决策树.

4.1 基本流程.

分治

训练集 $D=\{(x_1,y_1),(x_2,y_2),\cdots,(x_m,y_m)\}$;

属性集 $A=\{a_1,a_2,\cdots,a_d\}$.

TreeGenerate(D,A)

1. 生成结点 node;

无需划分 2. 若D中样本全属于同一类别C then

将node标为C类结点.

无法划分 3. 若A为空或D中样本在A上取值相同.

将node标记为叶结点,其类别标记为D中样本数最多的类.

4. 从A中选择最优划分属性 a_*;

从 a_* 中每一个值 a_*^v, do.

为node生成一个分支;令 D_v 表示D中在 a_* 取值为 a_*^v 的样本子集;

不能划分. 若 D_v 为空.

又结点的.

将分支结点标记为叶结点,其类别标记为D中样本最多的类.

若 D_v 非空.

A中去掉 a_*

以 TreeGenerate(D_v, $A\setminus\{a_*\}$) 为分支结点.

4.2 划分选择.

4.2.1 信息增益.

$Ent(D)=-\sum_{k=1}^{|y|}p_k\log_2 p_k$. D中第k类样本占比例为 p_k.

Ent(D) 越小,D纯度越高.

离散属性 a 有 V 个可能取值 $\{a^1, a^2, \cdots, a^V\}$. 使用 a 对样本集 D 划分, 产生 V 个分支结点. 第 V 个分支结点包含 D 中所有在属性 a 上取值为 a^V 的样本. 记为 D^V. 算出 D^V 的信息熵, 给分支结点赋权 $|D^V|/|D|$. 即样本越多, 分支结点影响越大.

信息增益 $Gain(D,a) = Ent(D) - \sum_{v=1}^{V} \frac{|D^v|}{|D|} Ent(D^v)$.

Gain 越大, 则使用属性 a 划分得到的纯度提升越大.

ID3 决策树 — Gain.

4.4.2. 增益率.

C4.5 决策树.

$$Gain\_ratio(D,a) = \frac{Gain(D,a)}{IV(a)}$$

$$IV(a) = -\sum_{v=1}^{V} \frac{|D^v|}{|D|} \log_2 \frac{|D^v|}{|D|}$$ 属性 a 的可取值.

取值数目越多, IV 越大.

4.4.3. 基尼指数.

CART 决策树.

$$Gini(D) = \sum_{k=1}^{|y|} \sum_{k' \neq k} p_k p_{k'} = 1 - \sum_{k=1}^{|y|} p_k^2$$ 越小越好.

$$Gini\_index(D,a) = \sum_{v=1}^{V} \frac{|D^v|}{|D|} Gini(D^v).$$

$$a_* = \arg\min_{a \in A} Gini\_index(D,a).$$

4.5 剪枝.

减少过拟合.

预剪枝: 生成过程中, 对每个结点在划分前进行估计. 若当前结点的划分不能带来决策树泛化性能提升, 则停止划分并将当前结点标记为叶结点.

后剪枝: 从训练集生成完整决策树. 自底向上考察. 若将该结点对应的子树替换为叶结点能带来泛化性能提升, 则将子树替换

为叶结点.

4.3.1 预剪枝. ——可能欠拟合.

使用留出法. 一半验证集. 一半训练集.

将训练集由需要剪枝的节点进行划分. D^v 为训练集该划中最多的类别. 再由训练集检测. 若子结点精度 > 父结点. 则保留结点.

4.3.2 后剪枝

从底层结点向上查看. 若剪枝后精度提高, 则剪枝

(未遵循的准则. 精度无提高也应剪枝).

缺点: 训练时间长.

4.4 连续与缺失值.

4.4.1 连续值处理.

连续属性离散化. (C4.5 决策树算法)

样本 D 和连续属性 a. 假定 a 在 D 上有 n 个不同取值. 将其从小到大排序. $\{a^1, a^2, \cdots, a^n\}$. 基于划分点 t 将 D 分为 D_t^-. D_t^+. D_t^- 包含在 a 上取值不大于 t 的样本. D_t^+ 包含属性 a 上取值大于 t 的样本. 对 a^i 和 a^{i+1} 来说, 在 $[a^i, a^{i+1})$ 中取任意值产生划分结果相同. 因此对连续属性 a, 我们考察包含 $n-1$ 个元素的候选划分集合 $T_a = \left\{ \frac{a^i + a^{i+1}}{2} \mid 1 \le i \le n-1 \right\}$.

以 $\frac{a^i + a^{i+1}}{2}$ 为划分点.

$$Gain(D, a) = \max_{t \in T_a} Gain(D, a, t) = \max_{t \in T_a} Ent(D) - \sum_{\lambda \in \{-,+\}} \frac{|D_t^\lambda|}{|D|} Ent(D_t^\lambda)$$

↓

最大化划分点.

※ 若当前划分属性为连续属性, 则后代仍可以此划分属性.

4.4.2 缺失值处理.

(1). 如何在属性值缺失的情况下进行划分属性？

$\tilde{D}$ 为 D 在属性 a 上没有缺失值的样本子集，我们可以用 $\tilde{D}$ 来判断 a 的好坏.

令 $\tilde{D}^v$ 表示 $\tilde{D}$ 中属性 a 上取值为 a^v 的样本子集. $\tilde{D}_k$ 表示 $\tilde{D}$ 中属于第 k 类 ($k=1,2,\cdots,|y|$) 的样本子集. 显然 $\tilde{D}=\bigcup_{k=1}^{|y|}\tilde{D}_k$, $\tilde{D}=\bigcup_{v=1}^{V}\tilde{D}^v$. 每一样本 x 赋予权重 w_x. 定义

无缺失样本占比例 $\rho=\dfrac{\sum_{x\in\tilde{D}} w_x}{\sum_{x\in D} w_x}$

无缺失样本k类占比 $\tilde{p}_k=\dfrac{\sum_{x\in\tilde{D}_k} w_x}{\sum_{x\in\tilde{D}} w_x}$ $(1\le k\le|y|)$

初始权重为1.

无缺失样本在属性a上取值a^v样本占比 $\tilde{r}_v=\dfrac{\sum_{x\in\tilde{D}^v} w_x}{\sum_{x\in D} w_x}$ $(1\le v\le V)$.

$\sum_{k=1}^{|y|}\tilde{p}_k=1$. $\sum_{v=1}^{V}\tilde{r}_v=1$.

$\therefore Gain(D,a)=\rho\times Gain(\tilde{D},a)$

$=\rho\times\left(Ent(\tilde{D})-\sum_{v=1}^{V}\tilde{r}_v Ent(\tilde{D}^v)\right)$.

$\downarrow$

$Ent(\tilde{D})=-\sum_{k=1}^{|y|}\tilde{p}_k\log_2\tilde{p}_k$

(2). 给定划分属性，若样本在该属性上值缺失，如何划分样本？

若 x 在划分属性 a 的取值已知，则将 x 划入其取值对应的子结点，且样本权值在子结点保持为 w_x.

若 x 在划分属性 a 的取值，则将 x 划入所有子结点，样本权值与属性值 a^v 对应的子结点调整为 $\tilde{r}_v\cdot w_x$，即样本以不同概率划到不同结点去. (第一次分类，权重 $=\dfrac{class\,n}{\sum class\,n}$).

4.5 多变量决策树. OC1. 感知机树.

决策树分类边界特点：轴平行.

若真实分类边界复杂，则使用多变量决策树. → "斜划分".

每个非叶结点是一个形如 $\sum_{i=1}^{d} w_i a_i = t$ 的线性分类器.

权重 属性. d: d个属性描述的样本.