

支持向量机(SVM) 与 AdaBoost

主讲人：付星宇，中山大学数学学院

• 提升方法 AdaBoost 算法

- 背景与直观
- 算法细节
- 例子

- 背景与直观

中国有句古话：“三个臭皮匠，顶个诸葛亮”，AdaBoost 正是将这个思想发挥到了极致。AdaBoost 通过反复训练，得到很多弱的分类器，然后组合这些弱分类器构成一个强分类器。

AdaBoost 的提出使得决策树 CART 的功能提升极高，一下子成为可以和 SVM 媲美的学习算法，它具有以下几个优点：

- 不容易 overfitting (弱学习算法都很简单且是加权投票).
- 可解释性强 (通常用决策树桩作为弱学习算法, 树解释性高)
- AdaBoosting 只是一个框架, 可拓展性高. (弱学习算法选择多).

- 算法细节

设训练集 $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ ，其中 $x_i \in \mathbb{R}^n$ 是特征向量 ($\forall i \in \{1, 2, \dots, N\}$)， $y_i \in \{-1, 1\}$ ($\forall i \in \{1, 2, \dots, N\}$) 是类标记。下面陈述 AdaBoosting 算法细节：

(1) 给这 N 个训练样本一样的权重，即：

$$\text{Weight}(i) = \frac{1}{N} \quad \forall i \in \{1, 2, \dots, N\}$$

注：一开始每一个训练

样本都同等关注。

(2) for m in range(M): % 训练 M 个弱分类器

(a) 利用权值分布 $\text{weight}(i)$ $\forall i \in \{1, 2, \dots, N\}$,
以及训练集 T , 得到弱分类器 G_m .

★ \rightarrow 注: 这里可以看出弱算法 i.e. $G_m: \mathbb{R}^n \rightarrow \{-1, 1\}$ 模型在
的选择很多, AdaBoosting 只能是树桩.
注: 训练集上错误
(b) 计算分类器 G_m 的重要性程度.

$$\text{i.e. } \alpha_m = \frac{1}{2} \ln\left(\frac{1-e_m}{e_m}\right)$$

$$\text{where: } e_m = \sum_{i=1}^N \text{weight}(i) \cdot I(G_m(x_i) \neq y_i)$$

(c) 更新训练样本的权重分布. \checkmark (old)
 $\text{i.e. } \text{weight}(i) = \frac{\text{weight}(i) \times \exp(-\alpha_m y_i G_m(x_i))}{Z_m}$

注: 若训练样本被错分类 (New)
则权重提高. 若训练样本被正确分类 (Old)
则权重下降. 其中调整幅度与
当前弱分类器的重要性有关.
这里 Z_m 是为了使更新后的权重呈一均匀分布.

(3) 完成(2)的迭代过程, 令 $f(x) = \sum_{m=1}^M \alpha_m G_m(x)$ \checkmark (最终决策是一个加权平均).

~~then we have~~ ~~$G(x) = \text{sign}(f(x)) = \begin{cases} +1, & \text{if } f(x) \geq 0 \\ -1, & \text{else} \end{cases}$~~

then we have $G: \mathbb{R}^n \rightarrow \{-1, 1\}$.

这里 $G(x)$ 为最终的成熟分类器.

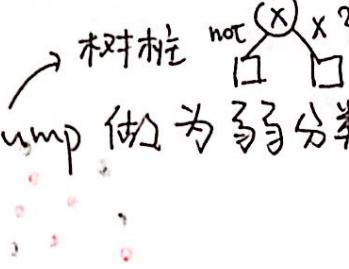
- 例子

在 AdaBoosting 的框架中，我们常用 Stump 作为弱分类器。
这是因为如下一些原因：

- Stump 很简单，不会 overfitting。

- Stump 可解释性很强，(树的 decision making 过程是可视化的)

具体算例见《统计学习方法》P140页 例 8.1.



★ • 支持向量机 (SVM)

• 背景与直观	linear
	non-linear
• 算法推导	数学准备
	线性可分支持向量机 线性支持向量机 非线性支持向量机 SMO (序列最小最优化算法)

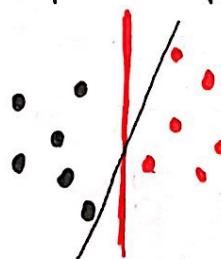
一 背景与直观

SVM 是监督学习中，最为优秀的算法之一，是深度学习出来之前最有力的机器学习工具。

下面直观解释一下 Linear 与 non-linear 的 SVM 工作原理。

Linear.

设我们有 2 种球：，我们想用棍子分开它们。



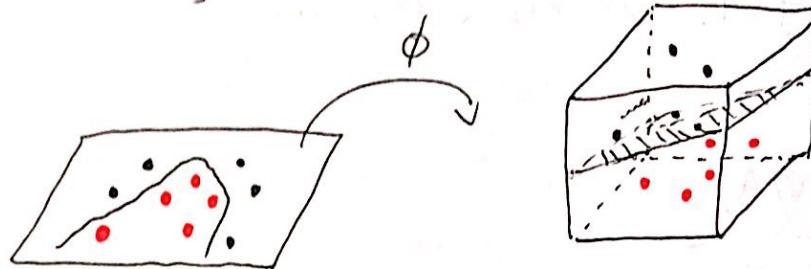
我们可以用上图的黑线分，亦可用红线分。显然红线也是一种更为好的线，因为红线直观上 最大程度分开了两类球。这样红线的泛化能力会更强。如何找红线是 SVM 的工作！（找黑线是感知机的工作）

• non-linear

若球堆如下：



则现在无法找根分开它们了，想象一个忍者用力一拍桌子，把球震到空中，你在空中拿一张纸 分开红星两类，i.e.：



将高维空间的分类结果投影下来，便有一个非线性划分。

注意：你用纸在空中划分小虫求时，亦是用 Linear 的 SVM 方法。

* < (难)

一 算法推导

* 数学准备：拉格朗日对偶性

- {① 原始问题
② 对偶问题
③ 原始问题与对偶问题的关系}

①. 原始问题：假设 $f(x)$, $c_i(x)$, $h_j(x)$ 是这在 R^n 上的连续可微

函数。考虑 约束最优化问题 → 优化目标

$$\begin{aligned} \text{原始问题} &\left\{ \begin{array}{l} \min_{x \in R^n} f(x) \\ c_i(x) \leq 0, \forall i \in \{1, 2, \dots, K\} \\ h_j(x) = 0, \forall j \in \{1, 2, \dots, L\} \end{array} \right. \\ &\quad \rightarrow \text{不等式约束} \\ &\quad \rightarrow \text{等式约束} \end{aligned}$$

引入拉格朗日函数

$$L(x, \alpha, \beta) = f(x) + \sum_{i=1}^K \alpha_i \cdot c_i(x) + \sum_{j=1}^L \beta_j \cdot h_j(x)$$

其中 $\alpha_i \geq 0$ 对 $\forall i \in K$.

(命题).

原始问题与 $\min_x \max_{\alpha, \beta} L(x, \alpha, \beta)$ 等价. 非常容易.
 (证明: 只用说明 $f(x)$ 与 $\max_{\alpha, \beta} L(x, \alpha, \beta)$ 的关系).

(2) 对偶问题

称 $\max_{\alpha, \beta} \min_x L(x, \alpha, \beta)$ 为原始问题和对偶问题.

(3) 原始问题与对偶问题的关系

定理 (同解存在性)

设 $f(x)$ 是 convex 的 (凸的), $c_i(x)$ 也是 convex 的, 且 $h_j(x)$ 是仿射函数 且 $\exists \vec{x} \in \mathbb{R}^n$, s.t. $c_i(\vec{x}) < 0$ 对 $\forall i \in \{1, 2, \dots, k\}$ 成立.

则有: $\exists x^*, \alpha^*, \beta^*$, where $x^* \in \mathbb{R}^n$, $\alpha^* \in \mathbb{R}^k$, $\beta^* \in \mathbb{R}^L$,

s.t. $L(x^*, \alpha^*, \beta^*)$ 同时是原始问题与对偶问题的解.

定理 (解的 KKT 条件).

设 $f(x)$, $c_i(x)$, $h_j(x)$ 满足如上条件, 则由上定理, $\exists \alpha^*, \beta^*, x^*$, s.t. $L(x^*, \alpha^*, \beta^*)$ 是原始问题与对偶问题的解.

且 x^*, α^*, β^* 是同解的充分必要条件 (KKT):

- $\nabla_x L(x^*, \alpha^*, \beta^*) = 0$
- $\nabla_\alpha L(x^*, \alpha^*, \beta^*) = 0$
- $\nabla_\beta L(x^*, \alpha^*, \beta^*) = 0$
- $\alpha_i^* c_i(x^*) = 0, \forall i \in \{1, 2, \dots, k\}$. KKT 对偶互补条件
- $\alpha_i^* \geq 0, \forall i \in \{1, 2, \dots, k\}$.

- $g_i(x^*) \leq 0, \forall i \in \{1, 2, \dots, k\}$.
- $h_j(x^*) = 0, \forall j \in \{1, 2, \dots, L\}$.

我们承认如上的凸优化理论，下面将正式研究 SVM.

* 线性可分支持向量机 *

若训练集 $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ 其中 $x_i \in \mathbb{R}^n$ 是特征向量， $y_i \in \{-1, +1\}$ 是类标记，是线性可分的，i.e. 存在超平面 π ，将正例与负例完全分开，则线性可分支持向量机将找到划分能力最强的分割超平面，事实上：(在线性可分假设下)

- 用感知机可找到训练集的某个划分超平面
- 用支持向量机可找到最能分割训练集的超平面(可能存在且唯一)

下面开始严格推导线性 SVM：

定义(盈数间隔).

给定 T ，给定超平面 (w, b) ，定义超平面 (w, b) 与样本点 (x_i, y_i) 的盈数间隔为：

$$\hat{\gamma}_i = y_i(w \cdot x_i + b)$$

这里 $\hat{\gamma} = \min_{i=1,2,\dots,N} \hat{\gamma}_i$ 为最小的盈数间隔。

注：若超平面 (w, b) 正确分类，则 y_i 与 $w \cdot x_i + b$ 同号，i.e. $y_i(w \cdot x_i + b) > 0$.

注： $|w \cdot x_i + b|$ 可理解为分类的置信度。

注：若 $(w, b) \rightarrow (zw, zb)$. 这实际上是一个分隔超平面，但盈数间隔却变为原来的 z 倍，这便启示我们要令 $\|w\|_2 = 1$.

定义(几何间隔).

给定 T ，给定 (w, b) ，定义

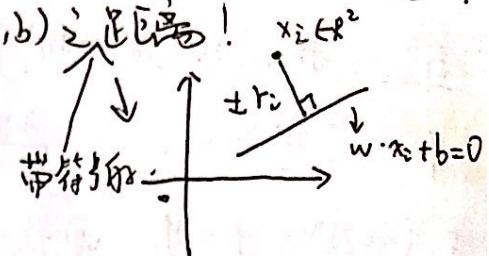
$$\gamma_i = y_i \left(\frac{w}{\|w\|_2} \cdot x_i + \frac{b}{\|w\|_2} \right).$$

$r = \min_{i=1,2,\dots,n} r_i$ (注: 取到最小的编号 j , 称 x_j 是支持向量)

注: 点 $(x_i) \in \mathbb{R}^n$, 超平面 $wx+b=0$. 则点到直线距离公式 $d = \frac{|wx_i+b|}{\|w\|_2}$.

所以几何间隔 r_i 可视为样本点 x_i 与 (w, b) 之距离!

注: 值 $\hat{r}_i = r_i \|w\|_2$, $\hat{r} = r \cdot \|w\|_2$



推导(线性可分SVM).

给定 $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$, $x_i \in \mathbb{R}^n$, $y_i \in \{-1, 1\}$. 线性可分,

我们希望找到能够正确划分数据集且几何间隔最大的分离超平面 (w^*, b^*) . 可以证明 (w^*, b^*) 是存在且唯一的. \hookrightarrow 可分割最难分类点.

i.e.

$$\begin{cases} \max_{w,b} \hat{r}(w,b,T) \\ \text{s.t. } y_i \left(\frac{w}{\|w\|_2} \cdot x_i + \frac{b}{\|w\|_2} \right) \geq \hat{r} \quad (\forall i \in \{1, 2, \dots, N\}) \end{cases}$$

$$\Leftrightarrow \begin{cases} \max_{w,b} \frac{\hat{r}}{\|w\|_2} (w,b, T) \\ \text{s.t. } y_i (w \cdot x_i + b) \geq \hat{r} \quad (\forall i \in \{1, 2, \dots, N\}) \end{cases}$$

因 \hat{r} 与 w 完全线性, 故可令 $\hat{r} = 1$, 有:

$$\Leftrightarrow (*) \begin{cases} \min_{w,b} \frac{1}{2} \|w\|^2 \\ \text{s.t. } y_i (w \cdot x_i + b) \geq 1 \end{cases}$$

硬间隔最大化.

$$1 - y_i (w \cdot x_i + b) \leq 0 \quad (\forall i \in \{1, 2, \dots, N\})$$

$$\text{拉格朗日函数 } L(w, b, \alpha) = \frac{1}{2} \|w\|^2 + \sum_{i=1}^N \alpha_i - \sum_{i=1}^N \alpha_i y_i (w \cdot x_i + b)$$

其中 $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_N)^T$, 且 $\alpha_i \geq 0$ 对 $\forall i \in \{1, 2, \dots, N\}$.

则 原始问题 (*) $\Leftrightarrow \min_{w,b} \max_{\alpha} L(w, b, \alpha)$. \leftarrow 数学准备部分.

不直接处理极小极大问题，我们处理原始问题的对偶问题

$$\max_{w,b} \min_{\alpha} L(w,b, \alpha).$$

首先，我们必须说明原始问题与对偶问题等价，这是因为

$$f(w) = \frac{1}{2} \|w\|^2, \quad \frac{\partial^2 f}{\partial w^2} = I > 0 \Rightarrow \text{凸函数}$$

$$c_i(w,b) = 1 - y_i(w \cdot x_i + b), \quad \frac{\partial^2 c_i}{\partial w^2} = 0 = \frac{\partial^2 C_i}{\partial b^2} = 0 \geq 0 \Rightarrow \text{凹},$$

故原始问题与对偶问题等价，若 (w^*, b^*) , α^* 是公共解，则

它们亦满足 KKT 条件。

现在求解对偶问题， $\max_{\alpha} \min_{w,b} L(w,b, \alpha)$

(1) 求 $\min_{w,b} L(w,b, \alpha)$

$$\nabla_w L(w,b, \alpha) = w - \sum_{i=1}^N \alpha_i y_i x_i = 0.$$

$$\nabla_b L(w,b, \alpha) = - \sum_{i=1}^N \alpha_i y_i = 0$$

$$\Leftrightarrow \begin{cases} w = \sum_{i=1}^N \alpha_i y_i x_i \\ \sum_{i=1}^N \alpha_i y_i = 0 \end{cases}$$

代入 $w = \sum_{i=1}^N \alpha_i y_i x_i$ 到 $L(w,b, \alpha)$ 的表达式中且考虑到 $\sum_{i=1}^N \alpha_i y_i = 0$

$$\Rightarrow L\left(\left(\sum_{i=1}^N \alpha_i y_i x_i, b\right), \alpha\right) = \sum_{i=1}^N \alpha_i - \sum_{i=1}^N \alpha_i y_i \left(\sum_{j=1}^N \alpha_j y_j (x_j \cdot x_i)\right) +$$

$$+ \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j (x_i \cdot x_j)$$

$$= \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j (x_i \cdot x_j)$$

(2) 求 $\max_{\alpha} \min_{(w,b)} (L)$

$$\text{i.e. } \begin{cases} \min_{\alpha} \left(\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) - \frac{N}{2} \alpha_i \right) \\ \text{s.t. } \alpha_i \geq 0 \quad \forall i \in \{1, 2, \dots, N\}, \sum_{i=1}^N \alpha_i y_i = 0 \end{cases}$$

现假设我们求解出上式, \Rightarrow 得到 α^* , 由 KKT 条件 (求 w^*, b^*).

$$\begin{cases} \nabla_w L(w^*, b^*, \alpha^*) = w^* - \sum_{i=1}^N \alpha_i^* y_i x_i = 0 \\ \nabla_b L(w^*, b^*, \alpha^*) = \sum_{i=1}^N \alpha_i^* y_i = 0 \\ \alpha_i^* \geq 0 \\ y_i (w^* \cdot x_i + b^*) - 1 \geq 0 \quad (\forall i) \\ \alpha_i^* [y_i (w^* \cdot x_i + b^*) - 1] = 0 \quad (\forall i) \rightarrow \text{KKT 相容条件.} \end{cases}$$

故 $w^* = \sum_{i=1}^N \alpha_i^* y_i x_i$.

至少 $\exists j \in \{1, 2, \dots, N\}$, s.t. $\alpha_j^* > 0$, 否则 $\Rightarrow w^* = 0$

$\Rightarrow y_j (w^* \cdot x_j + b^*) - 1 \geq 0$ 对 $\forall i \in \{1, 2, \dots, N\}$ 成立 (矛盾).

对此 j , since $\alpha_j^* > 0 \Rightarrow y_j (w^* \cdot x_j + b^*) = 1$

$$\Rightarrow \begin{cases} b^* = y_j - w^* \cdot x_j \\ (y_j = 1) = y_j - \sum_{i=1}^N \alpha_i^* y_i (x_i \cdot x_j) \end{cases}$$

总结: (线性可分 SVM)

$$(1) \begin{cases} \min_{\alpha} \left(\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) - \frac{N}{2} \alpha_i \right) \\ \text{s.t. } \alpha_i \geq 0 \quad \text{且} \sum_{i=1}^N \alpha_i y_i = 0 \end{cases}$$

解得 $\alpha^* = (\alpha_1^* \dots \alpha_N^*)^T$

$$(2) w^* = \sum_{i=1}^N \alpha_i^* y_i \cdot x_i$$

$$(3) \nexists j \text{ s.t. } \alpha_j^* > 0, \text{ 计算 } b^* = y_j - \sum_{i=1}^N \alpha_i^* y_i (x_i \cdot x_j)$$

14) 求得分离超平面: $w^* \cdot x + b^* = 0$

分类决策函数: $f(x) = \text{sign}(w^* \cdot x + b^*)$

* 线性支持向量机

若训练集不是线性可分, 则线性支持SVM的技术失效.

这是因为 $\begin{cases} \min_{w,b} \frac{1}{2} \|w\|^2 \\ \text{s.t. } y_i(w \cdot x_i + b) \geq 1 \quad \forall i \in \{1, \dots, N\} \end{cases}$

不能对 $\forall i$ 成立不等式约束条件. (有不可正确分类点, $y_i \leq w \cdot x_i + b \frac{\pm \epsilon}{2}$).

故我们引入松弛变量 $\xi_i \geq 0 \quad (\forall i \in \{1, \dots, N\})$

惩罚放松.

考虑优化问题:

$$\begin{cases} \min_{w,b} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \xi_i \\ \text{s.t. } \xi_i \geq 0, \forall i \\ y_i(w \cdot x_i + b) \geq 1 - \xi_i \end{cases} \quad \checkmark \text{ 软间隔最大化}$$

其中 (> 0 是事先给定的惩罚系数).

同样用拉格朗日对偶技术(省略过程), 得到算法:

(1) 给定 $C > 0$.

$$\min_{\alpha} \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^N \alpha_i$$

$$\text{s.t. } \sum_{i=1}^N \alpha_i y_i = 0 \quad \text{且 } 0 \leq \alpha_i \leq C$$

$$\Rightarrow \alpha^* = (\alpha_1^* \dots \alpha_N^*)$$

与线性可分情形不同.

$$(2) w^* = \sum_{i=1}^N \alpha_i^* y_i x_i$$

$$(3) \text{ 找 } j \text{ s.t. } 0 < \alpha_j^* < C, \text{ 算 } b^* = y_j - \sum_{i=1}^N y_i \alpha_i^* (x_i \cdot x_j)$$

(4) 得到分离平面: $w^* x + b^* = 0$

(5) 分类函数: $f(x) = \text{sign}(w^* x + b^*)$.

* 非线性支持向量机 *

当训练集的线性可分性极差时，前两种方法全失效！例如：



且这不能用直线划分两种类型。

$$(R^n \xrightarrow{\phi} H)$$

非线性支持向量机的做法是：将特征空间 R^n 映射到某一高维空间在高维空间中，训练集的线性可分性好，故在高维空间中，使用线性 SVM 进行分类。之后的分类决策过程中，第一步： $X \xrightarrow{\phi} \phi(X) \in H$

第二步：对 $\phi(X)$ 在 H 中利用训练好的 SVM 进行分类。

非线性支持向量机的巧妙之处在于：你不用显示构造 $\phi: R^n \rightarrow H$ 而是利用 kernel trick，做显示的高维映射。

定义(核函数)

$\phi: R^n \rightarrow H$. 若 \exists 核函数 $k: R^n \times R^n \rightarrow R$, s.t.

$$k(x, z) = \underbrace{\phi(x) \cdot \phi(z)}$$

则称 $k(x, z)$ 为 中核函数。

算法：

(1) 取 $\forall c > 0$, $k(x, z)$ 为核函数 $\phi(x) \cdot \phi(z)$

$$\begin{cases} \min_{\alpha} \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j k(x_i, x_j) - \sum_{i=1}^N \alpha_i \\ \text{s.t. } \sum_{i=1}^N \alpha_i y_i = 0, 0 < \alpha_i \leq c \quad \forall i \end{cases}$$

$$\Rightarrow \text{得到 } \alpha^* = (\alpha_1^* \dots \alpha_N^*)^T$$

(2) 取 j , s.t. $\alpha_j^* > 0$, $b^* = y_j - \sum_{i=1}^N \alpha_i^* y_i K(x_i, x_j)$.

(3) 决策: $f(x) = \text{sign}\left(\sum_{i=1}^N \alpha_i^* y_i K(x, x_i) + b^*\right)$.

注: (1)(2) 步相当于隐含地把 $T \xrightarrow{\phi} H$. 在 H 中学习了一个线性 SVM
划分 $\phi(T)$.

(3) 相当于把新输入 $x \xrightarrow{\phi} \phi(x)$, 对 $\phi(x)$ 利用 H 中
训练好的线性 SVM, 进行分类.

注: 用 $\phi(x) \cdot \phi(z)$ 替换 $K(x, z)$.

注: 在应用中, 我们给定 K , 不显示给 ϕ .

* 序列最小最优化算法 (SMO)

SMO 算法解

$$\begin{cases} \min_{\alpha} & \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j K(x_i, x_j) - \sum_{i=1}^N \alpha_i \\ \text{s.t.} & 0 \leq \alpha_i \leq C \\ & \sum_{i=1}^N \alpha_i y_i = 0 \end{cases}$$

这是一种启发式优化算法, 包括两步:

- 找变量, α_i, α_j $\begin{cases} \text{① 找 } \alpha_i \\ \text{② 通过 } \alpha_i \text{ 找 } \alpha_j \end{cases}$
- 固定其余 α_k , 优化更新 α_i 与 α_j ,

细节见《统计学习方法》