决策树剪枝

思想：

决策树的剪枝是通过极小化决策树整体的损失函数。(决策树的生成只考虑局部最优，决策树的剪枝考虑全局最优)

eg.

设树T的叶节点为 t，个数为 |T|，该叶节点有 Nt 个样本点，其中 k 类的样本点有 Ntk 个，k = 1,2,...,K，Ht(T)为叶节点 t 上的经验熵，α≥0为参数，则决策树的损失函数：

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

其中特征熵Ht(T)：

$$H_t(T) = -\sum_k \frac{N_{tk}}{N_t} \log \frac{N_{tk}}{N_t}$$

在损失函数中，式子右端的第一项记作：

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

这时损失函数为：

$$C_\alpha(T) = C(T) + \alpha|T|$$

C(T) 表示模型对训练数据的预测误差，即模型与训练数据拟合程度，|T| 表示模型复杂度，参数 α≥0控制两者之间的影响。较大的α促使选择简单的模型（树），较小的α促使选择复杂的模型（树），α=0只考虑模型与训练数据的拟合程度，不考虑模型的复杂度。

当α确定时，子树越大，与训练数据拟合越好，但模型复杂度越高；相反，子树越小，与训练数据拟合不好，但模型复杂度低。

上面两个决策树损失函数的极小化等价于正则化的极大似然估计。所以，利用损失函数最小原则进行剪枝就是用正则化的极大似然估计进行模型选择。

算法过程

输入：生成树T，参数α

输出：剪之后的子树T

1.计算每个节点的**特征*****熵****。

2.递归地从树的叶节点向上**回缩**。假设回缩之前和之后的树为TB，TA，对应的损失函数为Cα(TB)与Cα(TA)，如果Cα(TB)≤Cα(TA)，则进行剪枝，即父节点为新的叶节点。

3.返回2，直到不能继续为止，得到**损失函数最小**的子树Tα。