Trees

Suppose our data consists of p inputs and a response, for each of N observations: $(\vec{X}_i, y_i)$ for $i=1,\dots,N$, with $\vec{X}_i = (X_{i1}, X_{i2}, \dots, X_{ip})$.
To build a tree, we need an algorithm to automatically decide on (1) the splitting variables, (2) split points, and (3) what topology (shape) of the tree. Suppose we have a partition into M regions $R_1, R_2, \dots, R_m$, and we model the response as a constant c_m in R_m

$$f(x) = \sum_{m=1}^{M} c_m \mathbb{1}_{\{x \in R_m\}}$$

Let $Y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$ $Z = \begin{pmatrix} \mathbb{1}_{\{X_1 \in R_1\}} & \cdots & \mathbb{1}_{\{X_1 \in R_M\}} \\ \vdots & & \\ \mathbb{1}_{\{X_n \in R_1\}} & \cdots & \mathbb{1}_{\{X_n \in R_M\}} \end{pmatrix}$, then the least squares estimator of $\vec{C} = \begin{pmatrix} c_1 \\ \vdots \\ c_M \end{pmatrix}$ is $\hat{C} = (Z^T Z)^{-1} Z^T Y$. Note that

$$(\mathbb{1}_{\{X_1 \in R_i\}} \cdots \mathbb{1}_{\{X_n \in R_i\}}) \begin{pmatrix} \mathbb{1}_{\{X_1 \in R_j\}} \\ \vdots \\ \mathbb{1}_{\{X_n \in R_j\}} \end{pmatrix} = \begin{cases} 0 & \text{if } i \neq j \\ N_m & \text{if } i = j \end{cases} \quad \therefore (Z^T Z) = \begin{pmatrix} n_1 & & 0 \\ & \ddots & \\ 0 & & n_M \end{pmatrix}$$

where N_m is the number of $\vec{X}_i$ in R_m.

$$\therefore \hat{c}_m = \frac{1}{N_m} (\mathbb{1}_{\{X_1 \in R_m\}} \cdots \mathbb{1}_{\{X_n \in R_m\}}) \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} = \frac{1}{N_m} \sum_{\vec{X}_i \in R_m} y_i = \text{avg}(y_i \mid \vec{X}_i \in R_m)$$

Choosing $R_1, \dots, R_M$ to minimize $\|Y - Z\hat{C}\|_2^2$ is generally infeasible. Hence we proceed with a greedy algorithm. For a splitting variable j and split point s, define

$$R_1(j,s) = \{X \mid X_j \le s\} \quad \text{and} \quad R_2(j,s) = \{X \mid X_j > s\}$$

We want to find j and s such that

$$\min_{c_1} \sum_{\vec{X}_i \in R_1(j,s)} (y_i - c_1)^2 + \min_{c_2} \sum_{\vec{X}_i \in R_2(j,s)} (y_i - c_2)^2$$

$$= \sum_{\vec{X}_i \in R_1(j,s)} (y_i - \hat{c}_1)^2 + \sum_{\vec{X}_i \in R_2(j,s)} (y_i - \hat{c}_2)^2 \quad \text{where } \hat{c}_K = \text{avg}(y_i \mid \vec{X}_i \in R_K(j,s))$$

to be minimum. It can be done by checking all variable j's and

all the mid-points between two adjacent points in jth variable.
Having found the best split, we partition the data into the two resulting regions and repeat the splitting process on each of the two regions.

When should we stop the splitting process (growing a tree)?
We first grow a large tree T_0, stopping the splitting process only when some minimum node size (say 5) is reached.
Let a subtree $T \subset T_0$ be any tree that can be obtained by pruning T_0, that is, collapsing any number of its internal (non-terminal) nodes. We index terminal nodes by m, with node m representing region R_m. Let $|T|$ denote the number of terminal nodes in T and

$$N_m = \# \{X_i \in R_m\} \qquad \hat{c}_m = \frac{1}{N_m} \sum_{X_i \in R_m} y_i$$

$$Q_m(T) = \frac{1}{N_m} \sum_{X_i \in R_m} (y_i - \hat{c}_m)^2$$

We want to find the subtree $T_\alpha \subseteq T_0$ to minimize

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

Weakest link pruning
1. Start from T_0, successively collapse the internal node that produces the smallest increase in $\frac{1}{|T|} \sum_{m=1}^{|T|} N_m Q_m(T)$
2. Continue until the single-node (root) tree is produced. This gives a sequence of subtrees.
3. Choose the subtree, T_α, that minimize $C_\alpha(T)$

α can be estimated by 5 or 10 fold cross-validation

The error measure $Q_m(T)$ can be changed for different problems.

For regression (y_i's are continuous), we use $Q_m(T) = \frac{1}{N_m}\sum_{x_i \in R_m}(y_i - \hat{C}_m)^2$ but it is not suitable for classification ($y_i \in \{1,2,\dots,K\}$).

Let $\hat{p}_{mk} = \frac{1}{N_m}\sum_{x_i \in R_m} \mathbb{1}\{y_i = k\}$ be the proportion of class k observations in node m.

In R_m, observation is classified to be $k(m) = \arg\max_k \hat{p}_{mk}$

Different choices of $Q_m(T)$ are :

Misclassification error : $\frac{1}{N_m}\sum_{i \in R_m} \mathbb{1}\{y_i \neq k(m)\} = 1 - \hat{p}_{mk(m)}$

Gini index : $\sum_{k=1}^{K} \hat{p}_{mk}(1-\hat{p}_{mk}) = \sum_{k \neq \tilde{k}} \hat{p}_{mk}\hat{p}_{m\tilde{k}} = \sum_{k=1}^{K} \hat{p}_{mk}\left(\sum_{\tilde{k} \neq k} \hat{p}_{m\tilde{k}}\right)$

Cross-entropy or deviance : $-\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}$

For two classes, if p is the proportion in the second class, the three measures are $1-\max(p, 1-p)$, $2p(1-p)$ and $-p\log p - (1-p)\log(1-p)$

Cross-entropy and Gini index are more sensitive to sensitive to changes in the node probabilities.

Consider a two-class problem with 400 observations in each class (denote this by (400, 400)), suppose one split created nodes (300, 100) and (100, 300). In this case, $N_1 = 400$, $\hat{p}_1 = \frac{3}{4}$, $N_2 = 400$, $\hat{p}_2 = \frac{3}{4}$

Consider $\sum_{m=1}^{|T|} N_m Q_m(T) = N_1 Q_1(T) + N_2 Q_2(T)$ for different measures.

Misclassification : $400(\frac{1}{4}) + 400(\frac{1}{4}) = 200$

Gini index : $400(\frac{1}{4})(\frac{3}{4})(2) + 400(2)(\frac{1}{4})(\frac{3}{4}) = 300$

Cross-entropy: $400(-\frac{1}{4}\log\frac{1}{4} - \frac{3}{4}\log\frac{3}{4}) \times 2 = 449.9$

On the other hand, suppose other created nodes (200, 400) and (200, 0)

This case is more preferable as (200, 0) is a pure node

Again consider $N_1 Q_1(T) + N_2 Q_2(T)$ ($N_1 = 600$, $\hat{p}_1 = \frac{2}{3}$, $N_2 = 200$, $\hat{p}_2 = 1$)

Misclassification : $600(\frac{1}{3}) + 200(0) = 200$

Gini index : $600(\frac{1}{3})(\frac{2}{3})(2) + 0 = 266.7$

Cross-entropy : $600(-\frac{1}{3}\log\frac{1}{3} - \frac{2}{3}\log\frac{2}{3}) + 0 = 381.9$

Therefore the Gini index and cross-entropy report a lower $C_\alpha(T)$ for the second case ($\Rightarrow$ second spit is better), but misclassification can't distinguish. For this reason, either the Gini index or cross-entropy should be used when growing the tree.

We have been assuming that regions can be separated by $X_j \leq s$ and $X_j > s$. What if the jth entry of $\vec{x}$ is categorical (no natural ordering)?

In general, when splitting a predictor having q possible unordered values, there are $2^{q-1}-1$ possible partitions of the q values into 2 groups, and the computations become prohibitive for large q.

The computation can be simplified if $y_i \in \{0,1\}$

1. order the predictor classes according to the proportion falling in outcome class 1.
2. split this predictor as if it were an ordered predictor.

One can show that this gives the optimal split, in terms of cross-entropy or Gini index, among all $2^{q-1}-1$ splits.

The result also holds for y_i continuous and square error loss. The categories are ordered by increasing mean of the outcome.

One major problem with trees is their high variance. Often a small change in the data can result in a very different series of splits.

This problem can be reduced by Bagging, which averages many trees to reduce the variance.