决策树

廖星宇

2017 年 6 月 15 日

1 基本流程

决策树 (decision tree) 遵循 “分而治之” (divide-and-conquer) 策略，目的为了产生一棵泛化能力强的决策树。

三类情况会递归返回：

1. 当前节点所有样本属于同一类；
2. 当前属性集为空，无法划分，标记为当前结点最多的类；
3. 当前节点样本为空，标记为父节点最多的类

2 划分选择

2.1 信息增益

“信息熵” (information entropy) 定位为

$$Ent(D) = -\sum_{k=1}^{|y|} p_k log_2 p_k \tag{1}$$

其中 p_k 表示样本集合中 D 中第 k 类样本所占的比例，Ent(D) 的值越小，D 的纯度越高。

“信息增益” (information gain) 根据不同分支节点的样本数不同赋予不同的权重 $|D^v|/|D|$，即样本数越多的分支结点的影响越大，定义为

$$Gain(D, a) = Ent(D) - \sum_{v=1}^{V} \frac{|D^v|}{|D|} Ent(D^v) \tag{2}$$

信息增益越大，说明使用属性 a 划分获得的“纯度提升”越大，所以选择属性可以通过下面的公式来得到

$$a_* = \arg\max_{a \in A} \quad Gain(D, a) \tag{3}$$

2.2 增益率

信息增益对取值数目较多的属性有所偏好，为了减少这种偏好带来的影响，使用“增益率”(gain ratio) 来选择最优划分属性。

$$Gain\ ratio(D, a) = \frac{Gain(D, a)}{IV(a)} \tag{4}$$

其中

$$IV(a) = -\sum_{v=1}^{V} \frac{|D^v|}{|D|} log_2 \frac{|D^2|}{|D|} \tag{5}$$

需要注意的是，增益率准则对可取数目较少的属性有所偏好。

2.3 基尼指数

CART 决策树使用“基尼指数”(Gini index) 来选择划分属性，数据集 D 的纯度可用基尼值来度量

$$\begin{aligned} Gini(D) &= \sum_{k=1}^{|y|} \sum_{k' \neq k} p_k p_{k'} \\ &= 1 - \sum_{k=1}^{|y|} p_k^2 \end{aligned} \tag{6}$$

Gini(D) 反映了数据集中随机两个样本标定不一致的概率，所以 Gini(D) 越小，表示数据集 D 的纯度越高。

属性 a 的基尼指数定义为

$$Gini\ index(D, a) = \sum_{v=1}^{V} \frac{|D^v|}{|D|} Gini(D^v) \tag{7}$$

所以我们可以依据使得划分后基尼指数最小的属性作为最优划分属性，即

$$a_* = \arg\min_{a \in A} Gini\ index(D, a) \tag{8}$$

3 剪枝处理

剪纸 (pruning) 是决策树学习算法里面对付“过拟合”的主要手段，决策树为了尽可能正确分类训练样本，会将一些训练集自身的特点当做所有数据都具有的一般性质而导致过拟合，所以可以去掉一些分支来降低过拟合的风险。

常见的剪枝策略有“预剪枝” (prepruning) 和“后剪枝” (postpruning)：

（1）预剪枝实在决策树生成过程中对每个节点在划分前进行估计；

（2）后剪枝是先训练一个完整的决策树，然后自底向上对非叶节点进行考察

4 连续与缺失值

4.1 连续值处理

最简单的策略是采用二分法对连续属性进行处理，给定样本集 D 利用连续属性 a，假定 a 在 D 上出现了 n 个不同的取值，将这些信息从小到大排序，记为 $\{a^1, a^2, \cdots a^n\}$，基于划分点 t 可将 D 分为 D_t^- 和 D_t^+，分别表示在 a 上取值不大于 t 和大于 t 的样本，对于相邻属性取值 a^i 和 a^{i+1} 来说，t 在区间 $[a^i, a^{i+1})$ 中任意取值产生的划分都相同，所以对连续属性 a，我们考察包括 n-1 个元素的候选划分点集合

$$T_a = \{\frac{a^i + a^{i+1}}{2} \ |1 \leq i \leq n-1\} \tag{9}$$

将区间 $[a^i, a^{i+1})$ 的中点 $\frac{a^i+a^{i+1}}{2}$ 作为候选划分点.

$$\begin{aligned} Gain(D, a) &= \max_{t \in T_a} Gain(D, a, t) \\ &= \max_{t \in T_a} Ent(D) - \sum_{\lambda \in \{-,+\}} \frac{|D_t^\lambda|}{|D|} Ent(D_t^\lambda) \end{aligned} \tag{10}$$

4.2 缺失值处理

现实任务中会遇到不完整的样本，如果简单放弃不完整样本，对数据造成极大地浪费。

我们需要解决两个问题：（1）如何在属性值确实的情况下进行划分属性选择？（2）给定划分属性，如样本在该属性上的值缺失，如何对样本进行划分？

给定训练集 D 和属性 a，令 $\tilde{D}$ 表示 D 中在属性 a 上没有缺失值的样本子集，假定 a 有 V 个可取值 $\{a^1, a^2, \cdots, a^V\}$，令 $\tilde{D}^v$ 表示 $\tilde{D}$ 中在属性 a 上取值为 a^v 的样本子集，$\tilde{D}_k$ 表示 $\tilde{D}$ 中属于第 k 类的样本子集，假定每个样本 x 赋予一个权重 w_x，并定义

$$\rho = \frac{\sum_{x \in \tilde{D}} w_x}{\sum_{x \in D} w_x} \tag{11}$$

$$\tilde{\rho_k} = \frac{\sum_{x \in \tilde{D}_k} w_x}{\sum_{x \in \tilde{D}} w_x} \tag{12}$$

$$\rho = \frac{\sum_{x \in \tilde{D}} w_x}{\sum_{x \in D} w_x} \tag{13}$$

我们可以将信息增益的计算式推广为

$$\begin{aligned} Gain(D, a) &= \rho \times Gain(\tilde{D}, a) \\ &= \rho \times (Ent(\tilde{D}) - \sum_{v=1}^{V} \tilde{r}_v Ent(\tilde{D}^v)) \end{aligned} \tag{14}$$

若样本 x 在划分属性 a 上的取值已知，直接将其划入对应子结点，且样本权值在子结点保持 w_x. 若样本 x 在划分属性 a 上取值未知，将 x 同时划入所有子结点，且样本权重在与属性值 a^v 对应的子结点中调整为 $\tilde{r}_v \cdot w_x$