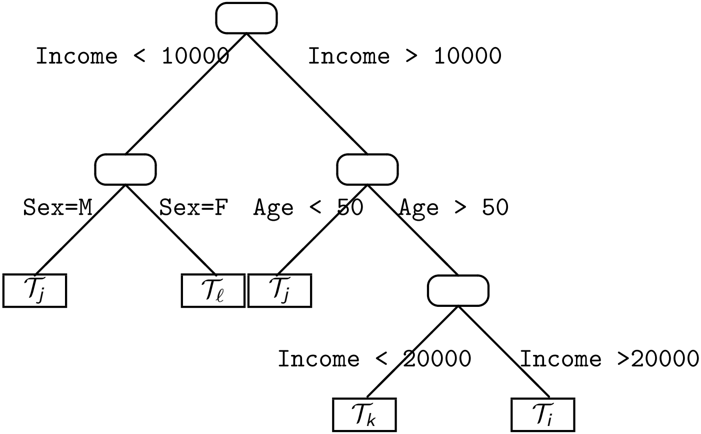
Classification And Regression Trees

Introduction

- Classification and regression trees (CART) : Breiman et al. (1984)
- X^j explanatory variables (quantitative or qualitative)
- Y qualitative with m modalities $\{\mathcal{T}_\ell; \ell = 1 \ldots, m\}$: classification tree
- Y quantitative : regression tree
- Objective : construction of a binary decision tree easy to interpret
- No assumption on the model : non parametric procedure.

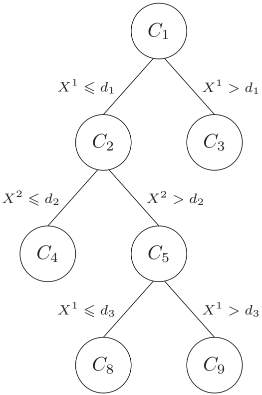
Example of binary classification tree

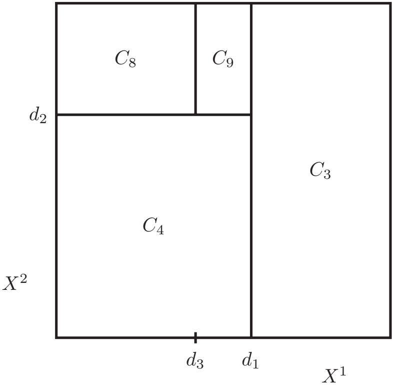


Definitions

- Determine an iterative sequence of *nodes*
- *Root* : initial note : the whole sample
- *Leaf* : Terminal node
- *Node* : choice of one variable and one *division* to proceed to a dichotomie
- *Division* : threshold value or group of modalities

Example of dyadic partition of the space





Rules

- We have to choose :
  1. Criterion for the “best” division among all *admissibles* ones (partition based on the values of one variable)
  2. Rules for a terminal node : *leaf*
  3. Rules to assign to a class $\mathcal{T}_\ell$ or one value for Y
- *Admissible* divisions : descendants $\neq \emptyset$
- X^j real or ordinal with c_j possible values : $(c_j - 1)$ possible divisions
- X^j nominal : $2^{(c_j-1)} - 1$ possible divisions
- Heterogeneity function D_κ of one node
  1. Null : a single modality for Y or Y is constant
  2. Maximal : all the modalities for Y or large variance

Division criterion

Optimal division

- Notations
  - κ : a node
  - κ_L and κ_R the two son nodes
- The algorithm retains the division which minimizes

$$D_{\kappa_L} + D_{\kappa_R}$$

- For each node κ in the construction of the tree :

$$\max_{\{Divisions\ of\ X^j; j=1,p\}} D_\kappa - (D_{\kappa_L} + D_{\kappa_R})$$

Stopping rule and affectation

Leaf and affectation

- A Node is a terminal node or a leaf, if it is :
  - Homogeneous
  - Number of observations below some threshold
- Affectation
  - Y quantitative, the value is the mean of the observations in the leaf
  - Y qualitative, each leaf is affected to one class $\mathcal{T}_\ell$ of Y by considering the conditional mode :
    - the mostly represented class in the node
    - The less cosltly class if cost for wrong classification are given

Heterogeneity criterion in regression

Y quantitative : heterogeneity in regression

Heterogeneity of the node κ :

$$D_\kappa = \sum_{i \in \kappa} (y_i - \overline{y}_\kappa)^2 = |\kappa| \frac{1}{|\kappa|} \sum_{i \in \kappa} (y_i - \overline{y}_\kappa)^2$$

where $|\kappa|$ is the cardinality of the node κ

Minimize the intra-class variance

The son nodes κ_L and κ_R minimize :

$$\frac{1}{n} \sum_{i \in \kappa_L} (y_i - \overline{y}_{\kappa_L})^2 + \frac{1}{n} \sum_{i \in \kappa_R} (y_i - \overline{y}_{\kappa_R})^2.$$

Heterogeneity criterion in classification

Y qualitative : $Y \in \{\mathcal{T}_\ell, l = 1, \ldots, m\}$.
Node κ.

- $n_\kappa^\ell = \text{Card } \{(X_i, Y_i), X_i \in \kappa, Y_i \in \mathcal{T}_\ell\}$.
- $n_\kappa = \text{Card } \{(X_i, Y_i), X_i \in \kappa\}$.
- p_κ^ℓ : probability that an observation is in class $\mathcal{T}_\ell$ given that it is in node κ.
- Estimated by $\frac{n_\kappa^\ell}{n_\kappa}$.

Heterogeneity criterion in classification

Y qualitative : heterogeneity in classification

Heterogeneity of node κ :

- Shannon Entropy with the notation $0\log(0) = 0$
  p_κ^ℓ : proportion of the class $\mathcal{T}_\ell$ of Y in the node κ.

$$E_\kappa = -\sum_{\ell=1}^{m} p_\kappa^\ell \log(p_\kappa^\ell)$$

  Maximal in $(\frac{1}{m}, \ldots, \frac{1}{m})$, minimal in $(1, 0, \ldots, 0), \ldots, (0, \ldots, 0, 1)$.

$$D_\kappa = -|\kappa| \sum_{\ell=1}^{m} p_\kappa^\ell \log(p_\kappa^\ell)$$

- Gini concentration : $D_\kappa = |\kappa| \sum_{\ell=1}^{m} p_\kappa^\ell (1 - p_\kappa^\ell)$

Pruning and optimal tree

Pruning : notations

- We look for a parcimonious tree
- *Complexity* of a tree : $K_A =$ numbers of leaves in A
- Adjustment error of A :

$$D(A) = \sum_{\kappa=1}^{K_A} D_\kappa$$

D_κ : heterogeneity of leaf κ

Sequence of embedded trees

- Adjustment error penalized by the complexity :

$$Crit_\gamma(A) = D(A) + \gamma \times K_A$$

- When $\gamma = 0$: $A_{\max}$ (maximal tree) minimizes $Crit_\gamma(A)$
- When γ increases, the division of A_H, for which the improvement of D is smaller than γ, is cancelled ; hence
  - two leaves are gathered (prunned)
  - there father node becomes a terminal node
  - A_K becomes A_{K-1}.
- After iteration of this process, we get a sequence of trees :

$$A_{\max} \supset A_K \supset A_{K-1} \supset \cdots A_1$$

Breiman sub-sequence

- A_K is the sub tree of $A_{\max}$ (maximal tree) obtained by pruning the nodes κ such that $D(\kappa) = D(\kappa_L) + D(\kappa_R)$.
- For each node in A_K, $D(\kappa) > D(\kappa_L) + D(\kappa_R)$ and $D(\kappa) > D(A_K^\kappa)$ where A_K^κ is the subtree of A_K from node κ.
- For γ small, $D(\kappa) + \gamma > D(A_K^\kappa) + \gamma |A_K^\kappa|$. This holds while $\gamma < (D(A_K^\kappa) - D(\kappa))/(|A_K^\kappa| - 1) = s(\kappa, A_K^\kappa)$ for all node κ of A_K.

$$\gamma_K = \inf_{\kappa \text{ node of } A_K} s(\kappa, A_K^\kappa)$$

- $Crit_{\gamma_K}(\kappa) = Crit_{\gamma_K}(A_K^\kappa)$ and the node κ becomes preferable to the subtree A_K^κ.
- $A_{K-1} = A_{\gamma_K}$ is the subtree obtained by pruning the branches from the nodes minimizing $s(\kappa, A_K^\kappa)$: this gives the second tree in the sub-sequence
- We iterate this process.

Optimal tree

Algorithm to select the optimal tree

- Maximal tree $A_{\max}$
- Imbedded sequence $A_{\max}, A_K \ldots A_1$ associated with an increasing sequence of values $\gamma_K \leq \ldots \leq \gamma_1$
- V-fold cross validation error :

  for $v = 1, \ldots, V$ **do**
  - Estimation of the sequence of trees associated to (γ_κ) with all the folds except v
  - Estimation of the error with the fold v.

  EndFor
- Sequence of the mean of these errors for each value of $\gamma_K, \ldots, \gamma_1$
- γ_{Opt} optimal value for the tuning parameter minimizing the mean of the errors
- Tree associated to γ_{Opt} in $A_K \ldots A_1$

Advantages

- Trees are easy to interpret
- Efficient algorithms to find the pruned trees
- Tolerant to missing data

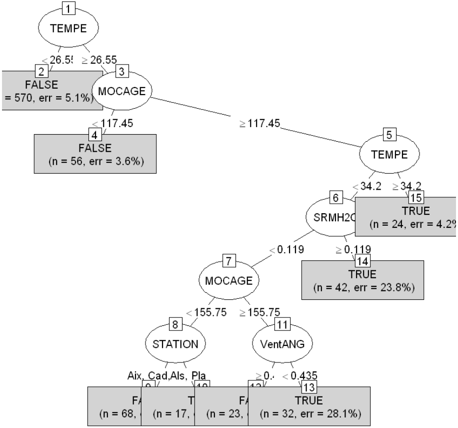
$\Longrightarrow$ Success of CART for practical applications

Warnings

- Variable selection : the selected tree only depends on few explanatory variables, trees are often (wrongly) interpreted as a variable selection procedure
- High instability of the trees : not robust to the learning sample, curse of dimensionality ..
- Prediction accuracy of a tree is often poor compared to other procedures

$\Longrightarrow$ Aggregation of trees : bagging, random forests

Example for Ozone data



Ozone : Classification tree pruned by cross-validation