



中國農業大學  
China Agricultural University

# 人工智能——经典监督学习算法

---

胡标





# 目录

## Contents

1. K最近邻

2. 朴素贝叶斯

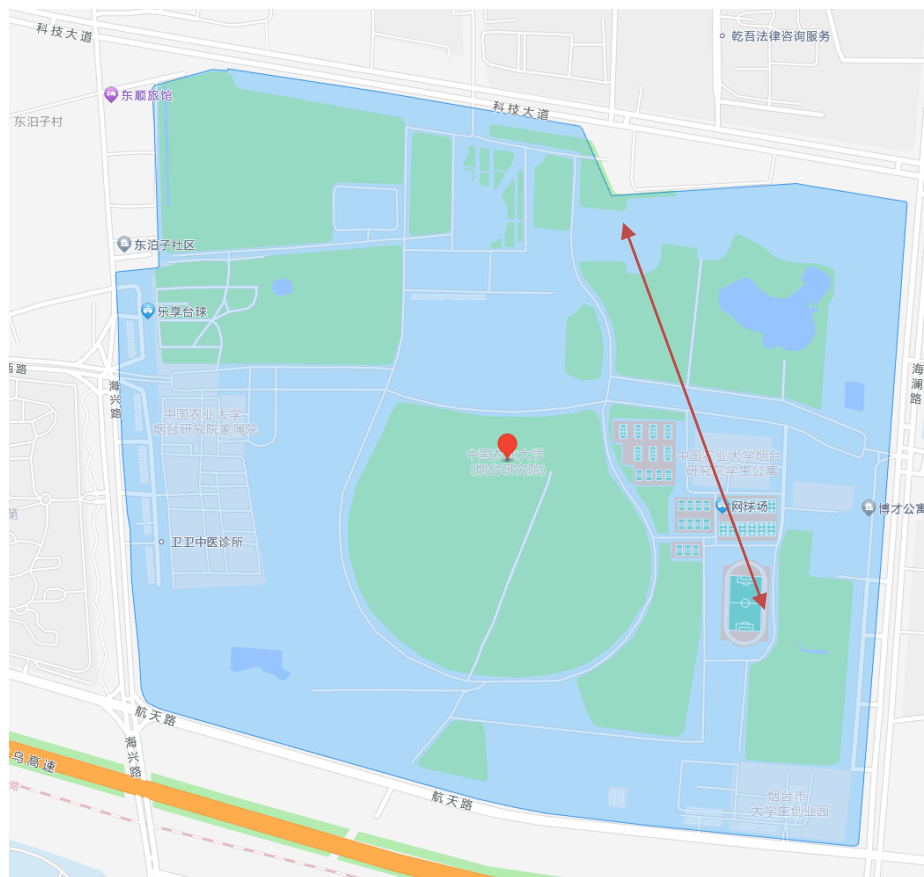
3. 决策树

4. 支持向量机

## 欧氏距离(Euclidean distance)

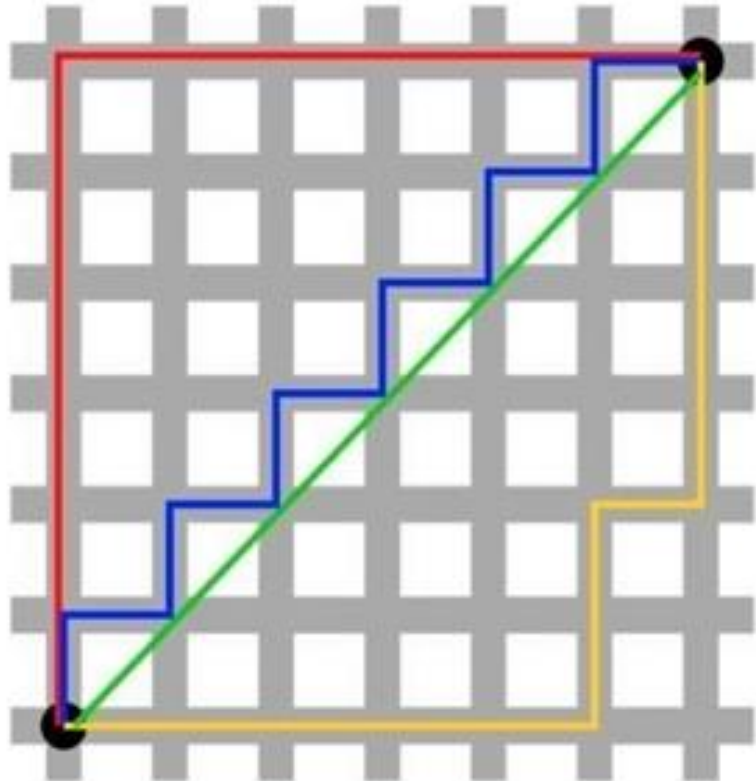
$$d(x, y) = \sqrt{\sum_i (x_i - y_i)^2}$$

欧几里得度量 (Euclidean Metric) (也称欧氏距离) 是一个通常采用的距离定义, 指在  $m$  维空间中两个点之间的真实距离, 或者向量的自然长度 (即该点到原点的距离)。在二维和三维空间中的欧氏距离就是两点之间的实际距离。



## 曼哈顿距离(Manhattan distance)


$$d(x, y) = \sum_i |x_i - y_i|$$



想象你在城市道路里，要从一个十字路口开车到另外一个十字路口，驾驶距离是两点间的直线距离吗？显然不是，除非你能穿越大楼。实际驾驶距离就是这个“曼哈顿距离”。而这也是曼哈顿距离名称的来源，曼哈顿距离也称为城市街区距离(CityBlock distance)。

## 切比雪夫距离(Chebyshev distance)

$$d(x, y) = \max_i |x_i - y_i|$$

	a	b	c	d	e	f	g	h	
8	5	4	3	2	2	2	2	2	8
7	5	4	3	2	1	1	1	2	7
6	5	4	3	2	1		1	2	6
5	5	4	3	2	1	1	1	2	5
4	5	4	3	2	2	2	2	2	4
3	5	4	3	3	3	3	3	3	3
2	5	4	4	4	4	4	4	4	2
1	5	5	5	5	5	5	5	5	1
	a	b	c	d	e	f	g	h	

二个点之间的距离定义是其各坐标数值差绝对值的最大值。

国际象棋棋盘上二个位置间的切比雪夫距离是指王要从一个位子移至另一个位子需要走的步数。

由于王可以往斜前或斜后方向移动一格，因此可以较有效率的到达目的的格子。上图是棋盘上所有位置距f6位置的切比雪夫距离。

## 闵可夫斯基距离(Minkows kidistance)

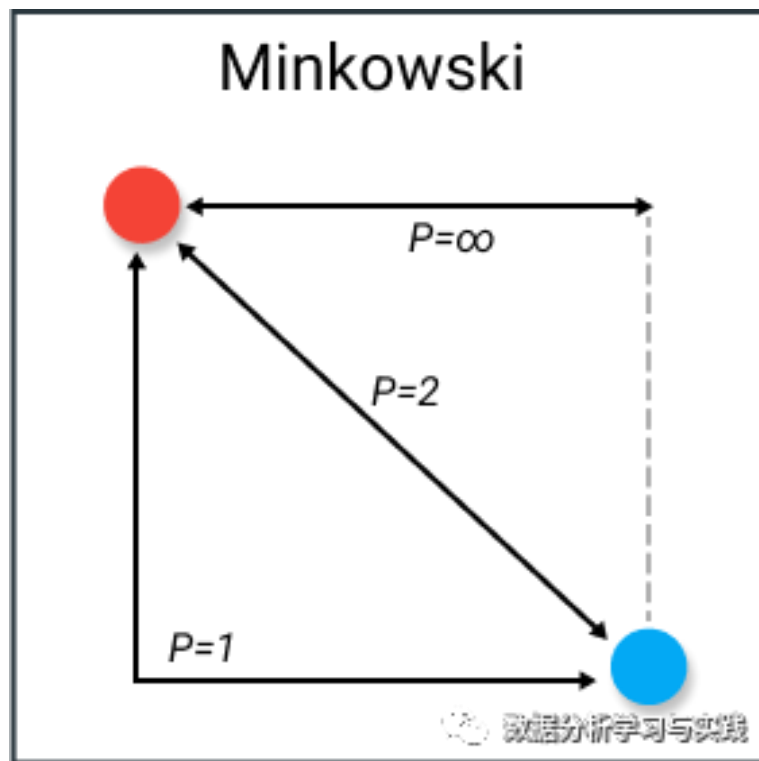
$$d(x, y) = \left( \sum_i (x_i - y_i)^p \right)^{\frac{1}{p}}$$

$p$ 取1或2时的闵氏距离是最为常用的

$p=2$ 即为欧氏距离,

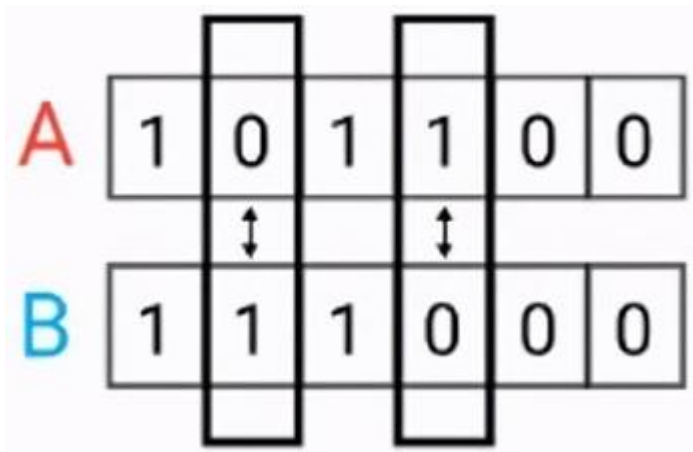
$p=1$ 时则为曼哈顿距离。

当 $p$ 取无穷时的极限情况下, 可以得到切比雪夫距离



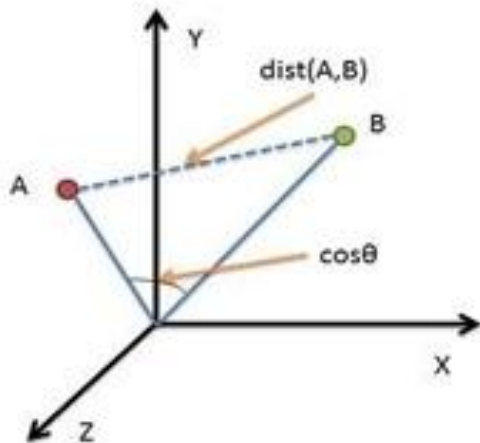
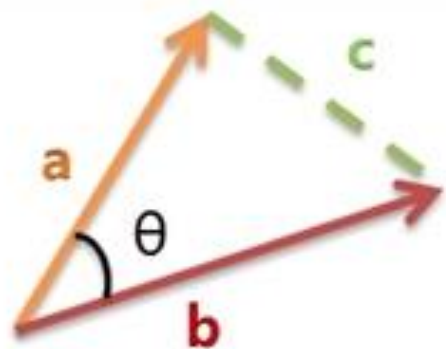
## 汉明距离(Hamming distance)

$$d(x, y) = \frac{1}{N} \sum_i 1_{x_i \neq y_i}$$



汉明距离是使用在数据传输差错控制编码里面的，汉明距离是一个概念，它表示两个（相同长度）字对应位不同的数量，我们以表示两个字之间的汉明距离。对两个字符串进行异或运算，并统计结果为1的个数，那么这个数就是汉明距离。

## 余弦相似度



假定 $\vec{A}$ 和 $\vec{B}$ 是两个n维向量， $\vec{A}$ 是 $[A_1, A_2, \dots, A_n]$ ， $\vec{B}$ 是 $[B_1, B_2, \dots, B_n]$ ，则 $\vec{A}$ 和 $\vec{B}$ 的夹角的余弦等于：

$$\cos(\theta) = \frac{\vec{A} \cdot \vec{B}}{\|\vec{A}\| \|\vec{B}\|}$$

两个向量有相同的指向时，余弦相似度的值为1；  
两个向量夹角为 $90^\circ$ 时，余弦相似度的值为0；  
两个向量指向完全相反的方向时，余弦相似度的值为-1。

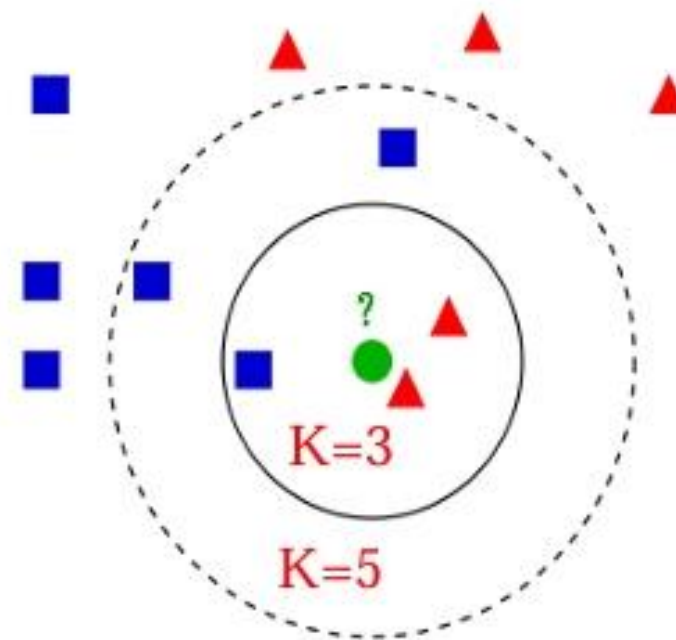


- $k$ 近邻法 (k-Nearest Neighbor, kNN) 是一种比较成熟也是最简单的机器学习算法，可以用于基本的分类与回归方法。
- 算法的主要思路：
  - 如果一个样本在特征空间中与 $k$ 个实例最为相似(即特征空间中最邻近)，那么这 $k$ 个实例中大多数属于哪个类别，则该样本也属于这个类别。
  - 对于分类问题：对新的样本，根据其 $k$ 个最近邻的训练样本的类别，通过多数表决等方式进行预测。
  - 对于回归问题：对新的样本，根据其 $k$ 个最近邻的训练样本标签值的均值作为预测值。

- $k$ 近邻法 (k-Nearest Neighbor, kNN) 是一种比较成熟也是最简单的机器学习算法，可以用于基本的分类与回归方法。
- 近邻法的三要素：
  - $k$ 值选择。
  - 距离度量。
  - 决策规则。

## □ 算法流程如下：

1. 计算测试对象到训练集中每个对象的距离
2. 按照距离的远近排序
3. 选取与当前测试对象最近的 $k$ 的训练对象，作为该测试对象的邻居
4. 统计这 $k$ 个邻居的类别频次
5.  $k$ 个邻居里频次最高的类别，即为测试对象的类别

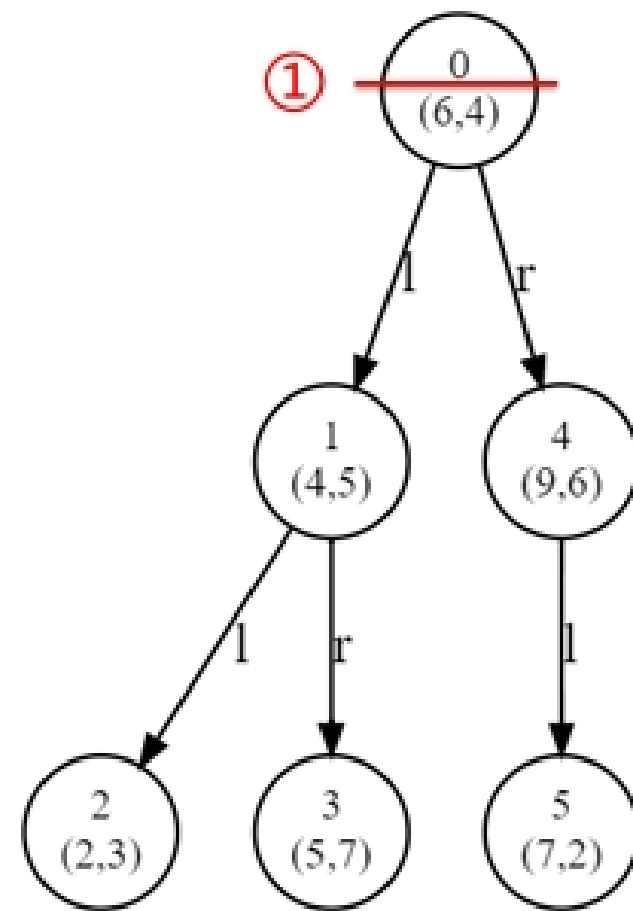


□ KD树(K-Dimension Tree), , 也可称之为K维树, 可以用更高的效率来对空间进行划分, 并且其结构非常适合寻找最近邻居和碰撞检测。

□ 假设有6 个二维数据点, 构建KD树的过程:

$$D = \{(2,3), (5,7), (9,6), (4,5), (6,4), (7,2)\}$$

① 从 $x$ 轴开始划分, 根据 $x$ 轴的取值2,5,9,4,6,7, 得到中位数为6, 因此切分线为:  $x = 6$ 。

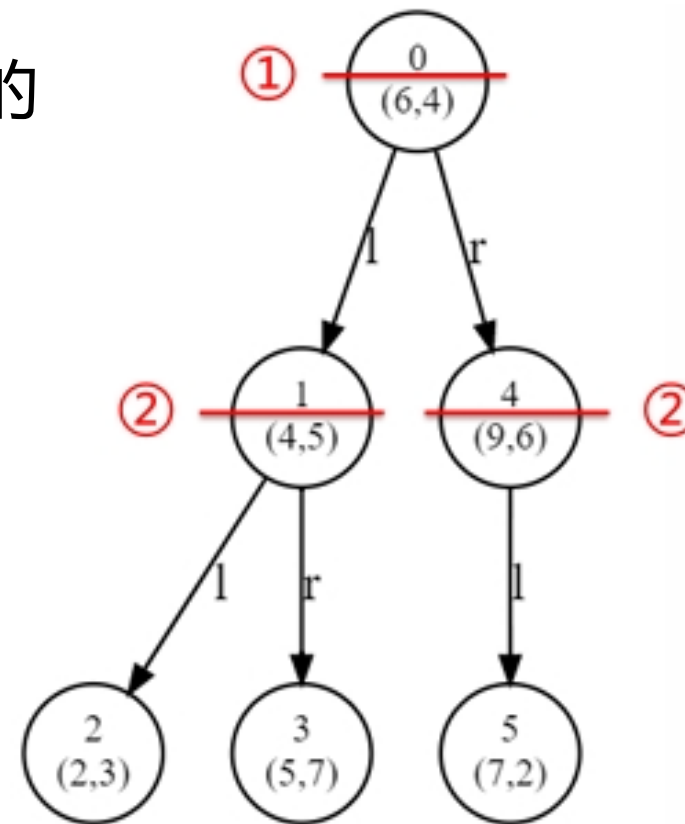


$$D = \{(2,3), (5,7), (9,6), (4,5), (6,4), (7,2)\}$$

② 可以根据 $x$ 轴和 $y$ 轴上数据的方差，选择方差最大的那个轴作为第一轮划分轴。

左子空间（记做 $D_1$ ）包含点 $(2,3), (4,5), (5,7)$ ，切分轴轮转，从 $y$ 轴开始划分，切分线为： $y=5$ 。

右子空间（记做 $D_2$ ）包含点 $(9,6), (7,2)$ ，切分轴轮转，从 $y$ 轴开始划分，切分线为： $y=6$ 。



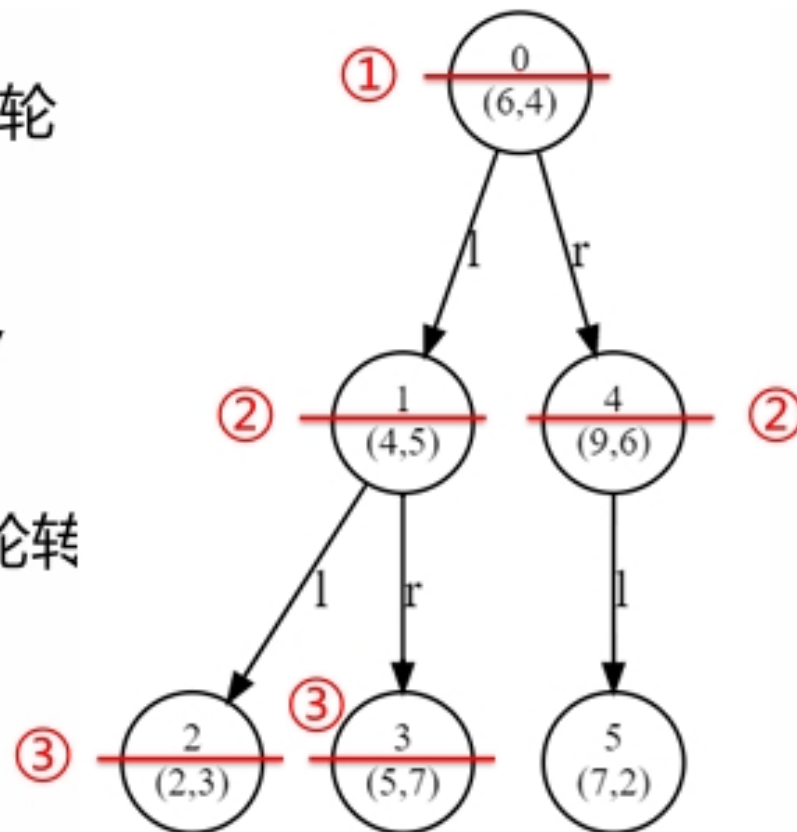
$$D = \{(2,3), (5,7), (9,6), (4,5), (6,4), (7,2)\}$$

③  $D_1$  的左子空间 (记做  $D_3$ ) 包含点  $(2,3)$ , 切分轴轮转, 从  $x$  轴开始划分, 切分线为:  $x = 2$ 。

其左子空间记做  $D_7$ , 右子空间记做  $D_8$ 。由于  $D_7, D_8$  都不包含任何点, 因此对它们不再继续拆分。

$D_1$  的右子空间 (记做  $D_4$ ) 包含点  $(5,7)$ , 切分轴轮转, 从  $x$  轴开始划分, 切分线为:  $x = 5$ 。

其左子空间记做  $D_9$ , 右子空间记做  $D_{10}$ 。由于  $D_9, D_{10}$  都不包含任何点, 因此对它们不再继续拆分。



$$D = \{(2,3), (5,7), (9,6), (4,5), (6,4), (7,2)\}$$

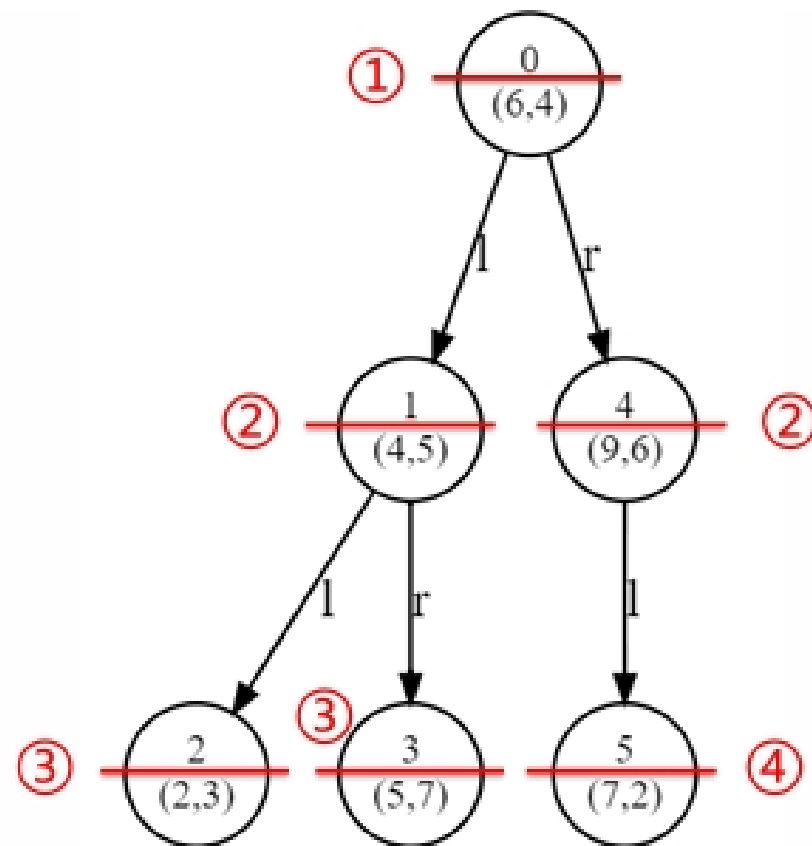
④  $D_2$  的左子空间 (记做  $D_5$ ) 包含点  $(7,2)$ , 切分轴轮转, 从  $x$  轴开始划分, 切分线为:  $x = 7$

。

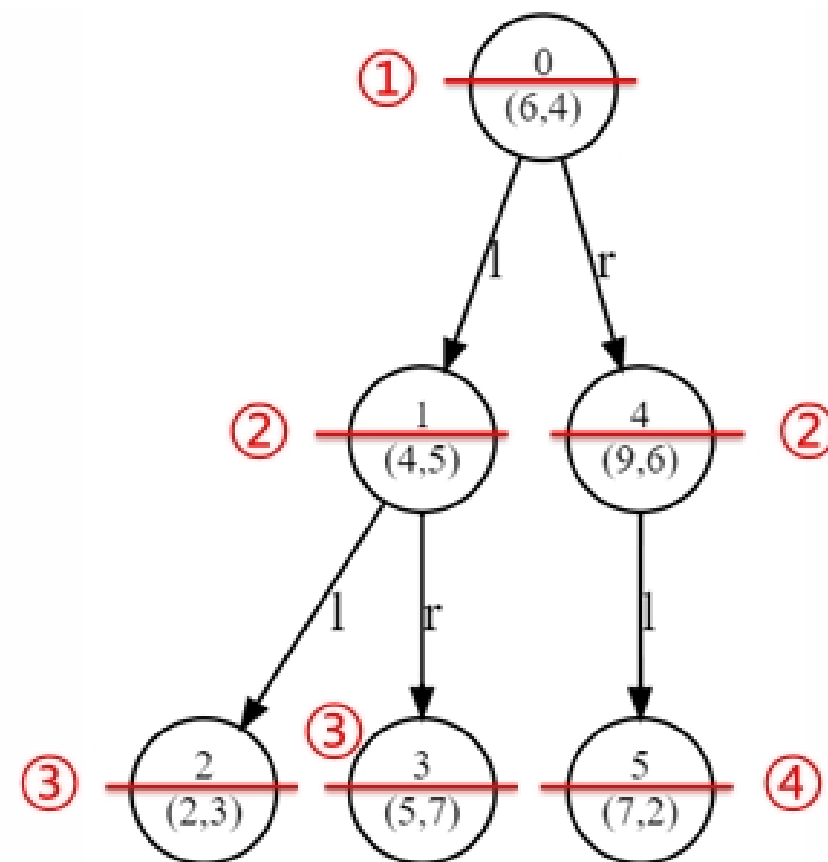
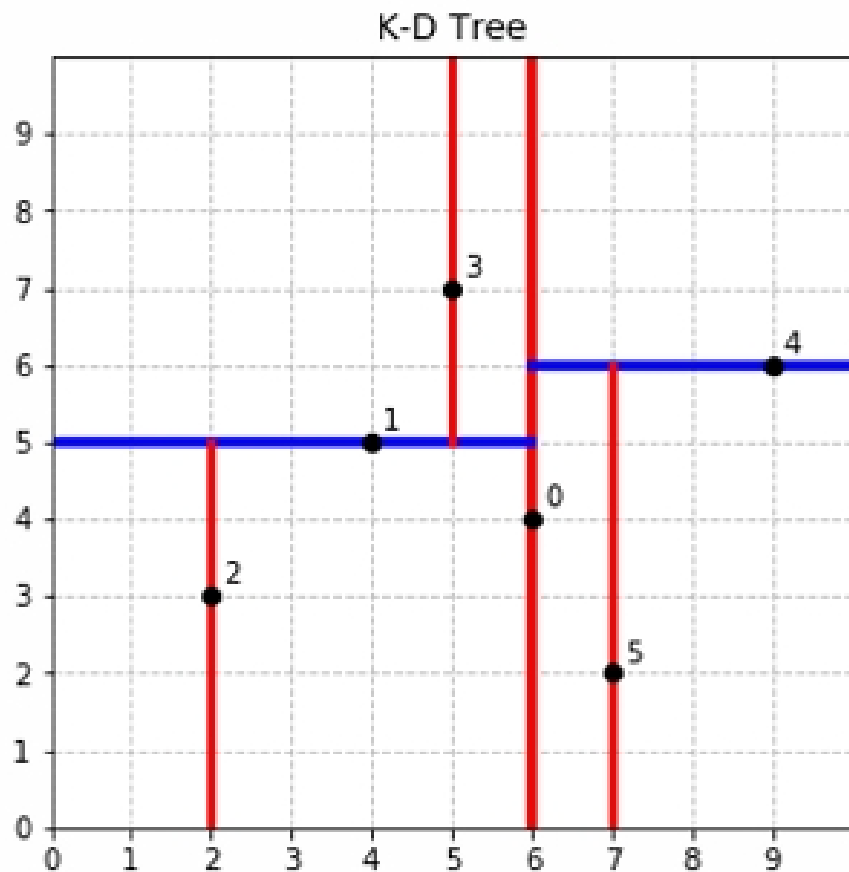
其左子空间记做  $D_{11}$ , 右子空间记做  $D_{12}$ 。

由于  $D_{11}, D_{12}$  都不包含任何点, 因此对它们不再继续拆分。

$D_2$  的右子空间 (记做  $D_6$ ) 不包含任何点, 停止继续拆分。

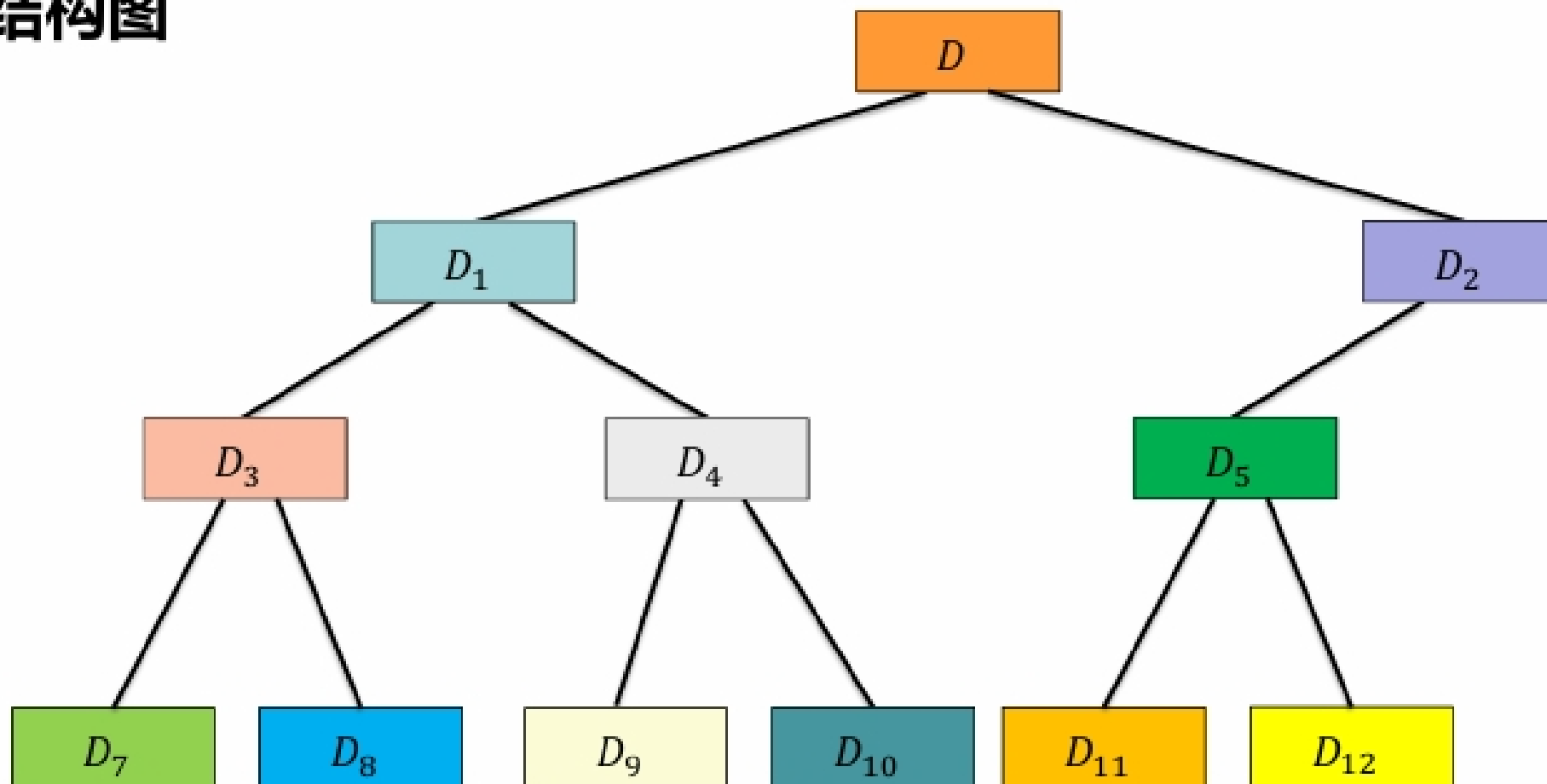


$$D = \{(2,3), (5,7), (9,6), (4,5), (6,4), (7,2)\}$$



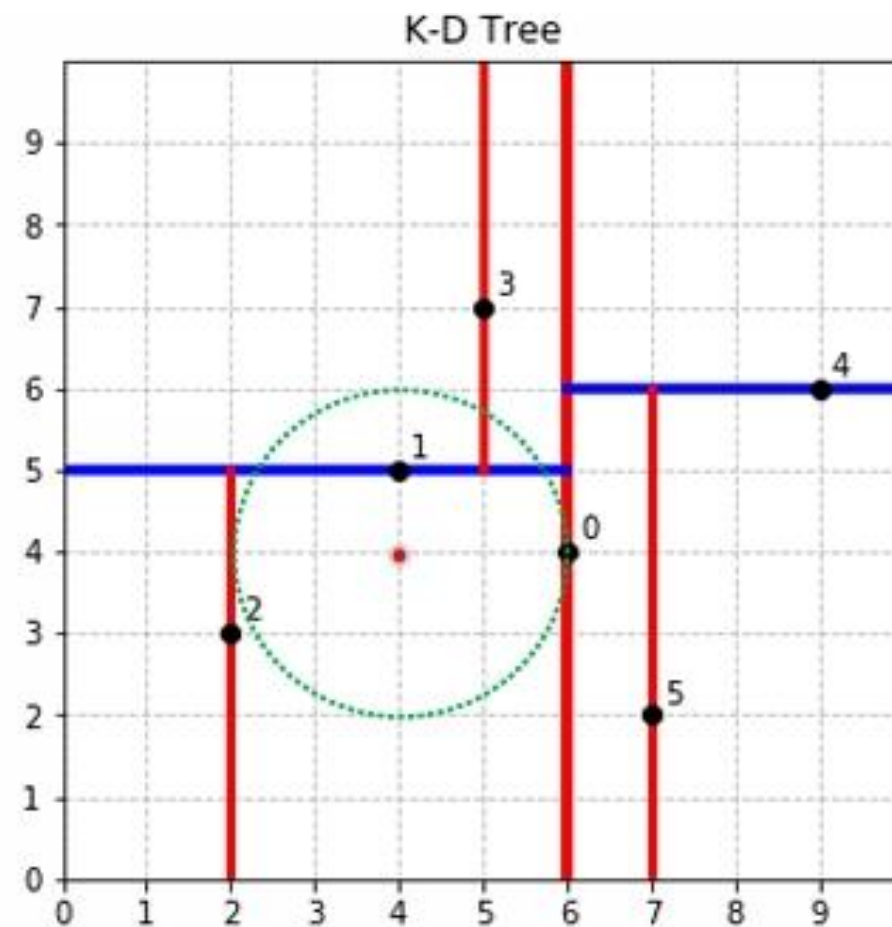


## 样本空间结构图



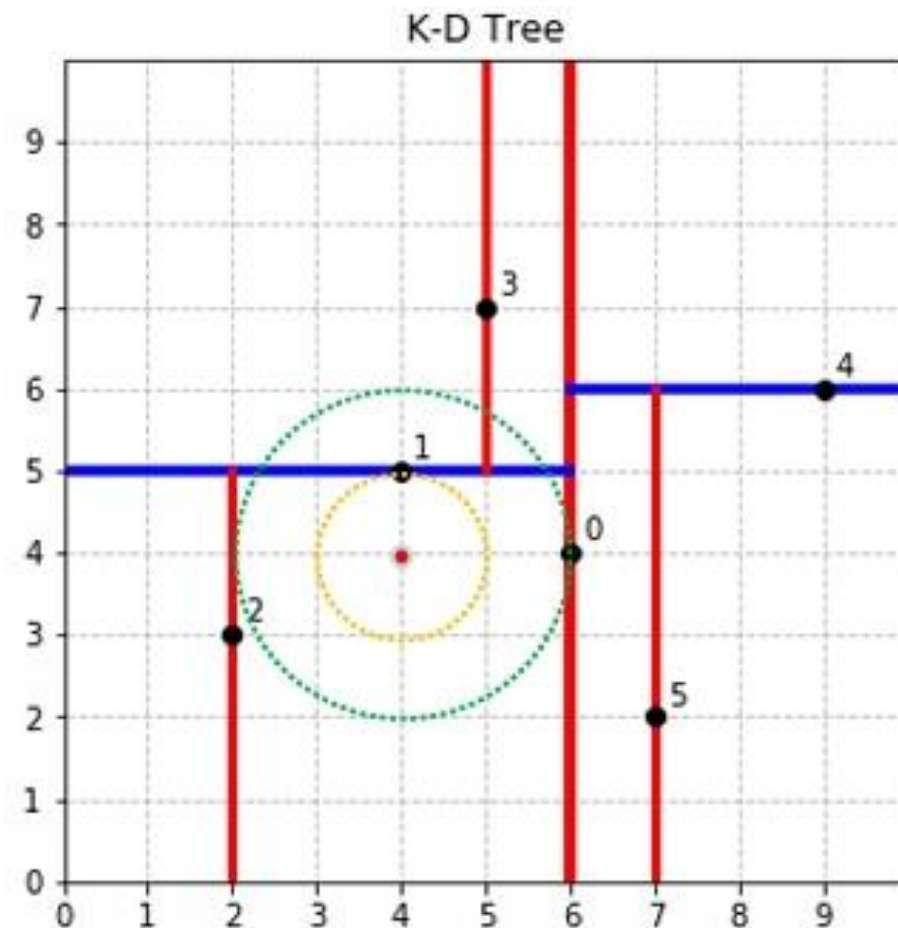
1. 首先要找到该目标点的叶子节点，然后以目标点为圆心，目标点到叶子节点的距离为半径，建立一个超球体，我们要找寻的最近邻点一定是在该球体内部。

搜索 (4,4) 的最近邻时。首先从根节点 (6,4) 出发，将当前最近邻设为 (6,4)，对该KD树作深度优先遍历。以 (4,4) 为圆心，其到 (6,4) 的距离为半径画圆（多维空间为超球面），可以看出 (7,2) 右侧的区域与该圆不相交，所以 (7,2) 的右子树全部忽略。



2. 返回叶子结点的父节点，检查另一个子节点包含的超矩形体是否和超球体相交，如果相交就到这个子节点寻找是否有更加近的近邻，有的话就更新最近邻。

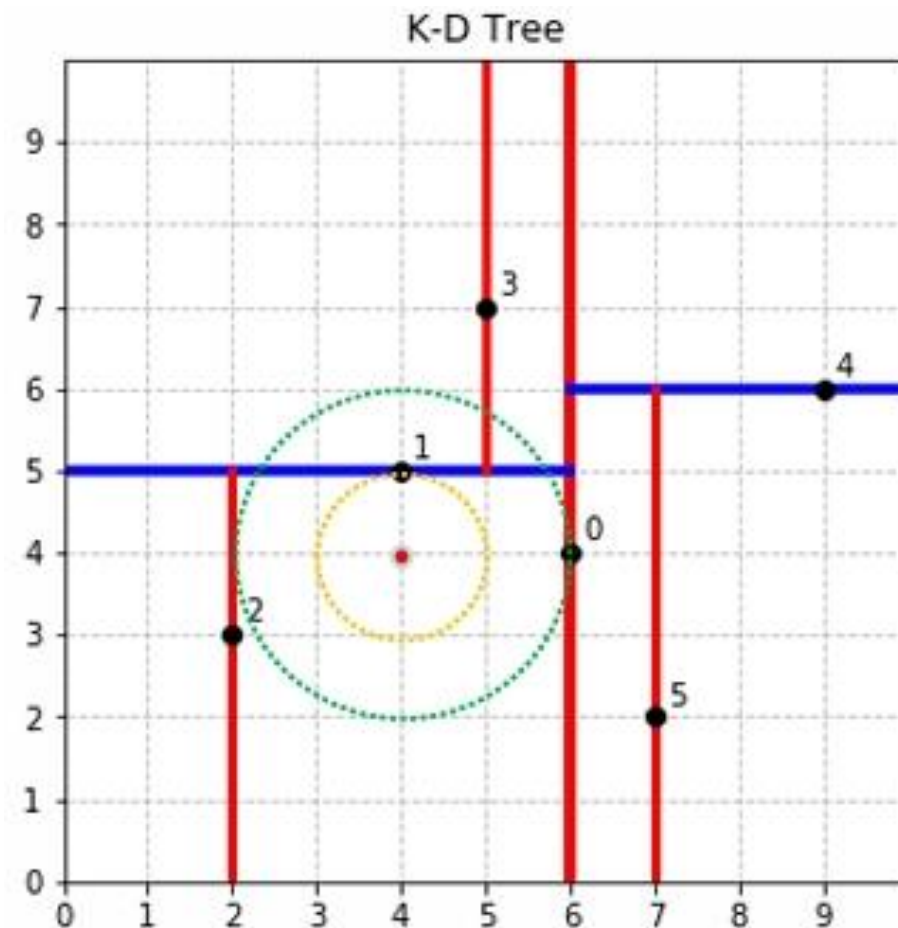
接着走到 (6,4) 左子树根节点 (4,5)，与原最近邻对比距离后，更新当前最近邻为 (4,5)。以 (4,4) 为圆心，其到 (4,5) 的距离为半径画圆，发现 (6,4) 右侧的区域与该圆不相交，忽略该侧所有节点，这样 (6,4) 的整个右子树被标记为已忽略。



3. 如果不相交直接返回父节点，在另一个子树继续搜索最近邻。
4. 当回溯到根节点时，算法结束，此时保存的最近邻节点就是最终的最近邻。

遍历完 (4,5) 的左右叶子节点，发现与当前最优距离相等，不更新最近邻。

所以 (4,4) 的最近邻为 (4,5)。





# 目录

## Contents

1. K最近邻

2. 朴素贝叶斯

3. 决策树

4. 支持向量机

- **贝叶斯分类：**贝叶斯分类是一类分类算法的总称，这类算法均以贝叶斯定理为基础，故统称为贝叶斯分类。
- **先验概率：**根据以往经验和分析得到的概率。我们用 $P(Y)$ 来代表在没有训练数据前假设 $Y$ 拥有的初始概率。
- **后验概率：**根据已经发生的事件来分析得到的概率。以 $P(Y|X)$ 代表假设 $X$ 成立的情下观察到 $Y$ 数据的概率，因为它反映了在看到训练数据 $X$ 后 $Y$ 成立的置信度。

- **联合概率：**联合概率是指在多元的概率分布中多个随机变量分别满足各自条件的概率。 $X$ 与 $Y$ 的联合概率表示为 $P(X,Y)$ 、 $P(XY)$  或 $P(X \cap Y)$ 。
- 假设 $X$ 和 $Y$ 都服从正态分布，那么 $P(X < 5, Y < 0)$ 就是一个联合概率，表示 $X < 5, Y < 0$ 两个条件同时成立的概率。表示两个事件共同发生的概率。

- 贝叶斯公式

$$P(Y|X) = \frac{P(X, Y)}{P(X)} = \frac{P(X|Y)P(Y)}{P(X)}$$

Diagram illustrating the components of Bayes' formula:

- 后验概率 (Posterior Probability) points to  $P(Y|X)$
- 似然度 (Likelihood) points to  $P(X|Y)$
- 先验概率 (Prior Probability) points to  $P(Y)$
- 边际似然度 (Marginal Likelihood) points to  $P(X)$

- 朴素贝叶斯法是典型的生成学习方法。生成方法由训练数据学习联合概率分布  $P(X, Y)$ ，然后求得后验概率分布  $P(Y|X)$
- 具体来说，利用训练数据学习  $P(X|Y)$  和  $P(Y)$  的估计，得到联合概率分布：

$$P(X, Y) = P(X|Y) P(Y)$$



- 判别模型和生成模型

## 判别模型 (Discriminative Model)

由数据直接学习决策函数 $Y=f(X)$ 或者条件概率分布 $P(Y|X)$ 作为预测的模型，即判别模型。基本思想是有限样本条件下建立判别函数，不考虑样本的产生模型，直接研究预测模型。

即：直接估计 $P(Y|X)$

线性回归、逻辑回归、感知机、决策树、支持向量机.....

## 生成模型 (Generative Model)

由训练数据学习联合概率分布 $P(X,Y)$ ，然后求得后验概率分布 $P(Y|X)$ 。具体来说，利用训练数据学习 $P(X|Y)$ 和 $P(Y)$ 的估计，得到联合概率分布： $P(X,Y) = P(Y)P(X|Y)$ ，再利用它进行分类。

即：估计 $P(X|Y)$ 然后推导 $P(Y|X)$

朴素贝叶斯、HMM、深度信念网络(DBN).....

## 1. 朴素贝叶斯法是典型的生成学习方法

- 生成方法由训练数据学习联合概率分布 $P(X,Y)$ ，然后求得后验概率分布 $P(Y|X)$ 。具体来说，利用训练数据学习 $P(X|Y)$ 和 $P(Y)$ 的估计，得到联合概率分布：

$$P(X,Y) = P(Y)P(X|Y)$$

- 概率估计方法可以是极大似然估计或贝叶斯估计。

## 2. 朴素贝叶斯法的基本假设是条件独立性

$$P(X = x|Y = c_k) = P(x^{(1)}, \dots, x^{(n)}|y^k) = \prod_{j=1}^n P(x^{(j)}|Y = c_k)$$

- $c_k$ 代表类别， $k$ 代表类别个数。
- 这是一个较强的假设。由于这一假设，模型包含的条件概率的数量大为减少，朴素贝叶斯法的学习与预测大为简化。因而朴素贝叶斯法高效，且易于实现。其缺点是分类的性能不一定很高。

## 3. 朴素贝叶斯法利用贝叶斯定理与学到的联合概率模型进行分类预测

– 我们要求的是 $P(Y|X)$ ，根据生成模型定义我们可以求 $P(X,Y)$ 和 $P(Y)$ 假设中的特征是条件独立的。这个称作朴素贝叶斯假设。形式化表示为，（如果给定 $Z$ 的情况下， $X$ 和 $Y$ 条件独立）：

$$P(X|Z) = P(X|Y, Z)$$

也可以表示为：

$$P(X, Y|Z) = P(X|Z)P(Y|Z)$$

## 3. 用于文本分类的朴素贝叶斯模型，这个模型称作多值伯努利事件模型

在这个模型中，我们首先随机选定了邮件的类型 $p(y)$ ，然后一个人翻阅词典的所有词，随机决定一个词是否出现依照概率 $p(x^{(1)}|y)$ ，出现标示为1，否则标示为0。假设有50000个单词，那么这封邮件的概率可以表示为：

$$\begin{aligned} & p(x^{(1)}, \dots, x^{(50000)}|y) \\ &= p(x^{(1)}|y)p(x^{(2)}|y, x^{(1)})p(x^{(3)}|y, x^{(1)}, x^{(2)}) \cdots p(x^{(50000)}|y, x^{(1)}, \dots, x^{(49999)}) \\ &= p(x^{(1)}|y)p(x^{(2)}|y)p(x^{(3)}|y) \cdots p(x^{(50000)}|y) \\ &= \prod_{i=1}^m p(x^{(i)}|y) \end{aligned}$$

- 假设我们正在构建一个分类器，该分类器说明文本是否与运动(Sports)有关。

我们的训练数据有5句话：

文本	标签
A great game	Sports
The election was over	Not Sports
Very clean match	Sports
A clean but forgettable game	Sports
It was a close election	Not Sports

- 我们想要计算句子 “A very close game” 是 Sports 的概率以及它不是 Sports 的概率。即 $P(\text{Sports} \mid \text{a very close game})$ 这个句子的类别是Sports的概率。

特征：单词的频率

已知贝叶斯定理 $P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$ ，则：

$$\begin{aligned} &P(\text{Sports} \mid \text{a very close game}) \\ &= \frac{P(\text{a very close game} \mid \text{Sports}) \times P(\text{Sports})}{P(\text{a very close game})} \end{aligned}$$

由于我们只是试图找出哪个类别有更大的概率，可以舍去除数，只是比较

$P(\text{a very close game} \mid \text{Sports}) \times P(\text{Sports})$  和

$P(\text{a very close game} \mid \text{Not Sports}) \times P(\text{Not Sports})$

我们假设一个句子中的每个单词都与其他单词无关。

$$\begin{aligned} &P(\text{a very close game}) \\ &= P(a) \times P(\text{very}) \times P(\text{close}) \times P(\text{game}) \end{aligned}$$

$$\begin{aligned} &P(\text{a very close game} | \text{Sports}) \\ &= P(a | \text{Sports}) \times P(\text{very} | \text{Sports}) \times P(\text{close} | \text{Sports}) \times P(\text{game} | \text{Sports}) \end{aligned}$$



计算每个类别的先验概率:

对于训练集中的给定句子,

$P(\text{Sports})$  的概率为  $\frac{3}{5}$ 。

$P(\text{Not Sports})$  是  $\frac{2}{5}$ 。

文本	标签
A great game	Sports
The election was over	Not Sports
Very clean match	Sports
A clean but forgettable game	Sports
It was a close election	Not Sports

然后, 在计算  $P(\text{game}|\text{Sports})$  就是 “game” 有多少次出现在 Sports 的样本, 然后除以 sports 为标签的文本的单词总数 ( $3+3+5=11$ )。

因此,  $P(\text{game}|\text{Sports}) = \frac{2}{11}$ 。

“close” 不会出现在任何 sports 样本中! 那就是说  $P(\text{close}|\text{Sports}) = 0$ 。

通过使用一种称为**拉普拉斯平滑**的方法：我们为每个计数加1，因此它永远不会为零。为了平衡这一点，我们将可能单词的数量添加到除数中，因此计算结果永远不会大于1。

14个单词

在这里的情况下，可能单词是['a', 'great', 'very', 'over', 'it', 'but', 'game', 'election', 'clean', 'close', 'the', 'was', 'forgettable', 'match']。

由于可能的单词数是14，因此应用平滑处理可以得到

$$P(\text{game} \mid \text{sports}) = \frac{2+1}{11+14}$$

**拉普拉斯平滑**是一种用于平滑分类数据的技术。引入拉普拉斯平滑法来解决零概率问题,通过应用此方法,先验概率和条件概率可以写为

$$P_{\lambda}(C_k) = P_{\lambda}(Y = C_k) = \frac{\sum_{i=1}^N I(y_i = C_k) + \lambda}{N + K\lambda}$$

$$P_{\lambda}(x_i = a_j | y = C_k) = \frac{\sum_{i=1}^N I(x_i = a_j, y_i = C_k) + \lambda}{\sum_{i=1}^N I(y_i = C_k) + A\lambda}$$

其中 $K$ 表示类别数量,  $A$ 表示 $a_j$ 中不同值的数量通常 $\lambda = 1$

加入拉普拉斯平滑之后, 避免了出现概率为0的情况, 又保证了每个值都在0到1的范围内, 又保证了最终和为1的概率性质。

Word	P (word   Sports)	P (word   Not Sports)
<b>a</b>	$(2 + 1) \div (11 + 14)$	$(1 + 1) \div (9 + 14)$
<b>very</b>	$(1 + 1) \div (11 + 14)$	$(0 + 1) \div (9 + 14)$
<b>close</b>	$(0 + 1) \div (11 + 14)$	$(1 + 1) \div (9 + 14)$
<b>game</b>	$(2 + 1) \div (11 + 14)$	$(0 + 1) \div (9 + 14)$

$$P(a | \text{Sports}) \times P(\text{very} | \text{Sports}) \times P(\text{close} | \text{Sports}) \times P(\text{game} | \text{Sports}) \times P(\text{Sports}) \\ = 2.76 \times 10^{-5} = 0.0000276$$

$$P(a | \text{Not Sports}) \times P(\text{very} | \text{Not Sports}) \times P(\text{close} | \text{Not Sports}) \\ \times P(\text{game} | \text{Not Sports}) \times P(\text{Not Sports}) \\ = 0.572 \times 10^{-5} = 0.00000572$$

由于0.0000276大于0.00000572，我们的分类器预测 “A very close game” 是Sport类。

■ 贝叶斯，全名托马斯·贝叶斯(Thomas Bayes)，出生于英国伦敦，死于1761年，英国新教徒，数学家，主要研究概率论。1763年12月23日，由理查德·普莱斯(Richard Price)在伦敦皇家学会会议上宣读了贝叶斯的遗世之作 - 《机遇理论中一个问题的解》(An essay towards solving a problem in the doctrine of chances)，提出了一种归纳推理的理论，从此贝叶斯定理诞生于世，后来的许多研究者将其不断完善，最终发展为一种系统的统计推理方法 - 贝叶斯方法。



■ 现有资料表明，贝叶斯是一位神职人员，长期担任英国坦布里奇韦尔斯地方教堂的牧师，他从事数学研究的目的是为了证明上帝的存在。他在1742年当选英国皇家学会会士，但没有记录表明他此前发表过任何科学或数学论文。他的提名是由皇家学会的重量级人物签署的，但为什么提名以及他为何能当选至今仍是谜。贝叶斯的研究工作和他本人在他生活的时代很少有人关注，贝叶斯定理出现后很快就被遗忘了，后来大数学家拉普拉斯使它重新被科学界所熟悉，但直到二十世纪随着统计学的广泛应用才备受瞩目。贝叶斯的出生年份至今也没有清楚确定，甚至关于如今广泛流传的他的画像是不是贝叶斯本人也仍存在争议。



# 目录

## Contents

1. K最近邻

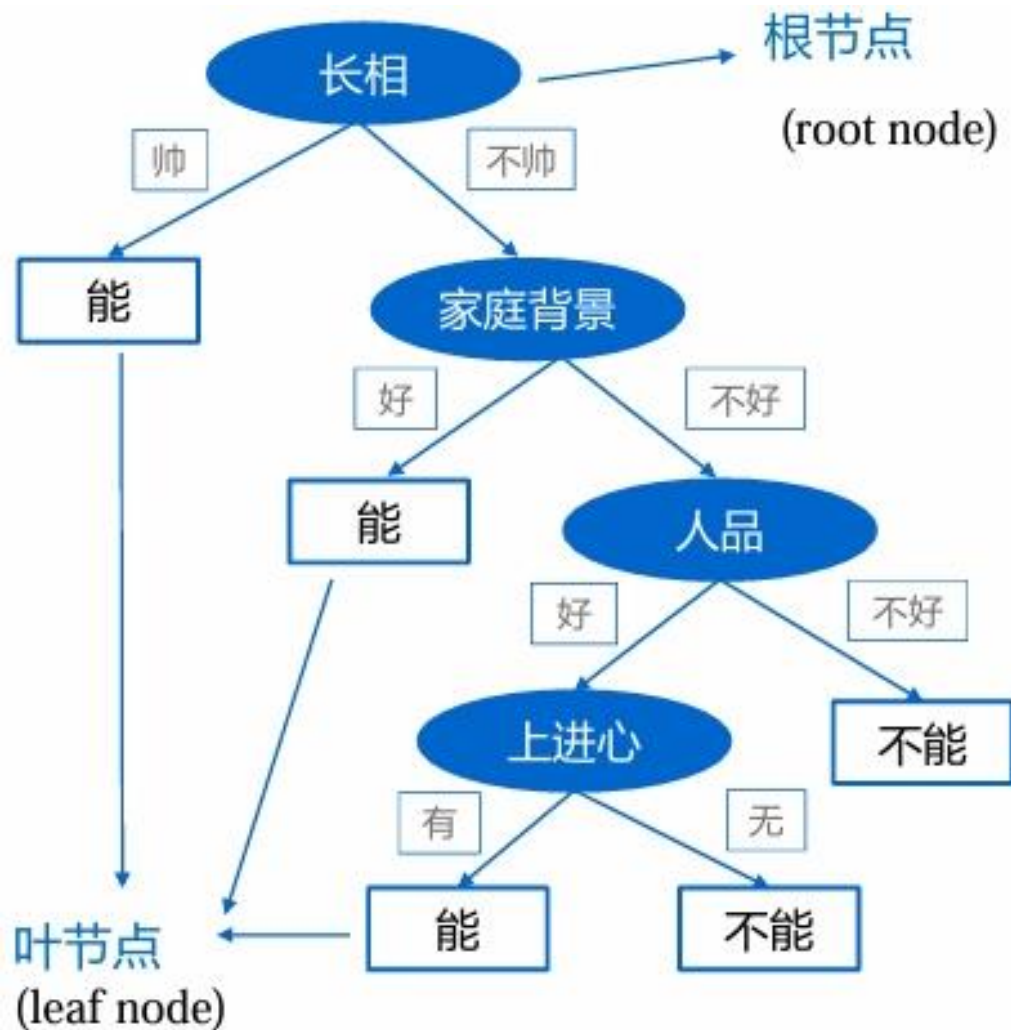
2. 朴素贝叶斯

3. 决策树

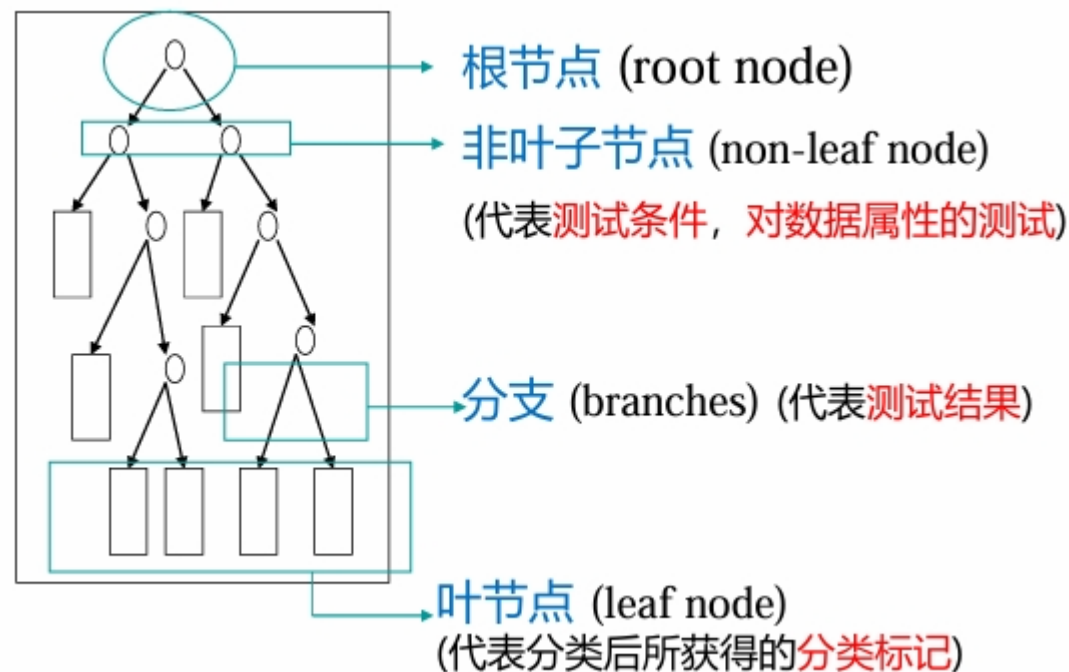
4. 支持向量机



- ❑ 决策树：从训练数据中学习得出一个树状结构的模型
- ❑ 决策树属于**判别模型**
- ❑ 决策树是一种树状结构，通过做出一系列决策（选择）来对数据进行划分，这类似于针对一系列问题进行选择
- ❑ 决策树的决策过程就是从根节点开始，测试待分类项中对应的特征属性，并按照其值选择输出分支，直到叶子节点，将叶子节点的存放的类别作为决策结果



- ❑ 决策树算法是一种归纳分类算法，它通过对训练集的学习，挖掘出有用的规则，用于对新数据进行预测
- ❑ 决策树算法属于**监督学习**方法
- ❑ 决策树归纳的基本算法是贪心算法，自顶向下来构建决策树
- ❑ 贪心算法：在每一步选择中都采取在当前状态下最好/优的选择
- ❑ 在决策树的生成过程中，分割方法即属性选择的度量是关键





## □ 优点:

- 推理过程容易理解，计算简单，可解释性强
- 比较适合处理有缺失属性的样本
- 可自动忽略目标变量没有贡献的属性变量，也为判断属性变量的重要性，减少变量的数目提供参考

## □ 缺点

- 容易造成过拟合，需要采用剪枝操作
- 忽略了数据之间的相关性
- 对于各类别样本数量不一致的数据，信息增益会偏向于那些更多数值的特征

- 建立决策树的关键，即在当前状态下选择哪个属性作为分类依据。根据不同的目标函数，建立决策树主要有以下三种算法：ID3(Iterative Dichotomiser)、C4.5、CART (Classification And Regression Tree)。

算法	支持模型	树结构	特征选择	连续值处理	缺失值处理	剪枝	特征属性多次使用
ID3	分类	多叉树	信息增益	不支持	不支持	不支持	不支持
C4.5	分类	多叉树	信息增益率	支持	支持	支持	不支持
CART	分类 回归	二叉树	基尼指数 均方差	支持	支持	支持	支持

- ID3 算法最早是由罗斯昆 (J.RossQuinlan) 于1975年提出的一种决策树构建算法，算法的核心是“信息熵”，期望信息越小，信息熵越大，样本纯度越低。
- ID3 算法是以信息论为基础，以信息增益为衡量标准，从而实现对数据的归纳分类。
- ID3 算法计算每个属性的信息增益，并选取具有最高增益的属性作为给定的测试属性。

**信息熵**

$$H(D) = - \sum_{k=1}^K \frac{|C_k|}{|D|} \log_2 \frac{|C_k|}{|D|}$$

$K$ 是类别,  $D$ 是数据集,  $C_k$ 是类别 $K$ 下的数据集

右边数据中:

数量	是	否	信息熵
15	9	6	0.971

$$H(D) = - \sum_{k=1}^K \frac{|C_k|}{|D|} \log_2 \frac{|C_k|}{|D|} = - \frac{9}{15} \log_2 \frac{9}{15} - \frac{6}{15} \log_2 \frac{6}{15}$$

$$= 0.971$$

	年龄	有工作	有房子	信用	类别
0	青年	否	否	一般	否
1	青年	否	否	好	否
2	青年	是	否	好	是
3	青年	是	是	一般	是
4	青年	否	否	一般	否
5	中年	否	否	一般	否
6	中年	否	否	好	否
7	中年	是	是	好	是
8	中年	否	是	非常好	是
9	中年	否	是	非常好	是
10	老年	否	是	非常好	是
11	老年	否	是	好	是
12	老年	是	否	好	是
13	老年	是	否	非常好	是
14	老年	否	否	一般	否



## 按年龄划分

年龄	数量	是	否	信息熵
青年	5	2	3	0.9710
中年	5	3	2	0.9710
老年	5	4	1	0.7219

$A_1$	年龄
$A_2$	有工作
$A_3$	有房子
$A_4$	信用

	年龄	有工作	有房子	信用	类别
0	青年	否	否	一般	否
1	青年	否	否	好	否
2	青年	是	否	好	是
3	青年	是	是	一般	是
4	青年	否	否	一般	否
5	中年	否	否	一般	否
6	中年	否	否	好	否
7	中年	是	是	好	是
8	中年	否	是	非常好	是
9	中年	否	是	非常好	是
10	老年	否	是	非常好	是
11	老年	否	是	好	是
12	老年	是	否	好	是
13	老年	是	否	非常好	是
14	老年	否	否	一般	否

$$H(D|A_1 = \text{青年}) = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5} = 0.971$$

$$H(D|A_1 = \text{中年}) = -\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5} = 0.971$$

$$H(D|A_1 = \text{老年}) = -\frac{4}{5}\log_2\frac{4}{5} - \frac{1}{5}\log_2\frac{1}{5} = 0.7219$$

**条件熵**  $H(D|A) = \sum_{i=1}^n \frac{|D_i|}{|D|} H(D_i)$

A是特征,  $i$ 是特征取值

$$\begin{aligned} H(D|\text{年龄}) &= \sum_{i=1}^n \frac{|D_i|}{|D|} H(D_i) \\ &= \frac{5}{15} \times 0.971 + \frac{5}{15} \times 0.971 + \frac{5}{15} \times 0.7219 \\ &= 0.8897 \end{aligned}$$

	年龄	有工作	有房子	信用	类别
0	青年	否	否	一般	否
1	青年	否	否	好	否
2	青年	是	否	好	是
3	青年	是	是	一般	是
4	青年	否	否	一般	否
5	中年	否	否	一般	否
6	中年	否	否	好	否
7	中年	是	是	好	是
8	中年	否	是	非常好	是
9	中年	否	是	非常好	是
10	老年	否	是	非常好	是
11	老年	否	是	好	是
12	老年	是	否	好	是
13	老年	是	否	非常好	是
14	老年	否	否	一般	否

信息增益  $g(D, A) = H(D) - H(D|A)$

其中,  $H(D|A) = - \sum_{i=1}^n \frac{|D_i|}{|D|} \log_2 \frac{|D_i|}{|D|}$ ,  $n$ 是特征 $A$ 的取值个数

$$\begin{aligned} &g(D, A_1 = \text{老年}) \\ &= H(D) - H(D|A_1 = \text{老年}) \\ &= 0.971 - 0.7219 = 0.2491 \end{aligned}$$

	年龄	有工作	有房子	信用	类别
0	青年	否	否	一般	否
1	青年	否	否	好	否
2	青年	是	否	好	是
3	青年	是	是	一般	是
4	青年	否	否	一般	否
5	中年	否	否	一般	否
6	中年	否	否	好	否
7	中年	是	是	好	是
8	中年	否	是	非常好	是
9	中年	否	是	非常好	是
10	老年	否	是	非常好	是
11	老年	否	是	好	是
12	老年	是	否	好	是
13	老年	是	否	非常好	是
14	老年	否	否	一般	否

## □ 其大致步骤为：

1. 初始化特征集合和数据集合；
2. 计算数据集合信息熵和所有特征的条件熵，选择信息增益最大的特征作为当前决策节点；
3. 更新数据集合和特征集合（删除上一步使用的特征，并按照特征值来划分不同分支的数据集合）
4. 重复2，3两步，若子集值包含单一特征，则为分支叶子节点。



## □ 缺点:

1. ID3 没有剪枝策略，容易过拟合；
2. 信息增益准则对可取值数目较多的特征有所偏好，类似“编号”的特征，其信息增益接近于1；
3. 只能用于处理离散分布的特征；
4. 没有考虑缺失值。

□ C4.5 算法是Ross 对ID3算法的改进：

1. 用信息增益率来选择属性。ID3选择属性用的是子树的信息增益，而C4.5用的是信息增益率。
2. 在决策树构造过程中进行剪枝。
3. 对非离散数据也能处理。
4. 能够对不完整数据进行处理。

**信息增益率**  $g_R(D, A) = \frac{g(D, A)}{H_A(D)}$

其中,  $H(D|A) = - \sum_{i=1}^n \frac{|D_i|}{|D|} \log_2 \frac{|D_i|}{|D|}$ ,  $n$ 是特征 $A$ 的取值个数

$$g(D, A_1 = \text{老年}) = H(D) - H(D|A_1 = \text{老年})$$

$$= 0.971 - 0.7219 = 0.2491$$

$$g_R(D, A_1 = \text{老年}) = \frac{g(D, A_1 = \text{老年})}{H_A(D)} = \frac{0.2491}{-\sum_{i=1}^n \frac{|D_i|}{|D|} \log_2 \frac{|D_i|}{|D|}}$$

$$= \frac{0.2491}{-\frac{9}{15} \log_2 \frac{9}{15} - \frac{6}{15} \log_2 \frac{6}{15}} = 0.2565$$

**备注：信息增益**  $g(D, A) = H(D) - H(D|A)$

	年龄	有工作	有房子	信用	类别
0	青年	否	否	一般	否
1	青年	否	否	好	否
2	青年	是	否	好	是
3	青年	是	是	一般	是
4	青年	否	否	一般	否
5	中年	否	否	一般	否
6	中年	否	否	好	否
7	中年	是	是	好	是
8	中年	否	是	非常好	是
9	中年	否	是	非常好	是
10	老年	否	是	非常好	是
11	老年	否	是	好	是
12	老年	是	否	好	是
13	老年	是	否	非常好	是
14	老年	否	否	一般	否

### □ 过拟合的原因：

1. 为了尽可能正确分类训练样本，节点的划分过程会不断重复直到不能再分，这样就可能对训练样本学习的“太好”了，把训练样本的一些特点当做所有数据都具有的一般性质，从而导致过拟合。
2. 通过剪枝处理去掉一些分支来降低过拟合的风险。
3. 剪枝的基本策略有“预剪枝”（prepruning）和“后剪枝”（post-pruning）

## □ 预剪枝 (prepruning)

1. 预剪枝不仅可以降低过拟合的风险而且还可以减少训练时间，但另一方面它是基于“贪心”策略，会带来欠拟合风险。

### 训练集

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	是
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	是
6	青绿	稍蜷	浊响	清晰	稍凹	软粘	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	是
10	青绿	硬挺	清脆	清晰	平坦	软粘	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	否
15	乌黑	稍蜷	浊响	清晰	稍凹	软粘	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	否
17	青绿	蜷缩	沉闷	稍糊	稍凹	硬滑	否

### 验证集

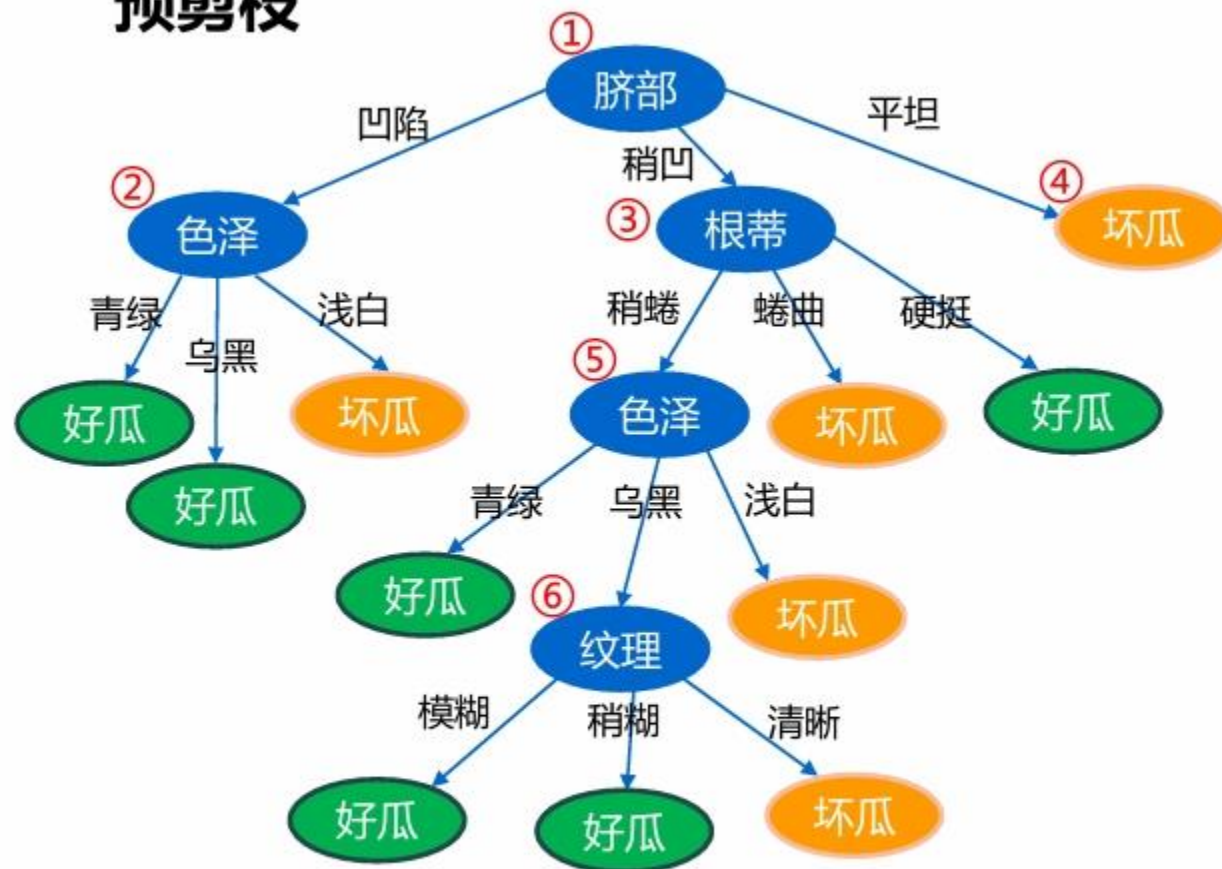
编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	是
9	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	否
11	浅白	硬挺	清脆	模糊	平坦	硬滑	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	否
13	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	否

## □ 剪枝策略

在节点划分前来确定是否继续增长，及早停止增长

1. 节点内数据样本低于某一阈值；
2. 所有节点特征都已分裂；
3. 节点划分前准确率比划分后准确率高。

### 预剪枝



基于表生成未剪枝的决策树

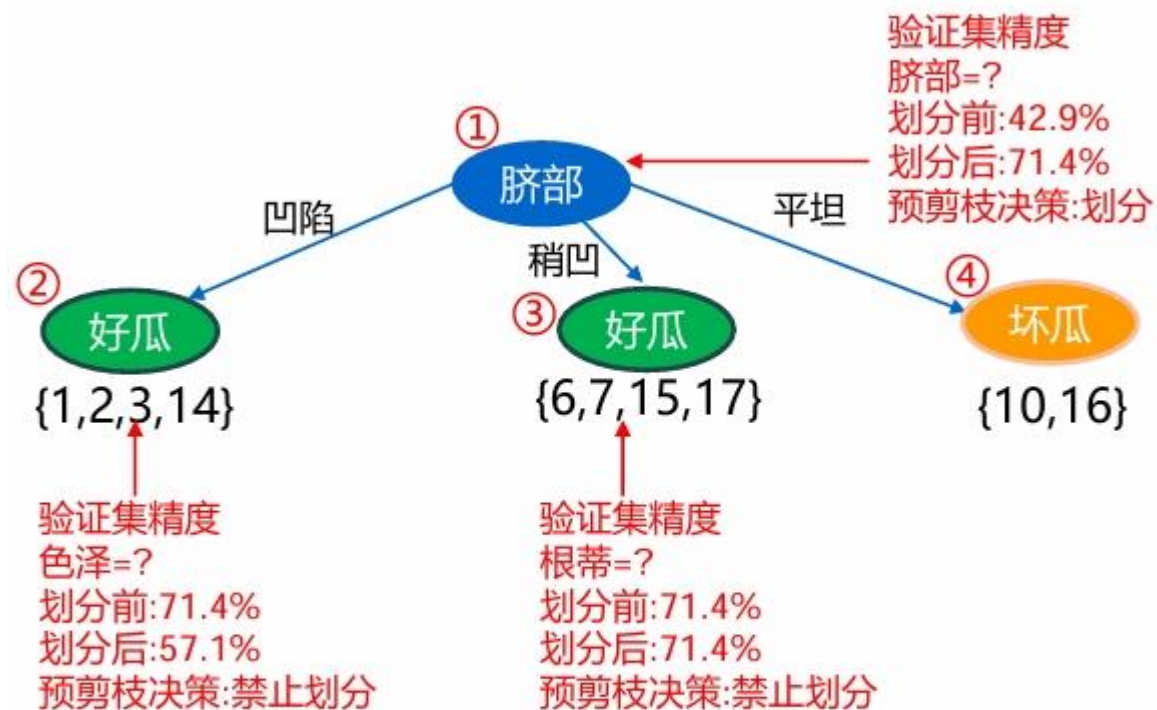


## 剪枝策略

在节点划分前来确定是否继续增长，及早停止增长

1. 节点内数据样本低于某一阈值；
2. 所有节点特征都已分裂；
3. 节点划分前准确率比划分后准确率高。

## 预剪枝



预剪枝的决策树

## □ 后剪枝

1. 在已经生成的决策树上进行剪枝，从而得到简化版的剪枝决策树。
2. 后剪枝决策树通常比预剪枝决策树保留了更多的分支。一般情况下，后剪枝的欠拟合风险更小，泛化性能往往优于预剪枝决策树。

### 训练集

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	是
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	是
6	青绿	稍蜷	浊响	清晰	稍凹	软粘	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	是
10	青绿	硬挺	清脆	清晰	平坦	软粘	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	否
15	乌黑	稍蜷	浊响	清晰	稍凹	软粘	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	否
17	青绿	蜷缩	沉闷	稍糊	稍凹	硬滑	否

### 验证集

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	是
9	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	否
11	浅白	硬挺	清脆	模糊	平坦	硬滑	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	否
13	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	否

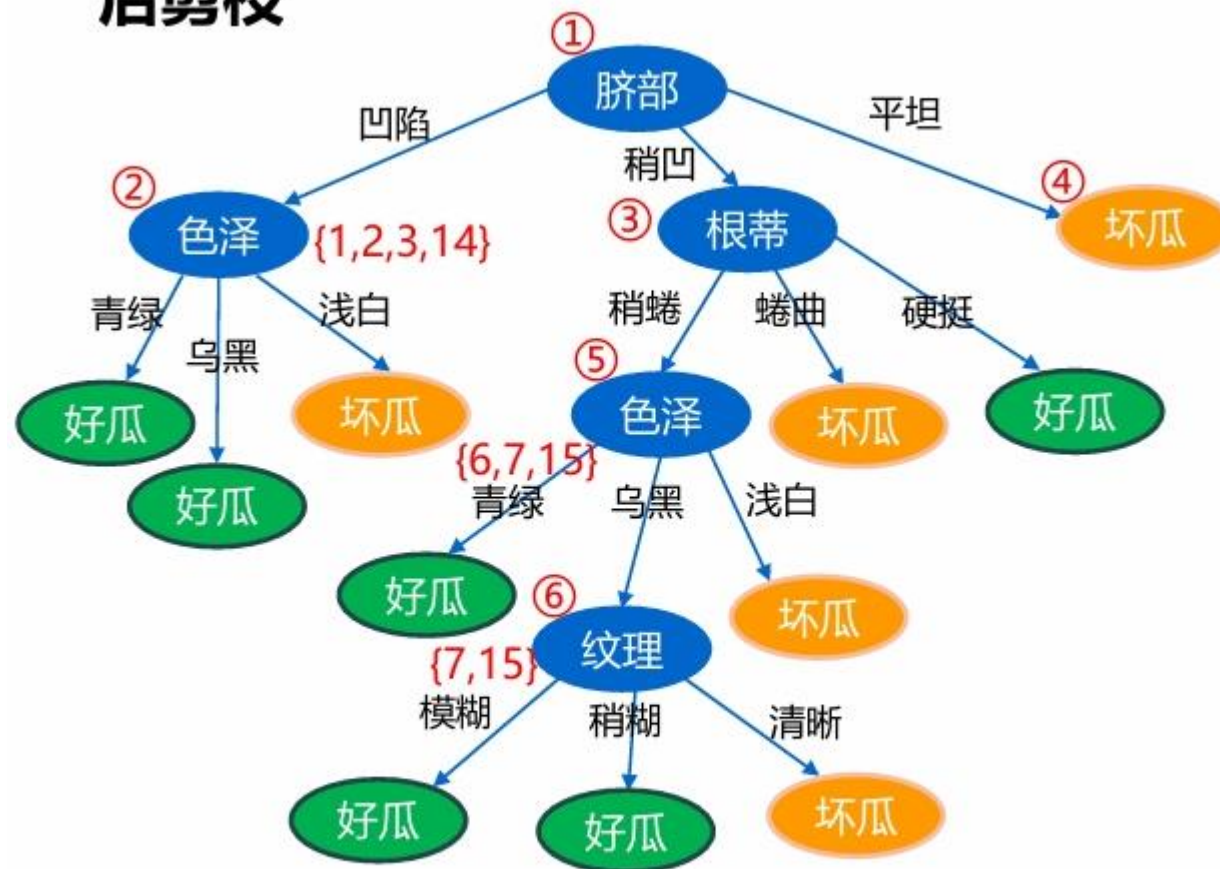


## □ 剪枝方法

在已经生成的决策树上进行剪枝，从而得到简化版的剪枝决策树

1. C4.5 采用的悲观剪枝方法，用递归的方式从低往上针对每一个非叶子节点，评估用一个最佳叶子节点去代替这棵子树是否有益。如果剪枝后与剪枝前相比其错误率是保持或者下降，则这棵子树就可以被替换掉。
2. C4.5 通过训练数据集上的错误分类数量来估算未知样本上的错误率。
3. 后剪枝决策树的欠拟合风险很小，泛化性能往往优于预剪枝决策树。

## 后剪枝



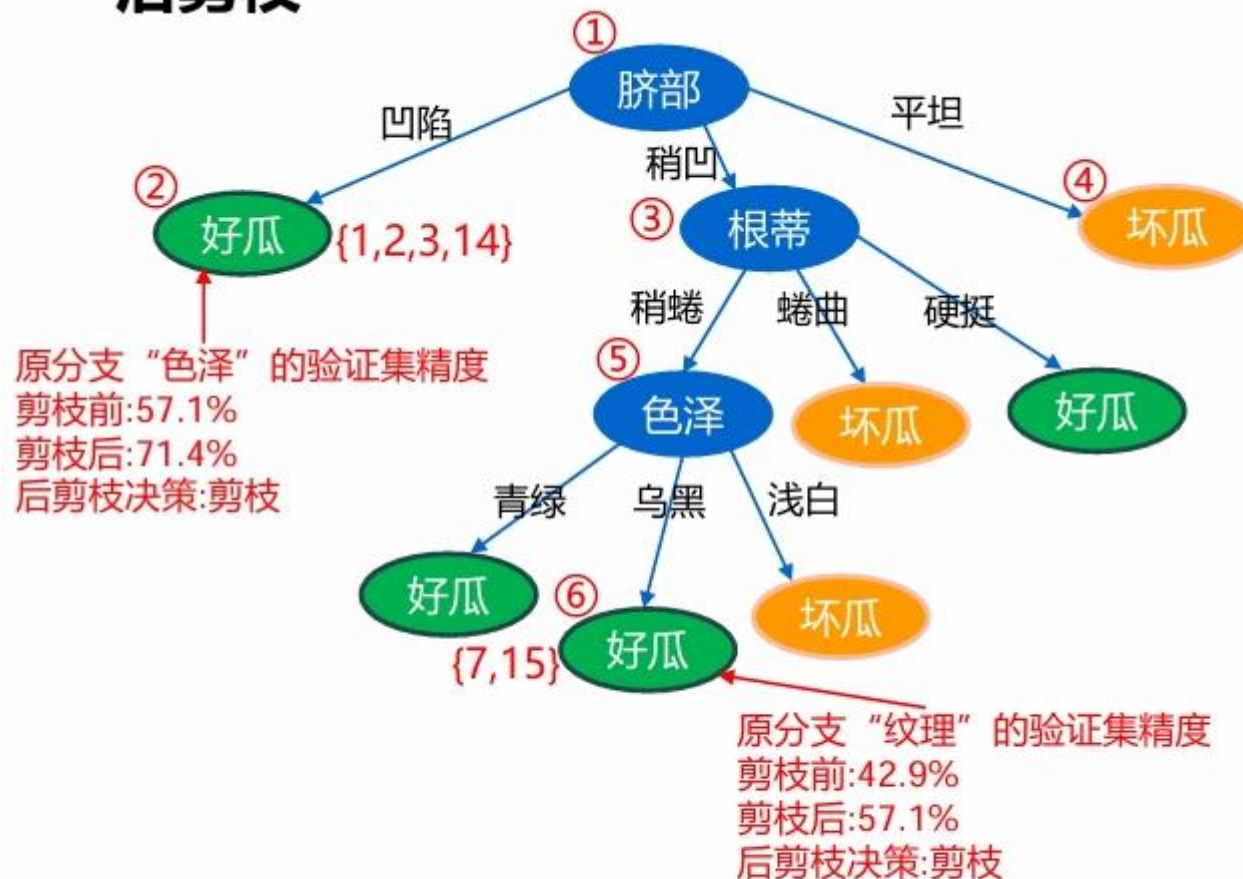
基于表生成未剪枝的决策树

## □ 剪枝方法

在已经生成的决策树上进行剪枝，从而得到简化版的剪枝决策树

1. C4.5 采用的悲观剪枝方法，用递归的方式从低往上针对每一个非叶子节点，评估用一个最佳叶子节点去代替这棵子树是否有益。如果剪枝后与剪枝前相比其错误率是保持或者下降，则这棵子树就可以被替换掉。
2. C4.5 通过训练数据集上的错误分类数量来估算未知样本上的错误率。
3. 后剪枝决策树的欠拟合风险很小，泛化性能往往优于预剪枝决策树。

## 后剪枝



## 后剪枝的决策树

- 说起决策树学习，就必然要谈到澳大利亚计算机科学家罗斯·昆兰 (J.Ross Quinlan.1943-)最初的决策树算法是心理学家兼计算机科学家E.B.Hunt1962年在研究人类的概念学习过程时提出的CLS(Concept Learning System),这个算法确立了决策树“分而治之”的学习策略。
- 罗斯·昆兰在Hunt 的指导下于1968年在美国华盛顿大学获得计算机博士学位,然后到悉尼大学任教。1978年他在学术假时到斯坦福大学访问，选修了图灵的助手D.Michie 开设的一门研究生课程.课上有一个大作业，要求写程序来学习出完备正确的规则，以判断国际象棋残局中一方是否会在两步棋后被将死。昆兰写了一个类似于CLS 的程序来完成作业,其中最重要的改进是引入了信息增益准则。后来他把这个工作整理出来在1979年发表这就是 ID3算法。1986年MachineLearning杂志创刊,昆兰应邀在创刊号上重新发表了ID3算法,掀起了决策树研究的热潮。短短几年间众多决策树算法问世，ID4、ID5等名字迅速被其他研究者提出的算法占用,昆兰只好将自己的 ID3 后继算法命名为 C4.0.在此基础上进一步提出了著名的 C4.5。





# 目录

## Contents

1. K最近邻

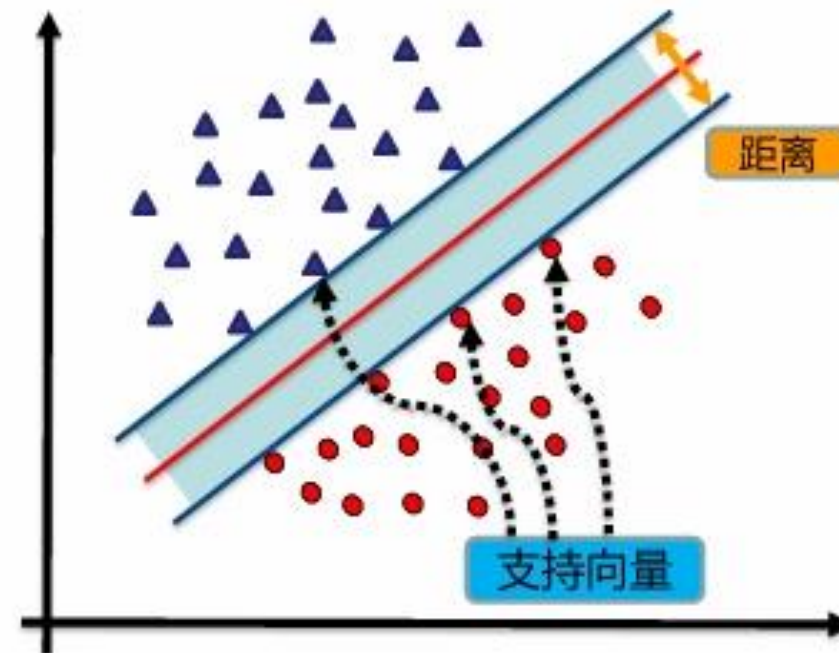
2. 朴素贝叶斯

3. 决策树

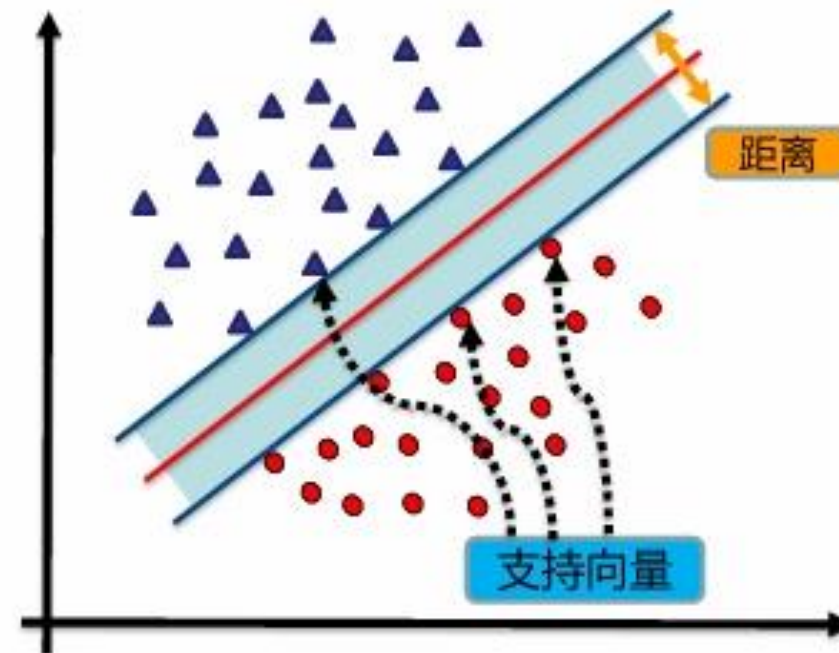
4. 支持向量机



- **支持向量机 (Support Vector Machine, SVM)** 是一类按监督学习 (supervised learning) 方式对数据进行二元分类的广义线性分类器 (generalized linear classifier), 其决策边界是对学习样本求解的最大边距超平面 (maximum-margin hyperplane)。
- 与对率回归和神经网络相比, 支持向量机, 在学习复杂的非线性方程时提供了一种更为清晰, 更距离支持向量加强大的方式。



- 找到集合边缘上的若干数据（称为支持向量（Support Vector）），用这些点找出一个平面（称为决策面），使得支持向量到该平面的距离最大。



## 背景知识

任意超平面可以用下面这个线性方程来描述：

$$w^T x + b = 0$$

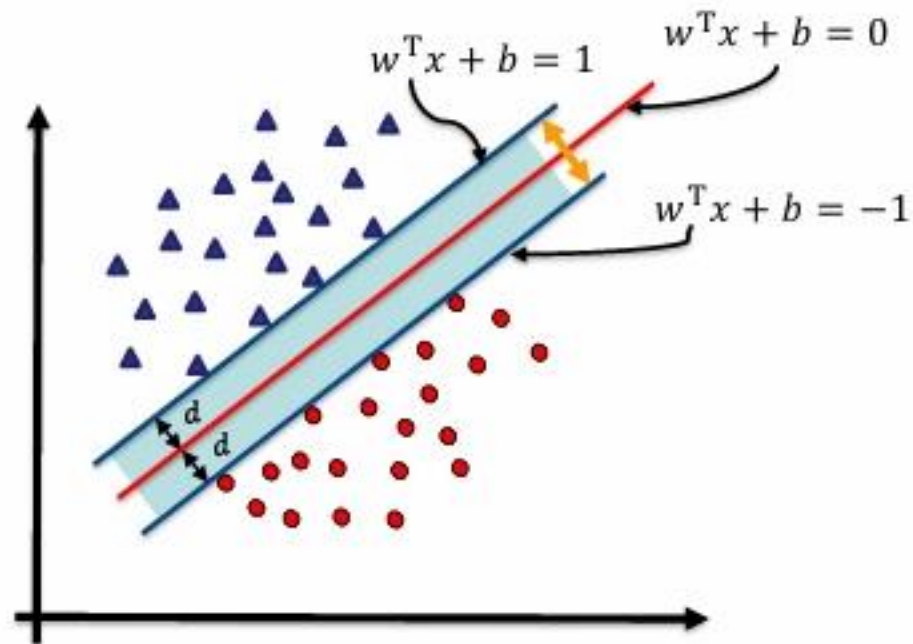
二维空间点  $(x, y)$  到直线  $Ax + By + C = 0$  的距离公式是：

$$\frac{|Ax + By + C|}{\sqrt{A^2 + B^2}}$$

扩展到  $n$  维空间后，点  $x = (x_1, x_2 \dots x_n)$  到超平面

$w^T x + b = 0$  的距离为：  $\frac{|w^T x + b|}{||w||}$

其中  $||w|| = \sqrt{w_1^2 + \dots w_n^2}$



如图所示，根据支持向量的定义我们知道，支持向量到超平面的距离为  $d$ ，其他点到超平面的距离大于  $d$ 。每个支持向量到超平面的距离可以写

$$\text{为： } d = \frac{|w^T x + b|}{||w||}$$

## 背景知识

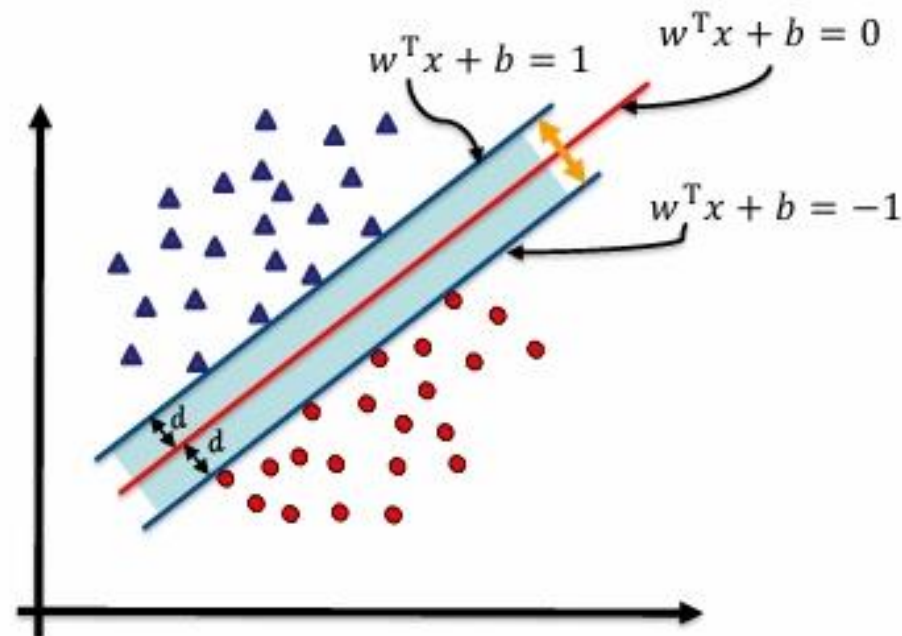
如图所示, 根据支持向量的定义我们知道, 支持向量到超平面的距离为  $d$ , 其他点到超平面的距离大于  $d$ 。

于是我们有这样的一个公式: 故: 
$$\begin{cases} \frac{w^T x + b}{\|w\|} \geq d & y = 1 \\ \frac{w^T x + b}{\|w\|} \leq -d & y = -1 \end{cases}$$

我们暂且令  $\|w\| \cdot d$  为1(之所以令它等于1, 是为了方便推导和优化且这样做对目标函数的优化没有影响)

将两个方程合并, 我们可以简写为:  $y(w^T x + b) \geq 1$

至此我们就可以得到最大间隔超平面的上下两个超平面:  $d = \frac{|w^T x + b|}{\|w\|}$





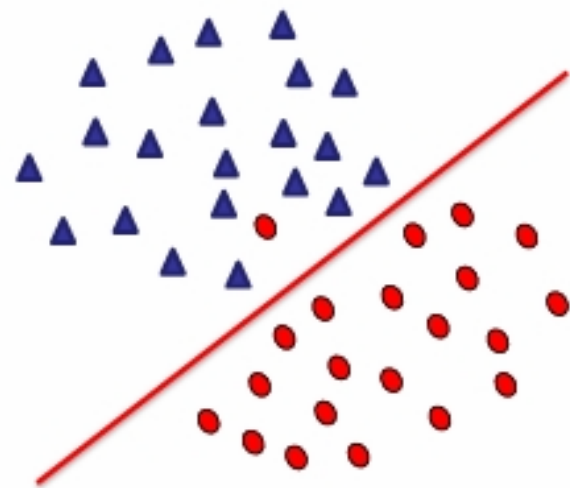
若数据线性不可分，则可以引入松弛变量 $\xi \geq 0$ ，使函数间隔加上松弛变量大于等于1，则目标函数：

$$\min_{w,b,\xi} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m \xi_i \quad s.t. \quad y_i(w^T x_i + b) \geq 1 - \xi_i$$

对偶问题：

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) = \min_{\alpha} \frac{1}{2} \sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^m \alpha_i \\ s.t. \quad & C \geq \alpha_i \geq 0, i = 1, 2, \dots, m \quad \sum_{i=1}^m \alpha_i y_i = 0 \end{aligned}$$

$C$ 为惩罚参数， $C$ 值越大，对分类的惩罚越大。跟线性可分求解的思路一致，同样这里先用拉格朗日乘子法得到拉格朗日函数，再求其对偶问题。



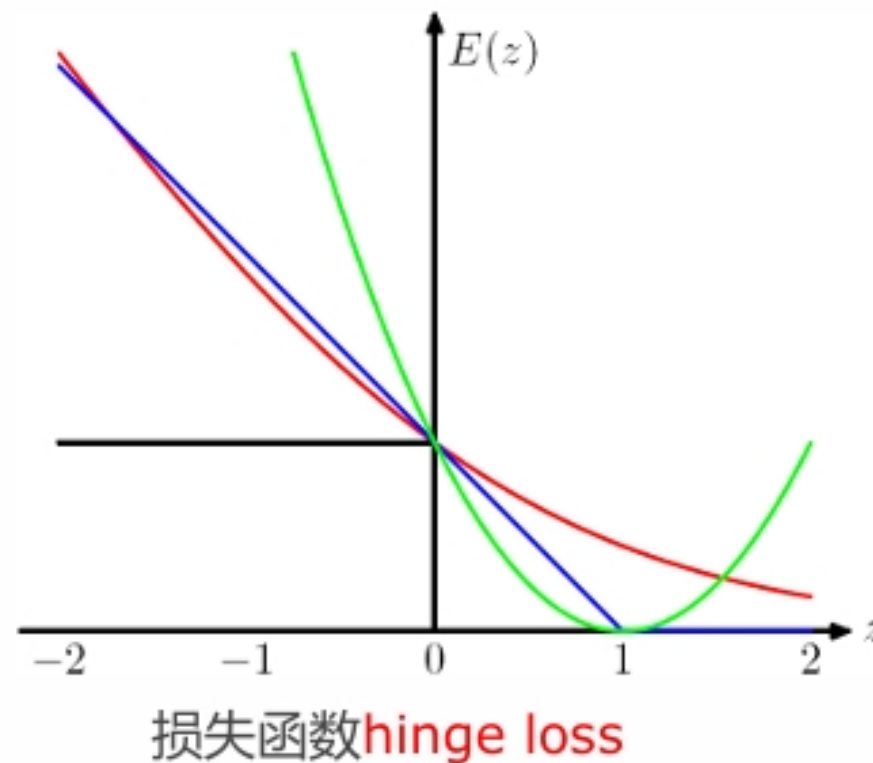
软间隔

$\xi$  为"松弛变量"

$$\xi_i = \max(0, 1 - y_i(w^T x_i + b))$$

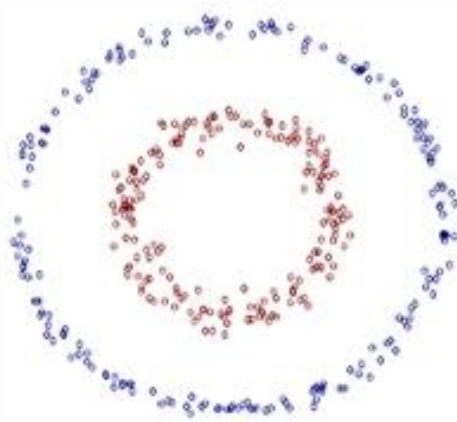
即。每一个样本都有一个对应的松弛变量，表征该样本不满足约束的程度。

绿色的线为 square loss  
蓝色的线为 hinge loss  
红色的线为负 log loss

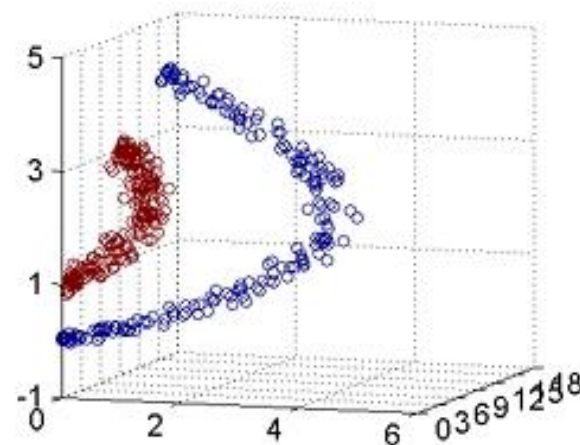


## 核技巧

- 在低维空间计算获得高维空间的计算结果，满足高维，才能在高维下线性可分。我们需要引入一个新的概念：核函数。它可以将样本从原始空间映射到一个更高维的特质空间中，使得样本在新的空间中线性可分。这样我们就可以使用原来的推导来进行计算，只是所有的推导是在新的空间，而不是在原来的空间中进行，即用核函数来替换当中的内积。



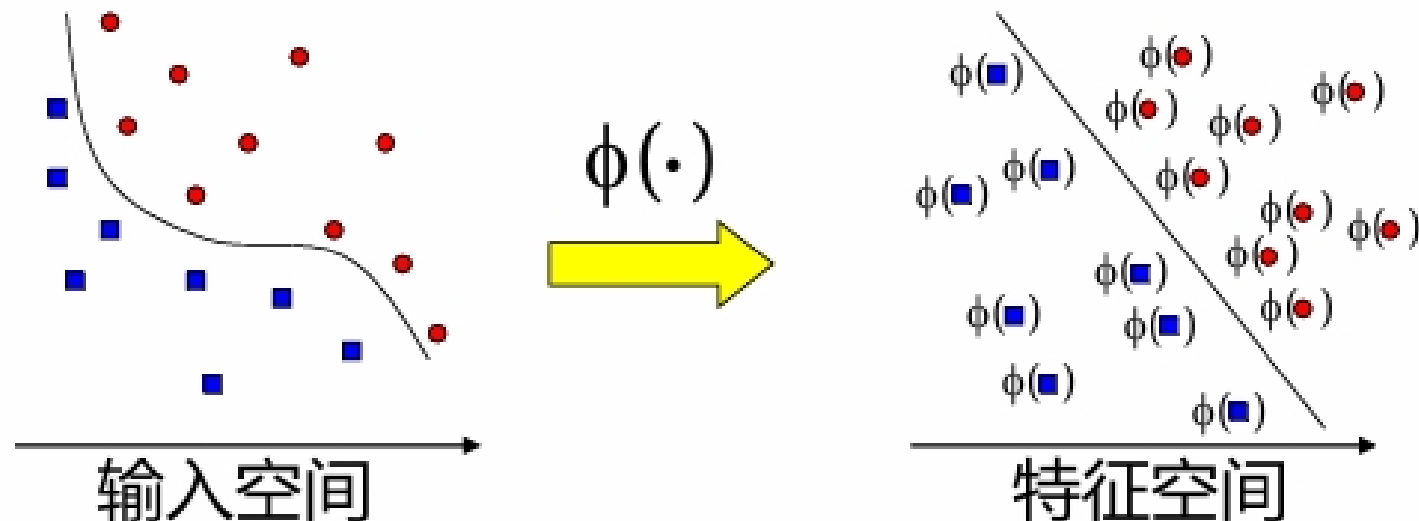
线性不可分



高维下线性可分

## 核技巧

□ 用核函数来替换原来的内积。



□ 即通过一个非线性转换后的两个样本间的内积。具体地， $K(x,z)$ 是一个核函数，或正定核，意味着存在一个从输入空间到特征空间的映射，对于任意空间输入的 $x,z$ 有：

$$K(x,z) = \phi(x) \cdot \phi(z)$$

## 核技巧

- 在线性支持向量机学习的对偶问题中，用核函数 $K(x,z)$ 替代内积，求解得到的就是非线性支持向量机

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

常用核函数有：

**线性核函数**

$$K(x_i, x_j) = x_i^T x_j$$

**多项式核函数**

$$K(x_i, x_j) = (x_i^T x_j)^d$$

**高斯核函数**

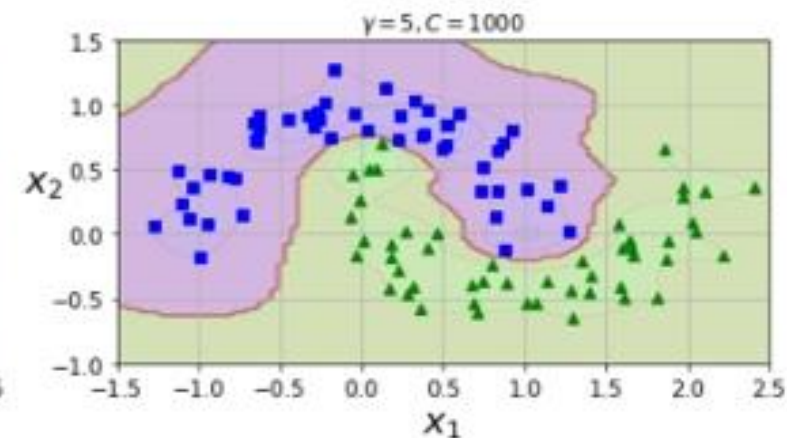
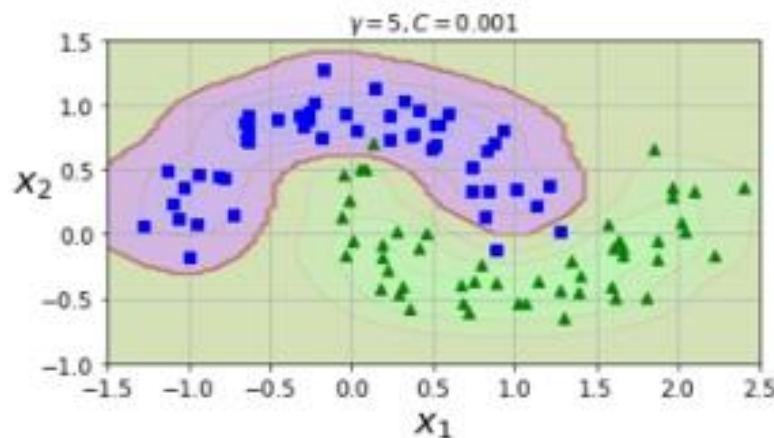
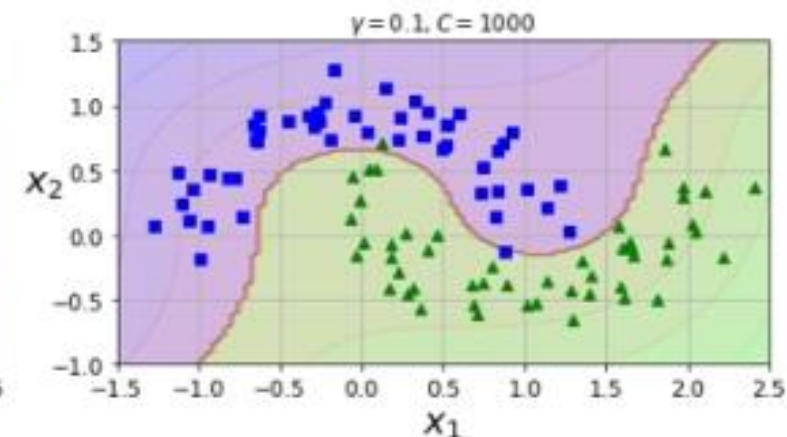
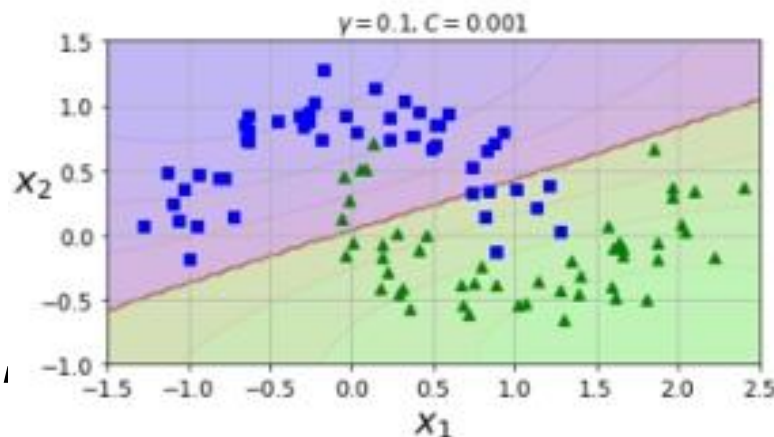
$$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\gamma^2}\right)$$

这三个常用的核函数中,只有高斯核函数是需要调参的。



## SVM的超参数

- $\gamma$  越大, 支持向量越少,  
 $\gamma$  值越小, 支持向量越多。其中  $C$  是惩罚系数, 即对误差的宽容度。  $C$  越高, 说明越不能容忍出现误差, 容易过拟合。  $C$  越小, 容易欠拟合。



- ❑ 弗拉基米尔·瓦普尼克(Vladimir N.Vapnik, 1936)是杰出的数学家、统计学家、计算机科学家。他出生于苏联，1958年在乌兹别克国立大学获数学硕士学位，1964年在莫斯科控制科学学院获统计学博士学位，此后一直在该校工作并担任计算机系主任，1990年(苏联解体的前一年)他离开苏联来到新泽西州的美国电话电报公司贝尔实验室工作，1995年发表了最初的 SVM 文章。当时神经网络正当红，因此这篇文章被权威期刊 Machine Learning 要求以“支持向量网络”的名义发表。



- ❑ 实际上，瓦普尼克在 1963 年就已提出了支持向量的概念，1968 年他与另一位苏联数学家 A.Chervonenkis 提出了以他们两人的姓氏命名的“VC 维”，1974 年又提出了结构风险最小化原则，使得统计学习理论在二十世纪七十年代就已成型，但这些工作主要是以俄文发表的，直到瓦普尼克随着东欧剧变和苏联解体导致的苏联科学家移民潮来到美国，这方面的研究才在西方学术界引起重视，统计学习理论、支持向量机、核方法在二十世纪末大红大紫。
- ❑ 瓦普尼克 2002 年离开美国电话电报公司加入普林斯顿的 NEC 实验室，2014 年加盟脸书(Facebook)公司人工智能实验室。1995 年之后他还在伦敦大学、哥伦比亚大学等校任教授。据说瓦普尼克在苏联根据一本字典自学了英语及其发音。他有一句名言被广为传诵：“Nothing is more practical than a good theory”。





中國農業大學  
China Agricultural University

最近邻法近求真， 距离权衡定乾坤。  
贝叶斯法藏机巧， 条件概率细辨分。  
决策树中枝叶繁， 信息增益绘路盘。  
支持向量高维跃， 超平面上隔峰峦。

