机器学习报告 # 1

从感知机到支撑向量机*

董晟渤, 统计 91, 2193510853

西安交通大学数学与统计学院

日期：2022 年 4 月

目录

*2021-2022 学年第二学期, 课程: 机器学习, 指导老师: 孟德宇.

1 概述

机器学习的一大任务是分类.

设正整数 $p \geq 1$. 假设在 p 维空间 $\mathbb{R}^p$ 中, 有训练数据 $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n \in \mathbb{R}^p$. 这些数据可以被分为两组, 并且对数据 $\boldsymbol{x}_i (1 \leq i \leq n)$, 引入实数 $y_i \in \{-1, 1\}$, 来表示第 i 个数据所属的组, 从而训练数据集可以表示为

$$T = \{(\boldsymbol{x}_i, y_i), 1 \leq i \leq n\}.$$

具体地说, 这些向量有可能:

- 是线性可分的, 也即存在一个 $p-1$ 维空间 $\mathbb{R}^{p-1}$ 中的超平面 (当 $p=2$ 时即为直线, 当 $p=3$ 时即为平面), 使得满足 $y_i = -1$ 和 $y_i = 1$ 的数据分别落在这个超平面的两侧;
- 或者, 不是线性可分的, 也即找不到满足上点的超平面.

设超平面的法向量 $\boldsymbol{w} \in \mathbb{R}^p$, 截距 $b \in \mathbb{R}$, 则超平面可以用 $\boldsymbol{w}^T \boldsymbol{x} + b = 0$ 表示. 为了方便, 也常常记 $\tilde{\boldsymbol{x}} = (\boldsymbol{x}, 1)^T \in \mathbb{R}^{p+1}$, $\tilde{\boldsymbol{w}} = (\boldsymbol{w}, b)^T \in \mathbb{R}^{p+1}$, 并设 $\|\tilde{\boldsymbol{w}}\| = 1$, 则

$$\boldsymbol{w}^T \boldsymbol{x} + b = 0 \iff \tilde{\boldsymbol{w}}^T \tilde{\boldsymbol{x}} = 0.$$

从而, 我们将线性可分的概念抽象为数学语言: 所谓的线性可分, 就是存在 $\gamma > 0$, 使得对任意的 $1 \leq i \leq n$, 都有

$$y_i \cdot \left(\boldsymbol{w}^T \boldsymbol{x_i} + b\right) > \gamma \iff y_i \cdot (\tilde{\boldsymbol{w}}^T \tilde{\boldsymbol{x}}_i) > \gamma.$$

- 若 $y_i = -1$, 则 $\tilde{\boldsymbol{w}}^T \tilde{\boldsymbol{x}}_i < 0$, 此时 $\tilde{\boldsymbol{x}}_i$ 落在 $\tilde{\boldsymbol{w}}^T \tilde{\boldsymbol{x}}_i = 0$ 的一侧, 或者说, $\boldsymbol{x}_i$ 落在 $\boldsymbol{w}^T \boldsymbol{x} + b = 0$ 的一侧;
- 若 $y_i = 1$, 则 $\tilde{\boldsymbol{w}}^T \tilde{\boldsymbol{x}}_i > 0$, 此时 $\tilde{\boldsymbol{x}}_i$ 落在 $\tilde{\boldsymbol{w}}^T \tilde{\boldsymbol{x}}_i = 0$ 的另一侧, 或者说, $\boldsymbol{x}_i$ 落在 $\boldsymbol{w}^T \boldsymbol{x} + b = 0$ 的另一侧;

我们要解决的问题是: 对于线性可分的数据, 能否找到满足上述条件的超平面 (直线、平面)? 更进一步, 能否找到能最好地区分这些数据的超平面? 而对于不是线性可分的数据, 能否找到最好地区分这些数据的超平面? 从另外一个角度看, 是否可以找到一个函数 $f : \mathbb{R}^d \to \{-1, 1\}$, 输入数据 $\boldsymbol{x}_i$, 输出的结果 $f(\boldsymbol{x}_i)$ 是这个数据的分类 y_i?

在本篇报告中引入的感知机模型, 可以对线性可分的数据找到满足条件的超平面, 但是找到的超平面可能有无穷多个, 对应的函数即为感知机; 而支撑向量机, 进一步解决了下一个问题, 也即无论数据是否是线性可分的, 都找到一个确定的超平面, 最好地区分这些数据.

2 感知机 (Perceptron)

2.1 感知机解决的问题

设训练数据 $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n \in \mathbb{R}^p$, 对数据 $\boldsymbol{x}_i (1 \leq i \leq n)$, 用 $y_i \in \{-1, 1\}$ 表示 $\boldsymbol{x}_i$ 所属的组, 考虑训练数据集 $T = \{(\boldsymbol{x}_i, y_i), 1 \leq i \leq n\}$. 在本节中, 我们仍记 $\tilde{x} = (\boldsymbol{x}, 1) \in \mathbb{R}^{p+1}$, 并且假定 T 满足以下假设:

- **线性可分**, 也即存在 $\tilde{\boldsymbol{w}} = (\boldsymbol{w}, b) \in \mathbb{R}^{p+1}$ 及 $\gamma > 0$, 使得对任意的 $1 \leq i \leq n$, 都有

$$y_i \cdot \left(\tilde{\boldsymbol{w}}^T \tilde{\boldsymbol{x}}_i\right) > \gamma;$$

- **有界**, 也即存在 $R > 0$, 使得对任意的 $1 \leq i \leq n$, 都有

$$\|\tilde{\boldsymbol{x}}_i\| \leq R.$$

对于线性可分且有界的训练数据, 我们希望能够找到一个超平面 $\tilde{\boldsymbol{w}}^T \tilde{\boldsymbol{x}} = 0$, 其对应着超平面 $\boldsymbol{w}^T \boldsymbol{x} + b = 0$, 可以将属于不同的组的数据分离开. 或者, 我们希望能够找到一个简单的函数 $f : \mathbb{R}^d \to \{-1, 1\}$, 使得对任意的 $1 \leq i \leq n$, 都有 $f(\boldsymbol{x}_i) = y_i$.

2.2 感知机的算法与原理

考虑以下算法:

算法 1: 感知机模型

$k \leftarrow 1$;

$\tilde{\boldsymbol{w}}_k \leftarrow \tilde{\boldsymbol{w}}_0$;

while 存在 $1 \leq i \leq n$, 使得 $y_i \cdot \left(\tilde{\boldsymbol{w}}_k^T \tilde{\boldsymbol{x}}_i\right) \leq 0$ **do**

  $\tilde{\boldsymbol{w}}_{k+1} \leftarrow \tilde{\boldsymbol{w}}_k + y_i \cdot \tilde{\boldsymbol{x}}_i$;

  $k \leftarrow k + 1$;

end

output $\dfrac{\tilde{\boldsymbol{w}}_k}{\|\tilde{\boldsymbol{w}}_k\|}$.

算法1看起来非常简单, 只有一个简单的迭代, 但是却是一个强大的算法: 它不但能找到一个满足条件的超平面 $\boldsymbol{w}^T \boldsymbol{x} = 0$, 并且只需要进行有限的步骤.

定理 2.1 (Novikoff)**.** 设训练数据集 $T = \{(\boldsymbol{x}_i, y_i), 1 \leq i \leq n\}$ 线性可分且有界.

(1) 存在超平面 $\tilde{\boldsymbol{w}}^T\tilde{\boldsymbol{x}}=0$ 及 $\gamma>0$, 使得 $\|\tilde{\boldsymbol{\omega}}\|=1$, 且对任意的 $1\le i\le n$, 都有

$$y_i\cdot\left(\tilde{\boldsymbol{w}}^T\tilde{\boldsymbol{x}}_i\right)>\gamma; \tag{1}$$

(2) 设对任意的 $1\le i\le n$, 都有 $\|\tilde{x}_i\|\le R$, 则算法1的计算次数

$$k\le\frac{R^2}{\gamma^2}.$$

证明. (1) 由 T 是线性可分的, 知存在 $\tilde{\boldsymbol{\omega}}\in\mathbb{R}^{p+1}$ 及 $\gamma>0$, 使得对任意的 $1\le i\le n$, 都有

$$y_i\cdot\left(\tilde{\boldsymbol{w}}^T\tilde{\boldsymbol{x}}_i\right)>\gamma;$$

若 $\|\tilde{\boldsymbol{\omega}}\|=1$, 则定理成立; 否则, 记 $\tilde{\boldsymbol{w}}'=\dfrac{\tilde{\boldsymbol{w}}}{\|\tilde{\boldsymbol{w}}\|}$, 则

$$\|\tilde{\boldsymbol{w}}'\|=\left\|\frac{\tilde{\boldsymbol{w}}}{\|\tilde{\boldsymbol{w}}\|}\right\|=\frac{\|\tilde{\boldsymbol{w}}\|}{\|\tilde{\boldsymbol{w}}\|}=1,$$

并且

$$y_i\cdot\left(\tilde{\boldsymbol{w}}'^T\tilde{\boldsymbol{x}}_i\right)=\frac{1}{\|\tilde{\boldsymbol{w}}\|}\cdot y_i\cdot\left(\tilde{\boldsymbol{w}}^T\tilde{\boldsymbol{x}}_i\right)>\frac{\gamma}{\|\tilde{\boldsymbol{w}}\|}>0.$$

记 $\gamma'=\min\limits_{1\le i\le n}y_i\cdot\left(\tilde{\boldsymbol{w}}'^T\tilde{\boldsymbol{x}}_i\right)>0$, 则 $\tilde{\boldsymbol{w}}'$ 和 γ' 满足方程(1).

(2) 设 $k\ge1$, 若存在 $1\le i\le n$, 使得 $y_i\cdot\left(\tilde{\boldsymbol{w}}_k^T\tilde{\boldsymbol{x}}_i\right)\le0$, 一方面, 计算得

$$\begin{aligned}\|\tilde{\boldsymbol{w}}_{k+1}\|^2&=\|\boldsymbol{w}_k\|^2+y_i^2\cdot\|\boldsymbol{x}_i\|^2+2y_i\cdot\left(\boldsymbol{w}_k^T\boldsymbol{x}_i\right)\\&\le\|\boldsymbol{w}_k\|^2+\|x_i\|^2+0\\&\le\|\boldsymbol{w}_k\|^2+R^2\\&\le\cdots\\&\le k\cdot R^2.\end{aligned}$$

另外一方面, 设 $\tilde{\boldsymbol{w}}$ 为定理2.1(1) 中满足条件的向量, 则有

$$\begin{aligned}\tilde{\boldsymbol{w}}^T\tilde{\boldsymbol{w}}_{k+1}&=\tilde{\boldsymbol{w}}^T\tilde{\boldsymbol{w}}_k+y_i\cdot\left(\tilde{\omega}^T\tilde{\boldsymbol{x}}_i\right)\\&\ge\tilde{\boldsymbol{w}}^T\tilde{\boldsymbol{w}}_k+\gamma\\&\ge\cdots\\&\ge k\gamma,\end{aligned}$$

因此

$$k^2\gamma^2\le\left\|\tilde{\boldsymbol{\omega}}^T\right\|^2\cdot\|\tilde{\boldsymbol{w}}_{k+1}\|^2=\|\tilde{\boldsymbol{w}}_{k+1}\|^2\le k\cdot R^2,$$

解得 $k\le\dfrac{R^2}{\gamma^2}$. □

定理2.1保证了算法1的收敛性. 算法1最终输出的将会是满足 $\|\tilde{\boldsymbol{w}}\|=1$ 的超平面 $\tilde{\boldsymbol{w}}^T\tilde{\boldsymbol{x}}=$

0 的法向量 $\tilde{\boldsymbol{w}} = (\boldsymbol{\omega}, b) \in \mathbb{R}^{p+1}$, 从而确定了 $\mathbb{R}^p$ 中的超平面 $\boldsymbol{w}^T\boldsymbol{x} + b = 0$, 其中

$$1 = \|\tilde{\boldsymbol{w}}\|^2 = \|\boldsymbol{w}\|^2 + b^2 \implies \|\boldsymbol{w}\| = \sqrt{1 - b^2}.$$

若需要得到法向量的范数为 1 的超平面, 则可以令

$$\boldsymbol{w}' = \frac{\boldsymbol{w}}{\sqrt{1-b^2}}, \quad b' = \frac{b}{\sqrt{1-b^2}},$$

则超平面 $\boldsymbol{w}'^T\boldsymbol{x} + b' = 0$ 满足条件. 至此, 我们找到了一个有效的算法, 对于线性可分且有界的训练数据集 T, 可以找到一个超平面 $\boldsymbol{w}^T\boldsymbol{x} + b = 0$, 将属于不同的组的数据分离开.

在上面的基础上, 我们进一步介绍什么是感知机. 注意到 $y_i \in \{-1, 1\}$, 因此条件

$$y_i \cdot \left(\tilde{\boldsymbol{w}}^T\tilde{\boldsymbol{x}}_i\right) > 0 \quad \text{或} \quad y_i \cdot \left(\boldsymbol{w}^T\boldsymbol{x} + b\right) > 0,$$

可以等价地写为

$$y_i = \operatorname{sgn}(\tilde{\boldsymbol{w}}^T\tilde{\boldsymbol{x}}_i) = \operatorname{sgn}(\boldsymbol{w}^T\boldsymbol{x}_i + b),$$

其中 $\operatorname{sgn}(x) = \begin{cases} 1, & x \geq -1, \\ 0, & x < 0 \end{cases}$ 是符号函数. 称从输入空间 $\mathbb{R}^d$ 到输出空间 $\{-1, 1\}$ 的函数

$$f(\boldsymbol{x}) = \operatorname{sgn}\left(\boldsymbol{w}^T\boldsymbol{x} + b\right) \tag{2}$$

为**感知机**, 其中 $\boldsymbol{w} \in \mathbb{R}^d$ 和 $b \in \mathbb{R}$ 称为感知机模型参数. 考虑我们在上一小节中提到的第二个目标, 我们想要找到一个形如方程(2)的函数 $f : \mathbb{R}^d \to \{-1, 1\}$, 使得对任意的 $1 \leq i \leq n$, 都有 $f(\boldsymbol{x}_i) = y_i$. 这是上面所介绍的感知机模型的另外一种形式, 可以直接用算法1求解.

至此, 我们完整地介绍了感知机模型及其原理.

2.3 使用模拟数据进行测试

在本小节中, 生成一些线性可分的模拟数据来测试算法1. 设训练数据 $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n \in \mathbb{R}^2$, 也即训练数据有两个属性. 我们需要生成线性可分且有界的训练数据, 从而假定初始的分割直线为 $y = x - 1$, 并设数据落在矩形 $[-5, 5] \times [-5, 5]$ 内. 编写程序, 随机地在 $y > x - 1$ 的部分取 6 个点, 并标记 $y_i = -1$; 再在 $y < x - 1$ 的部分取 6 个点, 并标记 $y_i = 1$.

表 1: 线性可分的模拟数据

y_i	$\boldsymbol{x}_i$					
-1	$(-3.8, 0.0)$	$(0.0, 2.0)$	$(3.9, 4.6)$	$(-3.5, -2.4)$	$(-1.5, 3.3)$	$(2.6, 2.5)$
1	$(3.0, -3.5)$	$(1.6, -4.6)$	$(4.5, 0.6)$	$(2.0, -4.7)$	$(-2.2, -4.5)$	$(1.9, -1.8)$

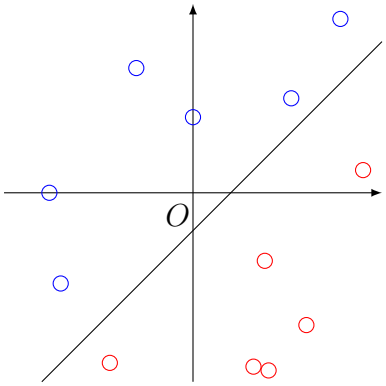


图 1: 根据模拟数据绘制的图像

观察**图 1**可得点 $(2.6, 2.5)$ 距离直线 $y = x - 1$ 最近, 计算得

$$\gamma = \frac{|2.6 - 2.5 - 1|}{\sqrt{1^2 + 1^2}} \approx 0.6364.$$

另外, 数据都落在 $[-5, 5]$ 内, 从而

$$\max_{1 \le i \le n} \|\tilde{\boldsymbol{x}}_i\| \le \sqrt{5^2 + 5^2 + 1^2} \approx 7.1414 := R.$$

根据定理2.1, 计算次数不超过 $\dfrac{R^2}{\gamma^2} \approx 126$. 在实际计算中, 我们分别取初值 $\tilde{\boldsymbol{w}}_0 = (1, 0, 0)^T$, $(0, 1, 0)^T$ 和 $(0, 0, 1)^T$, 发现仅需计算 6 次即可得到目标直线 (当然不一定是 $y = x - 1$).

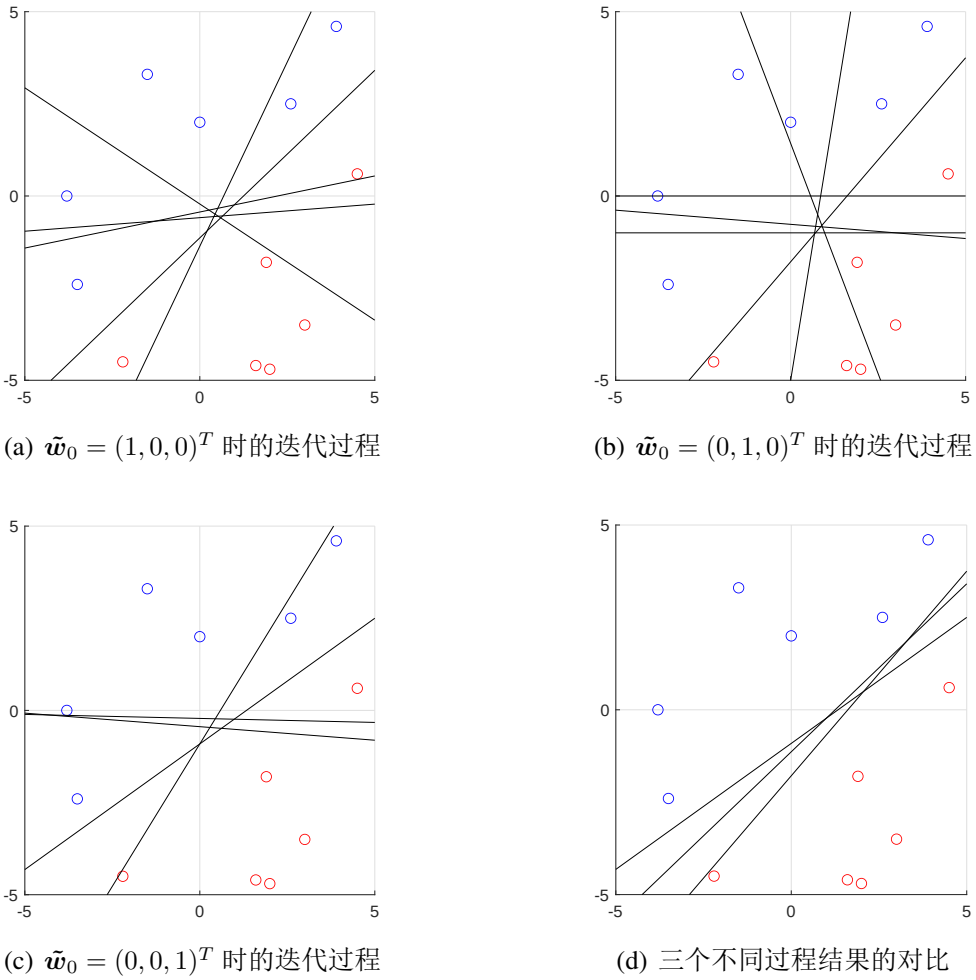
表 2: 使用模拟数据进行运算的结果

参数	$\tilde{\boldsymbol{w}}_0 = (1, 0, 0)^T$	$\tilde{\boldsymbol{w}}_0 = (0, 1, 0)^T$	$\tilde{\boldsymbol{w}}_0 = (0, 0, 1)^T$
$\tilde{\boldsymbol{w}}_1$	$(1, 0, 0)^T$	$(0, 1, 0)^T$	$(0, 0, 1)^T$
$\tilde{\boldsymbol{w}}_2$	$(-2.9, -4.6, -1)^T$	$(0, -1, -1)^T$	$(3.8, 0, 0)^T$
$\tilde{\boldsymbol{w}}_3$	$(0.9, -4.6, -2)^T$	$(3.5, 1.4 - 2)^T$	$(-0.1, -4.6, -1)^T$
$\tilde{\boldsymbol{w}}_4$	$(4.4, -2.2, -3)^T$	$(3.5, -0.6, -3)^T$	$(3.4, -2.2, -2)^T$
$\tilde{\boldsymbol{w}}_5$	$(0.5, -6.8, -4)^T$	$(-0.4, -5.2, -4)^T$	$(-0.5, -6.8, -3)^T$
$\tilde{\boldsymbol{w}}_6$	$(4, -4.4, -5)^T$	$(3.1, -2.8, -5)^T$	$(3, -4.4, -4)^T$

为了方便表现迭代的过程, 可以对三个不同的过程, 绘制每次迭代后对应的直线的图像, 如**图 2**所示. 注意到这时候结果并不是唯一的:

- 若 $\tilde{\boldsymbol{w}}_0 = (1, 0, 0)^T$, 则所得的直线为 $4x - 4.4y - 5 = 0$;
- 若 $\tilde{\boldsymbol{w}}_0 = (0, 1, 0)^T$, 则所得的直线为 $3.1x - 2.8y - 5 = 0$;
- 若 $\tilde{\boldsymbol{w}}_0 = (0, 0, 1)^T$, 则所得的直线为 $3x - 4.4y - 4 = 0$.

为了对比, 在图 2中绘制出这三条直线. 这也印证了我们最开始提到的感知机的特点: 其找到的超平面 (在这里为直线) 可能有无穷多个, 且无法给出一个确定的超平面, 能够最好地区分这些数据.



(a) $\tilde{\boldsymbol{w}}_0 = (1, 0, 0)^T$ 时的迭代过程

(b) $\tilde{\boldsymbol{w}}_0 = (0, 1, 0)^T$ 时的迭代过程

(c) $\tilde{\boldsymbol{w}}_0 = (0, 0, 1)^T$ 时的迭代过程

(d) 三个不同过程结果的对比

图 2: 初值不同时的迭代过程, 及最后结果的对比

下一节中, 我们需要考虑的问题就是, 能否确定一个能够最好地区分数据的超平面? 更进一步, 如果数据并不是线性可分的, 还能不能找到这样的超平面?

3 支撑向量机 (Support Vector Machines, SVM)

3.1 支撑向量机解决的问题

仍然设训练数据 $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n \in \mathbb{R}^p$, 且对数据 $\boldsymbol{x}_i (1 \le i \le n)$, 用 $y_i \in \{-1, 1\}$ 表示 $\boldsymbol{x}_i$ 所属的组, 在本节中考虑的训练数据集 $T = \{(\boldsymbol{x}_i, y_i), 1 \le i \le n\}$. 在这里, 我们希望能够找到一个超平面 $\boldsymbol{w}^T\boldsymbol{x} + b = 0$, 它不但能够将满足 $y_i = -1$ 和 $y_i = 1$ 的点分隔开, 也即对任意的 $1 \le i \le n$, 都有

$$\begin{cases} \boldsymbol{w}^T\boldsymbol{x}_i + b > 0, & y_i = 1, \\ \boldsymbol{w}^T\boldsymbol{x}_i + b < 0, & y_i = -1, \end{cases} \quad \text{或} \quad y_i \cdot \left(\boldsymbol{w}^T\boldsymbol{x}_i + b\right) > 0,$$

我们还希望这个超平面进一步满足

$$\begin{cases} \boldsymbol{w}^T\boldsymbol{x}_i + b \ge 1, & y_i = 1, \\ \boldsymbol{w}^T\boldsymbol{x}_i + b \le -1, & y_i = -1, \end{cases} \quad \text{或} \quad y_i \cdot \left(\boldsymbol{w}^T\boldsymbol{x}_i + b\right) \ge 1. \tag{3}$$

从而, 满足 $y_i = 1$ 的数据不但落在超平面 $\boldsymbol{w}^T\boldsymbol{x}_i + b = 0$ 的一侧, 还进一步落在超平面 $\boldsymbol{w}^T\boldsymbol{x}_i + b - 1 = 0$ 的一侧 (包含超平面本身); 而满足 $y_i = 1$ 的数据不但落在超平面 $\boldsymbol{w}^T\boldsymbol{x}_i + b = 0$ 的另一侧, 还进一步落在超平面 $\boldsymbol{w}^T\boldsymbol{x}_i + b + 1 = 0$ 与上面相反的另一侧 (包含超平面本身).

需要注意的是, 公式(3)中的 1 是必然的, 否则可以等比例地放大或缩小 $\boldsymbol{w}$ 和 b, 使得右式恰好达到 1, 并且可以取等. (3)取等时, 恰好各有几个满足 $y_i = -1$ 和 $y_j = 1$ 的数据 $\boldsymbol{x}_i$ 和 $\boldsymbol{x}_j$, 分别落在超平面 $\boldsymbol{w}^T\boldsymbol{x}_i + b - 1 = 0$ 和 $\boldsymbol{w}^T\boldsymbol{x}_i + b + 1 = 0$ 上, 如**图 3**所示. 满足这些性质的数据 $\boldsymbol{x}_i$ 和 $\boldsymbol{x}_j$ 被称为**支撑向量**.

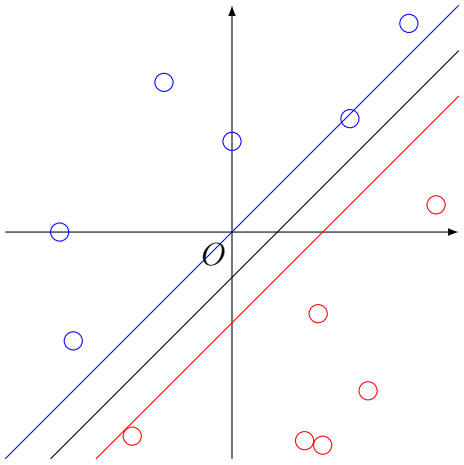


图 3: 支撑向量机模型示意图

在上面的基础上, 超平面 $\boldsymbol{w}^T\boldsymbol{x}_i + b - 1 = 0$ 和 $\boldsymbol{w}^T\boldsymbol{x}_i + b + 1 = 0$ 之间的距离

$$\gamma = \frac{2}{\|\boldsymbol{w}\|}.$$

直观上, 我们希望让 γ 最大, 这样得到的超平面可以最好地区分不同组的数据. 最大化 γ, 实际上就是最小化 $\|\boldsymbol{w}\|$. 通常情况下, 我们考虑的是最小化 $\frac{1}{2}\|\boldsymbol{w}\|^2$, 并据此写出优化问题

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2 \quad \text{s.t.} \quad y_i \cdot \left(\boldsymbol{w}^T\boldsymbol{x}_i + b\right) \geq 1, \quad 1 \leq i \leq n. \tag{4}$$

我们回忆起最开始我们考虑的另一个问题: 如果训练数据不是线性可分的, 怎么办? 如果数据并不是线性可分的, 那么就找不到一组 $\boldsymbol{w}$ 和 b, 使得不等式 $y_i \cdot \left(\boldsymbol{w}^T\boldsymbol{x}_i + b\right) \geq 1$ 严格成立. 这时, 我们可以对数据 $\boldsymbol{x}_i (1 \leq i \leq n)$, 引入松弛变量 $\xi_i \geq 0$, 将上述不等式放宽到

$$y_i \cdot \left(\boldsymbol{w}^T\boldsymbol{x}_i + b\right) \geq 1 - \xi_i.$$

记 $\boldsymbol{\xi} = (\xi_1, \xi_2, \cdots, \xi_n)$, 并设对所有的松弛变量 $\xi_i (1 \leq i \leq n)$ 之和的惩罚因子为 C, 则可以将优化问题改写成

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \frac{1}{2}\|\boldsymbol{w}\|^2 + C \cdot \sum_{i=1}^{n} \xi_i \quad \text{s.t.} \quad \begin{cases} y_i \cdot \left(\boldsymbol{w}^T\boldsymbol{x}_i + b\right) \geq 1 - \xi_i, \\ \xi_i \geq 0, \end{cases} \qquad 1 \leq i \leq n. \tag{5}$$

这时候, 若数据 $\boldsymbol{x}_i$ 满足 $y_i \cdot \left(\boldsymbol{w}^T\boldsymbol{x}_i + b\right) = 1 - \xi_i$, 则称其为**支撑向量**.

类比感知机, 我们在求解出 $\boldsymbol{w}$ 和 b 之后, 称从输入空间 $\mathbb{R}^d$ 到输出空间 $\{-1, 1\}$ 的函数

$$f(\boldsymbol{x}) = \operatorname{sgn}(\boldsymbol{w}^T\boldsymbol{x} + b)$$

为**支撑向量机**, 其中 $\boldsymbol{w} \in \mathbb{R}^d$ 和 $b \in \mathbb{R}$ 称为支撑向量机模型参数.

本节中剩下的部分, 我们将对优化问题(4)和(5)进行分析, 深入剖析支撑向量机.

3.2 优化问题及其对偶问题

3.2.1 线性可分情形

我们首先考虑优化问题(4), 也即

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2 \quad \text{s.t.} \quad y_i \cdot \left(\boldsymbol{w}^T\boldsymbol{x}_i + b\right) \geq 1, \quad 1 \leq i \leq n.$$

应用 Lagrange 乘子法, 设 $1 \leq i \leq n$, 对第 i 个约束, 设 Lagrange 乘子为 $\alpha_i \geq 0$, 并用 $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \cdots, \alpha_n)$ 表示 Lagrange 乘子的全体, 则 Lagrange 函数

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + \sum_{i=1}^{n} \alpha_i \cdot \left(1 - y_i \cdot \left(\boldsymbol{w}^T\boldsymbol{x}_i + b\right)\right).$$

原优化问题等价于最小化 $L(\boldsymbol{w}, b, \boldsymbol{\alpha})$. 而要求 L 的最小值, 可以先求导并令其导数为 0. 首先, 由 $\|\boldsymbol{w}\| = \boldsymbol{w}^T\boldsymbol{w}$, 对 $\boldsymbol{w}$ 求导得

$$\frac{\partial L}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_{i=1}^{n} \alpha_i y_i \cdot \boldsymbol{x}_i = 0 \implies \boldsymbol{w} = \sum_{i=1}^{n} \alpha_i y_i \cdot \boldsymbol{x}_i,$$

接下来, 再对 b 求导得

$$\frac{\partial L}{\partial b} = -\sum_{i=1}^{n} \alpha_i y_i = 0 \implies \sum_{i=1}^{n} \alpha_i y_i = 0.$$

将第一式代入目标函数, 再将第二式作为约束条件, 则原优化问题被转化为了对偶优化问题

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j \quad \text{s.t.} \quad \begin{cases} \sum_{i=1}^{n} \alpha_i y_i = 0, \\ \alpha_i \geq 0, \quad 1 \leq i \leq n. \end{cases} . \tag{6}$$

根据优化问题(6), 求解得到 $\boldsymbol{\alpha}$, 再解出 $\boldsymbol{w}$ 和 b, 即可得到我们所希望得到的超平面

$$\boldsymbol{w}^T\boldsymbol{x} + b = \sum_{i=1}^{n} \alpha_i y_i \cdot \boldsymbol{x}_i^T \boldsymbol{x} + b = 0. \tag{7}$$

接下来, 我们希望借助对偶优化问题, 来说明所求得的超平面 $\boldsymbol{w}^T\boldsymbol{x} + b = 0$ 与支撑向量之间的关系. 注意到原优化问题有不等式约束, 因此上述过程需要满足 KKT(Karush-Kuhn-Tucker) 条件, 也即

$$\begin{cases} \alpha_i \geq 0, \\ y_i \cdot \left(\boldsymbol{w}^T\boldsymbol{x}_i + b\right) \geq 1, & 1 \leq i \leq n, \\ \alpha_i \cdot \left(y_i \cdot \left(\boldsymbol{w}^T\boldsymbol{x}_i + b\right) - 1\right) = 0, \end{cases}$$

对于数据 $\boldsymbol{x}_i (1 \leq i \leq n)$, 约束 $\alpha_i \geq 0$ 和 $y_i \cdot \left(\boldsymbol{w}^T\boldsymbol{x}_i + b\right) \geq 1$ 是必然的; 在这里主要考虑的约束条件是 $\alpha_i \cdot \left(y_i \cdot \left(\boldsymbol{w}^T\boldsymbol{x}_i + b\right) - 1\right) = 0$, 其保证了:

- 一方面, 有可能 $\alpha_i = 0$, 则在超平面(7)中 $\boldsymbol{x}_i$ 没有贡献;
- 另外一方面, 若 $\alpha_i \neq 0$, 则一定有 $y_i \cdot \left(\boldsymbol{w}^T\boldsymbol{x}_i + b\right) - 1 = 0$, 则 $\boldsymbol{x}_i$ 恰好落在超平面 $\boldsymbol{w}^T\boldsymbol{x} + b - 1 = 0$ 或 $\boldsymbol{w}^T\boldsymbol{x} + b + 1 = 0$ 上, 也即 $\boldsymbol{x}_i$ 为上节中所提到的支撑向量.

这说明了: 超平面(7)是由支撑向量, 也即离超平面 $\boldsymbol{w}^T\boldsymbol{x} + b = 0$ 最近的数据所确定的. 这是支撑向量机模型的一个重要的特点, 也是其得此命名的原因.

3.2.2 非线性可分情形

对于非线性可分情形, 我们再考虑优化问题(5), 也即

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\cdot\sum_{i=1}^{n}\xi_i \quad \text{s.t.} \quad \begin{cases} y_i\cdot\left(\boldsymbol{w}^T\boldsymbol{x}_i+b\right)\geq 1-\xi_i, \\ \xi_i\geq 0, \end{cases} \qquad 1\leq i\leq n.$$

同样应用 Lagrange 乘子法, 对 $1\leq i\leq n$, 引入 Lagrange 乘子 $\alpha_i\geq 0, \mu_i\geq 0$, 并记 $\boldsymbol{\alpha}=(\alpha_1,\alpha_2,\cdots,\alpha_n), \boldsymbol{\mu}=(\mu_1,\mu_2,\cdots,\mu_n)$, 则 Lagrange 函数

$$L(\boldsymbol{w},b,\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\mu}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + C\cdot\sum_{i=1}^{n}\xi_i + \sum_{i=1}^{n}\alpha_i\cdot\left(1-\xi_i-y_i\cdot\left(\boldsymbol{w}^T\boldsymbol{x}_i+b\right)\right) - \sum_{i=1}^{n}\mu_i\xi_i.$$

在这里, 我们需要最小化 Lagrange 函数 $L(\boldsymbol{w},b,\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\mu})$. 对 $\boldsymbol{w}$ 求导得

$$\frac{\partial L}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_{i=1}^{n}\alpha_i y_i\cdot\boldsymbol{x}_i = 0 \implies \boldsymbol{w} = \sum_{i=1}^{n}\alpha_i y_i\cdot\boldsymbol{x}_i;$$

对 b 求导得

$$\frac{\partial L}{\partial b} = -\sum_{i=1}^{n}\alpha_i y_i = 0 \implies \sum_{i=1}^{n}\alpha_i y_i = 0;$$

对 $\xi_i(1\leq i\leq n)$ 求导得

$$\frac{\partial L}{\partial \xi_i} = C-\alpha_i-\mu_i = 0, \quad 1\leq i\leq n.$$

将以上几个式子代入原优化问题, 可将其转化为对偶优化问题

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \boldsymbol{x}_i^T\boldsymbol{x}_j \quad \text{s.t.} \quad \begin{cases} \sum_{i=1}^{n}\alpha_i y_i = 0, \\ 0\leq\alpha_i\leq C, \quad 1\leq i\leq n. \end{cases} \tag{8}$$

注意, 优化问题(8)与(6)的区别仅为: 将 $\alpha_i\geq 0$ 修改为 $0\leq\alpha_i\leq C$. 求解优化问题(8)得到 $\boldsymbol{\alpha}$, 再解出 $\boldsymbol{w}$ 和 b, 得到的超平面形式同方程(7).

在这里, 写出 KKT 条件

$$\begin{cases} \alpha_i\geq 0, \quad \mu_i\geq 0, \quad \xi_i\geq 0, \\ \mu_i\xi_i = 0, \\ y_i\cdot\left(\boldsymbol{w}^T\boldsymbol{x}+b\right)\geq 1-\xi_i, \\ \alpha_i\cdot\left(y_i\cdot\left(\boldsymbol{w}^T\boldsymbol{x}+b\right)-1+\xi_i\right) = 0, \end{cases} \qquad 1\leq i\leq n.$$

在此, 主要考虑的约束是 $\mu_i\xi_i=0$ 和 $\alpha_i\cdot\left(y_i\cdot\left(\boldsymbol{w}^T\boldsymbol{x}+b\right)-1+\xi_i\right)=0$. 对于后一个约束:

- 一方面, 有可能 $\alpha_i=0$, 则在超平面(7)中 $\boldsymbol{x}_i$ 没有贡献;
- 另外一方面, 若 $\alpha_i\neq 0$, 则一定有 $y_i\cdot\left(\boldsymbol{w}^T\boldsymbol{x}_i+b\right)-1+\xi_i=0$, 这时候 $\boldsymbol{x}_i$ 是上节中提到

的支撑向量.

这说明了超平面(7)同样是由支撑向量所决定的. 设 $\alpha_i \neq 0$, 也即 $0 < \alpha_i \leq C$, 我们进一步利用 KKT 条件, 来探讨哪些数据可以作为支撑向量:

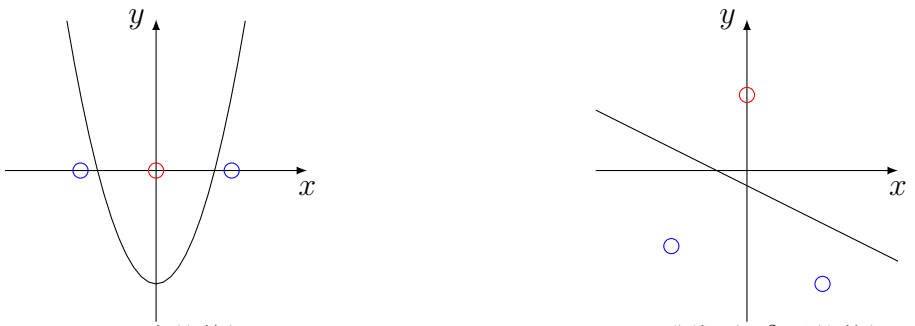
- 若 $0 < \alpha_1 < C$, 则 $\mu_i = C - \alpha_i \neq 0$, 此时一定有 $\xi_i = 0$, 这时候支撑向量 $\boldsymbol{x}_i$ 恰好在超平面 $\boldsymbol{w}^T\boldsymbol{x} + b - 1 = 0$ 或 $\boldsymbol{w}^T\boldsymbol{x} + b + 1 = 0$ 上;
- 若 $\alpha_1 = C$, 则 $\mu_i = C - \alpha_i = 0$, 这时候 $\xi_i \geq 0$, 支撑向量 $\boldsymbol{x}_i$ 有可能落在超平面 $\boldsymbol{w}^T\boldsymbol{x} + b - 1 = 0$ 与 $\boldsymbol{w}^T\boldsymbol{x} + b + 1 = 0$ 之间, 也可能在这个范围之外.

这进一步说明了, 支撑向量要么是恰好落在超平面 $\boldsymbol{w}^T\boldsymbol{x} + b - 1 = 0$ 或 $\boldsymbol{w}^T\boldsymbol{x} + b + 1 = 0$ 上的数据, 要么是落在超平面 $\boldsymbol{w}^T\boldsymbol{x} + b - 1 = 0$ 与 $\boldsymbol{w}^T\boldsymbol{x} + b + 1 = 0$ 之间的数据, 要么是导致数据线性不可分的点.

3.3 支撑向量机的核方法

在上面提到, 对于非线性可分的数据, 支持向量机可以选取那些导致数据非线性可分的点作为支撑向量. 这样训练得到的分类器仍然是线性的. 为了将该方法推广到非线性可分的情形, 本小节介绍的核方法.

考虑非线性可分的情形, 假设数据 $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n \in \mathbb{R}^p$, 如果找到一个合适的映射 ϕ, 将它们投影到一个高维的空间 $\mathbb{R}^m (m > p)$, 且让 $\phi(\boldsymbol{x}_1), \phi(\boldsymbol{x}_2), \cdots, \phi(\boldsymbol{x}_n)$ 是线性可分的, 如**图 4**所示, 那么就可以再度应用支撑向量机方法来求解问题.



(a) $\mathbb{R}$ 中的数据　　(b) 升维到 $\mathbb{R}^2$ 后的数据

图 4: $\mathbb{R}$ 中的数据和升维到 $\mathbb{R}^2$ 后的数据

构造从高维空间 $\mathbb{R}^m$ 到低维空间 $\mathbb{R}^p$ 的映射是容易的, 然而从低维空间 $\mathbb{R}^p$ 到高维空间 $\mathbb{R}^m$ 的映射 ϕ 并不容易. 为了简化问题, 我们需要另辟蹊径. 同样考虑原优化问题(4)或(5)的

对偶问题(6)或(8), 并且注意到它们的目标都是最大化函数

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \cdot \boldsymbol{x}_i^T \boldsymbol{x}_j.$$

其中, 涉及到数据 $\boldsymbol{x}_i$ 或 $\boldsymbol{x}_j$ 的部分仅有 $\boldsymbol{x}_i^T \boldsymbol{x}_j$. 若用 $\phi(\boldsymbol{x}_i)$ 代替 $\boldsymbol{x}_i$, $\phi(\boldsymbol{x}_j)$ 代替 $\boldsymbol{x}_j$, 则目标函数转化为

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \cdot \phi(\boldsymbol{x}_i)^T \phi(\boldsymbol{x}_j).$$

尽管 ϕ 作用的结果是高维空间 $\mathbb{R}^m$ 中的向量, 但是 $\phi(\boldsymbol{x}_i)^T \phi(\boldsymbol{x}_j)$ 的结果是实数. 这启发我们, 只要构造函数 $k : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}, (\boldsymbol{x}_i, \boldsymbol{x}_j) \mapsto \phi(\boldsymbol{x}_i)^T \phi(\boldsymbol{x}_j)$, 就可以实现升维的效果. 这里所提到的 k 被称为**核函数**. 按照以上方式构造的核函数, 实际上是高维空间的内积在低维空间的投影, 从而其满足: 对于任何的数据 $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n \in \mathbb{R}^n$, 矩阵

$$\begin{bmatrix} k(\boldsymbol{x}_1, \boldsymbol{x}_1) & k(\boldsymbol{x}_1, \boldsymbol{x}_2) & \cdots & k(\boldsymbol{x}_1, \boldsymbol{x}_n) \\ k(\boldsymbol{x}_2, \boldsymbol{x}_1) & k(\boldsymbol{x}_2, \boldsymbol{x}_2) & \cdots & k(\boldsymbol{x}_2, \boldsymbol{x}_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(\boldsymbol{x}_n, \boldsymbol{x}_1) & k(\boldsymbol{x}_n, \boldsymbol{x}_2) & \cdots & k(\boldsymbol{x}_n, \boldsymbol{x}_n) \end{bmatrix}$$

都是半正定的.

基于核函数, 我们可以将对偶优化问题(8)写成

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \cdot k(\boldsymbol{x}_i, \boldsymbol{x}_j) \quad \text{s.t.} \quad \begin{cases} \sum_{i=1}^{n} \alpha_i y_i = 0, \\ 0 \le \alpha_i \le C, \quad 1 \le i \le n. \end{cases} \tag{9}$$

通过选取合适的核函数 $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i)^T \phi(\boldsymbol{x}_j)$, 求解(9)得到 $\boldsymbol{\alpha}$, 并由

$$\boldsymbol{w} = \sum_{i=1}^{n} \alpha_i \cdot \phi(\boldsymbol{x}_i)$$

解出 $\boldsymbol{w}$ 和 b, 由此可以得到在高维空间 $\mathbb{R}^m$ 中关于 $\phi(\boldsymbol{x})$ 的超平面 $\boldsymbol{w}^T \phi(\boldsymbol{x}) + b = 0$, 其对应着低维空间关于 $\boldsymbol{x}$ 的非线性方程, 这时用于分类的函数相应地改为 $f(\boldsymbol{x}) = \operatorname{sgn}(\boldsymbol{w}^T \phi(\boldsymbol{x}) + b)$.

常用的核函数有:

- 线性核函数 $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i^T \boldsymbol{x}_j$, 即为 $\boldsymbol{R}^p$ 中的内积;
- Gauss 核函数 $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\dfrac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\sigma^2}\right)$, 其中 $\sigma > 0$;
- Laplace 核函数 $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\dfrac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}{\sigma}\right)$, 其中 $\sigma > 0$;
- Sigmoid 核函数 $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \tanh(\beta \cdot \boldsymbol{x}_i^T \boldsymbol{x}_j + \theta)$, 其中 $\beta > 0, \theta < 0$.

3.4 多分类的支撑向量机

上面我们所提到的支撑向量机方法, 都是用于二分类的. 然而, 现实中的分类常常不仅有两类. 基于二分类的支撑向量机, 我们可以实现对数据的多分类. 设分类的类别数 $k > 2$, 通常可以采用以下两种方法:

- “一类对一类”的分类方法, 需要建立 $\binom{k}{2}$ 个支撑向量机, 对所有的类两两比较, 最后将所有的支撑向量机整合起来;
- “一类对其他类”的分类方法, 需要建立 k 个支撑向量机, 对某个类, 将除去该类的剩下所有类并为一类, 最后将所有的支撑向量机整合起来.

至此, 我们已经介绍完了支撑向量机的原理和方法, 并将其推广到了非线性可分的情形和多分类的情形.

3.5 使用模拟数据进行测试

3.5.1 线性可分情形

为了验证支撑向量机模型, 首先使用上一节中所应用的线性可分的数据 (**表 1**和**图 1**所示). 使用 R 中的`e1071` 库可以很方便地实现支撑向量机. 计算的结果如**图 5**所示.

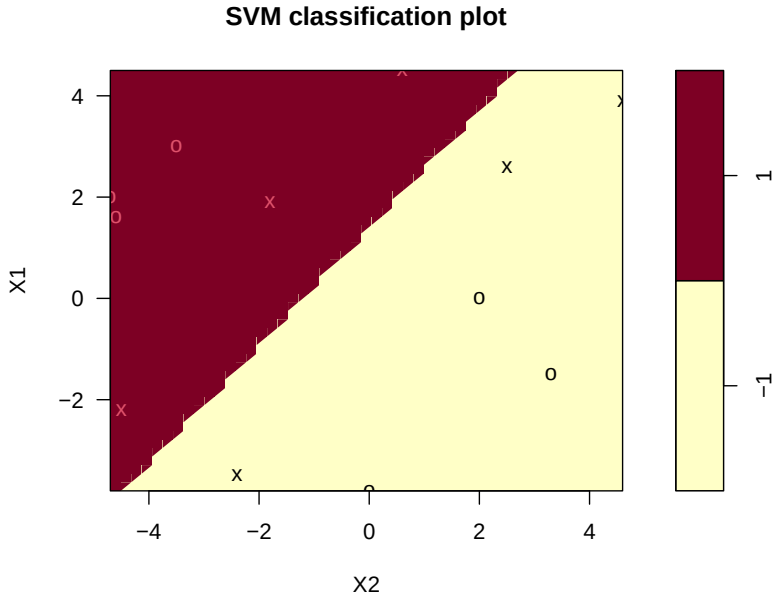


图 5: 线性可分数据的支撑向量机

3.5.2 非线性可分情形

表 3: 非线性可分的模拟数据

y_i	$\boldsymbol{x}_i$					
-1	$(-3.8, 0.0)$	$(3.9, 4.6)$	$(-3.5, -2.4)$	$(-1.5, 3.3)$	$(2.6, 2.5)$	$(1.9, -1.8)$
1	$(3.0, -3.5)$	$(1.6, -4.6)$	$(4.5, 0.6)$	$(2.0, -4.7)$	$(-2.2, -4.5)$	$(0.0, 2.0)$

支撑向量机的一个重要特点是, 可以处理非线性可分的数据. 在数据**表 1**的基础上进行修改, 即可得到非线性可分的数据. 利用**表 3**中的数据进行计算, 结果如**图 6**所示.

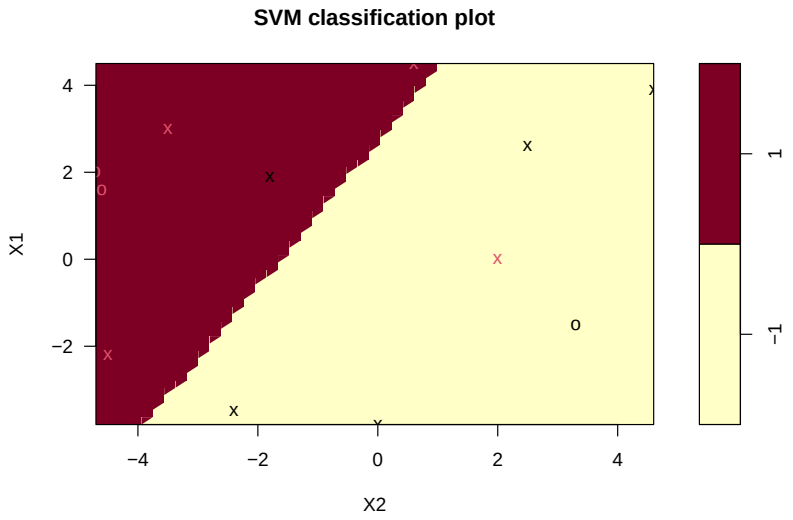


图 6: 非线性可分数据的支撑向量机

3.6 使用真实数据进行测试

为了验证支撑向量机在实际数据中的效果, 我们使用了知名的鸢尾花数据集, 并应用三分类的支撑向量机模型.

```
Call:
svm(formula = y ~ ., data = trainset, kernel = "linear", cost = 1,
    scale = FALSE)
```

```
Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  linear
       cost:  1

Number of Support Vectors:  27

 ( 3 12 12 )


Number of Classes:  3

Levels:
 setosa versicolor virginica
```

若选用鸢尾花数据集的后两个维度, 画出结果如图 7所示.

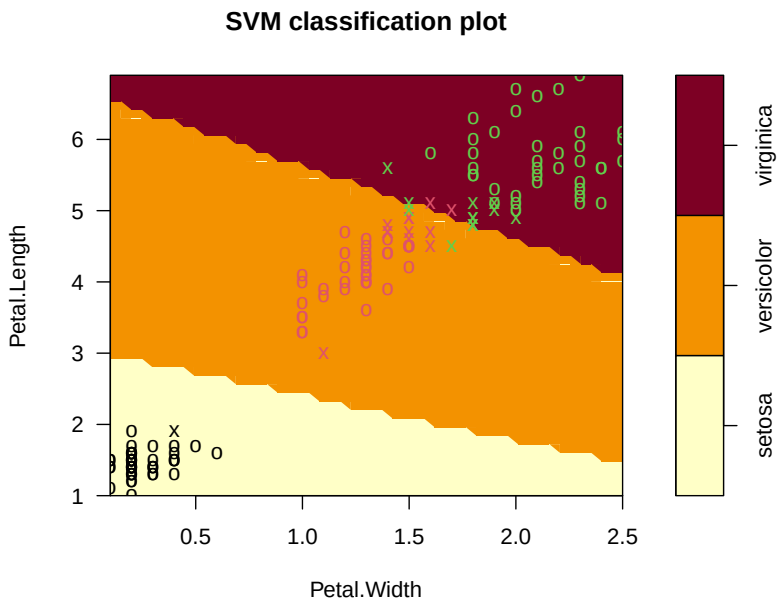


图 7: 真实数据的支撑向量机

参考文献

[1] 周志华. 机器学习 [M]. 北京: 清华大学出版社, 2016.

[2] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. 统计学习导论——基于 *R* 应用 [M]. 北京: 机械工业出版社.

附录

A 数据 1: 线性可分的模拟数据

以下数据存储在`data_1.csv` 中.

```
-3.8,0,3.9,-3.5,-1.5,2.6,3,1.6,4.5,2,-2.2,1.9
0,2,4.6,-2.4,3.3,2.5,-3.5,-4.6,0.6,-4.7,-4.5,-1.8
-1,-1,-1,-1,-1,-1,1,1,1,1,1,1
```

B 数据 2: 非线性可分的模拟数据

以下数据存储在`data_2.csv` 中.

```
-3.8,3.9,-3.5,-1.5,2.6,1.9,3,1.6,4.5,2,-2.2,0
0,4.6,-2.4,3.3,2.5,-1.8,-3.5,-4.6,0.6,-4.7,-4.5,2
-1,-1,-1,-1,-1,-1,1,1,1,1,1,1
```

C 数据 3: 鸢尾花数据集

R 中自带鸢尾花数据集, 只需要使用指令`iris`, 输出的结果如下:

```
Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
1            5.1         3.5          1.4         0.2     setosa
2            4.9         3.0          1.4         0.2     setosa
3            4.7         3.2          1.3         0.2     setosa
4            4.6         3.1          1.5         0.2     setosa
5            5.0         3.6          1.4         0.2     setosa
6            5.4         3.9          1.7         0.4     setosa
7            4.6         3.4          1.4         0.3     setosa
```

8	5.0	3.4	1.5	0.2	setosa
9	4.4	2.9	1.4	0.2	setosa
10	4.9	3.1	1.5	0.1	setosa
11	5.4	3.7	1.5	0.2	setosa
12	4.8	3.4	1.6	0.2	setosa
13	4.8	3.0	1.4	0.1	setosa
14	4.3	3.0	1.1	0.1	setosa
15	5.8	4.0	1.2	0.2	setosa
16	5.7	4.4	1.5	0.4	setosa
17	5.4	3.9	1.3	0.4	setosa
18	5.1	3.5	1.4	0.3	setosa
19	5.7	3.8	1.7	0.3	setosa
20	5.1	3.8	1.5	0.3	setosa
21	5.4	3.4	1.7	0.2	setosa
22	5.1	3.7	1.5	0.4	setosa
23	4.6	3.6	1.0	0.2	setosa
24	5.1	3.3	1.7	0.5	setosa
25	4.8	3.4	1.9	0.2	setosa
26	5.0	3.0	1.6	0.2	setosa
27	5.0	3.4	1.6	0.4	setosa
28	5.2	3.5	1.5	0.2	setosa
29	5.2	3.4	1.4	0.2	setosa
30	4.7	3.2	1.6	0.2	setosa
31	4.8	3.1	1.6	0.2	setosa
32	5.4	3.4	1.5	0.4	setosa
33	5.2	4.1	1.5	0.1	setosa
34	5.5	4.2	1.4	0.2	setosa
35	4.9	3.1	1.5	0.2	setosa
36	5.0	3.2	1.2	0.2	setosa
37	5.5	3.5	1.3	0.2	setosa
38	4.9	3.6	1.4	0.1	setosa
39	4.4	3.0	1.3	0.2	setosa
40	5.1	3.4	1.5	0.2	setosa
41	5.0	3.5	1.3	0.3	setosa
42	4.5	2.3	1.3	0.3	setosa
43	4.4	3.2	1.3	0.2	setosa

44	5.0	3.5	1.6	0.6	setosa
45	5.1	3.8	1.9	0.4	setosa
46	4.8	3.0	1.4	0.3	setosa
47	5.1	3.8	1.6	0.2	setosa
48	4.6	3.2	1.4	0.2	setosa
49	5.3	3.7	1.5	0.2	setosa
50	5.0	3.3	1.4	0.2	setosa
51	7.0	3.2	4.7	1.4	versicolor
52	6.4	3.2	4.5	1.5	versicolor
53	6.9	3.1	4.9	1.5	versicolor
54	5.5	2.3	4.0	1.3	versicolor
55	6.5	2.8	4.6	1.5	versicolor
56	5.7	2.8	4.5	1.3	versicolor
57	6.3	3.3	4.7	1.6	versicolor
58	4.9	2.4	3.3	1.0	versicolor
59	6.6	2.9	4.6	1.3	versicolor
60	5.2	2.7	3.9	1.4	versicolor
61	5.0	2.0	3.5	1.0	versicolor
62	5.9	3.0	4.2	1.5	versicolor
63	6.0	2.2	4.0	1.0	versicolor
64	6.1	2.9	4.7	1.4	versicolor
65	5.6	2.9	3.6	1.3	versicolor
66	6.7	3.1	4.4	1.4	versicolor
67	5.6	3.0	4.5	1.5	versicolor
68	5.8	2.7	4.1	1.0	versicolor
69	6.2	2.2	4.5	1.5	versicolor
70	5.6	2.5	3.9	1.1	versicolor
71	5.9	3.2	4.8	1.8	versicolor
72	6.1	2.8	4.0	1.3	versicolor
73	6.3	2.5	4.9	1.5	versicolor
74	6.1	2.8	4.7	1.2	versicolor
75	6.4	2.9	4.3	1.3	versicolor
76	6.6	3.0	4.4	1.4	versicolor
77	6.8	2.8	4.8	1.4	versicolor
78	6.7	3.0	5.0	1.7	versicolor
79	6.0	2.9	4.5	1.5	versicolor

80	5.7	2.6	3.5	1.0	versicolor
81	5.5	2.4	3.8	1.1	versicolor
82	5.5	2.4	3.7	1.0	versicolor
83	5.8	2.7	3.9	1.2	versicolor
84	6.0	2.7	5.1	1.6	versicolor
85	5.4	3.0	4.5	1.5	versicolor
86	6.0	3.4	4.5	1.6	versicolor
87	6.7	3.1	4.7	1.5	versicolor
88	6.3	2.3	4.4	1.3	versicolor
89	5.6	3.0	4.1	1.3	versicolor
90	5.5	2.5	4.0	1.3	versicolor
91	5.5	2.6	4.4	1.2	versicolor
92	6.1	3.0	4.6	1.4	versicolor
93	5.8	2.6	4.0	1.2	versicolor
94	5.0	2.3	3.3	1.0	versicolor
95	5.6	2.7	4.2	1.3	versicolor
96	5.7	3.0	4.2	1.2	versicolor
97	5.7	2.9	4.2	1.3	versicolor
98	6.2	2.9	4.3	1.3	versicolor
99	5.1	2.5	3.0	1.1	versicolor
100	5.7	2.8	4.1	1.3	versicolor
101	6.3	3.3	6.0	2.5	virginica
102	5.8	2.7	5.1	1.9	virginica
103	7.1	3.0	5.9	2.1	virginica
104	6.3	2.9	5.6	1.8	virginica
105	6.5	3.0	5.8	2.2	virginica
106	7.6	3.0	6.6	2.1	virginica
107	4.9	2.5	4.5	1.7	virginica
108	7.3	2.9	6.3	1.8	virginica
109	6.7	2.5	5.8	1.8	virginica
110	7.2	3.6	6.1	2.5	virginica
111	6.5	3.2	5.1	2.0	virginica
112	6.4	2.7	5.3	1.9	virginica
113	6.8	3.0	5.5	2.1	virginica
114	5.7	2.5	5.0	2.0	virginica
115	5.8	2.8	5.1	2.4	virginica

116	6.4	3.2	5.3	2.3	virginica
117	6.5	3.0	5.5	1.8	virginica
118	7.7	3.8	6.7	2.2	virginica
119	7.7	2.6	6.9	2.3	virginica
120	6.0	2.2	5.0	1.5	virginica
121	6.9	3.2	5.7	2.3	virginica
122	5.6	2.8	4.9	2.0	virginica
123	7.7	2.8	6.7	2.0	virginica
124	6.3	2.7	4.9	1.8	virginica
125	6.7	3.3	5.7	2.1	virginica
126	7.2	3.2	6.0	1.8	virginica
127	6.2	2.8	4.8	1.8	virginica
128	6.1	3.0	4.9	1.8	virginica
129	6.4	2.8	5.6	2.1	virginica
130	7.2	3.0	5.8	1.6	virginica
131	7.4	2.8	6.1	1.9	virginica
132	7.9	3.8	6.4	2.0	virginica
133	6.4	2.8	5.6	2.2	virginica
134	6.3	2.8	5.1	1.5	virginica
135	6.1	2.6	5.6	1.4	virginica
136	7.7	3.0	6.1	2.3	virginica
137	6.3	3.4	5.6	2.4	virginica
138	6.4	3.1	5.5	1.8	virginica
139	6.0	3.0	4.8	1.8	virginica
140	6.9	3.1	5.4	2.1	virginica
141	6.7	3.1	5.6	2.4	virginica
142	6.9	3.1	5.1	2.3	virginica
143	5.8	2.7	5.1	1.9	virginica
144	6.8	3.2	5.9	2.3	virginica
145	6.7	3.3	5.7	2.5	virginica
146	6.7	3.0	5.2	2.3	virginica
147	6.3	2.5	5.0	1.9	virginica
148	6.5	3.0	5.2	2.0	virginica
149	6.2	3.4	5.4	2.3	virginica
150	5.9	3.0	5.1	1.8	virginica

D 代码 1: 感知机

感知机可以用 MATLAB 实现. 以下代码存储在`code_1.m` 中, 且与`data_1.csv` 同目录.

```
%% 初始化

clc;
clear;
close;

%% 数据处理

data = csvread("data_1.csv"); % 读取模拟数据
x = data([1, 2], :); % 模拟数据
y = data(3, :); % 模拟数据的分类
p = size(x, 1); % 数据的维数
n = size(x, 2); % 数据的数量
xx = [x; ones(1, 12)]; % x加上1

%% 感知机

k = 1;
w = [1; 0; 0];
while isempty(find(y .* (w' * xx) < 0, 1)) == 0
    i = find(y .* (w' * xx) < 0, 1);
    w = w + y(i) * xx(:, i);
    k = k + 1;
end

%% 绘图

scatter(x(1, 1 : 6), x(2, 1 : 6), 'b');
hold on;
scatter(x(1, 7:12), x(2, 7 : 12), 'r');
hold on;
plot([-5, 5], [(5 * w(1) - w(3))/w(2), (-5 * w(1) - w(3))/w(2)], 'k');
grid on;
```

```
axis([-5, 5, -5, 5]);
```

另外, 感知机也可以用 R 实现. 以下代码存储在code_1.r 中, 且与data_1.csv 同目录.

```
# 读取数据

data = read.csv("data_1.csv", header = F) # 读取模拟数据
x = as.matrix(data[1 : 2, ]) # 模拟数据
y = as.matrix(data[3, ]) # 模拟数据的分类
p = dim(x)[1] # 数据的维数
n = dim(x)[2] # 数据的数量
xx = rbind(x, rep(1, n)); # x加上1

# 感知机

k = 1
w = c(1, 0, 0)
while(!(is.na(which(y * (w %*% xx) < 0)[1])))
{
i = which(y * (w %*% xx) < 0)[1]
w = w + y[i] * xx[, i]
k = k + 1
}
```

E 代码 2: 模拟数据的支撑向量机

使用 R 的e1071 库中的svm() 函数, 即可实现支撑向量机. 以下代码存储在code_1.r 中, 且与data_1.csv 和data_2.csv 同目录.

```
# 读取数据

data = read.csv("data_1.csv", header = F) # 读取模拟数据1
# data = read.csv("data_2.csv", header = F) # 读取模拟数据2
x = t(as.matrix(data[1 : 2, ]))
y = as.factor(data[3, ])
trainset = data.frame(x, y)
```

```
# 支撑向量机

library(e1071)
svmfit = svm(y~., data = trainset, kernel = 'linear')
plot(svmfit, dat)
summary(svmfit)
```

F 代码 3: 真实数据的支撑向量机

svm() 函数也可以用于多分类的支撑向量机.

```
# 读取数据

x = iris[, 1 : 4]
y = iris[, 5]
trainset = data.frame(x, y)

# 支持向量机

library(e1071)
svmfit = svm(y~., data = trainset, kernel = 'linear', cost = 1, scale
   = FALSE)
summary(svmfit)
```