



机器学习（研究生）

决策树与支持向量机

Decision Trees & Support Vector Machines

赵凯 上海大学

<https://kaizhao.net/teaching/shu-ml-grad>

本次课路线

两种经典的非线性分类方法：一个基于规则，一个基于间隔

Part 1 决策树

符号主义：从数据中学习 if-then 规则

1. 决策树的结构与训练框架
2. 用哪个属性划分：信息增益、增益率、基尼指数
3. 怎么划分：离散属性与连续属性
4. 剪枝
5. 集成学习：Bagging、随机森林、Boosting

Part 2 支持向量机

统计学习：最大化分类间隔

1. 间隔与最大间隔
2. 拉格朗日乘子法与 KKT 条件
3. SVM 的对偶问题
4. 核技巧：从线性到非线性

符号约定

整门课程统一使用以下记号

数据集	$D = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m)\}$, 共 m 个样本; $ D $ 表示集合中的样本个数
样本（特征向量）	$\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{id})^\top \in \mathbb{R}^d$, 粗体小写表示向量
属性（特征）	第 j 个属性记为 X_j , $j = 1, \dots, d$; x_{ij} 是样本 $\mathbf{x}_i$ 在属性 X_j 上的取值
标签	y_i ; 决策树中 $y_i \in \{1, \dots, K\}$ (K 个类别), 支持向量机中 $y_i \in \{-1, +1\}$
模型参数	$\mathbf{w} \in \mathbb{R}^d$ (权重向量), $b \in \mathbb{R}$ (偏置)

记号 $^\top$: 转置 (把行向量变成列向量); $\mathbb{R}^d$: d 维实数向量空间; $\in$: 属于。

符号约定：对应到西瓜数据集

后面决策树部分的所有例子都在西瓜数据集上计算

样本个数、属性个数	$m = 17$ 个西瓜, $d = 8$ 个属性
属性	$X_1 = \text{色泽}$, $X_2 = \text{根蒂}$, $\dots$, $X_7 = \text{密度}$, $X_8 = \text{含糖率}$
离散属性的取值	$x_{1,4} = \text{清晰}$: 1 号瓜的纹理是“清晰”
连续属性的取值	$x_{1,7} = 0.697$: 1 号瓜的密度是 0.697
标签	$y_1 = \text{好瓜}$; 决策树中 $K = 2$ 个类别 (好瓜、坏瓜)

注意

X_j (大写) 指一个属性, 即数据表中的一列; $\mathbf{x}_i$ (粗体小写) 指一个样本, 即数据表中的一行; x_{ij} 是第 i 行第 j 列的那个值。

01

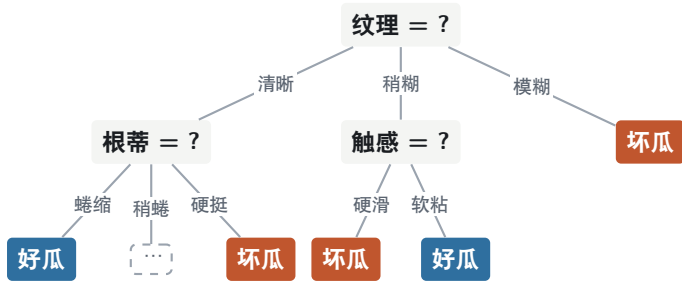
决策树

Decision Trees

结构与训练框架 · 划分属性的选择 · 离散与连续属性的划分 · 剪枝

决策树长什么样

以西瓜数据集为例：一连串“属性测试”，从根走到叶就得到预测



根结点：包含全部样本

内部结点：对一个属性做测试

分支：测试的一个结果

叶结点：给出预测的类别

每条根到叶的路径是一条规则：
若纹理 = 稍糊且触感 = 软粘，
则好瓜

数据：西瓜数据集 3.0

17 个样本，6 个离散属性 + 2 个连续属性；本节所有例子都在这张表上计算

编号 i	色泽 X_1	根蒂 X_2	敲声 X_3	纹理 X_4	脐部 X_5	触感 X_6	密度 X_7	含糖率 X_8	好瓜 y
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	0.697	0.460	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	0.774	0.376	是
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	0.634	0.264	是
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	0.608	0.318	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	0.556	0.215	是
6	青绿	稍蜷	浊响	清晰	稍凹	软粘	0.403	0.237	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	0.481	0.149	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	0.437	0.211	是
9	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	0.666	0.091	否
10	青绿	硬挺	清脆	清晰	平坦	软粘	0.243	0.267	否
11	浅白	硬挺	清脆	模糊	平坦	硬滑	0.245	0.057	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	0.343	0.099	否
13	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	0.639	0.161	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	0.657	0.198	否
15	乌黑	稍蜷	浊响	清晰	稍凹	软粘	0.360	0.370	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	0.593	0.042	否
17	青绿	蜷缩	沉闷	稍糊	稍凹	硬滑	0.719	0.103	否

记号 i : 样本编号 (第 i 行); X_j : 第 j 个属性 (第 j 列), $j = 1, \dots, 8$; y : 标签, “是” 为好瓜, “否” 为坏瓜。

训练决策树：每个结点要回答两个问题

从根结点开始，对当前结点的样本集合 D 反复回答这两个问题

问题 1：用哪个属性划分？

在候选属性 $X_1, \dots, X_d$ 中选一个“最好”的 X_{j^*}

需要一个度量划分好坏的指标：

信息增益 (ID3)、增益率 (C4.5)、基尼指数 (CART)

问题 2：怎么划分？

选定属性后，把 D 分成若干子集（记号见下页）

离散属性：每个取值一个分支（多叉），或把取值分成两组（二叉）

连续属性：选阈值 t ，分成 $x_{ij} \leq t$ 与 $x_{ij} > t$ 两支

两个问题一起回答：对每个属性先找到它最好的划分方式并算出指标，再比较所有属性，选指标最好的。

记号 D ：当前结点的样本集合； $X_1, \dots, X_d$ ：全部 d 个属性； X_{j^*} ：被选中的最优属性； x_{ij} ：样本 x_i 在属性 X_j 上的取值； t ：连续属性的划分阈值。

记号：划分后的子集 D^v

按属性 X_j 划分当前结点的样本集合 D ，每个分支得到 D 的一个子集

离散属性： 设 X_j 有 V 个取值 $a^1, \dots, a^V$

$$D^v = \{(\mathbf{x}_i, y_i) \in D : x_{ij} = a^v\}, \quad v = 1, \dots, V$$

例：按纹理 X_4 划分， $V = 3$

v	a^v	D^v (样本编号)	$ D^v $
1	清晰	1, 2, 3, 4, 5, 6, 8, 10, 15	9
2	稍糊	7, 9, 13, 14, 17	5
3	模糊	11, 12, 16	3

连续属性： 给定阈值 t ，分成两个子集

$$D_t^- = \{x_{ij} \leq t\}, \quad D_t^+ = \{x_{ij} > t\}$$

例：按密度 X_7 、 $t = 0.381$ 划分

子集 (样本编号)	个数
D_t^- : 10, 11, 12, 15	4
D_t^+ : 其余 13 个	13

上标 v 是取值的编号，不是指数。 各子集互不重叠，合起来恰好是 D ： $|D^1| + \dots + |D^V| = |D|$ （上例 $9 + 5 + 3 = 17$ ）。

记号 $(\mathbf{x}_i, y_i)$ ： 第 i 个样本及其标签； x_{ij} ：样本 $\mathbf{x}_i$ 在 X_j 上的取值； $|\cdot|$ ：集合中的样本个数； $D_t^\pm$ 的定义中省略了 “ $(\mathbf{x}_i, y_i) \in D$ ”。

训练算法：递归地分而治之

输入：当前结点的样本集合 D ，候选属性集合 $\mathcal{A}$ ；从根结点调用 $\text{TREEGENERATE}(D, \{X_1, \dots, X_d\})$ 。连续属性时 $V = 2$ ， $D^1 = D_t^-$ ， $D^2 = D_t^+$

```
1: function TREEGENERATE( $D, \mathcal{A}$ )
2:   生成结点 node
3:   if  $D$  中样本全属于同一类别  $k$  then 标记 node 为  $k$  类叶结点, return
4:   if  $\mathcal{A} = \emptyset$  或  $D$  在  $\mathcal{A}$  上取值全相同 then 标记 node 为  $D$  的多数类叶结点, return
5:   回答问题 1、2：选择最优划分属性  $X_{j^*}$  及其划分方式，把  $D$  分成子集  $D^1, \dots, D^V$ 
6:   for  $v = 1, \dots, V$  do
7:     if  $D^v = \emptyset$  then 生成叶结点，标记为  $D$  的多数类
8:     else 以  $\text{TREEGENERATE}(D^v, \mathcal{A}')$  的返回结点为分支
9:   end for
10:  return node
11: end function
```

记号 D ：当前结点的样本集合； $\mathcal{A}$ ：候选属性集合， $\mathcal{A}'$ ：传给子结点的候选属性集合； k ：类别编号； X_{j^*} ：选中的属性； D^v ：第 v 个分支的样本子集，共 V 个； D_t^- , D_t^+ ：连续属性按阈值 t 分成的两部分； $\emptyset$ ：空集。

什么时候停止划分？

三种情况下不再递归，直接生成叶结点

1. 样本全属同一类

已经完全“纯”了，不必再分。

例：纹理 = 模糊的 3 个瓜全是坏瓜 $\rightarrow$ 叶结点“坏瓜”

2. 无法再分

没有候选属性可用，或所有样本在候选属性上取值完全相同。

标记为 D 中的多数类（利用当前结点的后验分布）

3. 子集为空

某个取值在当前结点中没有样本。

标记为父结点 D 的多数类（把父结点的分布当作先验）

候选属性集合 $\mathcal{A}'$

离散属性按取值多叉划分后，子结点中它的取值都相同，因此从 $\mathcal{A}$ 中去掉；连续属性（以及 CART 的二叉划分）在后代结点中可以再次使用。

记号 D ：当前结点的样本集合； $\mathcal{A}$ ：当前的候选属性集合； $\mathcal{A}'$ ：传给子结点的候选属性集合。

问题 1 的直觉：划分后越“纯”越好

纯：同一分支内的样本尽量属于同一类。比较两种划分（● 好瓜，● 坏瓜）

按 X_6 （触感）划分

硬滑：6/12



软粘：2/5



两个分支仍然一半对一半：几乎没有用

(好瓜数 / 样本数)

按 X_4 （纹理）划分

清晰：7/9



稍糊：1/4



模糊：0/3



各分支明显更纯：好的划分

(好瓜数 / 样本数)

划分前：17 个样本，8 好 9 坏。如何把“纯度”变成一个数？ $\Rightarrow$ 信息熵

记号 X_4 ：纹理， X_6 ：触感；分数表示“好瓜数 / 样本数”。

信息熵：度量一个集合的纯度

设 D 中第 k 类样本所占比例为 p_k ($k = 1, \dots, K$)

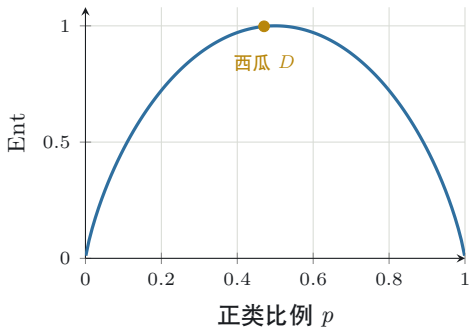
$$\text{Ent}(D) = - \sum_{k=1}^K p_k \log_2 p_k$$

其中 $p_k = |\{i: y_i = k\}| / |D|$

- 约定 $0 \log_2 0 = 0$
- 全部属于同一类： $\text{Ent}(D) = 0$ ，**最纯**
- K 类均匀分布： $\text{Ent}(D) = \log_2 K$ ，**最不纯**
- 二分类时最大值为 1（各占一半）

熵越小，纯度越高

二分类时的熵



记号 D : 样本集合, $|D|$: 其中样本个数; K : 类别个数, k : 类别编号; y_i : 样本 i 的标签; p_k : 第 k 类样本的比例; p (右图): 二分类时正类的比例。

算一个：整个数据集 D 的熵

17 个样本中 8 个好瓜、9 个坏瓜， $K = 2$

$$p_1 = \frac{8}{17}, \quad p_2 = \frac{9}{17}$$

$$\begin{aligned} \text{Ent}(D) &= - \left(\frac{8}{17} \log_2 \frac{8}{17} + \frac{9}{17} \log_2 \frac{9}{17} \right) = \\ &= -(0.471 \times (-1.087) + 0.529 \times (-0.918)) = 0.998 \end{aligned}$$

含义

0.998 非常接近二分类的最大值 1：好瓜、坏瓜几乎各占一半，不看任何属性时，我们对一个瓜的好坏几乎一无所知。

记号 D ：全部 17 个样本； $K = 2$ ：类别数； p_1 ：好瓜的比例， p_2 ：坏瓜的比例。

信息增益：划分后熵减少了多少

设离散属性 X_j 有 V 个取值 $a^1, \dots, a^V$

$$\text{Gain}(D, X_j) = \text{Ent}(D) - \sum_{v=1}^V \frac{|D^v|}{|D|} \text{Ent}(D^v), \quad D^v = \{(\mathbf{x}_i, y_i) \in D : x_{ij} = a^v\}$$

D^v : 属性 X_j 取第 v 个值 a^v 的样本子集。例：纹理 X_4

v	a^v	D^v (样本编号)	$ D^v $	$\frac{ D^v }{ D }$
1	清晰	1-6, 8, 10, 15	9	$\frac{9}{17}$
2	稍糊	7, 9, 13, 14, 17	5	$\frac{5}{17}$
3	模糊	11, 12, 16	3	$\frac{3}{17}$

$\frac{|D^v|}{|D|}$ 是分支的权重；求和项是划分后各分支熵的**加权平均**。
信息增益越大，“纯度提升”越大。

ID3 算法 (Quinlan, 1986)

问题 1：选信息增益最大的属性

$$X_{j^*} = \arg \max_{X_j \in \mathcal{A}} \text{Gain}(D, X_j)$$

问题 2：每个取值一个分支

记号 D : 当前样本集合; X_j : 第 j 个属性, V : 它的取值个数, a^v : 第 v 个取值; D^v : 取值为 a^v 的子集; x_{ij} : 样本 $\mathbf{x}_i$ 在 X_j 上的取值; $\mathcal{A}$: 候选属性集合; X_{j^*} : 选出的属性; $\arg \max$: 使函数取最大值的自变量。

例 1：用 X_4 （纹理）划分

纹理有 3 个取值：清晰、稍糊、模糊，产生 3 个分支

纹理 = 清晰

9 个：7 好 2 坏



$$\text{Ent}(D^1) = 0.764$$

纹理 = 稍糊

5 个：1 好 4 坏



$$\text{Ent}(D^2) = 0.722$$

纹理 = 模糊

3 个：0 好 3 坏



$$\text{Ent}(D^3) = 0 \quad (\text{全是坏瓜})$$

$$\text{Ent}(D^1) = -\frac{7}{9} \log_2 \frac{7}{9} - \frac{2}{9} \log_2 \frac{2}{9} = 0.764$$

$$\text{Ent}(D^2) = -\frac{1}{5} \log_2 \frac{1}{5} - \frac{4}{5} \log_2 \frac{4}{5} = 0.722$$

$$\text{Ent}(D^3) = -\frac{3}{3} \log_2 \frac{3}{3} = 0$$

$$\text{Gain}(D, X_4) = 0.998 - \left(\frac{9}{17} \times 0.764 + \frac{5}{17} \times 0.722 + \frac{3}{17} \times 0 \right) = 0.998 - 0.617 = 0.381$$

记号 D ：全部 17 个样本； X_4 ：纹理； D^1, D^2, D^3 ：纹理为清晰、稍糊、模糊的样本子集； $0.998 = \text{Ent}(D)$ 。

例 2：用 X_6 （触感）划分

触感只有 2 个取值：硬滑、软粘

触感 = 硬滑

12 个：6 好 6 坏



$$\text{Ent}(D^1) = -\frac{1}{2} \log_2 \frac{1}{2} - \frac{1}{2} \log_2 \frac{1}{2} = 1.000$$

触感 = 软粘

5 个：2 好 3 坏



$$\text{Ent}(D^2) = -\frac{2}{5} \log_2 \frac{2}{5} - \frac{3}{5} \log_2 \frac{3}{5} = 0.971$$

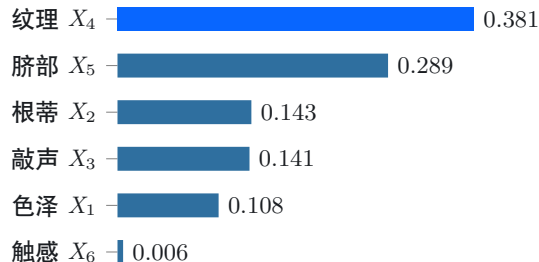
$$\text{Gain}(D, X_6) = 0.998 - \left(\frac{12}{17} \times 1.000 + \frac{5}{17} \times 0.971 \right) = 0.998 - 0.992 = 0.006$$

与直觉一致：按触感划分后，熵几乎没有下降。

记号 D ：全部 17 个样本； X_6 ：触感； D^1, D^2 ：触感为硬滑、软粘的样本子集； $0.998 = \text{Ent}(D)$ 。

比较所有离散属性：选出根结点

对 $X_1, \dots, X_6$ 分别计算 $\text{Gain}(D, X_j)$



- 纹理的信息增益最大 $\Rightarrow$ **根结点按纹理划分**
- 得到 3 个分支，各自继续递归
- 纹理 = 模糊的 3 个样本全是坏瓜：直接成为叶结点

注：色泽为 0.108，西瓜书为 0.109，差别来自中间结果的四舍五入。

记号 D ：全部 17 个样本； $X_1, \dots, X_6$ ：色泽、根蒂、敲声、纹理、脐部、触感。

递归：第二层怎么选

在“纹理 = 清晰”的 9 个样本上（7 好 2 坏， $\text{Ent} = 0.764$ ），对剩下的属性重复同样的计算

属性	各取值（好瓜数 / 样本数）	信息增益
根蒂 X_2	蜷缩 5/5，稍蜷 2/3，硬挺 0/1	0.458
脐部 X_5	凹陷 5/5，稍凹 2/3，平坦 0/1	0.458
触感 X_6	硬滑 6/6，软粘 1/3	0.458
敲声 X_3	浊响 5/6，沉闷 2/2，清脆 0/1	0.331
色泽 X_1	青绿 3/4，乌黑 3/4，浅白 1/1	0.043

要点

三个属性并列最大，任选其一（西瓜书选根蒂）。纹理已经用过，不再参与比较。每个分支都独立地重复“问题 1 + 问题 2”，直到满足停止条件。

记号 Ent ：“纹理 = 清晰”这 9 个样本的信息熵； X_1, X_2, X_3, X_5, X_6 ：色泽、根蒂、敲声、脐部、触感。

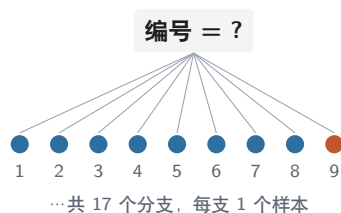
信息增益的缺陷：偏好取值多的属性

如果把“编号”也当作一个属性呢？

- “编号”有 17 个取值，每个分支只有 1 个样本
- 每个分支都是纯的： $\text{Ent}(D^v) = 0$
- 于是 $\text{Gain}(D, \text{编号}) = 0.998 - 0 = 0.998$ ，比纹理还大

但按编号划分毫无泛化能力：新来一个瓜，它的编号不在训练集里。

一般地：属性取值越多，每个分支越小，越容易“纯”，信息增益就越大。



记号 D ：全部 17 个样本； D^v ：编号为 v 的分支，只含 1 个样本。

增益率 (C4.5)

用属性的“固有值”对信息增益做归一化：这是专门为多叉划分设计的校正

$$\text{Gain_ratio}(D, X_j) = \frac{\text{Gain}(D, X_j)}{\text{IV}(X_j)}, \quad \text{IV}(X_j) = - \sum_{v=1}^V \frac{|D^v|}{|D|} \log_2 \frac{|D^v|}{|D|}$$

属性	Gain	IV	增益率
纹理 X_4	0.381	1.447	0.263
脐部 X_5	0.289	1.549	0.187
根蒂 X_2	0.143	1.402	0.102
敲声 X_3	0.141	1.333	0.106
编号	0.998	4.087	0.244

- IV 是“按 X_j 的取值分组”本身的熵：取值越多越分散，IV 越大
- 编号：IV = $\log_2 17 = 4.087$ ，增益率被压下来
- 但增益率又偏好取值少的属性

C4.5 的启发式：先找出信息增益高于平均水平的属性，再从中选增益率最高的。

记号 D ：当前样本集合； X_j ：第 j 个属性， V ：它的取值个数； D^v ：取第 v 个值的样本子集； $\text{IV}(X_j)$ ：属性 X_j 的固有值。

基尼指数 (CART)

从 D 中随机抽两个样本，它们类别不一致的概率

$$\text{Gini}(D) = \sum_{k=1}^K \sum_{k' \neq k} p_k p_{k'} = 1 - \sum_{k=1}^K p_k^2,$$

$$\text{Gini_index}(D, X_j) = \sum_{v=1}^V \frac{|D^v|}{|D|} \text{Gini}(D^v)$$

- $\text{Gini}(D)$ 越小，纯度越高；同一类时为 0
- 选划分后基尼指数最小的属性
- 只需平方，不需对数，计算更快

$$\text{Gini}(D) = 1 - (8/17)^2 - (9/17)^2 = 0.498$$

例：按纹理（三叉）划分

清晰： $1 - (7/9)^2 - (2/9)^2 = 0.346$

稍糊： $1 - (1/5)^2 - (4/5)^2 = 0.320$

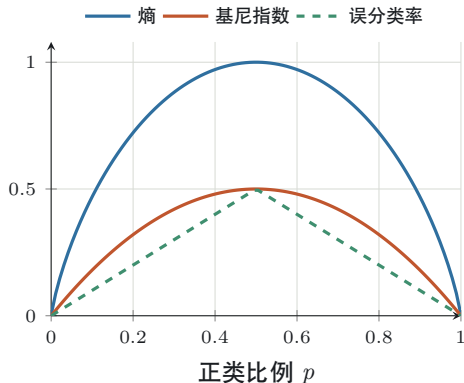
模糊： 0

$$\text{Gini_index} = \frac{9}{17} 0.346 + \frac{5}{17} 0.320 + 0 = 0.277$$

记号 D : 当前样本集合; K : 类别个数; k, k' : 类别编号; p_k : D 中第 k 类样本的比例; X_j : 第 j 个属性, V : 它的取值个数; D^v : 取第 v 个值的样本子集。

三种指标的对比

二分类时，把熵、基尼指数画成正类比例 p 的函数



- 三者都在 $p = 0$ 或 1 时为 0 ，在 $p = 0.5$ 时最大
- 熵与基尼指数形状非常接近，实践中选出的属性通常相同
- 误分类率 $\min(p, 1 - p)$ 不够“灵敏”，很少用作划分指标

要点

指标只负责给一个候选划分打分，不决定树分几叉，也不限制连续属性。指标的选择对结果影响不大；剪枝的影响大得多。

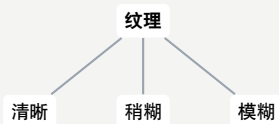
记号 p ：二分类时正类样本的比例，负类比例为 $1 - p$ 。

问题 2（离散属性）：多叉还是二叉？

以纹理为例：它有 3 个取值

多叉划分（ID3、C4.5）

每个取值一个分支



用过的离散属性不再使用

二叉划分（CART）

把取值分成两组，枚举所有分法，选基尼指数最小的：

分组	两组（好/总）		Gini_index
{清晰} vs {稍糊, 模糊}	7/9	1/8	0.286
{稍糊} vs {清晰, 模糊}	1/5	7/12	0.437
{模糊} vs {清晰, 稍糊}	0/3	8/14	0.403

V 个取值共有 $2^{V-1} - 1$ 种二分法；同一属性可以在后代结点中再次划分

记号 V ：属性的取值个数（纹理 $V = 3$ ，共 $2^{3-1} - 1 = 3$ 种二分法）。

问题 2（连续属性）：二分法

连续属性（如密度 X_7 ）的取值无穷多，不能每个值一个分支

1. 把 D 中 X_j 出现过的 n 个不同取值从小到大排序： $a^1 < a^2 < \dots < a^n$
2. 候选阈值取相邻两值的中点：

$$T_j = \left\{ \frac{a^s + a^{s+1}}{2} : s = 1, \dots, n-1 \right\}$$

3. 每个阈值 t 把 D 分成两部分： $D_t^- = \{(\mathbf{x}_i, y_i) \in D : x_{ij} \leq t\}$, $D_t^+ = \{(\mathbf{x}_i, y_i) \in D : x_{ij} > t\}$
4. 对每个 t 计算指标，取最好的：

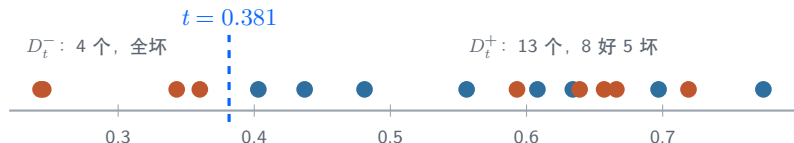
$$\text{Gain}(D, X_j) = \max_{t \in T_j} \text{Ent}(D) - \frac{|D_t^-|}{|D|} \text{Ent}(D_t^-) - \frac{|D_t^+|}{|D|} \text{Ent}(D_t^+)$$

阈值取在 $[a^s, a^{s+1})$ 内的任何位置，划分结果都一样，所以只需检查 $n-1$ 个中点。基尼指数的用法完全相同。

记号 D : 当前样本集合； X_j : 一个连续属性； n : X_j 在 D 中的不同取值个数， $a^1 < \dots < a^n$: 排序后的取值， s : 它们的下标； T_j : 候选阈值集合， t : 其中一个阈值； x_{ij} : 样本 $\mathbf{x}_i$ 在 X_j 上的取值； D_t^- , D_t^+ : 取值 $\leq t$ 与 $> t$ 的样本子集。

例 3：密度 X_7 的候选阈值

17 个样本的密度排序后画在数轴上（● 好瓜，● 坏瓜）



排序后的 17 个值产生 16 个候选阈值：

0.244, 0.294, 0.352, 0.381, 0.420, 0.459, 0.519,
0.575, 0.601, 0.621, 0.637, 0.648, 0.662, 0.682,
0.708, 0.747

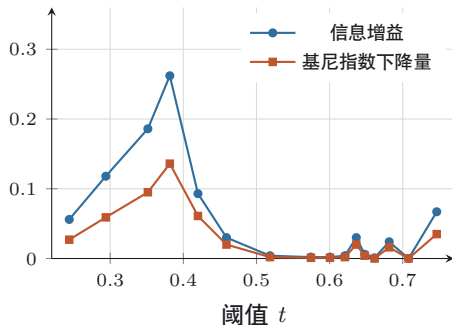
$$t = 0.381: \text{Gain} = 0.998 - \frac{4}{17} \cdot 0 - \frac{13}{17} \cdot 0.961 = 0.262$$

记号 X_7 : 密度; t : 阈值; D_t^- : 密度 $\leq t$ 的样本, D_t^+ : 密度 $> t$ 的样本; $0.998 = \text{Ent}(D)$, $0.961 = \text{Ent}(D_t^+)$ (8 好 5 坏)。

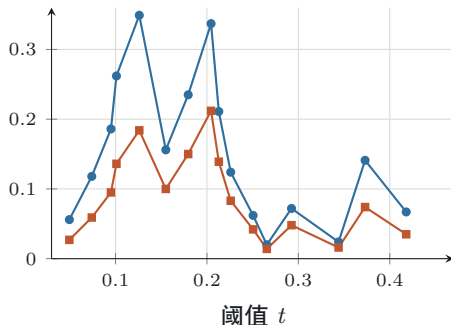
例 3（续）：扫描所有阈值

对每个候选阈值计算信息增益与基尼指数

密度 X_7



含糖率 X_8

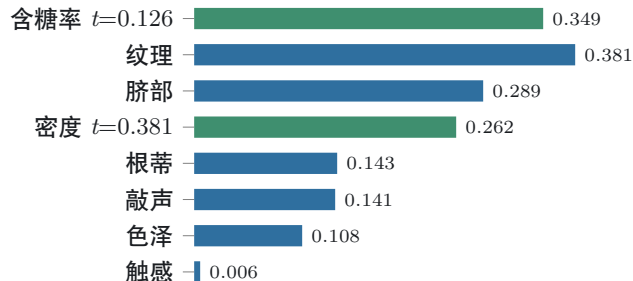


密度：两种指标都选 $t = 0.381$ （信息增益 0.262）。含糖率：信息增益选 $t = 0.126$ (0.349)，**基尼指数选 $t = 0.205$** (Gini_index = 0.286) ——指标不同，阈值可以不同。

记号 t ：候选阈值； X_7 ：密度， X_8 ：含糖率；基尼指数下降量 = $\text{Gini}(D) - \text{Gini_index}$ ，越大越好。

把离散与连续放在一起：根结点选谁？

西瓜数据集 3.0 上，以信息增益为指标



- 连续属性先用各自的最优阈值算出增益，再和离散属性一起比较
- 纹理 0.381 仍然最大，根结点按纹理划分
- 若在“纹理 = 清晰”分支中继续，连续属性可以被再次使用，例如先按密度 ≤ 0.381 ，再按密度 ≤ 0.56

记号 t ：连续属性各自的最优阈值（使信息增益最大）。

指标不决定树的形状

任何指标都只是给“一个候选划分”打分；分几叉、阈值怎么取，是问题 2 的事

	多叉	二叉	连续二分
	纹理三支	{清晰} vs {稍糊, 模糊}	密度 ≤ 0.381 vs > 0.381
信息增益（越大越好）	0.381	0.337	0.262
Gini_index（越小越好）	0.277	0.286	0.362

要点

表中 6 个格子都能算：熵和基尼指数都是“划分后各子集纯度的加权平均”，公式对 $V=2$ 、 $V=3$ 或阈值二分都成立。所以“ID3 只能处理离散属性”“CART 只能二分”都不是指标造成的，而是算法的设计选择（下两页）。

记号 V ：分支个数； D^v ：第 v 个分支的样本子集； $\text{Gini_index} = \sum_v \frac{|D^v|}{|D|} \text{Gini}(D^v)$ ：划分后的加权基尼指数；二叉一列中 $V=2$ ，连续二分一列中两个子集为 D_t^- , D_t^+ （阈值 $t=0.381$ ）。

小结：训练决策树的两个问题

同一棵树里，每个结点都按下表回答

	离散属性	连续属性
问题 2：怎么分	每个取值一支（多叉）；或枚举二分组（CART）	排序，取相邻中点为候选阈值，分成 $\leq t$ 与 $> t$
问题 1：用什么指标	信息增益 / 增益率 / 基尼指数	同左，对每个阈值计算后取最优
能否在后代中再用	多叉：不能；二叉：可以	可以

回归树

若 y 是连续值，CART 改用平方误差：选使 $\sum_{i \in D_t^-} (y_i - \bar{y}^-)^2 + \sum_{i \in D_t^+} (y_i - \bar{y}^+)^2$ 最小的划分；叶结点输出均值 $\bar{y}$ 。

记号 t ：阈值； y_i ：样本 i 的标签（回归时是实数）； D_t^- , D_t^+ ：按阈值 t 分成的两部分； $\bar{y}^-$, $\bar{y}^+$ ：两部分标签的平均值； $\bar{y}$ ：叶结点中样本标签的平均值。

ID3 · C4.5 · CART

第一行是指标；其余各行是算法的设计选择，与指标无关

	ID3 (1986)	C4.5 (1993)	CART (1984)
划分指标	信息增益	增益率（启发式）	基尼指数；回归用平方误差
离散属性怎么分	多叉：每个取值一支	多叉：每个取值一支	二叉：取值分成两组
连续属性怎么分	不支持	二分法（阈值 t ）	二分法（阈值 t ）
任务	分类	分类	分类 + 回归
剪枝	无	后剪枝	后剪枝

三者框架相同（递归地回答问题 1 和问题 2）。scikit-learn 的决策树实现的是 CART。

记号 t ：连续属性的划分阈值；年份为算法发表时间。

为什么三种算法的树形不同？

都是设计上的取舍，不是指标本身的限制

ID3：不支持连续属性

最初只为离散（符号）数据设计。

若把连续属性的每个取值当作一个分支，就和“编号”一样：分支极多、每支极纯，信息增益虚高。

需要先用阈值二分，这正是 C4.5 补上的。

C4.5：多叉 + 增益率

离散属性沿用多叉，连续属性用阈值二分。

多叉划分下，取值多的属性分支多，信息增益偏高，所以用增益率（除以 IV）来校正。

增益率是“多叉”带来的需要，不是反过来。

CART：一律二叉

多叉可以由连续几次二叉实现；而二叉有几个好处：

每次只切两份，数据碎片化更慢；

候选划分都只有两支，比较公平，不需要增益率校正；

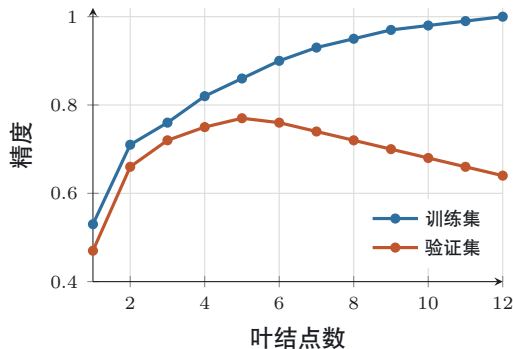
离散、连续、回归统一处理。

二叉的计算技巧：二分类时不必枚举 $2^{V-1} - 1$ 种分组，把 V 个取值按正类比例排序，只需检查 $V - 1$ 个切点（纹理：模糊 $0 < \text{稍糊 } 0.2 < \text{清晰 } 0.78$ ，最优为 {模糊, 稍糊} vs {清晰}）。

记号 IV：属性的固有值（按取值分组本身的熵）； V ：离散属性的取值个数；正类比例：该取值下好瓜所占的比例。

过拟合：树越深越好吗？

一直划分到每个叶结点都纯，训练误差可以降到 0
示意



- 分支过多时，树把训练样本自身的特点当成了一般规律
- 训练精度一直上升，验证精度先升后降
- 剪枝：主动去掉一些分支，降低过拟合风险
- 判断依据：在验证集上的精度

预剪枝：生成过程中就停下

每次划分前先问：划分后验证集精度会提高吗？不会就不分，当前结点直接作为叶结点

例：训练集为 1, 2, 3, 6, 7, 10, 14, 15, 16, 17 号（5 好 5 坏），验证集为 4, 5, 8, 9, 11, 12, 13 号（3 好 4 坏）

不划分

根结点作为叶结点，标记为训练集多数类（平票取好瓜）

验证集：3 个好瓜判对

精度 = $3/7 = 42.9\%$

按脐部 X_5 划分（训练集上增益最大）

凹陷：训练 3 好 1 坏 → 好瓜 验证 4, 5, 13 对 2

稍凹：训练 2 好 2 坏 → 好瓜 验证 8, 9 对 1

平坦：训练 0 好 2 坏 → 坏瓜 验证 11, 12 对 2

精度 = $5/7 = 71.4\% > 42.9\%$ ，划分

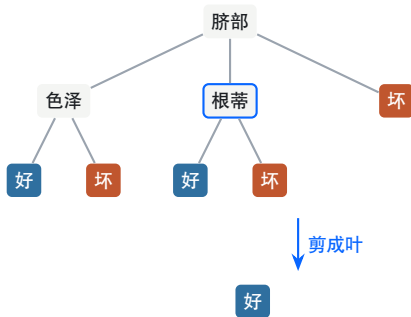
对每个新结点重复同样的判断，直到划分不再提高验证精度。

记号 X_5 ：脐部。

后剪枝：先长满，再自底向上剪

先生成完整的决策树，再逐个考察内部结点

1. 从最深的内部结点开始
2. 把它的整棵子树替换为一个叶结点（标记为子树样本的多数类）
3. 如果验证集精度**不下降**，就保留这次剪枝
4. 向上重复，直到根结点



预剪枝 vs 后剪枝

预剪枝

划分前用验证集判断，不提升就停

优点

很多分支根本不展开，训练和测试都快

缺点

“贪心”：当前划分不提升，不代表后续划分也没用，有欠拟合风险

后剪枝

先生成完整的树，再自底向上考察

优点

保留更多分支，欠拟合风险小，泛化性能通常更好

缺点

要先生成完整的树，再逐个考察，训练开销大得多

决策树：小结与延伸

优点

- 可解释：规则一目了然
- 不需要特征归一化
- 天然处理多分类和离散属性

缺点

- 不稳定：数据稍变，树可能大变
- 单棵树容易过拟合
- 决策边界与坐标轴平行

延伸

- 缺失值：按样本权重分到各分支
- 多变量决策树：用属性的线性组合划分
- 集成学习：随机森林、GBDT、XGBoost

单棵决策树很少直接使用，但它是随机森林、梯度提升树等表格数据上最强方法的基学习器。

02

集成学习

Ensemble Learning

为什么集成 · Bagging 与随机森林 · Boosting · 与 Dropout、MoE 的联系

从一棵树到一片森林

单棵决策树的问题：不稳定，方差大

- 训练数据稍有变化，树的结构就可能完全不同：根结点换一个属性，下面的整棵树都跟着变
- 例：“纹理 = 清晰”分支中，根蒂、脐部、触感的信息增益并列 0.458，任选其一就长出不同的树
- 单棵树的预测因此波动很大：方差大

集成学习

训练 T 个个体学习器 $h_1, \dots, h_T$ ，再把它们的预测结合起来：

$$H(\mathbf{x}) = \text{sign}\left(\sum_{t=1}^T h_t(\mathbf{x})\right) \quad (\text{投票})$$

$$H(\mathbf{x}) = \frac{1}{T} \sum_{t=1}^T h_t(\mathbf{x}) \quad (\text{回归：平均})$$

记号 $\mathbf{x}$ ：样本； T ：个体学习器的个数； h_t ：第 t 个个体学习器，二分类时 $h_t(\mathbf{x}) \in \{-1, +1\}$ ； H ：集成后的学习器； sign ：取符号。

集成为什么有效：好而不同

三个分类器 h_1, h_2, h_3 在三个测试样本上投票（✓ 分对，× 分错）

集成提升

	例 1	例 2	例 3
h_1	✓	✓	×
h_2	×	✓	✓
h_3	✓	×	✓
集成	✓	✓	✓
各 66.7% → 100%			

集成无效

	例 1	例 2	例 3
h_1	✓	✓	×
h_2	✓	✓	×
h_3	✓	✓	×
集成	✓	✓	×
三者一模一样：66.7%			

集成变差

	例 1	例 2	例 3
h_1	✓	×	×
h_2	×	✓	×
h_3	×	×	✓
集成	×	×	×
各 33.3% → 0%			

个体学习器要“好而不同”：准确率高於随机猜测，且犯的错误的不同。只有错误互相错开，投票才能把它们抵消。

记号 h_1, h_2, h_3 ：三个个体学习器；集成：三者多数投票的结果。例子取自周志华《机器学习》图 8.2。

多数投票：错误率随学习器个数指数下降

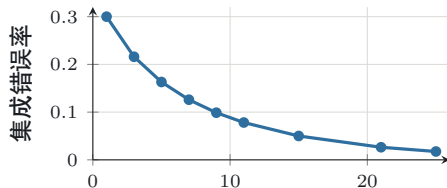
理想情况：每个个体学习器错误率都是 $\epsilon < 0.5$ ，且错误相互独立

$$P(H(\mathbf{x}) \neq y) = \sum_{k=\lceil T/2 \rceil}^T \binom{T}{k} \epsilon^k (1-\epsilon)^{T-k} \leq \exp\left(-\frac{1}{2} T(1-2\epsilon)^2\right)$$

超过半数个体分错，集成才会分错；不等式（Hoeffding）说明错误率随 T 指数下降。

T	1	5	11	15	25
集成错误率	0.300	0.163	0.078	0.050	0.018

($\epsilon = 0.3$)



个体学习器个数 T

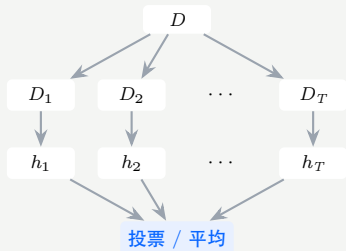
现实中：个体学习器用同一份数据训练，错误高度相关，“独立”并不成立。集成方法的核心问题是如何让个体学习器尽量不同。

记号 ϵ ：单个学习器的错误率； T ：个体学习器个数（取奇数，避免平票）； y ：真实标签； $H(\mathbf{x})$ ：多数投票结果； $\binom{T}{k}$ ：组合数； $\lceil \cdot \rceil$ ：向上取整。

两类集成方法

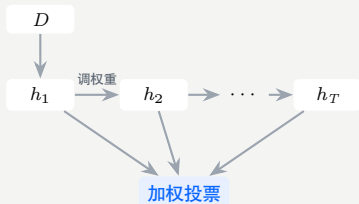
制造“不同”的两种思路

并行：Bagging、随机森林



各学习器独立训练，差异来自不同的数据（和属性）子集；主要降低方差

串行：Boosting



后一个学习器重点关注前一个分错的样本；主要降低偏差

记号 D : 训练集; D_t : 第 t 个学习器使用的训练集 (自助采样得到); h_t : 第 t 个个体学习器; T : 个体学习器个数。

Bagging (一): 自助采样

Bootstrap AGGregatING (Breiman, 1996): 用有放回抽样制造 T 个不同的训练集

自助采样: 从 m 个样本中有放回地抽 m 次, 得到 D_t

某个样本一次也没被抽中的概率:

$$\left(1 - \frac{1}{m}\right)^m \xrightarrow{m \rightarrow \infty} \frac{1}{e} \approx 0.368$$

所以每个 D_t 大约包含原数据中 **63.2%** 的不同样本;
其余约 36.8% 称为**包外样本** (out-of-bag)。

$m = 17$ 时: $1 - (16/17)^{17} = 0.643$

例: 西瓜数据集, $m = 17$, 抽一次

抽到的编号 (有重复):

1, 3, 5, 5, 5, 7, 8, 8, 8, 9, 12, 13, 16, 16, 16,
16, 17

不同样本: 10 个 (58.8%)

包外样本: 2, 4, 6, 10, 11, 14, 15

每次抽样都不同, 于是 T 棵树看到的数据各不相同

记号 m : 训练样本数; D_t : 第 t 个自助采样集 (大小也是 m); T : 采样次数, 即个体学习器个数; $e \approx 2.718$: 自然常数。

Bagging（二）：结合与包外估计

每个 D_t 训练一个学习器 h_t ，最后等权结合

结合：分类用多数投票，回归用平均

$$H(\mathbf{x}) = \arg \max_k \sum_{t=1}^T \mathbb{I}(h_t(\mathbf{x}) = k)$$

- 各学习器互不依赖，可以并行训练
- 训练代价约为单个学习器的 T 倍
- 对“不稳定”的学习器（如不剪枝的深决策树）效果最好

包外估计：免费的验证集

样本 $\mathbf{x}_i$ 只用没见过它的那些学习器（约 36.8%）来预测：

$$H^{\text{oob}}(\mathbf{x}_i) = \arg \max_k \sum_{t: \mathbf{x}_i \notin D_t} \mathbb{I}(h_t(\mathbf{x}_i) = k)$$

这些预测的错误率就是泛化误差的估计，不需要另留验证集。

记号 h_t ：用 D_t 训练的学习器； T ：学习器个数； k ：类别； $\mathbb{I}(\cdot)$ ：指示函数，条件成立为 1，否则为 0； $\arg \max_k$ ：得票最多的类别； $\mathbf{x}_i \notin D_t$ ：样本 i 不在第 t 个自助采样集中；oob：out-of-bag，包外。

随机森林：在 Bagging 上再加一层随机

Random Forest (Breiman, 2001): 基学习器是决策树，每个结点只在随机选出的部分属性中挑最优

对 $t = 1, \dots, T$:

1. 自助采样得到 D_t
 2. 在 D_t 上训练一棵决策树；在每个结点：
 - 从当前 d' 个候选属性中**随机选** k 个
 - 只在这 k 个属性中选最优划分
 3. 树长到底，通常不剪枝
- 最后对 T 棵树投票（回归取平均）

- $k = d'$: 退化为普通的 Bagging 决策树
- $k = 1$: 完全随机选一个属性
- 常用 $k = \log_2 d$ (西瓜书推荐) 或 $k = \sqrt{d}$ (scikit-learn 分类默认)
- 西瓜数据集 $d = 8$: 每个结点只看 $\log_2 8 = 3$ 个随机属性



三棵树的根结点各不相同

记号 T : 树的棵数; D_t : 第 t 个自助采样集; d : 属性总数; d' : 当前结点的候选属性个数; k : 每个结点随机选取的属性个数。

为什么随机选属性：降低树与树之间的相关性

平均 T 个预测，方差能降多少取决于它们彼此有多相关

$$\text{Var}\left(\frac{1}{T} \sum_{t=1}^T h_t(\mathbf{x})\right) = \rho \sigma^2 + \frac{1-\rho}{T} \sigma^2$$

	$T = 10$	$T = 100$
$\rho = 0$ （完全独立）	$0.10 \sigma^2$	$0.01 \sigma^2$
$\rho = 0.3$	$0.37 \sigma^2$	$0.31 \sigma^2$
$\rho = 0.5$	$0.55 \sigma^2$	$0.51 \sigma^2$

- 第二项随 T 增大趋于 0，第一项 $\rho \sigma^2$ 降不下去
- 只做 Bagging：若“纹理”特别强，几乎每棵树的根结点都是纹理，树与树高度相关
- 随机选属性后，有些树的根结点看不到纹理，只能用别的属性： ρ 变小

要点

随机森林用“每棵树稍差一点”换来“树与树更不相关”，整体方差反而更小。

记号 $h_t(\mathbf{x})$ ：第 t 棵树的预测，看作随机变量； σ^2 ：单棵树预测的方差； ρ ：任意两棵树预测之间的相关系数； T ：树的棵数； Var ：方差。

Boosting: 串行地纠正错误（以 AdaBoost 为例）

AdaBoost (Freund & Schapire, 1997): 每一轮加大分错样本的权重

初始化: $w_i = 1/m, i = 1, \dots, m$

对 $t = 1, \dots, T$:

1. 按权重 w_i 训练 h_t , 计算加权错误率

$$\epsilon_t = \sum_i w_i \mathbb{I}(h_t(\mathbf{x}_i) \neq y_i)$$

2. 学习器权重 $\alpha_t = \frac{1}{2} \ln \frac{1-\epsilon_t}{\epsilon_t}$

3. 更新 $w_i \leftarrow w_i \exp(-\alpha_t y_i h_t(\mathbf{x}_i)) / Z_t$

输出: $H(\mathbf{x}) = \text{sign}(\sum_t \alpha_t h_t(\mathbf{x}))$

例: 某轮 $\epsilon_t = 0.3$

$$\alpha_t = \frac{1}{2} \ln \frac{0.7}{0.3} = 0.424$$

分错的样本: 权重 $\times e^{0.424} = 1.53$

分对的样本: 权重 $\times e^{-0.424} = 0.65$

(再归一化)

梯度提升树

GBDT、XGBoost、LightGBM: 每一轮新加一棵树, 去拟合当前集成的残差 (负梯度)。表格数据上最常用的方法之一。

记号 m : 样本数; w_i : 样本 i 的权重; $y_i \in \{-1, +1\}$: 标签; $h_t(\mathbf{x}) \in \{-1, +1\}$: 第 t 轮的学习器; ϵ_t : 加权错误率; α_t : h_t 在最终投票中的权重; Z_t : 使权重之和为 1 的归一化常数; $\mathbb{I}$: 指示函数。

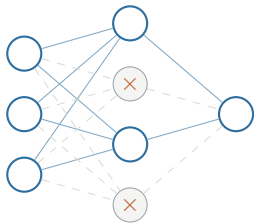
Bagging 与 Boosting 对比

	Bagging / 随机森林	Boosting
训练方式	并行，各学习器相互独立	串行，后一个依赖前一个
多样性来源	自助采样（+ 随机属性）	样本权重（或残差）
结合方式	等权投票 / 平均	加权投票 / 累加
主要降低	方差	偏差
典型基学习器	深树（低偏差、高方差）	浅树、树桩（高偏差）
代表方法	随机森林	AdaBoost、GBDT、XGBoost、LightGBM

树桩：只有一次划分的决策树。两类方法都常以决策树为基学习器，这也是决策树在今天仍然重要的原因。

与深度学习的联系（一）：Dropout

Dropout (Srivastava 等, 2014) 可以看作一种“共享参数的隐式集成”



这一步随机丢弃了 2 个隐藏单元

- **训练**：每一步以概率 p 随机丢弃每个隐藏单元，得到一个随机的子网络
- n 个可丢弃单元 $\Rightarrow$ 共有 2^n 个可能的子网络，它们共享同一套参数
- **测试**：用完整网络，把权重乘以保留概率 $1 - p$ （现代实现改为训练时除以 $1 - p$ ，等价），近似为对这 2^n 个子网络的预测取平均

与 Bagging 的异同

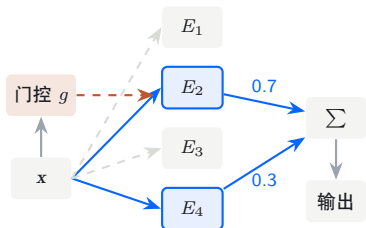
同：用随机性制造许多不同的模型，再平均。

异：Bagging 的模型相互独立、各自完整训练；Dropout 的子网络共享参数，每个只被训练一小步。

记号 p ：丢弃概率； $1 - p$ ：保留概率； n ：可被丢弃的单元个数； 2^n ：可能的子网络个数。

与深度学习的联系（二）：混合专家（MoE）

Mixture of Experts (Jacobs 等, 1991): 由一个门控网络决定, 每个输入交给哪些专家



例: $E = 4$ 个专家, 只激活 $k = 2$ 个

$$y = \sum_{i=1}^E g_i(\mathbf{x}) E_i(\mathbf{x})$$
$$g(\mathbf{x}) = \text{softmax}(\text{TopK}(W_g \mathbf{x}))$$

大模型中：MoE 层替代 Transformer 的前馈层, 每个 token 只交给 k 个专家 (Mixtral $8 \times 7\text{B}$: $E = 8$, $k = 2$)。参数量随 E 增长, 计算量只随 k 增长。

与集成的异同

同: 多个子模型的输出加权结合。异: 集成中每个成员处理所有输入、权重固定; MoE 的权重随输入变化, 专家各有分工, 与门控一起端到端训练。

记号 x : 输入 (大模型中为一个 token 的表示); E : 专家个数, E_i : 第 i 个专家网络; $g_i(x)$: 门控给第 i 个专家的权重; W_g : 门控参数; TopK: 只保留最大的 k 个分数, 其余置为 $-\infty$; y : 输出。

小结：集成的思想无处不在

方法	多样性从哪里来	怎么结合
Bagging	自助采样	等权投票 / 平均
随机森林	自助采样 + 每个结点随机选属性	等权投票 / 平均
Boosting	调整样本权重（或拟合残差）	加权投票 / 累加
Dropout	训练时随机丢弃神经元	测试时用完整网络，近似平均
MoE	按输入路由到不同专家	门控网络给出的权重

要点

表格数据：随机森林与梯度提升树（XGBoost、LightGBM）至今仍是首选的强基线。深度学习中，集成思想以 Dropout、MoE、模型平均等形式延续。

03

支持向量机：最大间隔

Support Vector Machines

间隔 · 为什么约束是 ± 1 · 最大间隔 · 支持向量

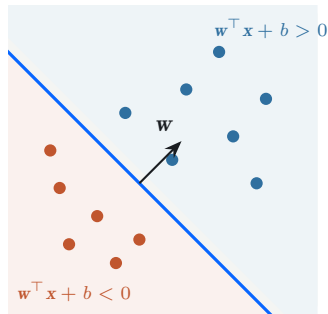
问题设定：线性二分类

标签 $y_i \in \{-1, +1\}$ ，用一个超平面把两类分开

超平面： $\mathbf{w}^\top \mathbf{x} + b = 0$

预测： $\hat{y} = \text{sign}(\mathbf{w}^\top \mathbf{x} + b)$

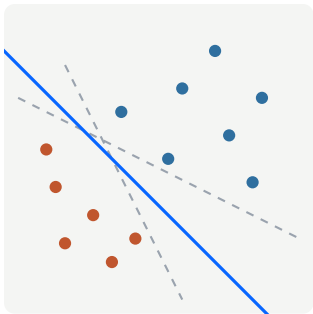
- $\mathbf{w} \in \mathbb{R}^d$: 法向量，决定超平面的方向
- $b \in \mathbb{R}$: 偏置，决定超平面的位置
- $\mathbf{w}^\top \mathbf{x} + b > 0$ 预测为 $+1$, < 0 预测为 -1
- 分类正确 $\iff y_i(\mathbf{w}^\top \mathbf{x}_i + b) > 0$



记号 $\mathbf{x} \in \mathbb{R}^d$: 样本（特征向量）， $\mathbf{x}_i$: 第 i 个训练样本； $y_i \in \{-1, +1\}$: 其标签； $\mathbf{w}$: 法向量， b : 偏置； $^\top$: 转置； $\hat{y}$: 预测的标签； sign : 取符号。

哪一个超平面最好？

能把训练集分开的超平面有无数个，训练误差都是 0



- 两条虚线离某些样本非常近：样本稍有扰动（噪声、测量误差）就会被分错
- 蓝色实线位于两类“正中间”，对扰动的容忍度最大

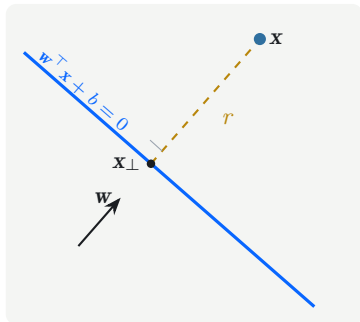
直觉

让离超平面最近的样本离得尽可能远。

要把直觉变成数学，先要会算“样本到超平面的距离”。

点到超平面的距离

推导：把 x 分解为超平面上的投影点加上沿法向量方向的一段



1. w 垂直于超平面：超平面上任意两点 x', x'' 满足 $w^T(x' - x'') = 0$
2. 于是 $x = x_{\perp} + r \frac{w}{\|w\|}$ ，其中 $x_{\perp}$ 在超平面上， r 是带符号的距离
3. 两边左乘 w^T 再加 b ，并利用 $w^T x_{\perp} + b = 0$ ：

$$w^T x + b = 0 + r \frac{w^T w}{\|w\|} = r \|w\|$$

$$r = \frac{w^T x + b}{\|w\|}$$

记号 x ：任意一点； $x_{\perp}$ ：它在超平面上的投影； r ：带符号的距离； x', x'' ：超平面上任意两点； w, b ：超平面参数； $\|w\| = \sqrt{w^T w}$ ： w 的长度。

函数间隔与几何间隔

把“距离”和“是否分对”合在一个量里：乘上标签 y_i

函数间隔

$$\hat{\gamma}_i = y_i(\mathbf{w}^\top \mathbf{x}_i + b)$$

- $\hat{\gamma}_i > 0 \iff$ 样本 i 被分对
- 数值大小不是距离：依赖于 w, b 的尺度

几何间隔

$$\gamma_i = \frac{y_i(\mathbf{w}^\top \mathbf{x}_i + b)}{\|\mathbf{w}\|} = \frac{\hat{\gamma}_i}{\|\mathbf{w}\|}$$

- 分对时，就是样本到超平面的真实距离
- 与 w, b 的尺度无关

整个训练集的（几何）间隔： $\gamma = \min_{i=1, \dots, m} \gamma_i$ ， 函数间隔 $\hat{\gamma} = \min_i \hat{\gamma}_i$ ， $\gamma = \hat{\gamma} / \|\mathbf{w}\|$

记号 $\mathbf{x}_i, y_i$ ：第 i 个样本及其标签（ ± 1 ）； $\mathbf{w}, b$ ：超平面参数； $\|\mathbf{w}\|$ ： $\mathbf{w}$ 的长度； m ：样本数； $\hat{\gamma}_i, \gamma_i$ ：样本 i 的函数间隔、几何间隔； $\hat{\gamma}, \gamma$ ：它们在训练集上的最小值。

关键观察：同一个超平面有无穷多种写法

把 $(\mathbf{w}, b)$ 同时乘以任意 $\lambda > 0$ ，超平面本身不变

$$\lambda \mathbf{w}^\top \mathbf{x} + \lambda b = 0 \iff \mathbf{w}^\top \mathbf{x} + b = 0$$

以超平面 $x_1 + x_2 - 5 = 0$ 和正类样本 $\mathbf{x} = (4.5, 3.5)^\top$ 为例：

$(\mathbf{w}, b)$	函数间隔 $\hat{\gamma}$	几何间隔 γ
$((1, 1), -5)$	3	$3/\sqrt{2} = 2.12$
$((2, 2), -10)$	6	$6/\sqrt{8} = 2.12$
$((\frac{1}{3}, \frac{1}{3}), -\frac{5}{3})$	1	$1/\sqrt{2/9} = 2.12$

要点

- 函数间隔随 λ 任意缩放：想要多大就有多大
- 几何间隔不变：它才是真正的距离
- 所以我们要最大化的是几何间隔 γ

这个“缩放自由度”正是下一步把约束写成 ± 1 的依据。

记号 $\lambda > 0$ ：任意缩放系数； x_1, x_2 ：二维向量 $\mathbf{x}$ 的两个坐标分量（不是样本）； $\hat{\gamma}, \gamma$ ：函数间隔、几何间隔； $\mathbf{w}, b$ ：超平面参数。

最大间隔问题：最初的形式

最大化训练集的几何间隔，同时要求每个样本的几何间隔都不小于它

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

用 $\gamma = \hat{\gamma} / \|\mathbf{w}\|$ 改写成函数间隔：

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

- 问题有三个“变量”： $\mathbf{w}$ ， b ，以及 $\hat{\gamma}$
- 但 $\hat{\gamma}$ 的值不影响最优超平面：把 $(\mathbf{w}, b)$ 缩放 λ 倍， $\hat{\gamma}$ 也缩放 λ 倍，目标 $\hat{\gamma} / \|\mathbf{w}\|$ 和约束都不变
- 所以可以**随便固定 $\hat{\gamma}$ 的值**——为了简单，固定为 1

记号 γ ：训练集的几何间隔， $\hat{\gamma}$ ：函数间隔； $\mathbf{x}_i, y_i$ ：第 i 个样本及其标签， m ：样本数； $\mathbf{w}, b$ ：超平面参数； λ ：缩放系数；s.t.：subject to, “满足约束”。

为什么是 ± 1 ，而不是 ± 2 或 ± 10000 ？

答： ± 1 是一种约定（归一化），不是说支持向量到超平面的距离为 1

- 取 $\hat{\gamma} = 1$ ：在同一超平面的无穷多种写法中，挑出使 $\min_i y_i(\mathbf{w}^\top \mathbf{x}_i + b) = 1$ 的那一种（称为规范超平面）
- 取 $\hat{\gamma} = 2$ 也完全可以：得到的 $(\mathbf{w}, b)$ 恰好是 $\hat{\gamma} = 1$ 时的 2 倍，超平面相同
- 真实的距离由 $\|\mathbf{w}\|$ 承担：支持向量到超平面的距离 $= 1/\|\mathbf{w}\|$

类比

“距离 = 1 个单位”，单位可以是米、也可以是公里。 ± 1 固定的是“单位”， $\|\mathbf{w}\|$ 决定这个单位有多长。

固定 $\hat{\gamma} = 1$ 后：

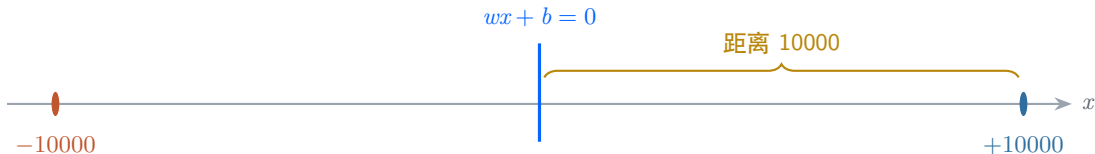
$$\gamma = \frac{\hat{\gamma}}{\|\mathbf{w}\|} = \frac{1}{\|\mathbf{w}\|}$$

最大化间隔 $\iff$ 最小化 $\|\mathbf{w}\|$

记号 $\hat{\gamma}$ ：函数间隔， γ ：几何间隔； $\mathbf{w}, b$ ：超平面参数， $\|\mathbf{w}\|$ ： $\mathbf{w}$ 的长度； $\mathbf{x}_i, y_i$ ：第 i 个样本及其标签。

例：两类相距 20000，支持向量怎么还是 ± 1 ？

一维数据：负类在 $x = -10000$ ，正类在 $x = +10000$ ，最优分界点显然是 $x = 0$



写法	正类处 $w x + b$	负类处 $w x + b$	间隔 $= \hat{\gamma}/ w $
$w = 1, b = 0$	$+10000$	-10000	$10000/1 = 10000$
$w = 0.5, b = 0$	$+5000$	-5000	$5000/0.5 = 10000$
$w = 10^{-4}, b = 0$ （规范写法）	$+1$	-1	$1/10^{-4} = 10000$

三种写法是同一个分界点。规范写法让 $w x + b = \pm 1$ ，而间隔 10000 体现在 $|w| = 10^{-4}$ 很小上：数据离得越远， $\|w\|$ 越小。

记号 一维情形： x ：样本（一个数）； w, b ：分界点 $w x + b = 0$ 的参数； $\hat{\gamma}$ ：函数间隔。

SVM 的基本型

固定 $\hat{\gamma} = 1$ 后，再做两步等价变换

$$\max_{\mathbf{w}, b} \frac{1}{\|\mathbf{w}\|} \quad \text{s.t. } y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 \quad \Longleftrightarrow \quad \min_{\mathbf{w}, b} \|\mathbf{w}\| \quad \text{s.t. } y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1$$

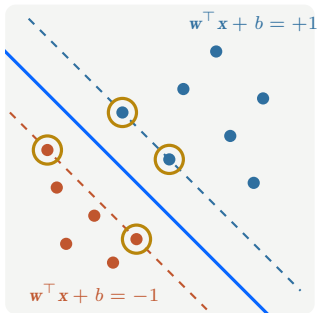
$$\min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 \quad \text{s.t. } y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1, \quad i = 1, \dots, m$$

- 最小化 $\|\mathbf{w}\|$ 与最小化 $\frac{1}{2} \|\mathbf{w}\|^2$ 的最优解相同（平方是单调变换）
- 平方后处处可导；系数 $\frac{1}{2}$ 只为求导后形式简洁
- 目标是凸二次函数，约束是 m 个线性不等式：凸二次规划，有全局最优解

记号 $\mathbf{w}, b$ ：超平面参数， $\|\mathbf{w}\|$ ： $\mathbf{w}$ 的长度； $\mathbf{x}_i, y_i$ ：第 i 个样本及其标签（ ± 1 ）； m ：样本数； $\hat{\gamma}$ ：函数间隔。

支持向量与间隔边界

恰好满足 $y_i(\mathbf{w}^\top \mathbf{x}_i + b) = 1$ 的样本



- 两条间隔边界： $\mathbf{w}^\top \mathbf{x} + b = \pm 1$
- **支持向量**：落在间隔边界上的样本（约束取等号）
- 间隔宽度：取边界上的点 $\mathbf{x}_+$, $\mathbf{x}_-$,

$$\mathbf{w}^\top (\mathbf{x}_+ - \mathbf{x}_-) = 2 \Rightarrow \frac{\mathbf{w}^\top}{\|\mathbf{w}\|} (\mathbf{x}_+ - \mathbf{x}_-) = \frac{2}{\|\mathbf{w}\|}$$

- 最优超平面只由支持向量决定：删掉其他样本，解不变

最后一点将由拉格朗日乘子法严格证明。

记号 $\mathbf{w}, b$ ：超平面参数； $\mathbf{x}_i, y_i$ ：第 i 个样本及其标签； $\mathbf{x}_+, \mathbf{x}_-$ ：分别落在 $\mathbf{w}^\top \mathbf{x} + b = +1$ 和 $= -1$ 上的点。

接下来：怎么解这个带约束的优化问题？

从“最大间隔”到“拉格朗日乘子法”

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

- 无约束问题：令梯度为 0 即可
- 这里有 m 个不等式约束，不能直接令梯度为 0
- 可以直接交给通用二次规划求解器，但当 m 很大时很慢

下一步

用拉格朗日乘子法把约束“吸收”进目标函数，得到对偶问题。好处：

1. 证明解只由支持向量决定
2. 样本只以内积 $\mathbf{x}_i^\top \mathbf{x}_j$ 出现 $\Rightarrow$ 核技巧
3. 有高效的专用算法（SMO）

记号 $\mathbf{w}, b$: 超平面参数; $\mathbf{x}_i, y_i$: 第 i 个样本及其标签; m : 样本数; $\mathbf{x}_i^\top \mathbf{x}_j$: 样本 i 与样本 j 的内积。

04

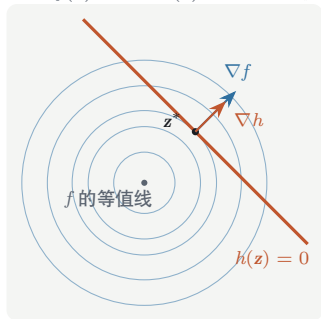
拉格朗日乘子法

Lagrange Multipliers & Duality

等式约束 · 不等式约束与 KKT 条件 · 对偶问题

等式约束：几何直觉

问题： $\min_z f(z)$ s.t. $h(z) = 0$ ，其中 z 是一般的优化变量



- 蓝色圆： f 的等值线；红线：可行的 z
- 沿约束曲线移动时，如果 ∇f 在切线方向上有分量， f 还能继续下降
- 最优点处， ∇f 必须垂直于约束曲线，也就是与 ∇h 平行：

$$\nabla f(z^*) + \lambda \nabla h(z^*) = \mathbf{0}, \text{ 某个 } \lambda \in \mathbb{R}$$

λ 称为拉格朗日乘子，可正可负。

记号 z ：优化变量； f ：目标函数； h ：约束函数； z^* ：最优点； ∇ ：梯度； λ ：拉格朗日乘子。

拉格朗日函数：把约束问题变成求驻点

定义 $L(\mathbf{z}, \lambda) = f(\mathbf{z}) + \lambda h(\mathbf{z})$ ，令它对所有变量的偏导为 0

$$\nabla_{\mathbf{z}} L = \nabla f(\mathbf{z}) + \lambda \nabla h(\mathbf{z}) = \mathbf{0} \quad (\text{梯度平行}), \quad \frac{\partial L}{\partial \lambda} = h(\mathbf{z}) = 0 \quad (\text{满足约束})$$

例： $\min z_1^2 + z_2^2 \quad \text{s.t.} \quad z_1 + z_2 = 1$

$$L = z_1^2 + z_2^2 + \lambda(z_1 + z_2 - 1)$$

$$\partial L / \partial z_1 = 2z_1 + \lambda = 0, \quad \partial L / \partial z_2 = 2z_2 + \lambda = 0$$

$$\partial L / \partial \lambda = z_1 + z_2 - 1 = 0$$

$$\Rightarrow z_1 = z_2 = \frac{1}{2}, \quad \lambda = -1, \quad \text{最小值 } \frac{1}{2}$$

要点

多个等式约束 $h_k(\mathbf{z}) = 0, \quad k = 1, \dots, p$ ：
每个约束一个乘子

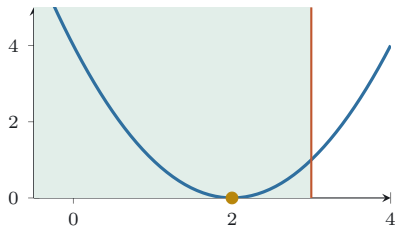
$$L(\mathbf{z}, \boldsymbol{\lambda}) = f(\mathbf{z}) + \sum_{k=1}^p \lambda_k h_k(\mathbf{z})$$

记号 f : 目标函数, h, h_k : 等式约束函数, p : 约束个数; $\mathbf{z} = (z_1, z_2, \dots)$: 优化变量; λ, λ_k : 拉格朗日乘子; L : 拉格朗日函数; $\nabla_{\mathbf{z}}$: 对 $\mathbf{z}$ 求梯度。

不等式约束：两种情形

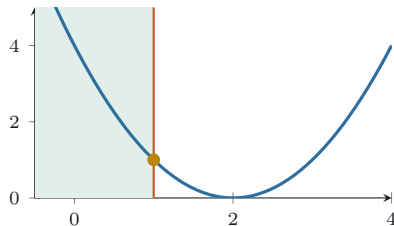
问题： $\min_z f(z)$ s.t. $g(z) \leq 0$ 。一维例子 $f(z) = (z-2)^2$

约束 $z \leq 3$ ：不起作用



最优点 $z^* = 2$ 在区域内部, $g(z^*) < 0$
和无约束时一样: $f'(z^*) = 0$, $\mu = 0$

约束 $z \leq 1$ ：起作用



最优点 $z^* = 1$ 在边界上, $g(z^*) = 0$
 $f'(1) = -2$, $\nabla g = 1$: $-2 + \mu \cdot 1 = 0 \Rightarrow \mu = 2 > 0$

绿色为可行域。两种情形下都有 $\mu g(z^*) = 0$ ：要么约束不起作用 ($\mu = 0$)，要么约束取等号 ($g = 0$)。

记号 z ：优化变量（此例为一个数）； f ：目标函数； g ：约束函数， $z \leq 3$ 即 $g(z) = z - 3 \leq 0$ ， $z \leq 1$ 即 $g(z) = z - 1 \leq 0$ ； z^* ：最优点； μ ：不等式约束的乘子。

KKT 条件

一般问题： $\min_{\mathbf{z}} f(\mathbf{z})$ s.t. $g_i(\mathbf{z}) \leq 0$ ($i = 1, \dots, m$), 拉格朗日函数 $L(\mathbf{z}, \boldsymbol{\mu}) = f(\mathbf{z}) + \sum_i \mu_i g_i(\mathbf{z})$

驻点条件 $\nabla f(\mathbf{z}^*) + \sum_i \mu_i \nabla g_i(\mathbf{z}^*) = \mathbf{0}$ 梯度平衡

原始可行 $g_i(\mathbf{z}^*) \leq 0$ 满足约束

对偶可行 $\mu_i \geq 0$ 乘子非负（方向不能反）

互补松弛 $\mu_i g_i(\mathbf{z}^*) = 0$ $\mu_i > 0 \Rightarrow g_i = 0; g_i < 0 \Rightarrow \mu_i = 0$

要点

对凸优化问题（ f 、 g_i 都是凸函数，且存在严格可行点），KKT 条件是最优解的充分必要条件。SVM 正属于这一类。

记号 f : 目标函数; g_i : 第 i 个不等式约束, 共 m 个; μ_i : 它的乘子, $\boldsymbol{\mu} = (\mu_1, \dots, \mu_m)$; L : 拉格朗日函数; $\mathbf{z}^*$: 最优解。

拉格朗日对偶问题

为什么要“对偶”：换一个等价的问题来解

1. 原问题可以写成 min-max

$$\min_z \max_{\mu \geq 0} L(\mathbf{z}, \mu)$$

若 $\mathbf{z}$ 违反某个约束 ($g_i > 0$)，让 $\mu_i \rightarrow \infty$ ，内层为 $+\infty$ ；
若 $\mathbf{z}$ 可行，内层最大值取 $\mu_i g_i = 0$ ，恰好等于 $f(\mathbf{z})$ 。

2. 交换 min 与 max，得到对偶问题

$$\max_{\mu \geq 0} \underbrace{\min_z L(\mathbf{z}, \mu)}_{\text{对偶函数 } \Gamma(\mu)}$$

弱对偶：对偶问题的最优值 $\leq$ 原问题的最优值（总成立）

强对偶：两者相等。凸问题且存在严格可行点（Slater 条件）时成立

SVM：目标凸，约束是线性的，数据线性可分时强对偶成立

内层 $\min_z L$ 是无约束问题，可以直接令梯度为 0 求出。

记号 $\mathbf{z}$ ：优化变量； f ：目标函数； g_i ：第 i 个不等式约束； $\mu \geq 0$ ：所有乘子非负； L ：拉格朗日函数； $\Gamma(\mu)$ ：对偶函数。

05

SVM 的对偶问题

The Dual Problem of SVM

拉格朗日函数 · 逐步推导对偶问题 · KKT 与支持向量 · 一个手算例子

SVM 为什么可以用拉格朗日乘子法

把基本型写成上一节的标准形式

$$\min_{\mathbf{w}, b} \underbrace{\frac{1}{2} \|\mathbf{w}\|^2}_{f(\mathbf{w}, b)} \quad \text{s.t.} \quad \underbrace{1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b)}_{g_i(\mathbf{w}, b)} \leq 0, \quad i = 1, \dots, m$$

- 优化变量是 $(\mathbf{w}, b)$ ，相当于上一节的 $\mathbf{z}$ ；数据 $\mathbf{x}_i, y_i$ 是已知常数
- f 是凸二次函数；每个 g_i 关于 $(\mathbf{w}, b)$ 是线性的
- 线性可分时存在严格可行点 $\Rightarrow$ KKT 条件充要，强对偶成立
- 每个样本对应一个约束，也就对应一个乘子。SVM 中习惯把 μ_i 记为 $\alpha_i \geq 0$

目标

求出对偶问题：只含 $\alpha = (\alpha_1, \dots, \alpha_m)^\top$ 的优化问题。

记号 f : 目标函数, g_i : 第 i 个约束; $\mathbf{w}, b$: 超平面参数; $\mathbf{x}_i, y_i$: 第 i 个样本及其标签, m : 样本数; α_i : 第 i 个约束 (样本) 的乘子, α : 所有乘子组成的向量。

第 1 步：写出拉格朗日函数

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2} \|\mathbf{w}\|^2 + \sum_{i=1}^m \alpha_i (1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b)), \quad \alpha_i \geq 0$$

展开，便于后面求导和代入：

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2} \mathbf{w}^\top \mathbf{w} + \sum_{i=1}^m \alpha_i - \sum_{i=1}^m \alpha_i y_i \mathbf{w}^\top \mathbf{x}_i - b \sum_{i=1}^m \alpha_i y_i$$

按对偶问题的定义，接下来要计算

$$\Gamma(\boldsymbol{\alpha}) = \min_{\mathbf{w}, b} L(\mathbf{w}, b, \boldsymbol{\alpha}), \quad \text{然后求 } \max_{\boldsymbol{\alpha} \geq \mathbf{0}} \Gamma(\boldsymbol{\alpha})$$

记号 $\mathbf{w}, b$: 超平面参数; $\mathbf{x}_i, y_i$: 第 i 个样本及其标签, m : 样本数; $\alpha_i \geq 0$: 第 i 个样本的乘子, $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_m)^\top$; L : 拉格朗日函数; $\Gamma(\boldsymbol{\alpha})$: 对偶函数。

第 2 步：对 w 和 b 求偏导，令其为 0

内层是关于 (w, b) 的无约束问题

对 w

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^m \alpha_i y_i x_i = 0$$

$$w = \sum_{i=1}^m \alpha_i y_i x_i$$

对 b

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

$$\sum_{i=1}^m \alpha_i y_i = 0$$

观察

w 是训练样本的线性组合，系数为 $\alpha_i y_i$ ：只有 $\alpha_i > 0$ 的样本对 w 有贡献。

记号 L ：拉格朗日函数； w, b ：超平面参数； α_i ：第 i 个样本的乘子； x_i, y_i ：第 i 个样本及其标签， m ：样本数； 0 ：零向量。

第 3 步：代回 L ，消去 $\mathbf{w}$ 和 b

把 $\mathbf{w} = \sum_j \alpha_j y_j \mathbf{x}_j$ 与 $\sum_i \alpha_i y_i = 0$ 代入展开式的每一项

$$\begin{aligned}\frac{1}{2} \mathbf{w}^\top \mathbf{w} &= \frac{1}{2} \left(\sum_i \alpha_i y_i \mathbf{x}_i \right)^\top \left(\sum_j \alpha_j y_j \mathbf{x}_j \right) = \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j \\ \sum_i \alpha_i y_i \mathbf{w}^\top \mathbf{x}_i &= \sum_i \alpha_i y_i \left(\sum_j \alpha_j y_j \mathbf{x}_j \right)^\top \mathbf{x}_i = \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j \\ b \sum_i \alpha_i y_i &= b \cdot 0 = 0\end{aligned}$$

第一项与第二项是同一个双重求和，只差系数 $\frac{1}{2}$ 与 1 。

记号 i, j : 样本下标，都取 $1, \dots, m$; α_i : 第 i 个样本的乘子; $\mathbf{x}_i, y_i$: 第 i 个样本及其标签; $\mathbf{x}_i^\top \mathbf{x}_j$: 样本 i 与 j 的内积; $\mathbf{w}, b$: 超平面参数。

第 3 步（续）：得到对偶函数

把上一页的三项代回 $L = \frac{1}{2} \mathbf{w}^\top \mathbf{w} + \sum_i \alpha_i - \sum_i \alpha_i y_i \mathbf{w}^\top \mathbf{x}_i - b \sum_i \alpha_i y_i$

$$\Gamma(\alpha) = \min_{\mathbf{w}, b} L = \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j + \sum_i \alpha_i - \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j - 0$$

$$\Gamma(\alpha) = \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j$$

要点

$\mathbf{w}$ 和 b 都消失了：对偶函数 $\Gamma(\alpha)$ 只含乘子 α_i 和数据。下一步对它求最大值。

记号 $\Gamma(\alpha)$ ：对偶函数； $\sum_{i,j}$ ： $\sum_{i=1}^m \sum_{j=1}^m$ 的简写； α_i ：乘子； $\mathbf{x}_i, y_i$ ：第 i 个样本及其标签； L ：拉格朗日函数。

第 4 步：SVM 的对偶问题

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j \\ \text{s.t.} \quad & \sum_{i=1}^m \alpha_i y_i = 0, \quad \alpha_i \geq 0, \quad i = 1, \dots, m \end{aligned}$$

- 变量： m 个 α_i （原问题是 $d+1$ 个）
- 约束 $\sum_i \alpha_i y_i = 0$ 来自 $\partial L / \partial b = 0$
- 约束 $\alpha_i \geq 0$ 来自乘子的对偶可行性
- 仍是凸二次规划

最重要的观察

训练样本在对偶问题中只以内积 $\mathbf{x}_i^\top \mathbf{x}_j$ 的形式出现，不需要单独用到 $\mathbf{x}_i$ 本身。

记号 α_i ：第 i 个样本的乘子， α ：所有乘子； m ：样本数， d ：样本维数； $\mathbf{x}_i, y_i$ ：第 i 个样本及其标签； $\mathbf{x}_i^\top \mathbf{x}_j$ ：样本 i 与 j 的内积； L ：拉格朗日函数。

第 5 步：KKT 条件告诉我们什么

强对偶成立时，最优解 $(\mathbf{w}^*, b^*, \alpha^*)$ 满足 KKT 条件

$$\alpha_i \geq 0, \quad y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1 \geq 0, \quad \alpha_i(y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1) = 0$$

由互补松弛，对每个样本只有两种可能：

$$\alpha_i = 0$$

样本对 $\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$ 没有贡献。

删掉它，解不变。

绝大多数样本属于这一类

$$\alpha_i > 0$$

必有 $y_i(\mathbf{w}^\top \mathbf{x}_i + b) = 1$ ：

样本落在间隔边界上，即支持向量。

这就严格证明了：**最终模型只由支持向量决定**。“支持向量机”因此得名。

记号 $(\mathbf{w}^*, b^*, \alpha^*)$ ：最优解； α_i ：第 i 个样本的乘子； $\mathbf{x}_i, y_i$ ：第 i 个样本及其标签； $\mathbf{w}, b$ ：超平面参数。

第 6 步：求出 w 、 b ，得到分类器

求 w ：直接由第 2 步

$$\mathbf{w} = \sum_{i=1}^m \alpha_i y_i \mathbf{x}_i = \sum_{i \in S} \alpha_i y_i \mathbf{x}_i$$

$S = \{i: \alpha_i > 0\}$ 为支持向量的下标集合

求 b ：任取支持向量 $s \in S$ ，有
 $y_s(\mathbf{w}^\top \mathbf{x}_s + b) = 1$ ，两边乘 y_s ($y_s^2 = 1$):

$$b = y_s - \sum_{i \in S} \alpha_i y_i \mathbf{x}_i^\top \mathbf{x}_s$$

实践中对所有支持向量取平均，更稳健

记号 α_i ：第 i 个样本的乘子； $\mathbf{x}_i, y_i$ ：第 i 个样本及其标签； S ：支持向量的下标集合， s ：其中任意一个； $\mathbf{x}$ ：待预测的新样本； $f(\mathbf{x})$ ：决策函数，预测标签为 $\text{sign } f(\mathbf{x})$ 。

分类器

$$f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b = \sum_{i \in S} \alpha_i y_i \mathbf{x}_i^\top \mathbf{x} + b$$

预测新样本 $\mathbf{x}$ 时，也只需要它与支持向量的内积。

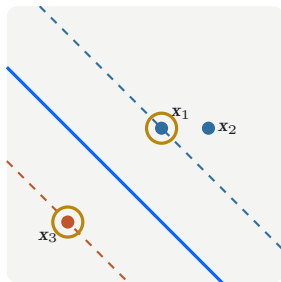
训练（对偶问题）和预测（ $f(\mathbf{x})$ ）都只用到内积——这是下一节核技巧的全部基础。

手算例子（一）：写出对偶问题

正类 $\mathbf{x}_1 = (3, 3)^\top$ 、 $\mathbf{x}_2 = (4, 3)^\top$ ；负类 $\mathbf{x}_3 = (1, 1)^\top$

内积矩阵 $K_{ij} = \mathbf{x}_i^\top \mathbf{x}_j$ ：

$$K = \begin{pmatrix} 18 & 21 & 6 \\ 21 & 25 & 7 \\ 6 & 7 & 2 \end{pmatrix}, \quad \mathbf{y} = (+1, +1, -1)$$



（图中为最终结果）

对偶问题（改写为最小化）：

$$\min_{\alpha} \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K_{ij} - \sum_i \alpha_i$$

$$\text{s.t. } \alpha_1 + \alpha_2 - \alpha_3 = 0, \quad \alpha_i \geq 0$$

用约束消去 $\alpha_3 = \alpha_1 + \alpha_2$ ，目标化为

$$s(\alpha_1, \alpha_2) = 4\alpha_1^2 + \frac{13}{2}\alpha_2^2 + 10\alpha_1\alpha_2 - 2\alpha_1 - 2\alpha_2$$

记号 $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ ：三个样本； K ：内积矩阵， K_{ij} ：第 i 行第 j 列； $\mathbf{y}$ ：标签向量， y_i ：其第 i 个分量； α_i ：乘子； s ：消去 α_3 后的目标函数。

手算例子（二）：求解并得到超平面

1. 无约束极值：令偏导为 0

$$8\alpha_1 + 10\alpha_2 - 2 = 0, \quad 10\alpha_1 + 13\alpha_2 - 2 = 0$$

得 $(\alpha_1, \alpha_2) = (1.5, -1)$ ，但 $\alpha_2 < 0$ ，不可行

2. 最优解在边界上，分别检查：

$$\alpha_1 = 0: \alpha_2 = \frac{2}{13}, \quad s = -\frac{2}{13} \approx -0.154$$

$$\alpha_2 = 0: \alpha_1 = \frac{1}{4}, \quad s = -\frac{1}{4} \text{ (更小)}$$

$$3. \text{ 解: } \alpha_1 = \frac{1}{4}, \alpha_2 = 0, \alpha_3 = \frac{1}{4}$$

$$\mathbf{w} = \frac{1}{4}(3, 3)^\top - \frac{1}{4}(1, 1)^\top = (\frac{1}{2}, \frac{1}{2})^\top$$

$$b = y_1 - \mathbf{w}^\top \mathbf{x}_1 = 1 - 3 = -2$$

$$\text{超平面: } \frac{1}{2}x_1 + \frac{1}{2}x_2 - 2 = 0$$

$$\text{检验: } y_i(\mathbf{w}^\top \mathbf{x}_i + b) \text{ 依次为 } 1, 1.5, 1$$

$\mathbf{x}_1, \mathbf{x}_3$ 在边界上 ($\alpha > 0$, 支持向量); $\mathbf{x}_2$ 不在 ($\alpha_2 = 0$)

$$\text{间隔} = 1/\|\mathbf{w}\| = \sqrt{2}$$

与 KKT 条件完全一致。真实问题中 m 很大，用下面的 SMO 算法求解。

记号 s : 消去 α_3 后的目标函数; α_i : 乘子; $\mathbf{w}, b$: 超平面参数; 超平面方程中的 x_1, x_2 是坐标分量, 粗体 $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ 是三个样本; m : 样本数。

求解对偶问题：SMO 算法

Sequential Minimal Optimization (Platt, 1998)

- 对偶问题有 m 个变量，通用 QP 求解器代价高
- 约束 $\sum_i \alpha_i y_i = 0$ 使得不能只改一个 α_i
- SMO: **每次只选两个变量** α_i, α_j ，固定其余，求解这个二变量子问题——有闭式解
- 反复选取、更新，直到满足 KKT 条件

重复：

1. 选一对违反 KKT 条件最严重的 α_i, α_j
2. 由 $\alpha_i y_i + \alpha_j y_j = \text{常数}$ ，化为单变量二次问题，闭式求解并裁剪到可行域
3. 更新 b

LIBSVM 与 scikit-learn 的 SVC 都基于 SMO 的变体。

记号 m : 样本数; α_i, α_j : 当前选中的两个乘子; y_i, y_j : 对应样本的标签; b : 偏置。

06

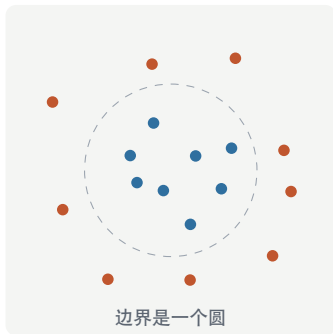
核技巧

The Kernel Trick

线性不可分 · 特征映射 · 映射的困难 · 核函数

问题：数据线性不可分

到目前为止，SVM 只能画一条直线（超平面）



- 正类在圆内，负类在圆外
- 任何一条直线都无法把两类分开：不存在满足全部约束 $y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1$ 的 $(\mathbf{w}, b)$
- 类似的还有“异或”问题

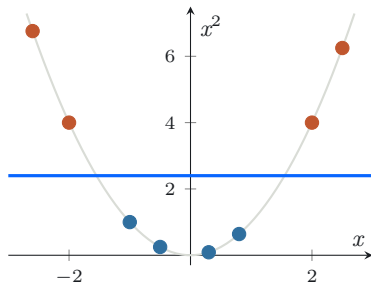
思路

不改变 SVM 本身，而是改变数据的表示：把样本映射到另一个空间，使它在那里线性可分。

记号 $\mathbf{x}_i, y_i$: 第 i 个样本及其标签; $\mathbf{w}, b$: 超平面参数。

特征映射：一维的例子

$\phi(x) = (x, x^2)$: 把一维的点映射到二维

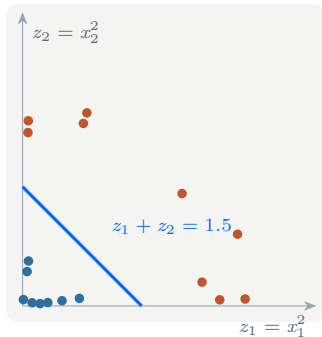
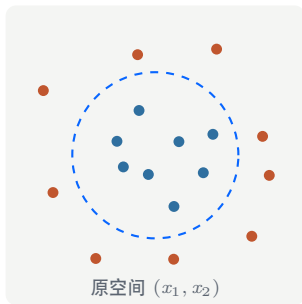


在 (x, x^2) 平面中，直线 $x^2 = 2.4$ 就能分开两类；回到原空间，对应的是两个阈值： $|x| < \sqrt{2.4}$ 。

记号 x : 一维样本（一个数）； ϕ : 特征映射，把 x 变成二维向量 (x, x^2) 。

特征映射：二维的例子

$\phi(\mathbf{x}) = (x_1^2, x_2^2)$: 圆 $x_1^2 + x_2^2 = r^2$ 变成直线 $z_1 + z_2 = r^2$



原空间中的圆形边界，在特征空间 $\mathbf{z} = \phi(\mathbf{x})$ 中是一条直线：线性不可分 $\Rightarrow$ 映射后线性可分。

记号 $\mathbf{x} = (x_1, x_2)$: 原空间的样本, x_1, x_2 为坐标分量; ϕ : 特征映射; $\mathbf{z} = \phi(\mathbf{x}) = (z_1, z_2)$: 特征空间中的点; r : 圆的半径。

在特征空间中做 SVM

把所有的 $\mathbf{x}$ 换成 $\phi(\mathbf{x}) \in \mathbb{R}^{\tilde{d}}$ ，其余推导一字不改

原问题

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

$\mathbf{w} \in \mathbb{R}^{\tilde{d}}$: 维数与特征空间相同

对偶问题

$$\begin{aligned} \max_{\alpha} \quad & \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j) \\ \text{s.t.} \quad & \sum_i \alpha_i y_i = 0, \quad \alpha_i \geq 0 \end{aligned}$$

分类器: $f(\mathbf{x}) = \sum_i \alpha_i y_i \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}) + b$

$\tilde{d}$ 是特征空间的维数，通常远大于原始维数 d 。

记号 ϕ : 特征映射; d : 原始维数, $\tilde{d}$: 特征空间维数; $\mathbf{w} \in \mathbb{R}^{\tilde{d}}$: 特征空间中的法向量, b : 偏置; $\mathbf{x}_i, y_i$: 样本及其标签; α_i : 乘子; $f(\mathbf{x})$: 决策函数。

困难：特征空间的维数爆炸

映射到“足够丰富”的特征空间才能线性可分，但显式写出 $\phi(\mathbf{x})$ 代价极高

把 d 维样本映射为所有次数 $\leq p$ 的单项式（多项式特征），维数为 $\binom{d+p}{p}$ ：

原始维数 d	次数 p	特征空间维数 $\tilde{d}$
2	2	6
100	2	5 151
100	3	176 851
100	5	96 560 646
1000	3	167 668 501

- 一张 10×10 的小图片 $d = 100$ ，5 次多项式特征就有近一亿维
- 每个样本都要存一亿维向量，每个内积都要算一亿次乘法
- 高斯核对应的特征空间是无穷维，根本写不出来

出路

对偶问题只需要内积 $\phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)$ 。能不能不算 ϕ ，直接算内积？

记号 d ：原始维数； p ：多项式的最高次数； $\tilde{d}$ ：特征空间维数； $\binom{d+p}{p}$ ：组合数； ϕ ：特征映射。

核函数：不算 ϕ ，直接算内积

若存在函数 κ 使 $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)$ ，称 κ 为核函数

例：二次多项式核， $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^2$

$$\begin{aligned}(\mathbf{x}^\top \mathbf{x}')^2 &= (x_1 x'_1 + x_2 x'_2)^2 \\&= x_1^2 x_1'^2 + 2x_1 x_2 x'_1 x'_2 + x_2^2 x_2'^2 \\&= \phi(\mathbf{x})^\top \phi(\mathbf{x}')\end{aligned}$$

其中 $\phi(\mathbf{x}) = (x_1^2, \sqrt{2}x_1 x_2, x_2^2)^\top$

数值验证： $\mathbf{x} = (1, 2)^\top$ ， $\mathbf{x}' = (3, 1)^\top$

直接算： $\mathbf{x}^\top \mathbf{x}' = 5$ ， $\kappa = 5^2 = 25$

经过 ϕ ： $(1, 2\sqrt{2}, 4) \cdot (9, 3\sqrt{2}, 1) = 9 + 12 + 4 = 25$

计算量对比

经过 ϕ ：先构造 $\tilde{d}$ 维向量，再做 $\tilde{d}$ 维内积

用核函数：在原空间做一次 d 维内积，再平方

$d = 100$ ， $p = 5$ ：一亿次乘法 vs 约一百次乘法

核函数隐式地定义了特征空间：我们从不需要知道 ϕ 长什么样。

记号 $\mathbf{x}, \mathbf{x}'$ ：两个样本， x_1, x_2 与 x'_1, x'_2 是它们的分量； ϕ ：特征映射； κ ：核函数； d ：原始维数， $\tilde{d}$ ：特征空间维数， p ：多项式次数。

核化的 SVM

把对偶问题和分类器中的内积全部换成核函数

$$\max_{\alpha} \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j \kappa(\mathbf{x}_i, \mathbf{x}_j) \quad \text{s.t.} \quad \sum_i \alpha_i y_i = 0, \alpha_i \geq 0$$

$$f(\mathbf{x}) = \sum_{i \in S} \alpha_i y_i \kappa(\mathbf{x}_i, \mathbf{x}) + b$$

- 训练：只需要 $m \times m$ 的核矩阵 $K_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$
- 预测：只需要新样本与支持向量的核函数值
- 算法（SMO）完全不变

要点

原空间中：非线性决策边界。特征空间中：仍是最大间隔超平面。

记号 κ : 核函数; α_i : 乘子, m : 样本数; $\mathbf{x}_i, y_i$: 第 i 个样本及其标签; S : 支持向量的下标集合; K : 核矩阵, $K_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$; $\mathbf{x}$: 待预测样本。

什么样的函数可以作为核函数？

不是任何二元函数都对应某个特征空间中的内积

Mercer 定理（简化版）

对称函数 $\kappa(\cdot, \cdot)$ 是核函数，当且仅当对任意样本 $\mathbf{x}_1, \dots, \mathbf{x}_m$ ，核矩阵

$$K = \begin{pmatrix} \kappa(\mathbf{x}_1, \mathbf{x}_1) & \cdots & \kappa(\mathbf{x}_1, \mathbf{x}_m) \\ \vdots & \ddots & \vdots \\ \kappa(\mathbf{x}_m, \mathbf{x}_1) & \cdots & \kappa(\mathbf{x}_m, \mathbf{x}_m) \end{pmatrix}$$

总是半正定的。

- 半正定保证对偶问题仍是凸问题，有全局最优解
- 核函数的组合仍是核函数： $\kappa_1 + \kappa_2$ ， $c\kappa_1$ ($c > 0$)， $\kappa_1\kappa_2$ 等

记号 κ ：核函数； $\mathbf{x}_1, \dots, \mathbf{x}_m$ ：任意 m 个样本； K ：核矩阵；半正定：对任意向量 $\mathbf{u}$ 有 $\mathbf{u}^\top K \mathbf{u} \geq 0$ ； κ_1, κ_2 ：任意两个核函数， c ：正常数。

常用核函数

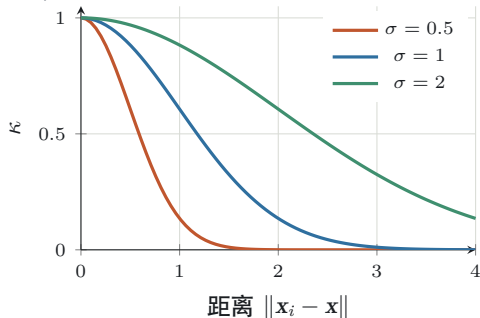
名称	表达式	参数
线性核	$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^\top \mathbf{x}_j$	即不做映射
多项式核	$\kappa(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^\top \mathbf{x}_j)^p$	$p \geq 1$ 为多项式次数
高斯核 (RBF)	$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\ \mathbf{x}_i - \mathbf{x}_j\ ^2}{2\sigma^2}\right)$	$\sigma > 0$ 为带宽
拉普拉斯核	$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\ \mathbf{x}_i - \mathbf{x}_j\ }{\sigma}\right)$	$\sigma > 0$
Sigmoid 核	$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\beta \mathbf{x}_i^\top \mathbf{x}_j + \theta)$	$\beta > 0, \theta < 0$

实践建议：先试线性核（特征多、样本多时通常够用），再试高斯核并用交叉验证选 σ 。

记号 $\mathbf{x}_i, \mathbf{x}_j$: 两个样本; $\|\mathbf{x}_i - \mathbf{x}_j\|$: 它们的欧氏距离; p, σ, β, θ : 核函数的参数（见右列）; $\exp$: 指数函数, $\tanh$: 双曲正切。

高斯核：相似度的视角

$\kappa(\mathbf{x}_i, \mathbf{x})$ 衡量 $\mathbf{x}$ 与支持向量 $\mathbf{x}_i$ 的相似程度



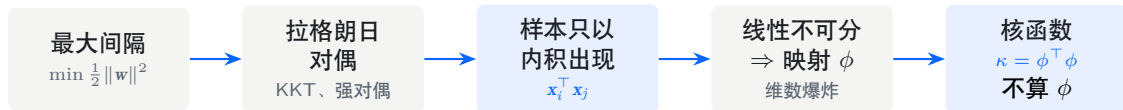
$$f(\mathbf{x}) = \sum_{i \in S} \alpha_i y_i \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}\|^2}{2\sigma^2}\right) + b$$

- 每个支持向量在自己周围放一个“钟形”影响，正类加、负类减
- σ 小：影响范围窄，边界弯曲，易过拟合
- σ 大：影响范围宽，边界平滑，接近线性
- 对应的特征空间是无穷维，只能通过核函数使用

记号 $\mathbf{x}$: 待预测样本, $\mathbf{x}_i$: 支持向量; σ : 高斯核带宽; S : 支持向量的下标集合; α_i : 乘子, y_i : 标签, b : 偏置。

核技巧的来龙去脉

每一步都依赖上一步



为什么 SVM 能用核技巧

拉格朗日对偶之后，训练和预测都只依赖样本间的内积。

为什么需要核技巧

映射到高维才线性可分，但显式计算 ϕ 太贵甚至不可能。

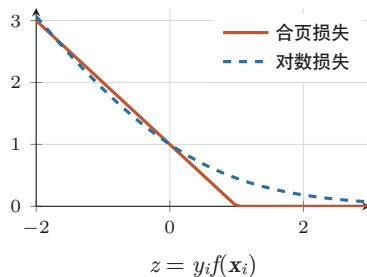
记号 w : 法向量; $x_i^T x_j$: 样本内积; ϕ : 特征映射; κ : 核函数。

实践中：软间隔（预告）

真实数据常有噪声，不必要求每个样本都满足 $y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1$

$$\min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^m \ell_{\text{hinge}}(y_i(\mathbf{w}^\top \mathbf{x}_i + b)), \quad \ell_{\text{hinge}}(z) = \max(0, 1 - z)$$

- 允许少量样本进入间隔甚至分错，但要付出代价
- C 大：不容忍错误，接近硬间隔，易过拟合
- C 小：间隔宽，容忍更多错误
- 对偶问题只多一个约束： $0 \leq \alpha_i \leq C$ ；核技巧照样可用



与逻辑回归只差在损失函数：合页损失 vs 对数损失。详见西瓜书 6.4 节。

记号 $C > 0$ ：惩罚系数； ℓ_{hinge} ：合页损失，自变量 $z = y_i f(x_i)$ 是样本 i 的函数间隔（一个数）； $f(x) = \mathbf{w}^\top \mathbf{x} + b$ ； α_i ：乘子。

07

小结

Summary

两种方法的对比 · 课后作业

两种经典方法的对比

	决策树	支持向量机
学到什么	一组 if-then 规则	一个最大间隔超平面
训练方式	贪心：逐结点选属性与划分	凸优化：全局最优
非线性	天然：逐层划分	核函数映射到高维
可解释性	强	弱
特征缩放	不敏感	敏感，需要标准化
防止过拟合	剪枝	间隔 + 软间隔参数 C

下次课：神经网络与深度学习。核方法用人为选定的特征映射 ϕ ，神经网络则从数据中学习特征。

记号 C ：软间隔的惩罚系数； ϕ ：特征映射。

课后作业

1. 决策树手算

在西瓜数据集 3.0 的“纹理 = 稍糊”分支（5 个样本）上，分别计算 X_1, X_2, X_3, X_5, X_6 的信息增益，以及密度 X_7 的最优阈值与信息增益，确定该结点的划分。

2. 支持向量机公式推导

从几何间隔出发，说明为何可以令 $\hat{\gamma} = 1$ ；写出拉格朗日函数，逐步推导对偶问题与 KKT 条件，并说明为什么解只依赖支持向量。

参考：周志华《机器学习》第 4、6、8 章；李航《统计学习方法》第 5、7 章。

记号 $X_1, X_2, X_3, X_5, X_6, X_7$ ：色泽、根蒂、敲声、脐部、触感、密度； $\hat{\gamma}$ ：函数间隔。