$$R_\alpha(T) = R(T) + \alpha|T|$$

Some Facts

- For a pair of leaf nodes (t_L, t_R), there exists α^*, such that

  1. for any $\alpha \geq \alpha^*$, we would like to collapse them to just node t;
  2. for any $\alpha < \alpha^*$, keep the two leaf nodes.

  That is, α^* is the maximal price we would like to pay to keep that split.

  Next we extend this calculation to compute the maximal price we would like to pay to keep a branch T_t.

- For any non-leaf node t, do the following calculation to find out the maximal price we'd like to pay for keeping the whole branch T_t.

  Focus only on samples at node t.

  - Cost for keeping branch T_t: $R_\alpha(T_t) = R(T_t) + \alpha|T_t|$
  - Cost for cutting branch T_t: $R_\alpha(\{t\}) = R(\{t\}) + \alpha$
  - Calculate

$$\alpha^* = \frac{R(\{t\}) - R(T_t)}{|T_t| - 1}.$$

  That is, if the given $\alpha > \alpha^*$, then it is too expensive to keep this branch and we would like to cut the whole branch and make t a leaf node.

Weakest-Link Pruning

The weakest-link pruning algorithm.

- Start with $T_0 = T_{\max}$ and $\alpha_0 = 0$.
- For any non-leaf node t, denote the maximal price we'd like to pay to keep T_t by $\alpha(t)$.
- $\alpha_1 = \min_t \alpha(t)$. The corresponding (non-terminal node) t_1 is called the weakest link. Cut the branch at t_1.
- Next update the maximal price for each non-leaf node (we only need to recompute the maximal price for nodes that are parents/grandparents of t_1). Find α_2 and cut the branch at the 2nd weakest link. Keep doing this until we get to the root.

The steps above generate a Solution Path:

$$T_{\mathsf{max}} = T_0 \succ T^*(\alpha_1) \succ T^*(\alpha_2) \succ \cdots \succ \{\text{root node}\}$$

$$0 = \alpha_0 < \alpha_1 < \alpha_2 < \cdots$$

All possible values of α are grouped into $(m+1)$ intervals:

$$\begin{aligned} I_0 &= [0, \alpha_1) \\ I_1 &= [\alpha_1, \alpha_2) \\ &\vdots \\ I_m &= [\alpha_m, \infty) \end{aligned}$$

where all $\alpha \in I_i$ share the same optimal subtree $T^*(\alpha_i)$.