

第四章 统计机器学习方法

◆什么是机器学习?

- “如果一个系统能够通过执行某个过程改进它的性能，这就是学习”——西蒙

- 统计学习就是计算机系统通过运用数据及统计方法提高系统性能的机器学习

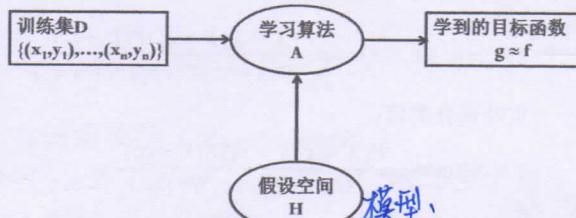
◆统计学习从数据出发，提取数据的特征，抽象出数据的模型，发现数据中的知识，又回到对数据的分析与预测中。

◆统计学习的目标就是考虑学习什么样的模型和如何学习，以使模型能对数据进行准确的预测和分析

什么是统计机器学习?

- 输入: $x \in X$
- 输出: $y \in Y$
- 未知的目标函数: $f: X \rightarrow Y$
- 训练集: $D = \{(x_1, y_1), \dots, (x_n, y_n)\}$,
D由产生 (可能具有噪音)
- 假设空间: $H = \{h_k\}$
- 学到的目标函数: $g \in H$
- 学习算法: A

是真正的
那个!



模型：
假定大概是什么样 (如：“线性”，
会有各种的
 $ax + b$)

- 学习算法A根据训练集D从假设空间H中选择一个那空间中最好的 $g \approx f$

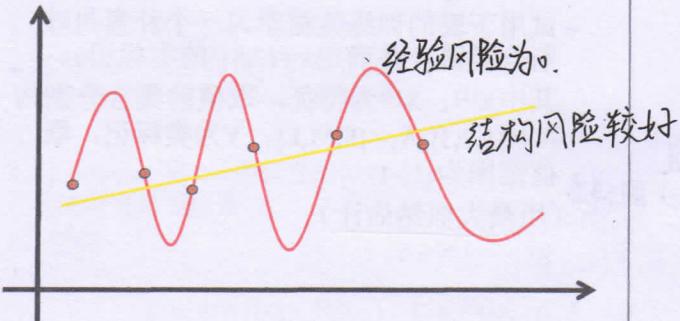
◆统计学习三要素：

- 模型：学习什么样的模型
 - 条件概率分布、决策函数
- 策略：模型选择的准则
 - 经验风险最小化、结构风险最小化
Error 加了正则项
- 算法：模型学习的算法
 - 一般归结为一个最优化问题

◆统计机器学习分类：

- 这门课讲
- 监督学习 有标志的
 - 非监督学习 无标志 (机器自己区分出分类)
 - 半监督学习 一半左右有标志 (两部分都可以利用)
 - 弱监督 (标记的数据很少或非常笼统)

过拟合与泛化能力



统计机器学习的应用

- 应用广泛，信息处理的各个方面几乎都要用到机器学习
 - 文字、语音识别，输入法
 - 搜索引擎
 - 推荐、广告
 - 文本处理、机器翻译
 - 图像、视频处理
 -

4.1 朴素贝叶斯法 (Naïve Bayes)

◆ 朴素贝叶斯法属于一种分类方法，基于特征条件独立假设学习输入/输出的联合概率分布，以此为模型，对于给定的输入 x ，利用贝叶斯定理求出后验概率最大的输出 y 。

◆ 简单有效，是一种常用的机器学习方法

◆ 设输入空间 $X \subseteq \mathbb{R}^n$ 为 n 维向量的集合

◆ 输出空间为类标记集合 $Y = \{c_1, c_2, \dots, c_k\}$ 分类共 k 个类

◆ X 是定义在输入空间上的随机变量

◆ Y 是定义在输出空间上的随机变量

假设：◆ $P(X, Y)$ 是 X 和 Y 的联合概率分布

$$T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$$

是由 $P(X, Y)$ 独立同分布产生的训练集

即样本应有一定的代表性。

贝叶斯法则

$$\Rightarrow P(Y = c_k | X = x) = \frac{P(X = x | Y = c_k)P(Y = c_k)}{\sum_k P(X = x | Y = c_k)P(Y = c_k)}$$

贝叶斯分类器：

$$y = \arg \max_{c_k} \frac{P(X = x | Y = c_k)P(Y = c_k)}{\sum_k P(X = x | Y = c_k)P(Y = c_k)}$$

把 y 取遍 $= \arg \max_{c_k} P(X = x | Y = c_k)P(Y = c_k)$ 归一化因子。
每个 c_k 那个 每个 c_k 都一样。
概率最大就 $P(X = x | Y = c_k) = P(X^{(1)} = x^{(1)}, \dots, X^{(n)} = x^{(n)} | Y = c_k)$
用哪个。 具有指数量级的参数，实际上不可计算 多个特征，

“组合爆炸” 每个特征多个取值。

- ① 很难找到足够多的样本
 - ② 样本数够了组合数也过多，不好统计。
- 参数估计——极大似然估计

$$P(Y = c_k) = \frac{\sum_{i=1}^N I(y_i = c_k)}{N}, \quad k = 1, 2, \dots, K$$

$$\text{其中 } I(y_i = c_k) = \begin{cases} 1, & \text{当 } y_i = c_k \\ 0, & \text{当 } y_i \neq c_k \end{cases}$$

◆ 为此引入独立性假设，即：

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

没有组合的关系了，可算。

◆ 得到朴素贝叶斯分类器：

$$y = \arg \max_{c_k} P(Y = c_k) \prod_{j=1}^n P(X^{(j)} = x^{(j)} | Y = c_k)$$

参数估计——贝叶斯估计

默认先有 λ 个

$$P_\lambda(X^{(j)} = a_{jl} | Y = c_k) = \frac{\left(\sum_{i=1}^N I(x_i^{(j)} = a_{jl}, y_i = c_k)\right) + \lambda}{\sum_{i=1}^N I(y_i = c_k) + S_j \lambda}$$

常取 $\lambda = 1$ ，称为拉普拉斯平滑

S_j 是第 j 个特征的取值数

为 $X^{(j)}$ 的每个取值
都增加 λ 个 $y_i = c_k$ 的样本

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

$$P(X^{(j)} = a_{jl} | Y = c_k) = \frac{\sum_{i=1}^N I(x_i^{(j)} = a_{jl}, y_i = c_k)}{\sum_{i=1}^N I(y_i = c_k)}$$

其中： $j = 1, 2, \dots, n$; $l = 1, 2, \dots, S_j$; $k = 1, 2, \dots, K$

$\{a_{jl}, a_{j2}, \dots, a_{JS_j}\}$ 为第 j 个特征 $x^{(j)}$ 可能取值的集合

举例

◆ 试用下表的训练数据学习一个朴素贝叶斯分类器，并确定 $x = (2, S)^T$ 的类标记 y 。

其中 $X^{(1)}$ 、 $X^{(2)}$ 为特征，取值的集合分别为

$A_1 = \{1, 2, 3\}$, $A_2 = \{S, M, L\}$, Y 为类标记，取

值范围为 $\{1, -1\}$

(用最大似然估计)

$$\lambda = 0$$

训练数据

	1	2	3	4	5	6	7	8
X ⁽¹⁾	1	1	✓	✓	1	2	2	2
X ⁽²⁾	S	M	M	S	S	S	M	M
Y	-1	-1	1	1	-1	-1	-1	1

	9	10	11	12	13	14	15	
X ⁽¹⁾	2	2	3	3	3	3	3	
X ⁽²⁾	L	L	L	M	M	L	L	
Y	1	1	1	1	1	1	-1	

对于给定的 $x = (2, S)^T$ 计算：

$$P(Y=1)P(X^{(1)}=2|Y=1)P(X^{(2)}=S|Y=1) \\ = \frac{9}{15} \cdot \frac{3}{9} \cdot \frac{1}{9} = \frac{1}{45} \quad (A)$$

$$P(Y=-1)P(X^{(1)}=2|Y=-1)P(X^{(2)}=S|Y=-1) \\ = \frac{6}{15} \cdot \frac{2}{6} \cdot \frac{3}{6} = \frac{1}{15} \quad (B)$$

由于 $(B) > (A)$, 所以 $y = -1$

训练过程

◆ Learn_naive_Bayes_text(Examples, C)

◆ Examples 为一组文本文档以及它们的类标记。
 $C = \{c_k\}$ 为所有可能类标记的集合

◆ 此函数的功能是学习概率项 $P(w_j|c_k)$, 它描述了从类别 c_k 中的一个文档中随机抽取的一个单词为单词 w_j 的概率。该函数也同时学习类别的先验概率 $P(c_k)$ 。

类别的先验概率

1, Vocabulary \leftarrow 在 Examples 中任意文本文档中出现的所有单词及记号的集合

2, 对 C 中每个类标记 c_k

- $docs_k \leftarrow$ Examples 中类标记为 c_k 的文档子集
 - $P(c_k) \leftarrow |docs_k| / |Examples|$
 - $Text_k \leftarrow$ 将 $docs_k$ 中所有单词连接起来建立的单个文档 (为了写算法方便, 实际不必)
 - $n \leftarrow Text_k$ 的长度 (以词为单位)
 - 对 Vocabulary 中每个单词 w_j
 - $n_j \leftarrow$ 单词 w_j 出现在 $Text_k$ 中的次数
 - $P(w_j|c_k) \leftarrow \frac{n_j + 1}{n + |Vocabulary|}$
- Laplace 平滑.

分类过程

◆ Classify_naive_Bayes_text(Doc)

◆ 对文档 Doc 返回其估计的类标记。 a_i 代表在 Doc 中的第 i 个位置上出现的单词。

1, positions \leftarrow 在 Doc 中包含的能在 Vocabulary 中找到的所有单词位置

2, 返回

$$y = \arg \max_{c_k \in C} P(c_k) \prod_{i \in positions} P(a_i | c_k)$$

不包含于 Vocabulary 的忽略

遍历 (存在于 Vocabulary 的).
 每个位置上的单词

练习题

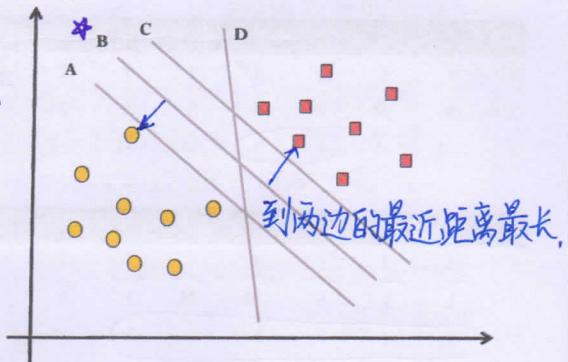
◆ 用朴素贝叶斯方法编程实现一个新闻分类系统。

4.2 支持向量机 (SVM)

◆ Support Vector Machines, SVM

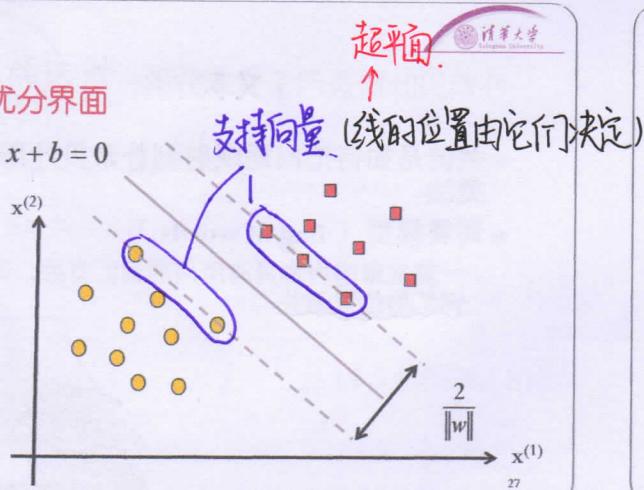
- ◆ 二类分类器 由多个二类分类器可以组成多类分类器!
- ◆ 特征空间上的间隔最大化线性分类器 因此后面都讲二类...
- ◆ 通过核技巧可实现非线性分类
- ◆ 根据模型的复杂程度可划分为:
 - 线性可分支持向量机 超平面，所有的都能分开
 - 线性支持向量机 大部分能分开
 - 非线性支持向量机

线性可分支持向量机



最优分界面

$$w \cdot x + b = 0$$



定义4.1：给定线性可分训练集：

$$\text{其中: } T = \{(x_1, y_1), \dots, (x_N, y_N)\}$$

$$x_i \in X = \mathbb{R}^n, y_i \in Y = \{+1, -1\}, i=1, 2, \dots, N$$

这里 x_i 为第 i 个特征向量, y_i 为 x_i 的类标记, +1 表示正类, -1 表示负类
只有两类.

通过间隔最大化得到分类超平面:

$$w^* \cdot x + b^* = 0$$

相应的决策函数: 符号函数(大于0的分到1, 小于0的分到-1)
 $f(x) = \text{sign}(w^* \cdot x + b^*)$

称为线性可分支持向量机

函数间隔

- ◆ 设训练集 T 和超平面 (w, b) , 定义超平面 (w, b) 关于样本点 (x_i, y_i) 的函数间隔为:

由于 y_i 取 ± 1 .
 且在线性上 $w \cdot x_i + b$
 取正. 因此 $\gamma_i = |w \cdot x_i + b|$.
 $\gamma_i = y_i(w \cdot x_i + b)$ 是一个可变大小的 $(wx+b=0)$
 到超平面的距离. 与 $kwx+kb=0$
 是同一个超平面).

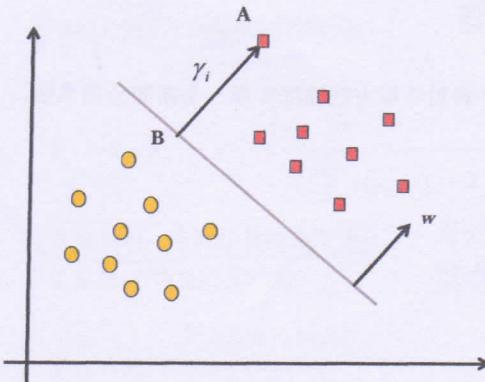
- ◆ 定义超平面关于 T 的函数间隔为:

$$\gamma_i = y_i \left(\frac{w}{\|w\|} \cdot x_i + \frac{b}{\|w\|} \right) \text{ 有了范数, 因此几何间隔固定.}$$

$$\gamma = \min_i \gamma_i$$

其中 $\|w\|$ 为 w 的 L_2 范数

意义: 垂线距离



函数间隔与几何间隔的关系

$$\gamma_i = \frac{\gamma_i}{\|w\|}$$

$$\gamma = \frac{\gamma}{\|w\|}$$

$\alpha_i > 0$

定理：设 $\alpha^* = (\alpha_1^*, \alpha_2^*, \dots, \alpha_N^*)$ 是对偶问题的解，则存在下标 j 使得 $\alpha_j^* > 0$ ，可按下式求得原始最优化问题的解：

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

就能求出来 $\alpha_i < 0$

如何调整呢？

$\alpha_i^* = 0$, 对应的 (x_i, y_i)
是打酱油的。
 $\alpha_i^* > 0$ 的, 对应的 (x_i, y_i)
是支持向量
(用哪个求结果都一样)

因此线性可分支持向量机就是求解如下的优化问题：

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

$$s.t. \sum_{i=1}^N \alpha_i y_i = 0$$

$$\alpha_i \geq 0, i = 1, 2, \dots, N$$

将 $\alpha_3 = \alpha_1 + \alpha_2$ 代入，并记为：

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

通过求偏导并令其为0，易知 $s(\alpha_1, \alpha_2)$ 在点

$$\left(\frac{3}{2}, -1\right)^T$$

所以最小值应该在边界上。

根据前面的公式得到：

$$w_1^* = w_2^* = \frac{1}{2}$$

$$b^* = -2$$

$$\text{分离超平面为: } \frac{1}{2}x^{(1)} + \frac{1}{2}x^{(2)} - 2 = 0$$

$$\text{分类决策函数为: } f(x) = \text{sign}\left(\frac{1}{2}x^{(1)} + \frac{1}{2}x^{(2)} - 2\right)$$

由定理知，分离超平面可写成：

$$\sum_{i=1}^N \alpha_i^* y_i (x \cdot x_i) + b^* = 0$$

分类决策函数可以写成：

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

由此可知：分类决策函数只依赖于输入 x 和训练样本输入的内积。

上式称为线性可分支持向量机的对偶形式，对于 $\alpha_i > 0$ 的实例 x_i 就是支持向量。

例：设正例： $x_1 = (3, 3)^T, x_2 = (4, 3)^T$ ，

负例： $x_3 = (1, 1)^T$ ，

用对偶问题求线性可分支持向量机。

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

$$= \min_{\alpha} \frac{1}{2} (18\alpha_1^2 + 25\alpha_2^2 + 2\alpha_3^2 + 42\alpha_1\alpha_2 - 12\alpha_1\alpha_3 - 14\alpha_2\alpha_3) - \alpha_1 - \alpha_2 - \alpha_3$$

$$s.t. \alpha_1 + \alpha_2 - \alpha_3 = 0$$

$$\alpha_i \geq 0, i = 1, 2, 3$$

当 $\alpha_1 = 0$ 时，最小值 $s\left(0, \frac{2}{13}\right) = -\frac{2}{13}$

当 $\alpha_2 = 0$ 时，最小值 $s\left(\frac{1}{4}, 0\right) = -\frac{1}{4}$

于是 $s(\alpha_1, \alpha_2)$ 在 $\alpha_1 = \frac{1}{4}, \alpha_2 = 0$ 时达到最小，

$$\text{此时 } \alpha_3 = \alpha_1 + \alpha_2 = \frac{1}{4}$$

这样 α_1, α_3 对应的实例点 x_1, x_3 是支持向量

练习题 考试的话大概也是这种题。(要用对偶的办法)

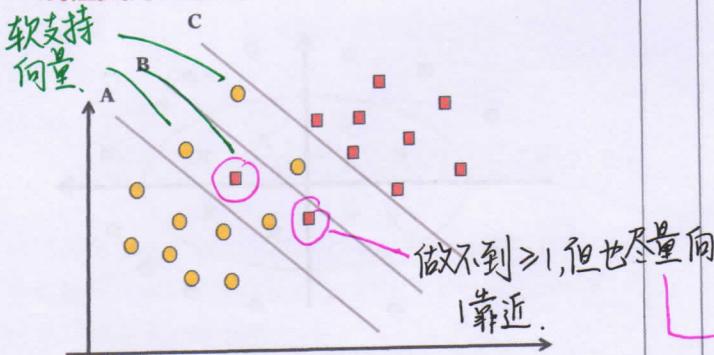
设：

正例： $x_1 = (1, 0)^T$

负例： $x_2 = (4, 3)^T, x_3 = (2, 1)^T$

求最大间隔超平面

线性支持向量机



- 为使 ξ_i 尽可能的小，优化目标增加惩罚项，变为：

$$\min_{w, b, \xi} \left(\frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \xi_i \right)$$

C → ∞: 线性可分的

- 称为软间隔最大化
- 其中 $C > 0$ 为惩罚参数， C 大时对误分类的惩罚增加， C 小时对误分类的惩罚减少。
- 上式的含义：间隔尽量最大，同时误分类的点数尽可能小。（二者由 C 调和。）

$$\frac{1}{2} \|w\|^2$$

- 同样，通过求解对偶问题求解原始问题

线性支持向量机的对偶问题：

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

$$s.t. \sum_{i=1}^N \alpha_i y_i = 0$$

$$0 \leq \alpha_i \leq C, i = 1, 2, \dots, N$$

若最小值不满足 $0 \leq \alpha_i \leq C$ ，那么满足条件的最小值人
为 α_i 在边界上（0 或 C ）时取到。
使原式成立

- 分类超平面：

$$w^* \cdot x + b^* = 0$$

都是支持向量

- 分类决策函数为：

$$f(x) = \text{sign}(w^* \cdot x + b^*)$$

回顾：线性可分支持向量机

$$\min_{w, b} \frac{1}{2} \|w\|^2$$

$$s.t. y_i (w \cdot x_i + b) \geq 1, i = 1, 2, \dots, N$$

◆ 某些点线性不可分，意味着这些点不满足函数间隔大于等于 1 的条件。

◆ 为此引入松弛变量 ξ_i ，使得：

$$y_i (w \cdot x_i + b) \geq 1 - \xi_i, i = 1, 2, \dots, N, \xi_i \geq 0$$

线性可分： $\xi_i = 0$

线性支持向量机就转化为如下的优化问题
(原始问题)：

$$\min_{w, b, \xi} \left(\frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \xi_i \right)$$

$$s.t. y_i (w \cdot x_i + b) \geq 1 - \xi_i, i = 1, 2, \dots, N$$

$$\xi_i \geq 0, i = 1, 2, \dots, N$$

求得最优解 $\alpha^* = (\alpha_1^*, \alpha_2^*, \dots, \alpha_N^*)^T$

$$\text{计算: } w^* = \sum_{i=1}^N \alpha_i^* y_i x_i$$

(选择一个 $0 < \alpha_j^* < C$ ，计算)

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

即使线性可分的问题，如果 C 选取不当，直接设等于更人
也可能存在软支持向量（先求出 α_i ，再设 C 小于 α_i 中最大值
那么那个 α_i 会在间隔

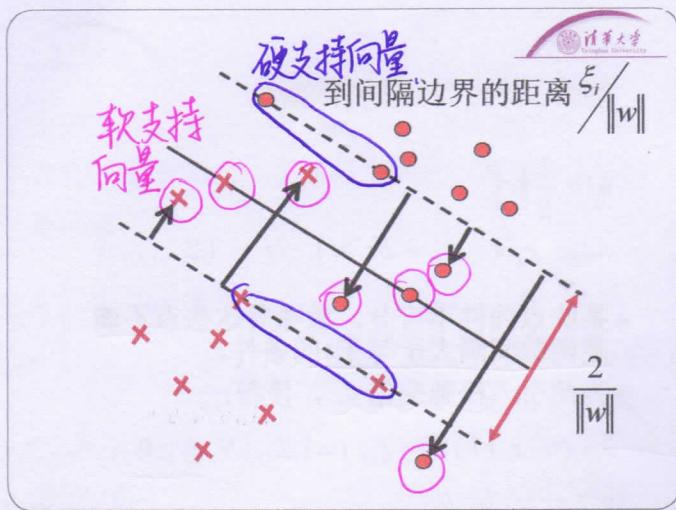
$\alpha_i^* > 0$ 所对应的样本 x_i 称为支持向量（软间隔的支持向量）

若 $\alpha_i^* < C$ ，则 $\xi_i = 0$ ， x_i 在间隔边界上

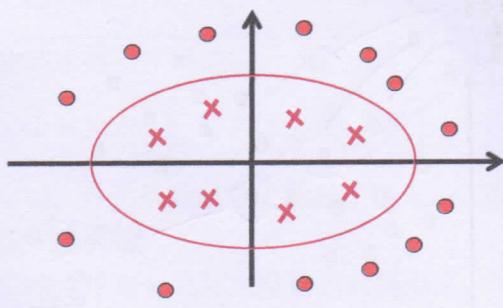
若 $\alpha_i^* = C$ ， $0 < \xi_i < 1$ ，则分类正确， x_i 在间隔边界与分离超平面之间

若 $\alpha_i^* = C$ ， $\xi_i = 1$ ，则 x_i 在超平面上

若 $\alpha_i^* = C$ ， $\xi_i > 1$ ，则 x_i 位于误分一侧



非线性支持向量机



设变换:

$$z = \phi(x) = ((x^{(1)})^2, (x^{(2)})^2)^T$$

原空间的椭圆:

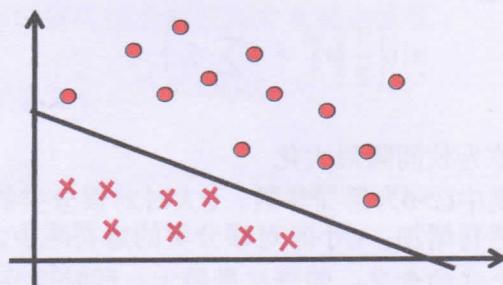
$$w_1(x^{(1)})^2 + w_2(x^{(2)})^2 + b = 0$$

变换为新空间中的直线:

$$w_1 z^{(1)} + w_2 z^{(2)} + b = 0$$

这样原空间的非线性可分问题
变为了新空间线性可分问题。

成对齐的点数



◆用线性分类的方法求解非线性分类问题

(1) 使用一个变换，将原空间数据映射到新空间；

(2) 在新空间用线性分类方法从训练数据中学习分类模型

· 不可解特空间
维数可以无限
(欧式以级有限)

核技巧应用于支持向量机

- 通过一个非线性变换将输入空间 X (欧式空间或者离散集合) 对应于一个特征空间 H (希尔伯特空间)，使得在输入空间 X 的超曲面模型对应于特征空间 H 中的超平面模型 (支持向量机)。
- 分类问题的学习就可以通过在 H 空间中求解线性支持向量机完成

线性支持向量机的对偶问题:

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

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

$$0 \leq \alpha_i \leq C, \quad i = 1, 2, \dots, N$$

\Rightarrow

非线性支持向量机的对偶问题:

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

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

$$0 \leq \alpha_i \leq C, \quad i = 1, 2, \dots, N$$

清华大学

Tsinghua University

问题: 中很难找
——能不能在不知中
Z_i · Z_j 的情况下得到
点积呢?

核函数

设 X 是输入空间, H 是特征空间, 如果存在

$$\phi(x): X \rightarrow H$$

使得对所有 $x, z \in X$, 函数 $K(x, z)$ 满足:

$$K(x, z) = \phi(x) \cdot \phi(z), \quad (\cdot \cdot \cdot \text{表示内积})$$

则称 $K(x, z)$ 为核函数, $\phi(x)$ 为映射函数
与点积是一样的!

中具体是啥不重要.

核函数举例：

设核函数是 $K(x, z) = (x \cdot z)^2$, 试找出映射 $\phi(x)$

记 $x = (x^{(1)}, x^{(2)})$, $z = (z^{(1)}, z^{(2)})$, 由于

$$(x \cdot z)^2 = (x^{(1)}z^{(1)} + x^{(2)}z^{(2)})^2$$

$$= (x^{(1)}z^{(1)})^2 + 2x^{(1)}z^{(1)}x^{(2)}z^{(2)} + (x^{(2)}z^{(2)})^2 = \phi(x) \cdot \phi(z)$$

所以可取映射: $\phi(x) = ((x^{(1)})^2, \sqrt{2}x^{(1)}x^{(2)}, (x^{(2)})^2)^T$

也可以取映射: $\phi(x) = ((x^{(1)})^2, x^{(1)}x^{(2)}, x^{(1)}x^{(2)}, (x^{(2)})^2)^T$

可见, 映射 $\phi(x)$ 并不唯一

但都是通过升维来实现。
(这样新空间中才能求得超平面)

不用构造映射 $\phi(x)$ 能否判断给定的函数 $K(x, z)$ 是核函数呢?

定理(正定核的充要条件)

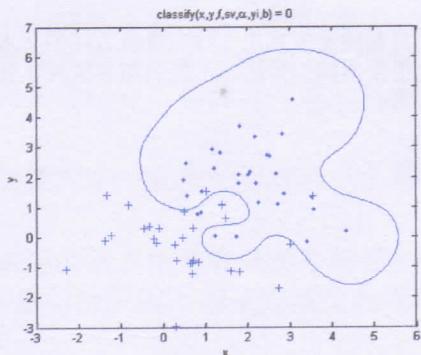
设 $X \subset \mathbb{R}^n$, $K(x, z)$ 是定义在 $X \times X$ 上的对称函数,

则 $K(x, z)$ 为正定核的充要条件是对任意 $x_i \in X$, $i = 1, \dots, m$, $K(x, z)$ 对应的 Gram 矩阵

$K = [K(x_i, x_j)]_{m \times m}$ 是半正定矩阵。

通常所说的核函数指的是正定核函数简称正定核

一个非线性分类的例子



在原始空间中分界面
也可能是多连通域
(很多个圈).

y 只能为 $+1$ 或 -1 .

SVM 用于求解多类问题

一对多

有 n 个类时分别取一个为正例构造 n 个 SVM

某类为正例, 其余类为负例。分类时将未知样本分类为具有最大分类函数值的那类 (离间隔边界越远越好)。

一对一 C^n 个 SVM

任意两类构造一个 SVM, 分类时采取投票法决定类别

层次法 $1+2+4+\dots+n$ ($O(n^2)$ 级别)

问题: 若在某一层分错了所有类先分成两类, 每类再分为两类.....

某一层分错了

则无法在后面挽救。



非线性支持向量机算法:

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

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

$$0 \leq \alpha_i \leq C, i = 1, 2, \dots, N$$

求得最优解: $\alpha^* = (\alpha_1^*, \alpha_2^*, \dots, \alpha_N^*)^T$

选择一个 α^* 的分量 $0 < \alpha_j^* < C$, 计算:

$$b^* = y_j - \sum_{i=1}^N \alpha_i^* y_i K(x_i, x_j) \quad (\text{W}^* \text{ 没有直接的公式了, 就不用它来表示了})$$

$$\text{决策函数: } f(x) = \text{sign}\left(\sum_{j=1}^N \alpha_j^* y_j K(x_j, x) + b^*\right)$$

常用的核函数

多项式核函数:

$$K(x, z) = (x \cdot z + 1)^p \quad (p > 0)$$

到

高斯核函数: 最常用. (中是无穷维空间的).

$$K(x, z) = \exp\left(-\frac{\|x - z\|^2}{2\sigma^2}\right)$$

一般来说调好 σ 的话 σ : 方差. (超参数).
高斯核函数挺好用的.

序列最小最优化算法SMO

网络课堂上有有关统计学习的 pdf 中

- 支持向量机的学习问题是一个凸二次规划问题, 具有全局最优解。
- 有很多算法可以求解这类问题, 但是当样本数多时, 往往非常低效, 以致无法使用。
- 为此, 提出了许多快速算法, SMO 由微软的 Platt 与 1998 年提出, 当时最快。

SVM 应用举例: 文本分类

文本的向量空间模型

- 文本表达为一个向量
- $(w_{1,j}, w_{2,j}, \dots, w_{n,j})^T$ 维数: Vocabulary 的维数
- w_{ij} 表示词项 i 在文档 j 中的权重

词项频率 tf_{ij} 权重

tf-idf 权重

◆ tf_{ij} 权重
对冲第 i 项
 $w_{ij} = \underline{tf}_{ij}$
◆ tf_{ij} 表示第 i 个词项在第 j 个文档中出现的次数
在文档中出现的频率

不是特别好用：“的”“地”“得”等与分类无关的
出现非常频繁。

人工调整
词语相关权重

交叉验证

- ◆ 当数据充分时，可以随机地将样本划分为三类：
 - 训练集
 - 验证集
 - 测试集
- ◆ 现实中数据往往不足

总的文档数。
◆ $tf\text{-}idf$ 权重
看是在某些中用的多，还是在所有文档中
次数都多。
逆文档频率： $idf_i = \log(N/df_i)$
比较专有的 df_i 小， idf_i 大。
(与在文档中出现的具体次数无关)。
◆ (1) $w_{ij} = tf_{ij} * idf_i$
◆ (2) $w_{ij} = (1 + \log tf_{ij}) * idf_i$, 当 $tf_{ij} = 0$ 时 $w_{ij} = 0$
◆ 第二种更常用，此外还有很多变形

◆ 交叉验证的基本思想就是重复使用数据

- ◆ 简单交叉验证
 - 将数据集划分为训练集和测试集，通过测试集选择模型
- ◆ S 折交叉验证 $k\text{-fold}$
 - 随机地将数据集划分为 s 个子集， $s-1$ 个子集用于训练，一个子集用于测试，重复 s 次
- ◆ 留一交叉验证
 - 当 $s=N$ 时的特殊情况，其中 N 为数据集的规模

实际中的问题

- ◆ 分类体系的建立 别有模棱两可的(如“电子”与“计算机”)
◆ 数据的收集 两类。
- ◆ 预处理

分词
停用词 (Stop word) 处理 如“的”“地”“得”。

词干化 (Stemming) (但要保证别分词分错),
特征选择 “的士”变成“士”

从总的 Vocabulary 中选一部分做特征，
省得特征数太大。
to swim 与 swimming 转为词干。

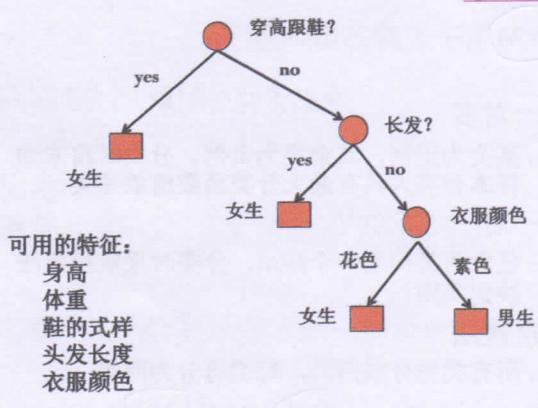
4.3 决策树

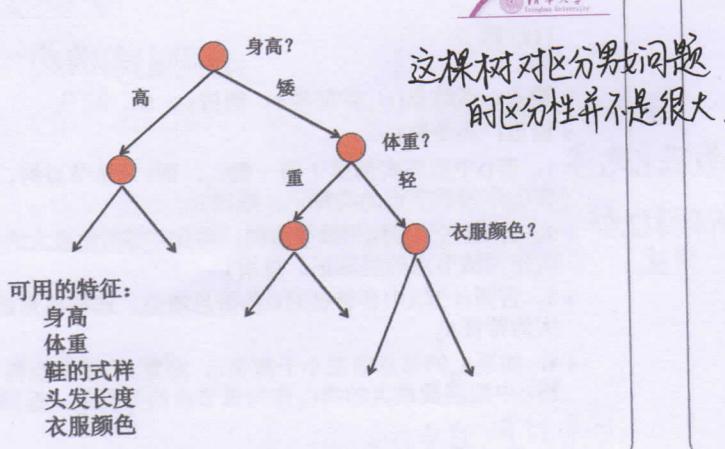
- ◆ 决策树模型是一种描述对实例进行分类的树形结构，由节点和有向边组成。节点有两种类型：内部节点和叶节点。内部节点表示一个特征或者属性，叶节点表示一个类。

可能为多叉树。

练习题

- ◆ 使用工具系统实现基于支持向量机方法的文本分类，并对比采用不同的特征、不同的超参数时，分类性能的优劣。





决策树学习

给定训练集： $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$
 其中 $x_i = (x_i^{(1)}, \dots, x_i^{(n)})$ 为输入实例， n 为特征个数，
 $y_i \in \{1, 2, \dots, K\}$ 为类标记
 $i = 1, 2, \dots, N$, N 为样本容量

- ◆ 决策树学习就是从训练集中归纳出一组分类规则，得到一个与训练集矛盾较小的决策树

- ◆ 对于给定的训练集，可以构造出多个决策树，一般以损失函数最小化作为优化目标
- ◆ 从所有决策树中选取最优决策树是一个 NPC 问题，所以一般采用启发式方法，得到一个近似解

特征个数稍微多一些，
就会出现组合爆炸的问题
——启发式

哪个效果好就用哪个，如何定义“好”？

特征选择

- ◆ 一个问题中可能有不同的特征，不同的特征具有不同的分类能力，特征选择就是如何选取出那些分类能力强的特征。
- ◆ 决策树中一般按照信息增益选择特征
- ◆ 所谓的信息增益就是某个特征 A 对数据集 D 进行分类的不确定性减少的程度

- ◆ 特征 A 对数据集 D 的信息增益定义为：

$$g(D, A) = H(D) - H(D|A) \quad (\text{熵减少了多少})$$

- ◆ 表示特征 A 对数据集 D 的分类的不确定减少的程度
- ◆ 信息增益大的特征具有更强的分类能力

决策树学习包括

- ◆ 特征选择（先用后用）
- ◆ 决策树生成
- ◆ 决策树剪枝

信息更多则猜

概率更

信息增益

评价数据的混乱程度。两类各占一半

随机变量 X 的熵：类别数目 * 标签的 (取值)

$$H(X) = -\sum_{i=1}^n p_i \log_2 p_i, \quad \text{其中 } p_i = P(X = x_i), \text{ 也记作 } H(p).$$

(=0: 纯净)

当概率由数据集 D 估计得到时，记作 $H(D)$

条件熵：这里指特征取这个值的概率

$$H(Y|X) = \sum_{i=1}^n p_i H(Y|X=x_i)$$

表示已知 X 时 Y 的不确定性

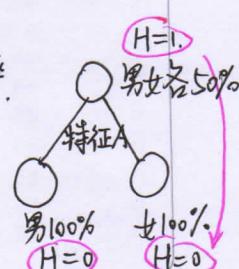
这里指特征！

$H(X)$ 的 X 表示 —— ?

随机变量取值可用 D 模拟概率

最乱：

两类各占一半



- ◆ 设训练集 D, K 个类 C_k , 特征 A 有 n 个不同的取值 $\{a_1, \dots, a_n\}$, A 的不同取值将 D 划分为 n 个子集 D_1, \dots, D_n , D_i 中属于类 C_k 的样本的集合为 D_{ik} , $| \cdot |$ 表示样本个数。

- ◆ 信息增益计算如下：

$$H(D) = -\sum_{k=1}^K \frac{|C_k|}{|D|} \log_2 \frac{|C_k|}{|D|} \quad k: \text{标签编号.}$$

$$H(D|A) = \sum_{i=1}^n \frac{|D_i|}{|D|} H(D_i) = -\sum_{i=1}^n \frac{|D_i|}{|D|} \sum_{k=1}^K \frac{|D_{ik}|}{|D_i|} \log_2 \frac{|D_{ik}|}{|D_i|}$$

$$g(D, A) = H(D) - H(D|A)$$

决策树的生成

◆两个常用的算法

◆ID3

- 一个基本的决策树生成算法

◆C4.5

- 对ID3的改进

数据已纯净

所有特征都用过了

ID3算法

- 输入：训练集D，特征集A，阈值 $\varepsilon > 0$
- 输出：决策树T
- 1. 若D中所有实例属于同一类 C_k ，则T为单节点树，将 C_k 作为该节点的类标记，返回T
- 2. 若A为空，则T为单节点树，将D中实例数最大的类 C_k 作为该节点的类标记，返回T
- 3. 否则计算A中各特征对D的信息增益，选择信息最大的特征 A_g
- 4. 如果 A_g 的信息增益小于阈值 ε ，则置T为单节点树，将D中实例数最大的类 C_k 作为该节点的类标记，返回T (同2)

在递归过程中D会

不断变为分类几次后的数据
(子节点)

- 5. 否则对 A_g 的每一可能值 a_i ，依 $A_g = a_i$ 将D分割为若干子集 D_i ，作为D的子节点

- 6. 对于D的每个子节点 D_i ，如果 D_i 为空，则将D中实例最大的类作为标记，构建子节点

- 7. 否则以 D_i 为训练集，以 $A - \{A_g\}$ 为特征集，递归地调用步1~步6，得到子树 T_i ，返回 T_i

特征的某一个
训练数据
都没有。

...但实际使用时
可能会有样本落到
这个节点嘛！

因此还要给它估一个标签——按D中的数量最多的标签来。

ID	年龄 A1	有工作 A2	有房子 A3	信贷情况 A4	类别
1	青年	否	否	一般	否
2	青年	否	否	好	否
3	青年	是	否	好	是
4	青年	是	是	一般	是
5	青年	否	否	一般	否
6	中年	否	否	一般	否
7	中年	否	否	好	否
8	中年	是	是	好	是
9	中年	否	是	非常好	是
10	中年	否	是	非常好	是
11	老年	否	是	非常好	是
12	老年	否	是	好	是
13	老年	是	否	好	是
14	老年	是	否	非常好	是
15	老年	否	否	一般	否

已纯净

A_3 作为根节点，将D划分为 $D_1(A_3 = \text{是})$ 和

$D_2(A_3 = \text{否})$ ， D_1 成为叶节点

对 D_2 从特征 A_1, A_2, A_4 中选择特征

$g(D_2, A_1) = 0.251$

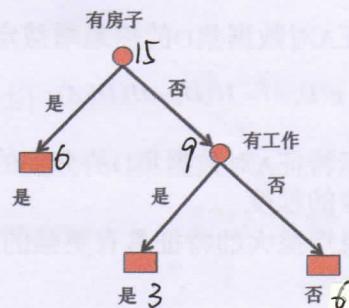
$g(D_2, A_2) = 0.918$ 信息增益最大

$g(D_2, A_4) = 0.474$

选取 A_2 作为节点的特征，根据其两个取值，可以得到两个子节点，一个对应“有工作”，并且是一个叶节点，标记类别为“是”。另一个节点对应“无工作”，且样本属于同一类，也是一个叶节点，标记类别为“否”

向下建分支后
发现两个分支均
纯净

生成的决策树如下：



小于阈值则认为没什么分类能力。
节点。

ID3存在的问题

- ◆ 信息增益倾向于选择分枝比较多的属性，就更可能在子节点获得干净的数据。
- ◆ 比如前面贷款的例子中，如果用 ID 做属性，将获得最大的信息增益值 索列编号…

信息增益比

$$g_R(D, A) = \frac{g(D, A)}{H_A(D)}$$

$$H_A(D) = -\sum_{k=1}^n \frac{|D_k|}{|D|} \log_2 \frac{|D_k|}{|D|}$$

- ◆ 其中 A 为属性，A 的不同取值将 D 划分为 n 个子集 D_1, \dots, D_n

$H(A)$: 分的类别越多熵越大。
(多分枝多的特征进行惩罚).

C4.5的生成算法

- ◆ 除了根据信息增益比选择特征外，C4.5 算法与 ID3 基本一样。

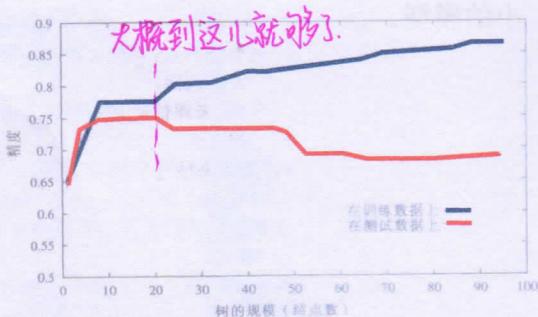
- ◆ 同时 C4.5 增加了对连续值属性的处理，对于连续值属性 A，找到一个属性值 a_0 ，将 $\leq a_0$ 的划分到左子树， $> a_0$ 的划分到右子树 区间。

ID3也可用，只是提出时没说

A 在训练集数据中有几个取值：如 1 | 3 | 6 | 8 | 9 | 12

比较现在哪 (如 6、8 之间 g_R 有 max，则 $a_0 = 7$)
带来的信息增益比最大，则 a_0 取在这一段

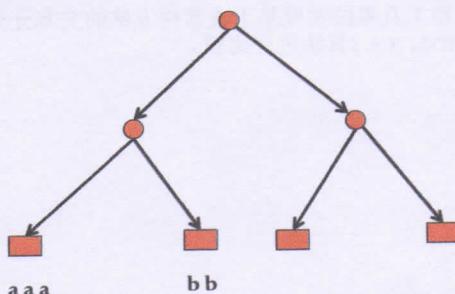
过拟合问题



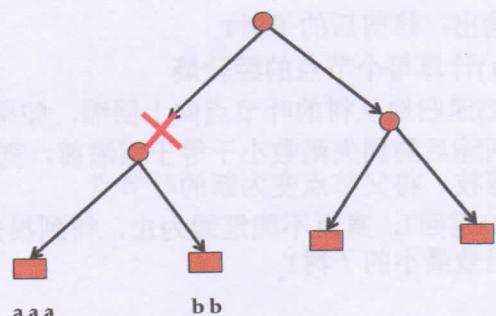
- ◆ 为了防止出现过拟合，对生成的决策树进行简化的过程称为剪枝。也就是从已经生成的树上裁掉一些子树或者叶节点，将其父节点作为新的页节点，用其实例数最大的类别作为标记。

◆ 这种先生成树再剪枝的方法称为后剪枝。

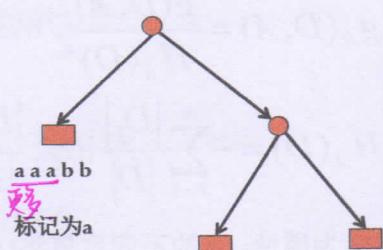
后剪枝方法示意



后剪枝方法示意



后剪枝方法示意



决策树的剪枝

当数据量大时：

- 将数据划分为训练集、验证集和测试集

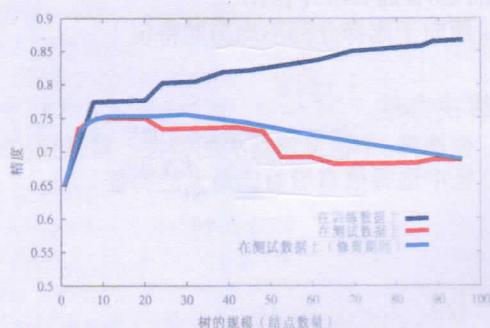
用训练集训练得到决策树

从下向上逐步剪枝

在验证集上测试性能，直到性能下降为止
之前应该剪了会提升。

最后在测试集上的性能作为系统的性能

剪枝的效果



决策树的剪枝

当数据量小时：

- 直接利用训练集进行剪枝

树T的叶节点个数为 $|T|$, t 是树T的叶节点, 该节点有 N_t 个样本, 其中k类的样本点有 N_{tk} 个 ($k=1, \dots, K$), $H_t(T)$ 为叶节点t上的经验熵, $a \geq 0$ 为参数

用训练集
(α) 超
算得的熵?
是的, 再下次哦.

定义损失函数

与误差总量有关 (数量多且熵大 → 误差总量大)

$$C_a(T) = \sum_{t=1}^{|T|} N_t H_t(T) + a|T|$$

↑
与模型复杂度有关
此消彼长
寻求 $C_a(T)$ 最小的平衡点

其中经验熵为: $H_t(T) = -\sum_k \frac{N_{tk}}{N_t} \log \frac{N_{tk}}{N_t}$

记: $C(T) = \sum_{t=1}^{|T|} N_t H_t(T) = -\sum_{t=1}^{|T|} \sum_{k=1}^K N_{tk} \log \frac{N_{tk}}{N_t}$

有: $C_a(T) = C(T) + a|T|$

$C(T)$ 表示模型对训练数据的预测误差, $|T|$ 表示模型的复杂程度

剪枝, 就是当 α 确定时, 选择损失函数最小的模型。

决策树的剪枝算法

- 输入: 生成算法产生的整个树T, 参数a
- 输出: 修剪后的子树 T_a
- (1)计算每个节点的经验熵
- (2)递归地从树的叶节点向上回缩, 如果回缩后的损失函数小于等于回缩前, 则剪枝, 将父节点变为新的叶节点
- (3)返回2, 直至不能继续为止, 得到损失函数最小的子树 T_a

练习题

- 使用工具系统实现基于决策树方法的文本分类, 并对ID3、C4.5算法进行比较。

周志华老师用它来实现深度学习.

- 随机森林 解决. 甚至不用剪枝. 可解释.
- ◆ 决策树容易过拟合
 - ◆ 随机森林是由多个决策树组成的分类器
 - ◆ 通过投票机制改善决策树 每棵树要不一样.
 - ◆ 单个决策树的生成
 - 有放回的数据采样 一靠“随机”办到.
 - 属性(特征)的采样 (不用全部特征做备选).
 - ◆ 集外数据的使用
 - 单个决策树未用到的数据 可以用来做验证集等别的东西.

小结

- ◆ 什么是统计机器学习方法?
- ◆ 朴素贝叶斯方法
- ◆ 支持向量机
 - 线性可分支持向量机
 - 线性支持向量机
 - 非线性支持向量机
- ◆ 决策树
 - ID3算法
 - C4.5算法

应用: 用DL做决策树
(从每个节点到叶节点的分类用DL).
(认为可以提升深度学习的可解释性).

关于ID3算法, 请选择以下正确的说法 B, D.

- A. 在生成决策树过程中必须使用所有的特征. (X)
- B. 允许生成的决策树叶节点实例类别不一样 (V)
- C. 同一个特征只能用在一个节点上 (X)
- D. 得到的决策树不能保证最优 (V)

因为是启发式搜索啦!

没有遍历所有选项哦~.

