

# 间隔与支持向量

## 安全声明

事实上，支持向量机是我在很早之前就已经学过一遍的知识点，只不过当时是在跟 CS229 学，吴恩达老师的笔记有一部分是缺失的（比如这里的支持向量机并没有讨论软间隔情况）。所以 **我按照周志华老师的西瓜书对这部分进行了重写，重写部分将参考西瓜书进行排布，并给出比西瓜书更详细的公式推导。**

注：本章将参考西瓜书和 CS229，详细推导参考部分知乎陆小亮的文章和视频（如 SMO 算法、核函数下 SVM 的推导、核函数 + 软间隔下 SVM 的推导）。话不多说，直接开始！

## 前言

现在，让我们想想我们学过的一个分类模型——LDA。还记得么？**我们的 LDA(线性判别分析)有这么一个公式：最大化分母“类与类之间”的“距离”，最小化分子“类中样本”的“距离”以此来实现模型算法，于是我们有了这个公式：**

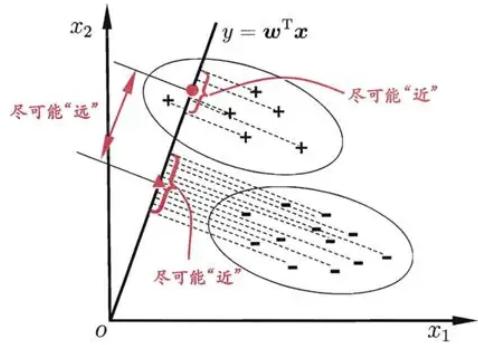
$$J(\mathbf{w}) = \frac{\sigma_B^2}{\sigma_W^2} = \frac{\mathbf{w}^T S_B \mathbf{w}}{\mathbf{w}^T S_W \mathbf{w}}$$

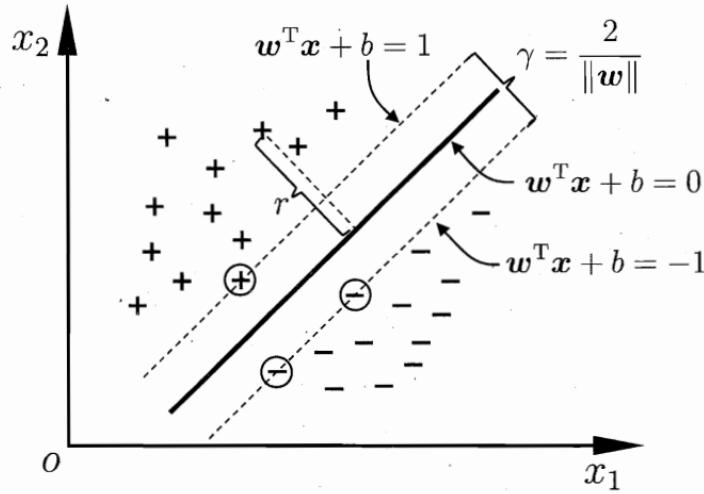
忘记了自己回去看 [线性判别分析 \(LDA\)](#)

这就是我们的 LDA 的核心思想。但是，现在请 delete your 🧠。因为我们要介绍一种全新的分类模型，这就是我们今天要讲的 **SVM(Support Vector Machine, 支持向量机)**。

什么是 SVM？很简单，想想我们下象棋时的楚汉河界，我们希望找到一个界限使得这个界限能够**完完全全**分开两侧的象棋，当然我们这里假设楚汉两方的棋是**线性可分**的（你看到了我标明了两处斜体红字。是的，我不可能一上来就谈**软间隔和核函数**下的 SVM，这既不合理、也不人性）。

那么问题来了，我们怎么表示这个界呢？这个界怎么来的？不急，我们先上图（来自西瓜书）：





在这张图上，这个“界”叫做**划分超平面**（实际上我个人觉得也可以叫做决策平面，只不过这是决策树那部分的叫法，**这里叫“超”可能是因为多维度的原因**）；而产生这个“界”的要素在于距离这个“界”两侧最近的点。也就是 $\oplus$ 和 $\ominus$ ，这类点叫做**支持向量**，也是**为什么这个模型叫做支持向量机的原因！**

## 间隔、支持向量与超平面

**下面全是重点，我就不标红了，注意理解！**

对于上图，我们在样本空间中，划分超平面可通过下面这个线性方程表示：

$$\mathbf{w}^\top \mathbf{x} + b = 0, \quad (6.1)$$

其中  $\mathbf{w} = (w_1, w_2, \dots, w_d)$  为法向量，决定了超平面的方向； $b$  为位移项，决定了超平面与原点之间的距离。显然，划分超平面可被法向量  $\mathbf{w}$  和位移  $b$  确定，我们先记为  $(\mathbf{w}, b)$ ，样本空间中任意一点  $x$  超平面  $(\mathbf{w}, b)$  距离可写为（这部分其实就是高中的点到直线距离 Pro Max 版本，直接对比着学会好理解

$$r = \frac{|\mathbf{w}^\top \mathbf{x} + b|}{\|\mathbf{w}\|}, \quad (6.2)$$

注：在 Latex 公式中  $\|\cdot\|$  代表向量的模，而不是绝对值！

假设超平面  $(\mathbf{w}, b)$  能将训练样本正确分类，即对于  $(x_i, y_i) \subseteq D$ ，若  $y_i = +1$ ，则有  $\mathbf{w}^\top x_i + b > 0$ ；若  $y_i = -1$ ，则有  $\mathbf{w}^\top x_i + b < 0$ 。所以我们得到

$$\begin{cases} \mathbf{w}^\top x_i + b \geq +1, & y_i = +1 \\ \mathbf{w}^\top x_i + b \leq -1, & y_i = -1 \end{cases}, \quad (6.3)$$

使这个方程组满足等号的，也就是我们上边说的支持向量 $\oplus$ 和 $\ominus$ 。两个异类支持向量到超平面的距离之和：

$$\gamma = \frac{2}{\|\mathbf{w}\|}, \quad (6.4)$$

这被我们称之为“间隔”。我们的目标就是最大化间隔，这与 LDA 不同，LDA 聚焦在“类与类之间”和“相同类中元素之间”（上文开头）。总之我们的 SVM 目标就出现了，它就是

$$\begin{aligned} \max_{\mathbf{w}, b} \quad & \gamma = \frac{2}{\|\mathbf{w}\|} \\ \text{s.t.} \quad & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1, \quad i = 1, 2, \dots, m \end{aligned}, \quad (6.5)$$

注：s.t. 代表约束 subject to，在什么什么的约束下；在约束中的  $y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1$  这个式子，是由 (6.3) 推导而来。

也即是最大化  $\gamma$ ，或者我们标准化一下，最大化间隔等同于最大化  $\|\mathbf{w}\|^{-1}$ ，同时等价于最小化  $\|\mathbf{w}\|^2$ 。于是我们改写上述公式

$$\begin{aligned} \min_{\mathbf{w}, b} \quad & \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{s.t.} \quad & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1, \quad i = 1, 2, \dots, m. \end{aligned}, \quad (6.6)$$

注：你要知道一个非常关键的问题，就是  $\mathbf{w}$ 、 $b$  和间隔之间的联系。我们在线性模型的时候就遇到过把  $b$  扔进  $\mathbf{w}$  中做矩阵运算的情况。所以不要只觉得间隔只与  $\mathbf{w}$  有关， $b$  也是一个影响关键。

这就是支持向量机的一般形式，我们通过最简单的楚汉河界将这个东西引了出来。事实上，在神经网络还没大肆流行之前，支持向量机一直是人工智能算法的焦点。即便现在神经网络胜出了，但是在其他领域中依然可以看到支持向量机的影子。