CS 4780/5780 Homework 8

Due: Thursday 29/11/18 11:55pm on Gradescope

Problem 1: Trade-off Between Impurity and Tree Size for Regression Trees

You are given a dataset $D = \{(-3, -20), (-2, -20), (-1, -17), (0, 15), (1, 25), (2, 26)\}$ and you want to build a regression tree for this dataset.

a Recall that the impurity for the regression tree model is defined as

$$L(S) = \frac{1}{|S|} \sum_{(x_i, y_i) \in S} (y_i - \bar{y}_S)^2,$$

where $\bar{y}_S = \frac{1}{|S|} \sum_{(x_i, y_i) \in S} y_i$. Draw the regression tree T_0 built by the ID3-Algorithm which was introduced in class.(There are multiple correct thresholds. Choose one of them to draw.)

b **Notation.** Let us first introduce some notation for a regress tree T (See Figure 1):

- Terminal node V_m: the m^{th} node we stop to split.
- Region R_m: the region of x defined by the path from root to terminal node V_m.
- $|T|$: the number of leaf nodes in the tree T.
- N_m: $|\{(x, y) \in D : x \in R_m\}|$.
- $L_m(T)$: is the impurity of $\{(x, y) \in D : x \in R_m\}$.
- Subtree T': a subtree $T' \subseteq T$ is any tree that can be obtained by pruning T, that is, collapsing any number of its internal (non-terminal) nodes. For example the tree in Figure 2 is a subtree of the tree in Figure 2.

Criterion. One way to regulate the bias variance trade-off in regression trees is to limit the number of leaf nodes $|T|$. If $|T| = n$, the bias of a classifier will be 0, but the variance will be very high. Conversely, if the number of leaf nodes is small, $|T| \ll n$, the tree will have low variance but suffer from high bias.

To find the right tradeoff between bias and variance, we define the cost complexity criterion for tree T:

$$C_\alpha(T) = \sum_{m=1}^{|T|} N_m L_m(T) + \alpha |T|,$$

where $\alpha \geq 0$ is the tuning parameter regulating the tradeoff between bias and variance. Note, this is very similar to regularization in empirical risk minimization.

We would like to find $\min_T C_\alpha(T)$. One complication is that trees are myopic, that means sometimes splits do not decrease the loss, but increase $|T|$. Such splits strictly *increase* $C_\alpha(T)$ but are necessary to get a low value at a later stage. So instead of optimization C_α directly, a common strategy is to first build a full tree T_0 (which minimizes C_0) and then prune it back to optimize C_α for some given $\alpha > 0$.

Weakest Link. The *weakest link* pruning procedure is one effective strategy to prune a tree:

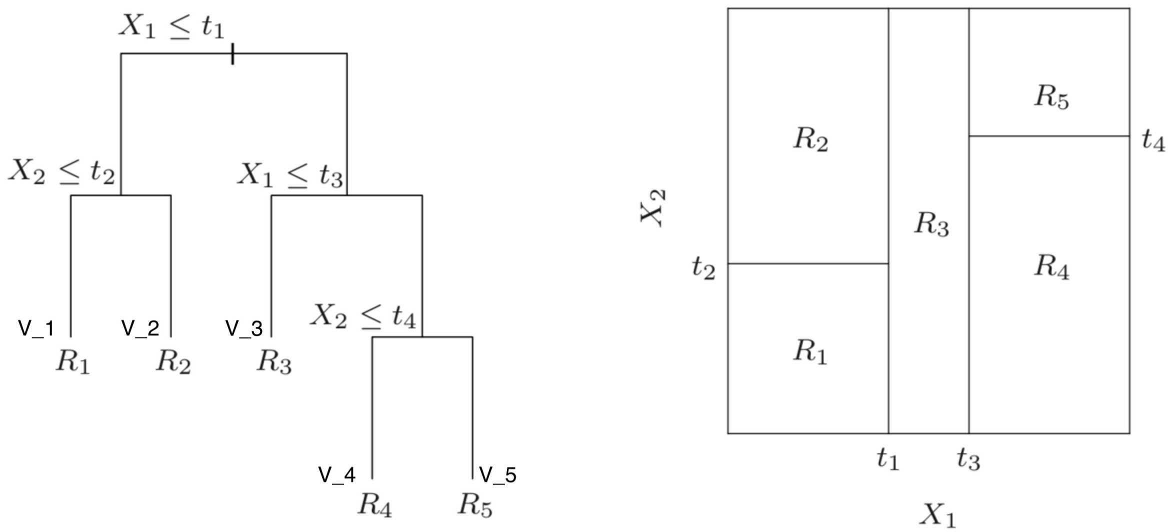

Figure 1: An example for Regression Tree

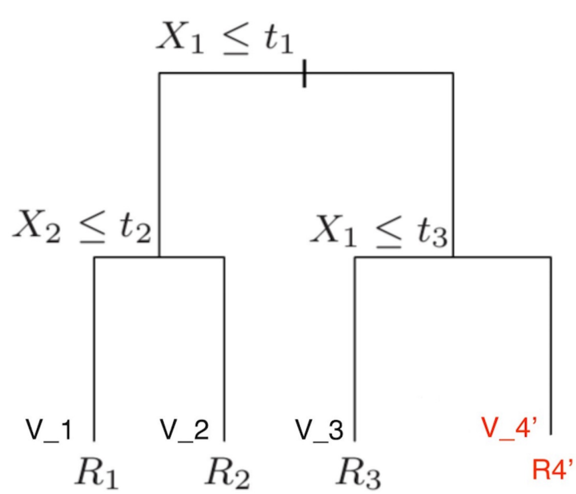

Figure 2: An example for Subtree

- successively collapse the internal node that produces the smallest per-node increase in $\sum_{m=1}^{|T|} N_m L_m(T)$.
- continue until we produce the single-node (root) tree.

Existed Results. You can use the following results directly without proof: for each α, there is a unique smallest subtree $T_\alpha \subseteq T_0$ that minimizes $C_\alpha(T)$. Moreover, the sequence of subtrees obtained by pruning under the *weakest link*, must contain T_α.

Problem. Please find the $T_\alpha \subseteq T_0$ with $\alpha = \frac{1}{2}$, where the tree T_0 is what you computed in (a).

Problem 2: Normalization Update in Adaboost

In the Adaboost, we keep $\sum_{i=1}^{n} w_t^i = 1$. In the iteration t of the algorithm, we update w_t^i as follow:

$$w_{t+1}^i \leftarrow \frac{w_t^i \cdot e^{-\alpha_{t+1} h_{t+1}(x_i) y_i}}{2\sqrt{\epsilon_{t+1}(1-\epsilon_{t+1})}}$$

where $\alpha_{t+1} = \frac{1}{2} \log\left(\frac{1-\epsilon_{t+1}}{\epsilon_{t+1}}\right)$ and $\epsilon_{t+1} = \sum_{i:h_{t+1}(x_i) \neq y_i} w_t^i$. Prove that if $\sum_{i=1}^{n} w_t^i = 1$, $\sum_{i=1}^{n} w_{t+1}^i = 1$, i.e. $\sum_{i=1}^{n} w_t^i \cdot e^{-\alpha_{t+1} h_{t+1}(x_i) y_i} = 2\sqrt{\epsilon_{t+1}(1-\epsilon_{t+1})}$. (Remember in the Adaboost, $h_{t+1}(x_i), y_i \in \{+1, -1\}$.)