Author：Liu Jian

Time：2020-01-03

机器学习2-决策树

数据集记为 D ，属性的集合 $A=\{a_1,a_2,\cdots,a_n\}$ ，属性 a_i 可能的取值个数为 v_i ，即 a_i 的可能取值为 $\{a_i^1,a_i^2,\cdots,a_i^{v_i}\}$ 。样本标记空间为 $\mathcal{Y}=\{y_1,y_2,\cdots,y_m\}$ ，即样本标记的可能取值有 $m=|\mathcal{Y}|$ 个。决策树的目的是得到一颗树，能根据属性判定标记。

以标记 y 为研究对象，考察每种标记在数据集 D 中出现的频率，即得数据集 D 中标记 y 的概率分布。根据标记 y 的概率分布，引入信息熵 $H_D(y)$ 表征数据集 D 中标记 y 的不确定性。熵越大，数据集中标记 y 的不确定性越大，数据集中标记 y 的纯度就越低。

类似地，以某一属性 a_i 为研究对象，考察属性 a_i 各种可能取值出现的频率，即得数据集 D 中属性 a_i 的概率分布。根据属性 a_i 的概率分布，引入信息熵 $H_D(a_i)$ 表征数据集 D 中属性 a_i 的不确定性。

1. ID3决策树学习算法

属性选择策略： 在属性集中选择使信息增益最大的那个属性。

信息增益：

$$Gain(D,a)=H_D(y)-H_D(y|a)$$

即在数据集 D 中，标记 y 的信息熵减去给定属性 a 后 y 的条件熵，反映了给定属性 a 后标记 y 取值不确定性下降的程度，或者说纯度上升的程度。

算法性质： 偏好选择可取值数目较多（ v_i 较大）的属性，会得到庞大且浅的树，会存在过拟合的现象导致泛化性能低。

2. C4.5决策树学习算法

是对ID3决策树算法偏好选择可取值数目较多的属性的一种改进。在信息增益的基础上，进一步引入了**增益率** 这一指标：

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

其中，在数据集 D 中，属性 a 的信息熵 $H_D(a)$ 又称为属性 a 的固有值(intrinsic value)。由信息熵的性质可知，一般地，属性 a 的可取值个数越多， $H_D(a)$ 越大。

属性选择策略： 若只采用增益率作为为属性选择的指标，则会对可取值数目较少的属性有所偏好。C4.5算法采用了一种启发式策略：**先从候选划分属性中找出信息增益高于平均水平的属性，再从中选择增益率最高的。**

3. CART决策树算法

CART决策树采用基尼指数来选择划分属性，既可以用于分类，又可以用于回归。

3.1 分类算法

数据集 D 中，标记 y 的**基尼值**：

$$Gini(D) = \sum_{j=1}^{m} \sum_{j \neq j'} p(y_j)p(y_{j'}) = 1 - \sum_{j=1}^{m} p_j^2$$

基尼值反应了从数据集 D 中，随机抽取两个样本，其标记 y 不一致的概率。基尼值越小，数据集 D 中，标记 y 的纯度越高。

数据集 D 中，属性 a 对标记 y 的**基尼系数**：

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

选定属性 a ，根据属性 a 的所有 v 种可能取值对数据集 D 进行划分，得到 v 个不相交的子集。计算每个子集 D^j 中，标记 y 的基尼值，并加权平均，得基尼系数。属性 a 对应的基尼系数越小，表明采用属性 a 对数据集 D 进行划分后，标记 y 的平均纯度越高。

属性选择策略：
选择对应基尼系数最小的属性：

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

3.2 回归算法

CART决策树回归算法：

1. 根据以下公式找出最优划分特征a_*和最优划分点a_*^v：

$$a_*, a_*^v = \arg\min_{a, a^v} \left[\min_{c_1} \sum_{x_i \in D_1(a, a^v)} (y_i - c_1)^2 - \min_{c_2} \sum_{x_i \in D_2(a, a^v)} (y_i - c_2)^2 \right]$$

   其中，$D_1(a, a^v)$表示在属性a上取值小于等于a^v的样本集合，$D_2(a, a^v)$表示在属性a上取值大于a^v的样本集合，c_1表示D_1的样本输出均值，c_2表示D_2的样本输出均值。

2. 根据划分点a_*^v将集合D划分为D_1和D_2两个集合 (节点)；
3. 对集合D_1和D_2重复步骤1和步骤2，直至满足停止条件。

上述三种决策树生成算法中，只阐明了决策树每一层的生成中如何选择划分属性，整棵树的生成过程是一个迭代的过程。

因为假设每个属性的取值都是离散的，所以每个属性在从根节点到叶节点的路径中最多只出现一次。

---

4. 决策树剪枝

对决策树进行剪枝地目的是为了防止过拟合。剪枝就是判断是否使用某个属性进行划分，判断依据是属性划分前后的决策树在验证集上精度是否有提升。存在预剪枝和后剪枝两种方法：

- 预剪枝：在决策树生成过程中自顶向下进行，开销小，是一种贪心算法，因而存在欠拟合的风险。
- 后剪枝：在决策树生成后自底向上进行，开销大，欠拟合风险小，泛化性能好。

《统计学习方法》中给出了关于剪枝的几种稍微复杂的算法，可供参考。

5. 其他问题

- 连续值的处理
- 缺失值的处理
- 多变量决策树/斜决策树