	9/10	1/9
class2	8/9	1/10

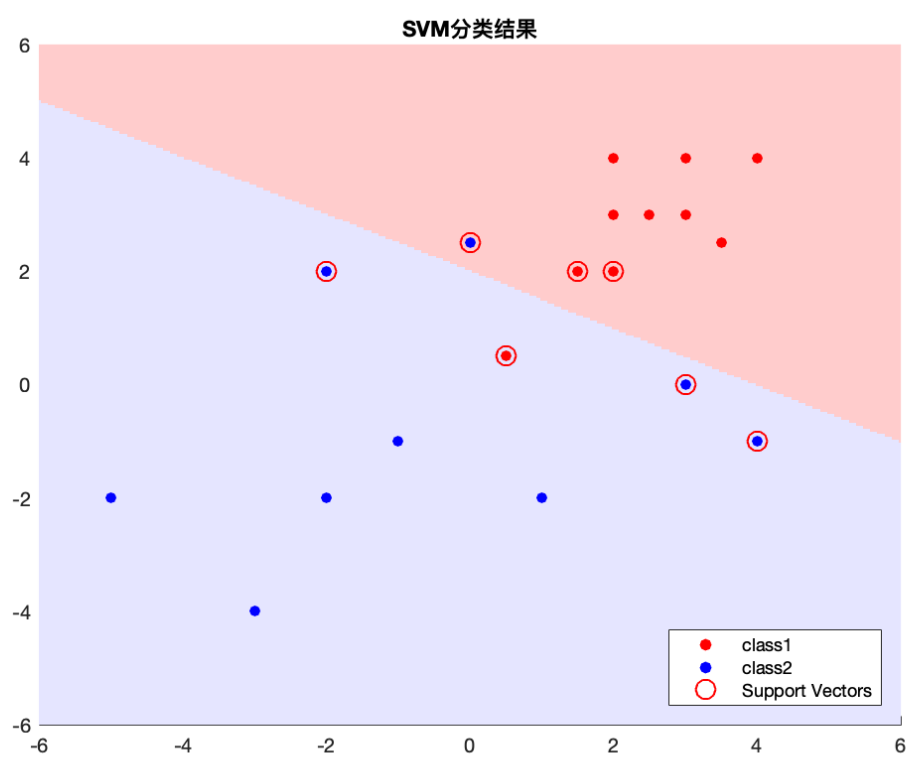


图 1.2: 使用最小欧式距离分类所得的分类界面示意图

第 2 章 非线性分类问题

2.1 问题描述

如图2.1所示，求分离●和▲的函数

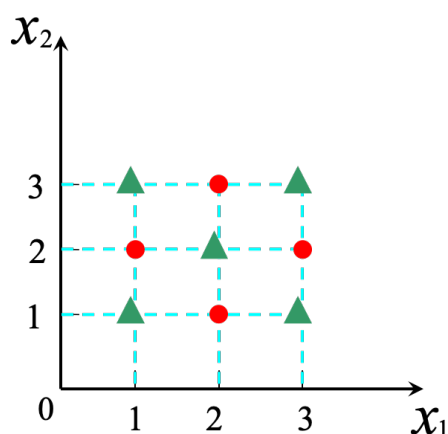


图 2.1: 非线性分类问题

2.2 非线性映射法

在本小节，我们首先使用非线性映射，将二维数据升到三维，然后再使用 Algorithm 1 中提到的最小欧式距离分类算法进行分类。

2.2.1 算法设计

观察数据点的分布可以看出，正负样本可以通过两个等轴双曲线进行分类，因此，对任意的数据点 $\mathbf{x} = [x_1, x_2]^T$ 我们首先设计了如下的非线性映射函数：

$$\phi_1(x_1, x_2) = \left[x_1, x_2, \left((x_1 - 2)^2 - (x_2 - 2)^2 \right)^2 \right] \quad (2.1)$$

在经过非线性映射之后，我们便可以将其看做是三维空间的线性分类问题，使用算法 Algorithm 1 或者线性 SVM 便可求出分类界面。

从数据点的分布我们还可以看出，可以通过两个同心圆对数据进行分类，因

此我们设计了如下的非线性映射函数：

$$\phi_2(x_1, x_2) = \left[x_1, x_2, \left((x_1 - 2)^2 + (y_1 - 2)^2 - 1 \right)^2 \right] \quad (2.2)$$

同样，转换之后可以用线性分类的方法进行分类。

2.2.2 实验结果

线性最小欧氏距离 + 非线性映射实验结果

使用上一小节设计的两种非线性映射函数进行线性最小欧氏距离分类实验，可以得到如图2.2和2.3所示的分类界面，可以看出分类结果正确，使用非线性映射的方式成功将线性不可分的问题转化为线性可分问题。

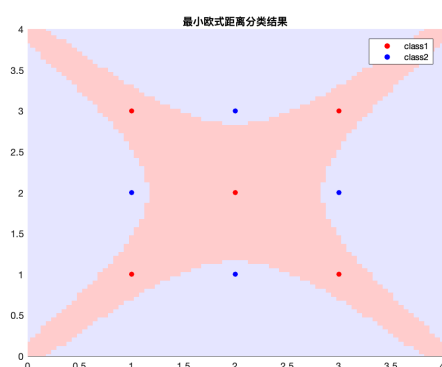


图 2.2: 最小欧式距离分类 + 非线性映射 1 的分类界面示意图

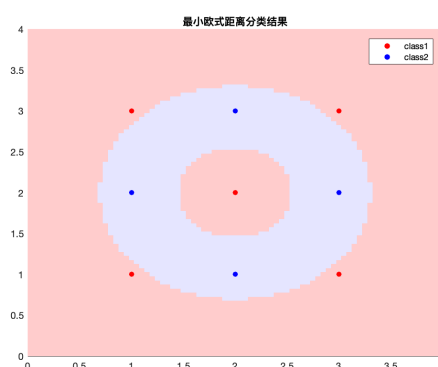


图 2.3: 最小欧式距离分类 + 非线性映射 2 的分类界面示意图

SVM+ 非线性映射实验结果

使用上一小节设计的两种非线性映射函数进行线性 SVM 分类实验，可以得到如图2.4和2.5所示的分类界面，可以看出分类结果正确，使用非线性映射的方式成功将线性不可分的问题转化为线性可分问题。

2.3 SVM+ 核函数方法

在使用 SVM 中，我们可以使用核函数来代替非线性映射的方式对非线性问题进行分类，具体而言，构造核函数 $k(x_1, x_2)$ 使得：

$$k(x_1, x_2) = \phi(x_1)^T \phi(x_2) \quad (2.3)$$

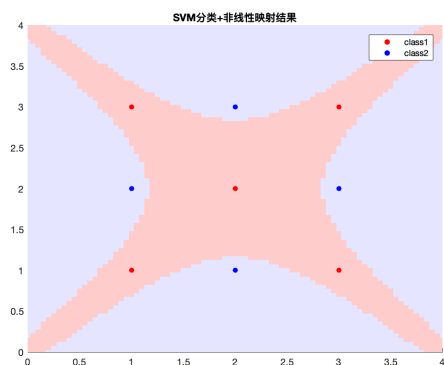


图 2.4: SVM 分类 + 非线性映射 1 的分类界面示意图

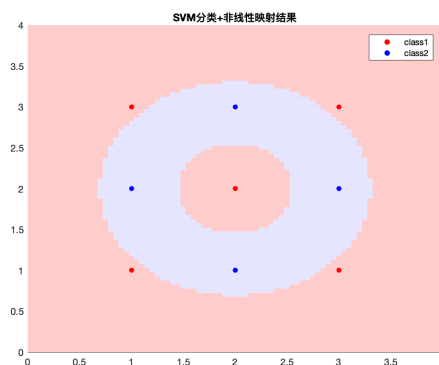


图 2.5: SVM 分类 + 非线性映射 2 的分类界面示意图

这样便可以将线性 SVM 中最后的优化问题转换为：

$$\begin{aligned}
 \min \quad & -\frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j k(x_i, x_j) + \sum_{i=1}^l \alpha_i \\
 \text{s.t.} \quad & \alpha_i \geq 0, \quad i = 1, 2, \dots, l \\
 & \sum_{i=1}^l \alpha_i y_i = 0
 \end{aligned} \tag{2.4}$$

常见的核函数有 RBF、Gaussian、polynomial 等核函数，在此我们选择 polynomial 核函数进行实验，绘制出分类界面如图2.6所示，可以看出并没有完全分类正确，因此，使用 polynomial 核函数的 SVM 相比于自己设计非线性映射的效果更差一些。

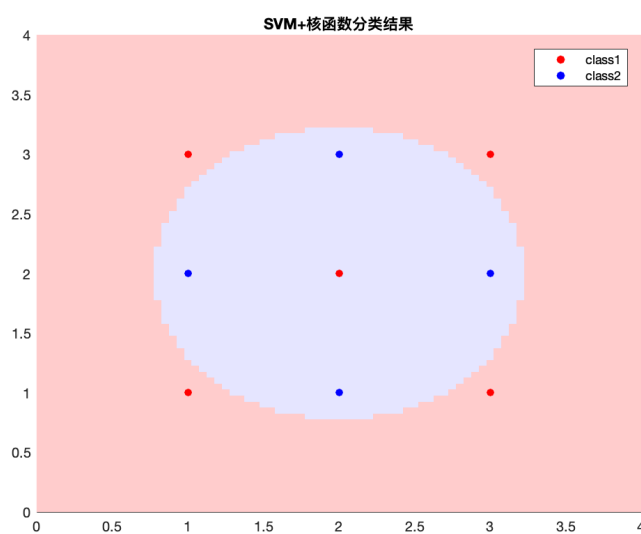


图 2.6: SVM 分类 + 核函数的分类界面示意图

2.4 总结

从提出的三种非线性分类的算法来看，设计好的非线性映射能够带来好的分类效果，使用最小欧式距离分类 + 非线性映射算法和 SVM 分类 + 非线性映射算法分类效果较好，而使用 `polynomial` 核函数的分类效果就比较差了。因此，对于不同的非线性分类问题，设计一种较好的非线性映射是一个较好的方法，但是需要针对不同问题分别设计，这样鲁棒性不高。

第 3 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：

- linearclassify.m: 线性分类的主程序，直接执行便可以得到分类界面；
- plot_boundary.m: 绘制线性分类器的分类界面；
- nonlinearclassify.m: 非线性分类的主程序，直接执行便可以得到分类界面；
- mykernel.m: 设计的非线性映射函数。