Cost-Complexity Pruning Process

Assuming a CART or QUEST tree has been grown successfully using a learning sample, this document describes the automatic cost-complexity pruning process for both CART and QUEST trees. Materials in this document are based on *Classification and Regression Trees* by Breiman et al (1984). Calculations of the risk estimates used throughout this document are given in “Assignment and Risk Estimation” (TREE-assignment-risk.pdf).

Cost-Complexity Risk of a Tree *T*

Given a tree T and a real number α, the cost-complexity risk of T with respect to α is

$$R_\alpha(T) = R(T) + \alpha|\tilde{T}|,$$

where $|\tilde{T}|$ is the number of terminal nodes and $R(T)$ is the resubstitution risk estimate of T.

Smallest Optimally Pruned Subtree

Pruned subtree: For any tree T, T' is a pruned subtree of T if T' is a tree with the same root node as T and all nodes of T' are also nodes of T. Denote $T' \preceq T$ if T' is a pruned subtree of T.

Optimally pruned subtree: Given α, a pruned subtree T' of T is called an optimally pruned subtree of T with respect to α if $R_\alpha(T') = \min_{T'' \preceq T} R_\alpha(T'')$. The optimally pruned subtree may not be unique.

Smallest optimally pruned subtree: If $T' \preceq T''$ for any optimally pruned subtree $T'' \preceq T_0$ such that $R_\alpha(T') = R_\alpha(T'')$, then T' is the smallest optimally pruned subtree of T_0 with respect to α, and is denoted by $T_0(\alpha)$.

Cost-Complexity Pruning Process

Suppose that a tree T_0 was grown. The cost-complexity pruning process consists of two steps:

1. Based on the **learning sample**, find a sequence of pruned subtrees $\{T_k\}_{k=0}^{K}$ of T_0 such that $T_0 \succ T_1 \succ T_2 \succ \ldots \succ T_K$, where T_K has only the root node of T_0.
2. Find an “honest” risk estimate $\hat{R}(T_k)$ of each subtree. Select a right sized tree from the sequence of pruned subtrees.

Generate a sequence of smallest optimally pruned subtrees

To generate a sequence of pruned subtrees in step 1, the cost-complexity pruning technique developed by Breiman et. al. (1984) is used. In generating the sequence of subtrees, only the learning sample is used. Given any real value $\alpha_{\min}$ ($\alpha_{\min} = 0$ in any SPSS implementation) and an initial tree T_0, there exists a sequence of real values $-\infty < \alpha_1 = \alpha_{\min} < \alpha_2 < \cdots < \alpha_K < +\infty$ and a sequence of pruned subtrees $T_0 \succ T_1 \succ \cdots \succ T_K$, such that the smallest optimally pruned subtree of T_0 for a given α is

$$T_0(\alpha) = \begin{cases} T_0 & \alpha < \alpha_1 \\ T_0(\alpha_k) = T_k & \alpha_k \le \alpha < \alpha_{k+1} \quad 1 \le k < K \\ T_0(\alpha_K) = T_K & \alpha_K \le \alpha \end{cases},$$

where

$$\alpha_{k+1} = \min_{t \in T_k} g_k(t)\,,\;\; T_{k+1} = \{t \in T_k : g_k(s) > \alpha_{k+1} \text{ for all ancestors of } t\},$$

$$g_k(t) = \begin{cases} \dfrac{R(t) - R(T_{k,t})}{|\tilde{T}_{k,t}| - 1} & t \in T_k - \tilde{T}_k \\ +\infty & t \in \tilde{T}_k \end{cases},$$

$\tilde{T}_{k,t}$ is the branch of T_k stemming from node *t*, and *R(t)* is the resubstitution risk estimate of node *t* based on the learning sample.

Explicit algorithm

The algorithm can be used to generate a sequence of subtrees of T_0 for a given initial value $\alpha = \alpha_{\min}$, and an initial tree T_0 = {1, …, #T_0} where #T_0 is the number of nodes in T_0. For node *t*, let

$$lt(t) = \begin{cases} 0 & t \text{ is terminal} \\ \text{left child of } t & \text{otherwise} \end{cases},\quad rt(t) = \begin{cases} 0 & t \text{ is terminal} \\ \text{right child of } t & \text{otherwise} \end{cases},$$

$$pa(t) = \begin{cases} 0 & t \text{ is root node} \\ \text{parent of } t & \text{otherwise} \end{cases}$$

$$\tilde{N}(t) = \begin{cases} 1 & t \text{ is terminal} \\ |\tilde{T}_{k,t}| & \text{otherwise} \end{cases}, \quad S(t) = \begin{cases} R(t) & t \text{ is terminal} \\ R(T_{k,t}) & \text{otherwise} \end{cases},$$

$$G(t) = \min_{s \in T_{k,t}} g_k(s).$$

The explicit algorithm is shown below.

1. Set k = 1, $\alpha = \alpha_{\min}$.

   For $t = \#T_0$ to 1 {

   if t is a terminal node, set

   $\tilde{N}(t) = 1$, $S(t) = R(t)$, $g(t) = +\infty$, $G(t) = +\infty$,

   else (i.e., if t is not a terminal node), set

   $\tilde{N}(t) = \tilde{N}(lt(t)) + \tilde{N}(rt(t))$

   $S(t) = S(lt(t)) + S(rt(t))$

   $g(t) = (R(t) - S(t))/(\tilde{N}(t) - 1)$

   $G(t) = \min\{g(t), G(lt(t)), G(rt(t))\}$

   }

2. If $G(1) > \alpha$,

   $$\alpha_k = \alpha \text{ and } T_k = \{t \in T_{k-1} : g(s) > \alpha_k \text{ for all ancestor s of t}\}.$$

   $$\alpha = G(1),\ k = k + 1.$$

   Else

   if $\tilde{N}(1) = 1$, terminate this process.

3. Set t = 1.

   While $G(t) < g(t)$, $t = \begin{cases} lt(t) & G(t) = G(lt(t)) \\ rt(t) & otherwise \end{cases}$.

4. Make current node t terminal by setting

   $\tilde{N}(t) = 1$, S(t) = R(t), g(t) = $+\infty$, G(t) = $+\infty$.

5. Update ancestor's information of current node t.

   While $t > 1$ (i.e. t is not the root node) {

   $t = pa(t)$

   $\tilde{N}(t) = \tilde{N}(lt(t)) + \tilde{N}(rt(t))$

   $S(t) = S(lt(t)) + S(rt(t))$

   $g(t) = (R(t) - S(t))/(\tilde{N}(t) - 1)$

   $G(t) = \min\{g(t), G(lt(t)), G(rt(t))\}$

   }

6. Then repeat steps 2 to 6 until the termination condition $\tilde{N}(1) = 1$ in Step 2 is satisfied.

Selecting the Right Sized Subtree

To select the right sized pruned subtree from the sequence of pruned subtrees $\{T_k\}_{k=0}^{K}$ of T_0, an "honest" method is used to estimate the risk $\hat{R}(T_k)$ and its standard error $se\left(\hat{R}(T_k)\right)$ of each subtree T_k. Two methods can be used: the resubstitution estimation method and the test sample estimation method. Resubstitution estimation is used if there is no test sample. Test sample estimation is used if there is a testing sample. Select the subtree T_{k^*} as the right sized subtree of T_0 based on one of the following rules.

Simple rule

The right sized tree is selected as the $k^* \in \{0, 1, 2, \ldots, K\}$ such that

$$\hat{R}(T_{k^*}) = \min_k \hat{R}(T_k) .$$

The b-SE rule

For any nonnegative real value b (default $b = 1$), the right sized tree is selected as the largest $k^{**} \in \{0, 1, 2, \ldots, K\}$ such that

$$\hat{R}(T_{k^{**}}) \leq \hat{R}(T_{k^*}) + b \cdot se\left(\hat{R}(T_{k^*})\right).$$

References

Breiman, L., Friedman, J.H., Olshen, R., and Stone, C.J., 1984. *Classification and Regression Trees* Wadsworth & Brooks/Cole Advanced Books & Software, Pacific California.