CART Algorithm

This document describes the tree growing process of the CART algorithm. The algorithm is based on *Classification and Regression Trees* by Breiman et al (1984). A CART tree is a binary decision tree that is constructed by splitting a node into two child nodes repeatedly, beginning with the root node that contains the whole learning sample.

Notations

Y	The dependent variable, or target variable. It can be ordinal categorical, nominal categorical or continuous. If Y is categorical with *J* classes, its class takes values in C = {1, …, *J*}.
X_m, m = 1, …, M	The set of all predictor variables. A predictor can be ordinal categorical, nominal categorical or continuous.
$\hbar = \{\mathbf{x}_n, y_n\}_{n=1}^{N}$	The whole learning sample.
$\hbar(t)$	The learning samples that fall in node *t*.
w_n	The case weight associated with case *n*.
f_n	The frequency weight associated with case *n*. Non-integral positive value is rounded to its nearest integer.
$\pi(j)$, j = 1, …, J	Prior probability of $Y = j$, j = 1, …, J.
$p(j,t)$, j = 1, …, J	The probability of a case in class *j* and node *t*.
$p(t)$	The probability of a case in node *t*.
$p(j \mid t)$, j = 1, …, J	The probability of a case in class *j* given that it falls into node *t*.
$C(i \mid j)$	The cost of miss-classifying a class *j* case as a class *i* case. Clearly $C(j \mid j) = 0$.

Tree Growing Process

The basic idea of tree growing is to choose a split among all the possible splits at each node so that the resulting child nodes are the "purest". In this algorithm, only univariate splits are considered. That is, each split depends on the value of only one predictor variable. All possible splits consist of possible splits of each predictor. If *X* is a *nominal categorical* variable of *I* categories, there are $2^{I-1}-1$ possible splits for this predictor. If *X* is an *ordinal categorical* or *continuous* variable with K different values, there are *K* - 1 different splits on *X*. A tree is grown starting from the root node by repeatedly using the following steps on each node.

1. Find each predictor's best split.

For each continuous and ordinal predictor, sort its values from the smallest to the largest. For the sorted predictor, go through each value from top to examine each candidate split point (call it v, if x ≤ v, the case goes to the left child node, otherwise, goes to the right.) to determine the best. The best split point is the one that maximize the splitting criterion the most when the node is split according to it. The definition of splitting criterion is in later section.

For each nominal predictor, examine each possible subset of categories (call it A, if $x \in A$, the case goes to the left child node, otherwise, goes to the right.) to find the best split.

2. Find the node's best split.

   Among the best splits found in step 1, choose the one that maximizes the splitting criterion.

3. Split the node using its best split found in step 2 if the stopping rules are not satisfied.

Splitting criteria and impurity measures

At node *t*, the best split *s* is chosen to maximize a splitting criterion $\Delta i(s,t)$. When the impurity measure for a node can be defined, the splitting criterion corresponds to a decrease in impurity. In SPSS products, $\Delta I(s,t) = p(t)\Delta i(s,t)$ is referred to as the improvement.

Categorical dependent variable

If *Y* is categorical, there are three splitting criteria available: Gini, Twoing, and ordered Twoing criteria.

At node *t*, let probabilities $p(j,t)$, $p(t)$ and $p(j \mid t)$ be estimated by

$$p(j,t) = \frac{\pi(j)N_{w,j}(t)}{N_{w,j}},$$

$$p(t) = \sum_j p(j,t),$$

$$p(j \mid t) = \frac{p(j,t)}{p(t)} = \frac{p(j,t)}{\sum_j p(j,t)}.$$

where

$$N_{w,j} = \sum_{n \in \hbar} w_n f_n I(y_n = j)$$

$$N_{w,j}(t) = \sum_{n \in \hbar(t)} w_n f_n I(y_n = j),$$

with $I(a = b)$ being indicator function taking value 1 when $a = b$, 0 otherwise.

Gini criterion

The Gini impurity measure at a node *t* is defined as

$$i(t) = \sum_{i,j} C(i \mid j) p(i \mid t) p(j \mid t).$$

The Gini splitting criterion is the decrease of impurity defined as

$$\Delta i(s,t) = i(t) - p_L i(t_L) - p_R i(t_R),$$

where p_L and p_R are probabilities of sending a case to the left child node t_L and to the right child node t_L respectively. They are estimated as $p_L = p(t_L)/p(t)$ and $p_R = p(t_R)/p(t)$.

Note: When user-specified costs are involved, the altered priors can be used to replace the priors (optional). When altered priors are used, the problem is considered as if no costs are involved. The altered prior is defined as $\pi'(j) = \dfrac{C(j)\pi(j)}{\sum_j C(j)\pi(j)}$, where $C(j) = \sum_i C(i \mid j)$.

Twoing Criterion

$$\Delta i(s,t) = p_L p_R \left[\sum_j | p(j \mid t_L) - p(j \mid t_R) | \right]^2.$$

Ordered Twoing Criterion

Ordered Twoing is used only when *Y* is ordinal categorical. Its algorithm is as follows:

1. First separate the class $C = \{1, \ldots, J\}$ of *Y* as two super-classes C_1 and $C_2 = C - C_1$ such that C_1 is of the form $C_1 = \{1, \ldots, j_1\}$, $j_1 = 1, \ldots, J - 1$.
2. Using the 2-class measure $i(t) = p(C_1 \mid t)p(C_2 \mid t)$, find the split $s^*(C_1)$ that maximizes $\Delta i(s,t)$,

$$\Delta i(s,t) = i(t) - p_L i(t_L) - p_R i(t_R) = p_L p_R \left[\sum_{j \in C_1} \{ p(j \mid t_L) - p(j \mid t_R) \} \right]^2 .$$

3. Find the super-class C^*_1 of C_1 which maximizes $\Delta i(s^*(C_1), t)$.

Continuous dependent variable

When *Y* is continuous, the splitting criterion $\Delta i(s,t) = i(t) - p_L i(t_L) - p_R i(t_R)$ is used with the Least Squares Deviation (LSD) impurity measures

$$i(t) = \frac{\sum_{n \in \hbar(t)} w_n f_n (y_n - \bar{y}(t))^2}{\sum_{n \in \hbar(t)} w_n f_n},$$

where

$$p_L = N_w(t_L) / N_w(t), \quad p_R = N_w(t_R) / N_w(t), \quad N_w(t) = \sum_{n \in \hbar(t)} w_n f_n ,$$

$$\bar{y}(t) = \frac{\sum_{n \in \hbar(t)} w_n f_n y_n}{N_w(t)} .$$

Stopping Rules

Stopping rules control if the tree growing process should be stopped or not. The following stopping rules are used:

- If a node becomes pure; that is, all cases in a node have identical values of the dependent variable, the node will not be split.
- If all cases in a node have identical values for each predictor, the node will not be split.
- If the current tree depth reaches the user-specified maximum tree depth limit value, the tree growing process will stop.
- If the size of a node is less than the user-specified minimum node size value, the node will not be split.
- If the split of a node results in a child node whose node size is less than the user-specified minimum child node size value, the node will not be split.

- If for the best split s^* of node *t*, the improvement $\Delta I(s^*, t) = p(t)\Delta i(s^*, t)$ is smaller than the user-specified minimum improvement, the node will not be split.

Surrogate Split

Given a split $X^* \le s^*$, its surrogate split is a split using another predictor variable *X*, $X \le s_X$ (or $X > s_X$), such that this split is most similar to it and is with positive predictive measure of association. There may be multiple surrogate splits. The bigger the predictive measure of association is, the better the surrogate split is.

Predictive measure of association

Let $\hbar_{X^* \cap X}$ (resp. $\hbar_{X^* \cap X}(t)$) be the set of learning cases (resp. learning cases in node *t*) that has non-missing values of both X^* and *X*. Let $p(s^* \approx s_X \mid t)$ be the probability of sending a case in $\hbar_{X^* \cap X}(t)$ to the same child by both s^* and s_X, and $\tilde{s}_X$ be the split with maximized probability $p(s^* \approx \tilde{s}_X \mid t) = \max_{s_X}\left(p(s^* \approx s_X \mid t)\right)$.

The *predictive measure of association* $\lambda(s^* \approx \tilde{s}_X \mid t)$ between s^* and $\tilde{s}_X$ at node *t* is

$$\lambda(s^* \approx \tilde{s}_X \mid t) = \frac{\min(p_L, p_R) - (1 - p(s^* \approx \tilde{s}_X \mid t))}{\min(p_L, p_R)},$$

where p_L (resp. p_R) is the relative probability that the best split s^* at node *t* sends a case with non-missing value of X^* to the left (resp. right) child node, $p_L = p(t_L)/p(t)$ and $p_R = p(t_R)/p(t)$ respectively. And where

$$p(s^* \approx s_X \mid t) = \begin{cases} \sum_j \dfrac{\pi(j) N_{w,j}(s^* \approx s_X, t)}{N_{w,j}(X^* \cap X)} & \text{if } Y \text{ is categorical} \\ \dfrac{N_w(s^* \approx s_X, t)}{N_w(X^* \cap X)} & \text{if } Y \text{ is continuous} \end{cases},$$

with

$$N_w(X^* \cap X) = \sum_{n \in \hbar_{X^* \cap X}} w_n f_n \ , \ N_w(X^* \cap X, t) = \sum_{n \in \hbar_{X^* \cap X}(t)} w_n f_n$$

$$N_w(s^* \approx s_X, t) = \sum_{n \in \hbar_{X^* \cap X}(t)} w_n f_n I(n : s^* \approx s_X)$$

$$N_{w,j}(X^* \cap X) = \sum_{n \in \hbar_{X^* \cap X}} w_n f_n I(y_n = j),\ N_{w,j}(X^* \cap X) = \sum_{n \in \hbar_{X^* \cap X}(t)} w_n f_n I(y_n = j)$$

$$N_{w,j}(s^* \approx s_X, t) = \sum_{n \in \hbar_{X^* \cap X}(t)} w_n f_n I(y_n = j) I(n : s^* \approx s_X)$$

and $I(n : s^* \approx s_X)$ being the indicator function taking value 1 when both splits s^* and s_X send the case *n* to the same child, 0 otherwise.

Missing Value Handling

If the dependent variable of a case is missing, this case will be ignored in the analysis. If all predictor variables of a case are missing, this case will also be ignored. If the case weight is missing, zero, or negative, the case is ignored. If the frequency weight is missing, zero, or negative, the case is ignored.

The surrogate split method is otherwise used to deal with missing data in predictor variables. Suppose that $X^* < s^*$ is the best split at a node. If value of X^* is missing for a case, the best surrogate split (among all non-missing predictors associated with surrogate splits) will be used to decide which child node it should go. If there are no surrogate splits or all the predictors associated with surrogate splits for a case are missing, the majority rule is used.

Reference

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