Modelos Estadísticos para la Regresión y la Clasificación

Clase 9: Arboles de clasificación y regresión (CART) - Metodos de Agregación

Mathias Bourel

Instituto de Matemática y Estadística Prof. Rafael Laguardia (IMERL)
Facultad de Ingeniería, Universidad de la República, Uruguay

18 de septiembre de 2024

Plan

1. Classification and Regression Trees
2. Pruning algorithm
3. Aggregation Methods
4. Bagging
5. Random Forest
6. Boosting

Figura: Construcción geométrica. Joaquin Torres García (1929)

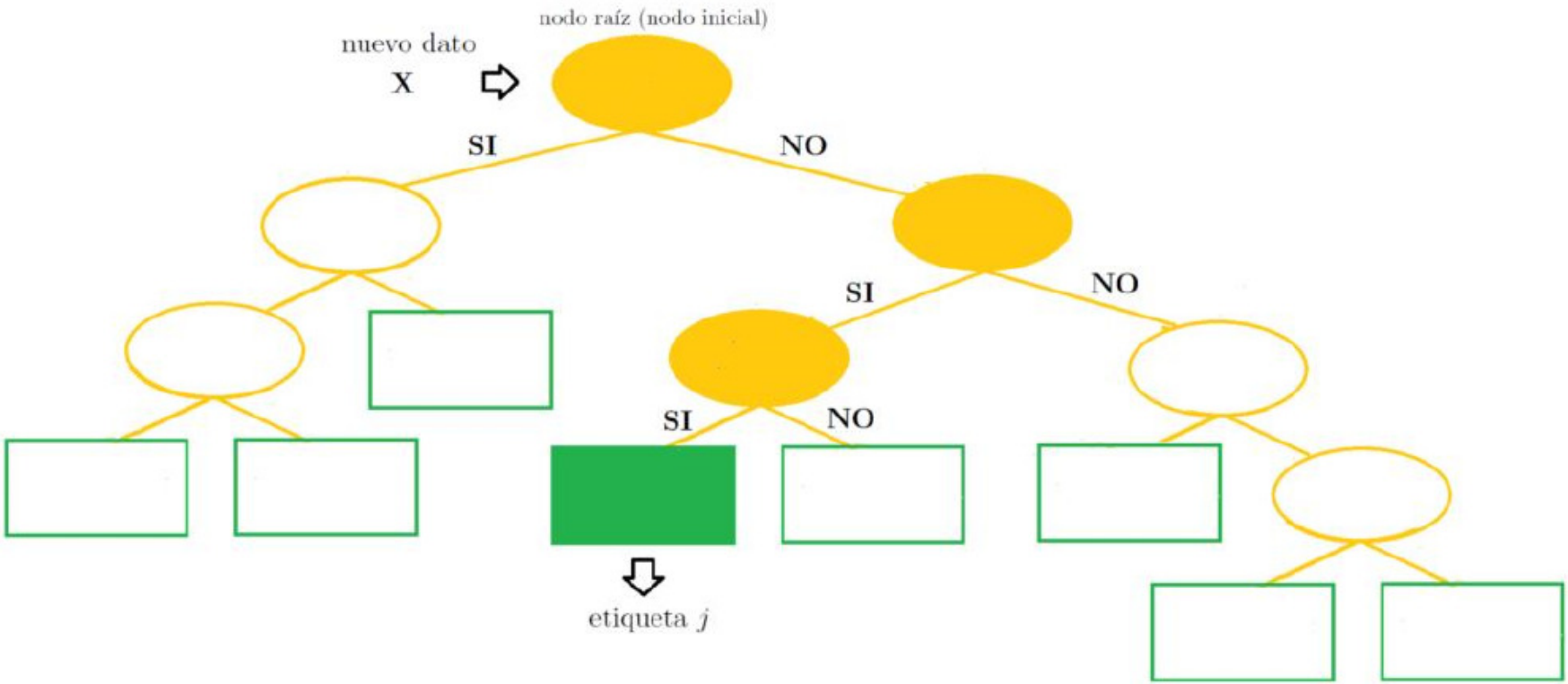
- It is a Machine Learning method.
- The idea of binary trees consists of recursively dividing the data set until reaching k terminal nodes (sheets)
- Explanatory variables can be quantitative or qualitative.
- At each stage of the algorithm, the best rule that divides the node into two is sought, in the most homogeneous way possible. This rule is of the type:

$$X_i \leq c \quad \text{vs} \quad X_i > c \text{ if } X_i \text{ is quantitative}$$

$$X_i \in \mathcal{A} \quad \text{vs} \quad X_i \in \mathcal{A} \text{ if } X_i \text{ is categorical}$$

- It is sought in each division to reduce the impurity of the parent node when its two children nodes appear.
- A stopping criterion is needed: for example minimum quantity of observations in the leaves or threshold on the criterion of impurity.

Classification and Regression Trees



The construction of a tree requires defining:

- A partition criterion: how to perform binary subdivisions.
- A stop criterion: to consider when a node is considered terminal and the process is stopped.
- An assignment criterion: for the assignment of the label to each sheet.

A partition of space X is found and we assign a value (regression problem) or a category (classification problem) at each elements of the partition. This can be written linearly as

$$\mathbb{E}(Y|X=x) = \sum_{j=1}^{q} c_j \mathbb{1}_{N_j}(x)$$

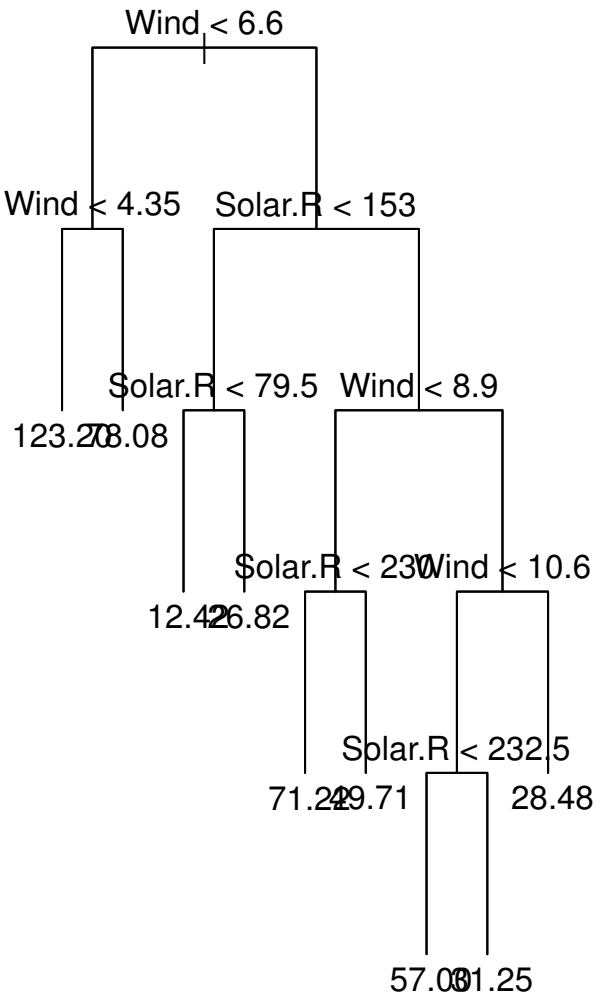
where

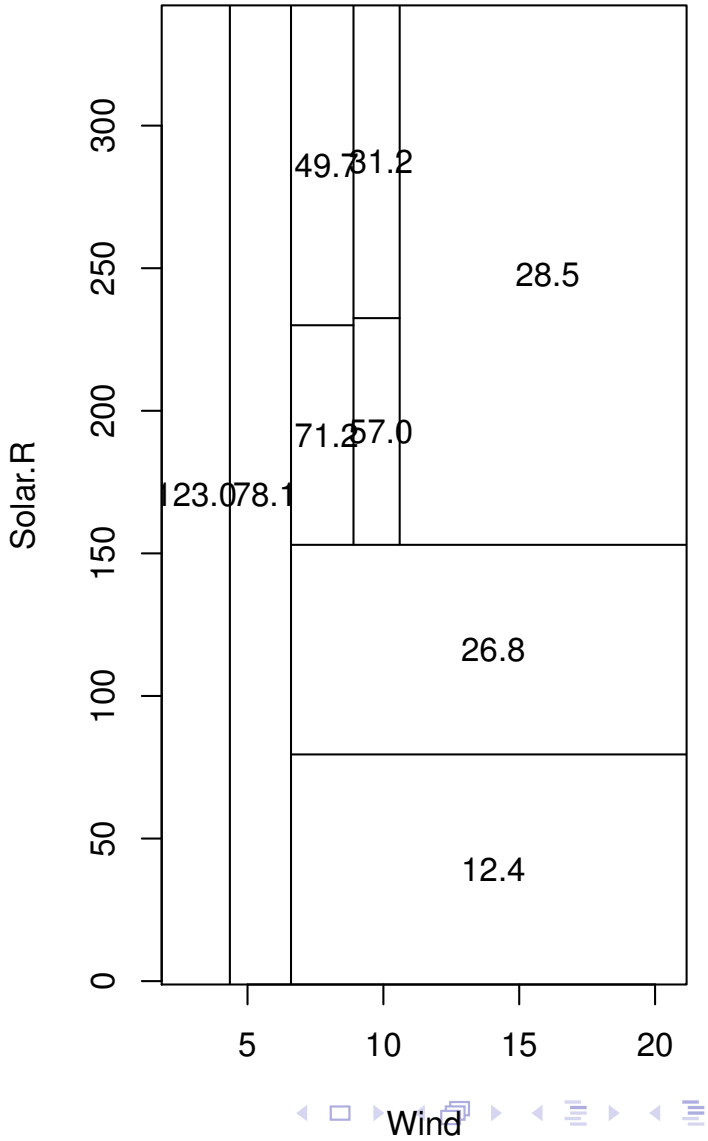
$$\widehat{c}_j = \frac{\sum_{i:X_i \in N_j} Y_i}{\#N_j} \quad \text{(regression)}$$

$$\widehat{c}_j = \text{majority class in } N_j \quad \text{(classification)}$$

Class: piecewise constant functions

Classification and Regression Trees

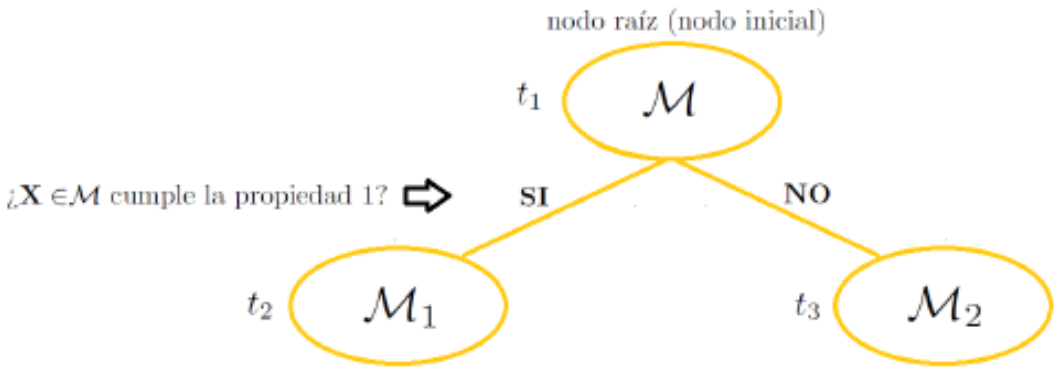




Partition criteria

All observations are in a root node. By means of a criterion of partition (criterion that involves the characteristics of the observations) this root is divided into two sub-samples, that is, two child nodes so that the children are more homogeneous in relation to Y that the parent node (decrease in impurity). And the process is repeated again.

A node is pure or homogeneous if it contains a single class. Otherwise is impure or heterogeneous.



Partition criteria (classification)

An impurity function $\phi : \{p = (p_1, \ldots, p_K) \in \mathbb{R}^K : p_i \geq 0, \sum_{i=1}^{K} p_i = 1\} \to \mathbb{R}$ must:

- be symmetric (that is, if the p_j is swapped, the function does not change).
- have minimums in the canonical basis of $\mathbb{R}^K$.
- its only maximum is $(\frac{1}{K}, \frac{1}{K}, \ldots, \frac{1}{K})$

Example of impurity functions:

1. $\phi(p) = 1 - ||p||_\infty = 1 - \max_k \{p_1, \ldots, p_K\}$ (classification error)
2. $\phi(p) = -\sum_{k=1}^{K} p_k \log(p_k)$ (entropy)
3. $\phi(p) = 1 - \sum_{k=1}^{K} p_k^2 = \sum_{k=1}^{K} p_k(1 - p_k) = \sum_{k \neq k'} p_k p_{k'}$ (Gini Index)

Note that:

- The entropy of a node with a single class is zero, because the probability is one and $\log(1) = 0$ (log is in base 2). Entropy reaches maximum ($\log(K)$) value when all classes have the same probability.
- Gini index of a node with a single class is zero. Gini index reaches maximum $(1 - 1/K)$ value when all classes have the same probability.
- Classification error of a node with a single class is zero. Classification error reaches maximum $(1 - 1/K)$ value when all classes have the same probability.

Gini index

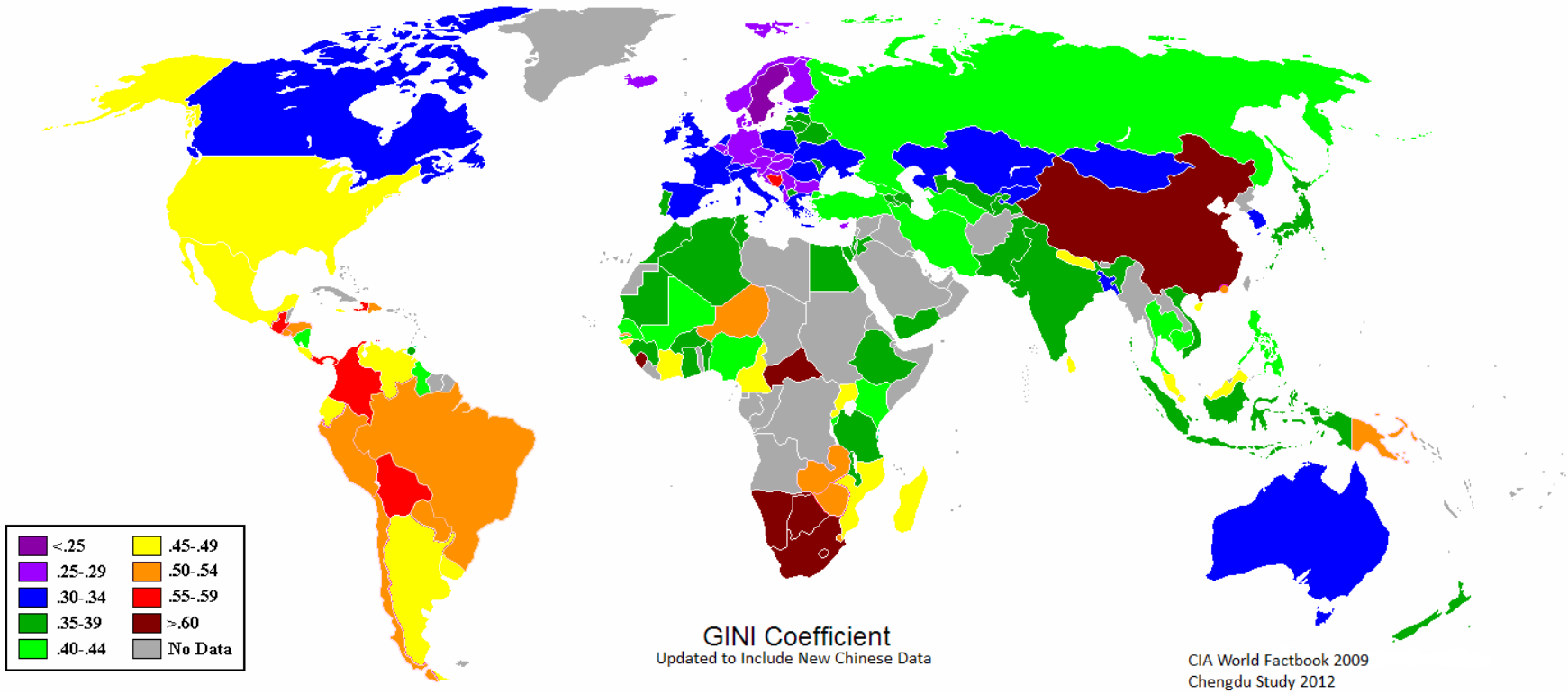


Figura: Gini coefficient map 2009

For a discrete probability distribution with probability mass function p_i, $i = 1, \dots n$, where p_i is the fraction of the population with income or wealth $y_i > 0$, the Gini coefficient is

$$G = \frac{1}{2\mu} \sum_{i=1}^{n} \sum_{j=1}^{n} p_i p_j |y_i - y_j| \text{ where } \mu = \sum_{i=1}^{n} y_i p_i.$$

Partition criteria (classification)

For example, assume we have a database with 100 observations: 40 are red, 30 are blue and 30 are green. Based on these data we can compute probability of each class. Since probability is equal to frequency relative:

$$\mathbb{P}(\text{red}) = \frac{40}{100} = 0,4 \quad \mathbb{P}(\text{blue}) = \frac{30}{100} = 0,3 \quad \mathbb{P}(\text{green}) = \frac{30}{100} = 0,3$$

- the entropy is $-0,4 \times \log(0,4) - 0,3 \times \log(0,3) - 0,3 \times \log(0,3) = 1,571$
- the gini index is $1 - (0,4^2 + 0,3^2 + 0,3^2) = 0,660$
- the classification error is $1 - 0,4 = 0,6$

Partition criteria (classification)

The Gini index and the entropy are more sensitive to changes in the probabilities of the nodes than the classification error (the latter may have many ties).

The Gini index $\sum_{k=1}^{K} p_k(1-p_k)$ is a measure of the total variation on the K classes. If all probabilities are close to 0 or 1, the Gini index is low (+ purity). Idem for entropy.

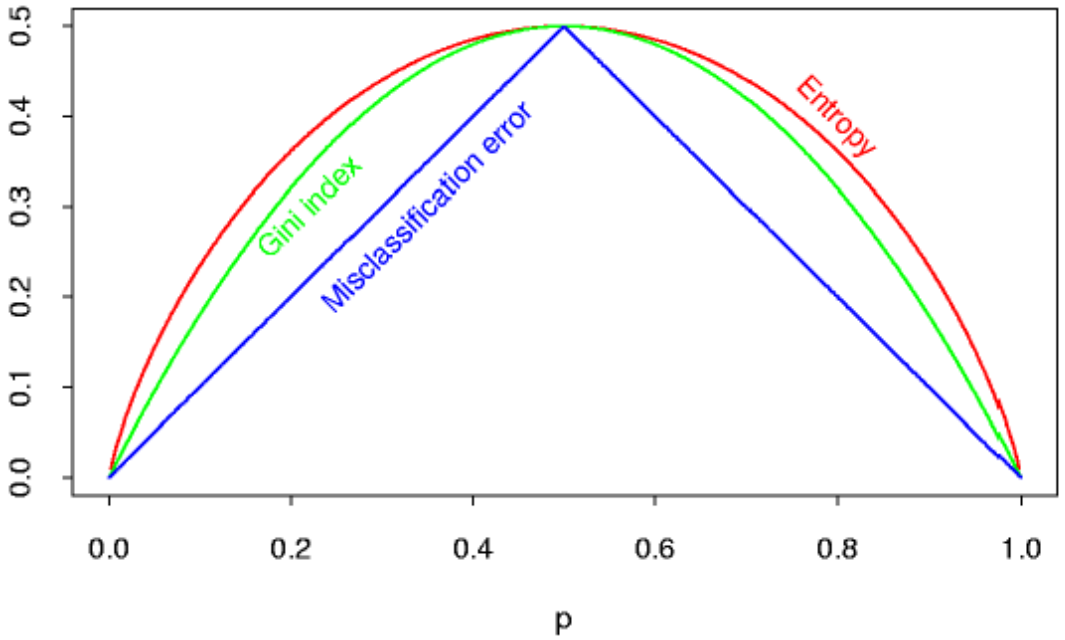


Figura: Hastie et al. (2001)

Partition criteria (classification) Node impurity

If $N(t)$ is the quantity of observations of $\mathcal{L}$ that belong in node t and $N_k(t)$ is the number of observations in t that have label k ($k \in \{1, \ldots, K\}$), then the probability that an observation of t belongs to class k is

$$p_k(t) = \frac{N_k(t)}{N(t)}$$

If ϕ an impurity function, the node's impurity of t is defined as:

$$i(t) = \phi\big(p_1(t), p_2(t), \ldots, p_K(t)\big)$$

For example if $\phi = 1 - ||p||_\infty$ then

$$i(t) = 1 - \max_k \{p_1(t), p_2(t), \ldots, p_K(t)\} = 1 - \max_k \left\{ \frac{N_1(t)}{N(t)}, \frac{N_2(t)}{N(t)}, \ldots, \frac{N_K(t)}{N(t)} \right\}$$

$$i(t) = \frac{N(t) - N_{j^*}(t)}{N(t)} = \frac{\text{misclassified in } t}{N(t)}$$

where j^* the majority class in t.

Partition criteria (classification)

The impurity variation of node t respect to its two children t_L and t_R when performing the s partition is:

$$\Delta\, i(t,s) = \underbrace{i(t) - p_L i(t_L) - p_R i(t_R)}_{i(t)-\text{expected impurity}} \geq 0$$

$$\Delta\, i(t,s) = i(t) - \frac{N(t_L)}{N(t)} i(t_L) - \frac{N(t_R)}{N(t)} i(t_R)$$

For example, if the impurity function is the classification error, then

$$\Delta\, i(t,s) = \frac{\text{misclassified in } t - \text{misclassified in } t_L - \text{misclassified in } t_R}{N(t)}$$

We choose then within all possible partitions $\mathcal{S}_t$ of t, on values and characteristic variables, the one that verifies that

$$s^*(t) = \underset{s \in \mathcal{S}_t}{\text{Argmax}}\, \Delta\, i(t,s)$$

Partition criteria (classification)

The classification error is the same for the two trees but the Gini index and entropy are defined by the tree with a pure terminal node (exercise).

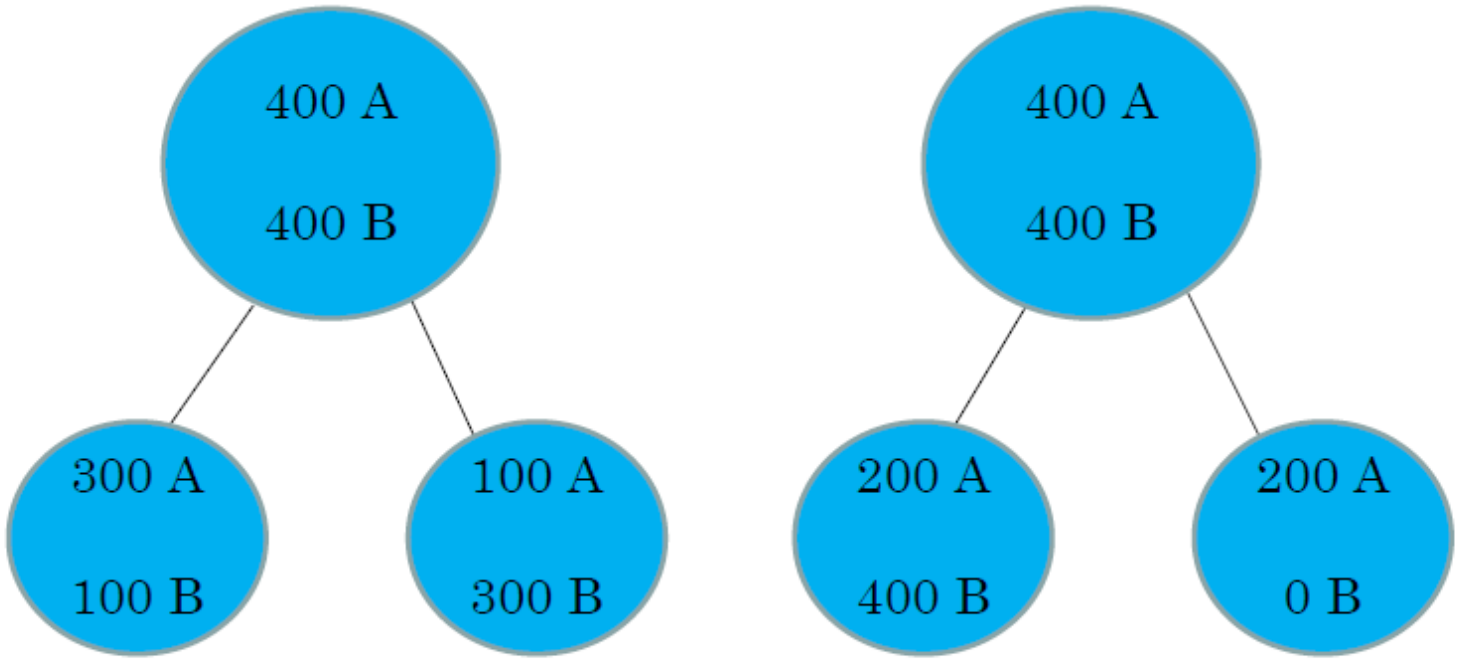


Figura: Breiman et. al, [1]

Partition criteria (classification)

The global impurity of the tree T is

$$I(T) = \sum_{t \in \tilde{T}} \underbrace{p(t)i(t)}_{R(t)}$$

where $\tilde{T}$ is the set of leaves of T, $p(t)$ is the probability of belonging to the node t and $i(t)$ is the impurity of t.

In [1], Breiman et. they prove that maximizing the impurity difference in each node is equivalent to minimizing the global impurity of the tree.

Partition criteria (regression)

In regression, deviance in the node is used to measure heterogeneity of a node

$$R(t) = \frac{1}{N} \sum_{X_i \in t} (Y_i - \overline{Y}(t))^2$$

where $\overline{Y}(t) = \frac{1}{\#t} \sum_{X_i \in t} Y_i$.

Observe that $R(t) = \frac{1}{N} \sum_{X_i \in t} (Y_i - \overline{Y}(t))^2 = \frac{\#t}{N} \frac{1}{\#t} \sum_{X_i \in t} (Y_i - \overline{Y}(t))^2 = p(t)\text{var}(t)$

We choose then within all possible partitions $\mathcal{S}_t$ of t, on values and characteristic variables, the one that verifies that minimizae the internal variance after the split

$$s^*(t) = \underset{s \in \mathcal{S}}{\text{Argmax}}\, \Delta\, R(t, s)$$

where

$$\Delta\, R(t, s) = R(t) - R(t_L) - R(t_R) \geq 0$$

Stop criterion

If i is the classification error $i(T) = R(T)$ is the estimation of the classification error and look for the tree that has smaller global impurity equivalent to the one that has $R(T)$ smaller and this will be T_{max}.
The stop criterion is defined by the user beforehand. It must be chosen so that the tree is not too large on the one hand and does not conform too much to the sample from which the tree develops. There are two main criteria:

1. Choose a threshold from which we decide that a node is pure, that is, a β such that if $i(t) \geq \beta$ we continue with the partition of t an if $i(t) < \beta$ we stop partition in t.
   If β is very small, this increases the complexity of the tree since the number of leaves can be close to N (the size of the sample) and we lose in generalization (one sheet for each observation).
2. Decide that a node does not divide more if it contains less than m observations.

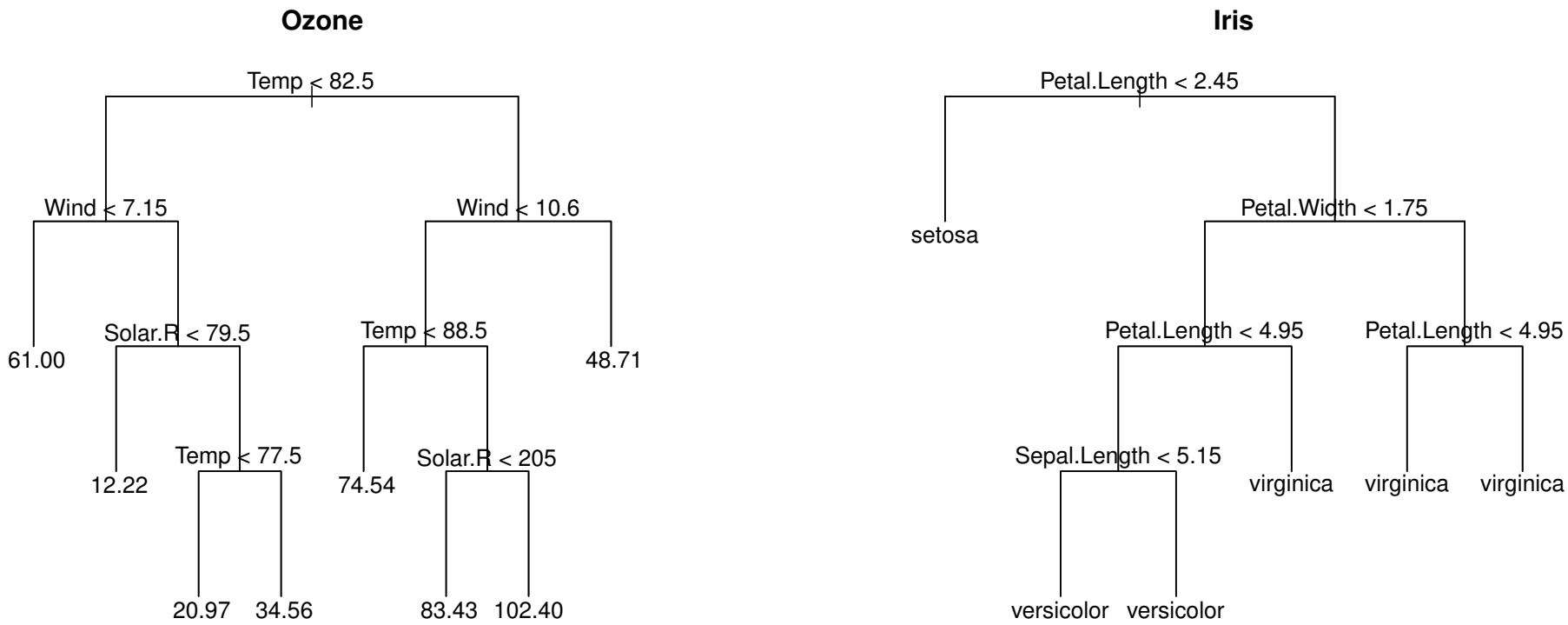
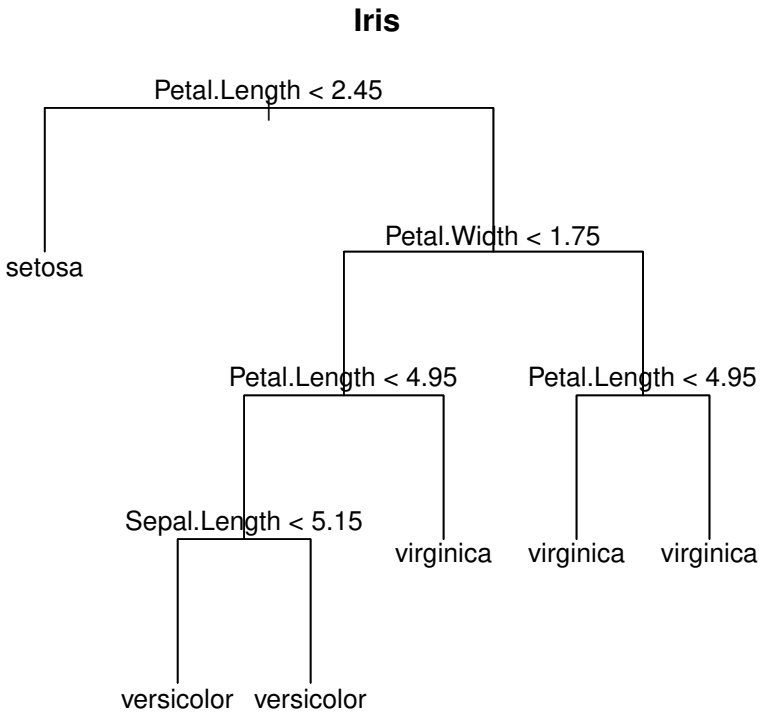


Figura: Classification and Regression Trees



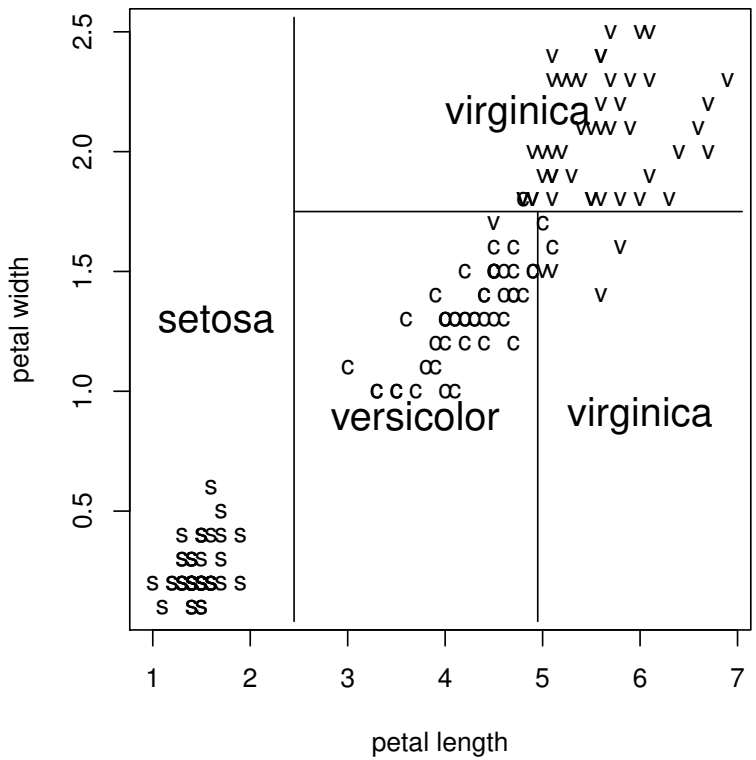


Figura: Classification and Regression Trees

Assignment criteria

In a terminal node:

- For classification:
  The class that is most represented in each terminal node is chosen (simple majority vote. If the maximum is reached for two or more classes, this class is assigned randomly.
- For regression:
  The average of the values of the dependent variable in the leaf

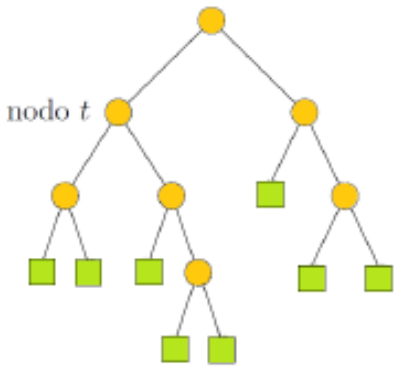
Plan

1. Classification and Regression Trees
2. Pruning algorithm
3. Aggregation Methods
4. Bagging
5. Random Forest
6. Boosting

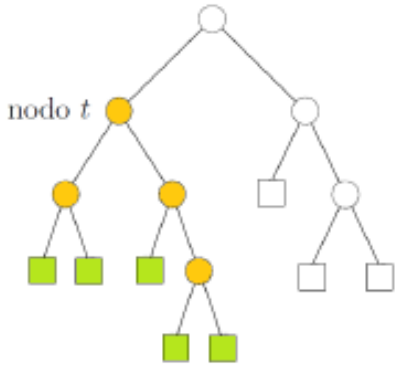
Pruning algorithm

Let t a node of T and we call branch coming form t to the subtree T_t of T that have t. The pruning of branch T_t consists in suppressing all the descendant nodes of t (except t). The tree obtained is noted by $T - T_t$. If T' is obtained from T by successive pruning of branches, we say that T' is a subtree of T and we note

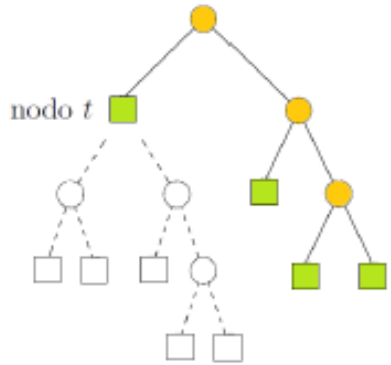
$$T' < T$$



árbol T



rama T_t



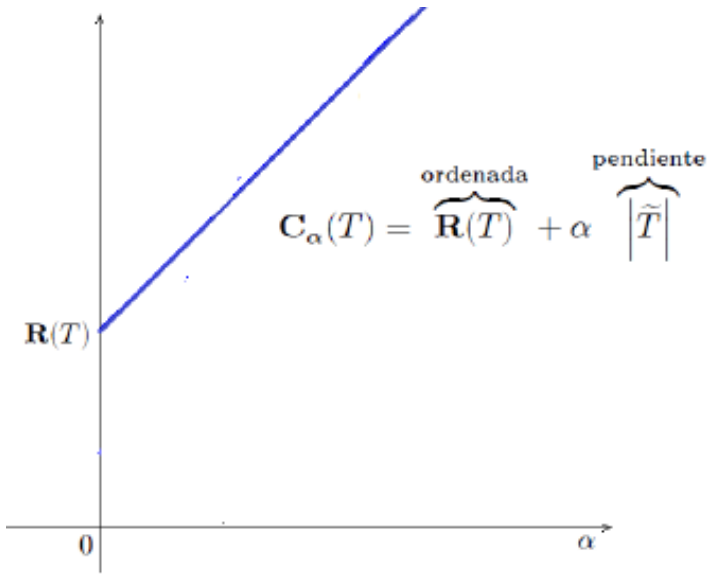
árbol podado $T - T$

Pruning algorithm

In addition to the classification error of tree T, its complexity is measured by $|\tilde{T}|$ (the number of sheets). We want to establish a trade off of combining good classification and simplicity of the classifier.

Let $\alpha \geq 0$. The cost-complexity measure of parameter α associated with tree T is:

$$C_\alpha(T) = R(T) + \alpha|\tilde{T}| \quad \text{(function of } \alpha\text{)}$$



where $R(T)$ is the classification error and $\tilde{T}$ the complexity of T (number of leaves).
Big values of α will penalize trees with many leaves, while small values of α will give little importance to the size. In the case that $\alpha = 0$ we are left with the maximal tree that minimizes the error. As we increase the value of α the size will be penalized, and then we get trees that are getting smaller and smaller but with a big error.

Pruning algorithm

In regression the cost-complexity measure is

$$C(T) = \sum_{t=1}^{|\tilde{T}|} \sum_{x_i \in t} (y_i - \hat{y}_t)^2 + \alpha |\tilde{T}|$$

(the first sum is over the leaves of T).

The pruning methods return a sequence of trees built together with their respective cost-complexity levels:

$$T_1 > T_2 > \cdots > \{t_1\} = T_K$$

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

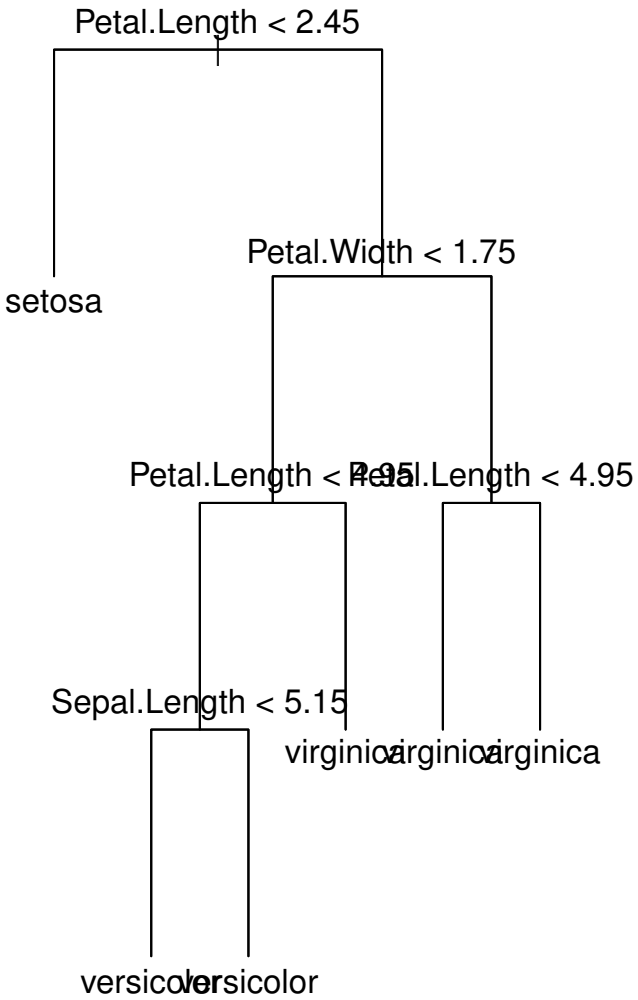
T_1 is more complex but has the less error and $\{t_1\} = T_K$ is more simple but have a big error.

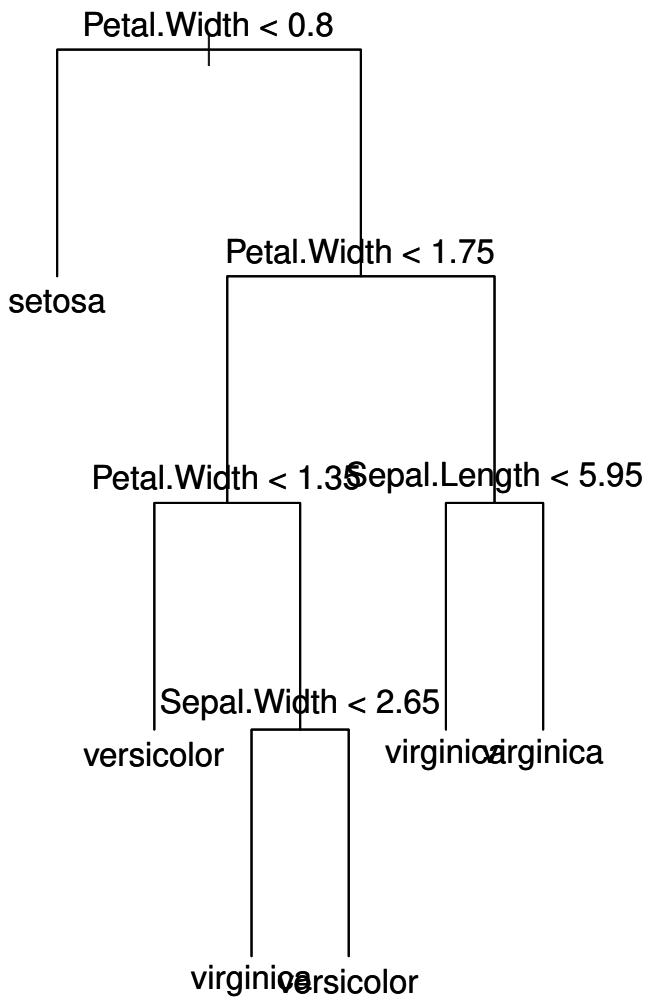
The best of these subtrees is find by the 1-SE rule using a cross validation procedure.

Final considerations

- In [1] the consistency of CART is proved: if the number of observations is increased, the classification error of the model converges to the classification error.
- CART is easy to interpret.
- CART serves both for classification and for regression.
- CART performs well with missing data (surrogate variables).
- CART is an algorithm of the greedy type: it uses the best partition at every moment and therefore can leave aside variables that can be important in explaining variability of the data because they are highly correlated with variables that were used.
- CART is unstable: aggregation methods to stabilize it (Bagging, Boosting, Random Forest).

CART instability





CART vs LM

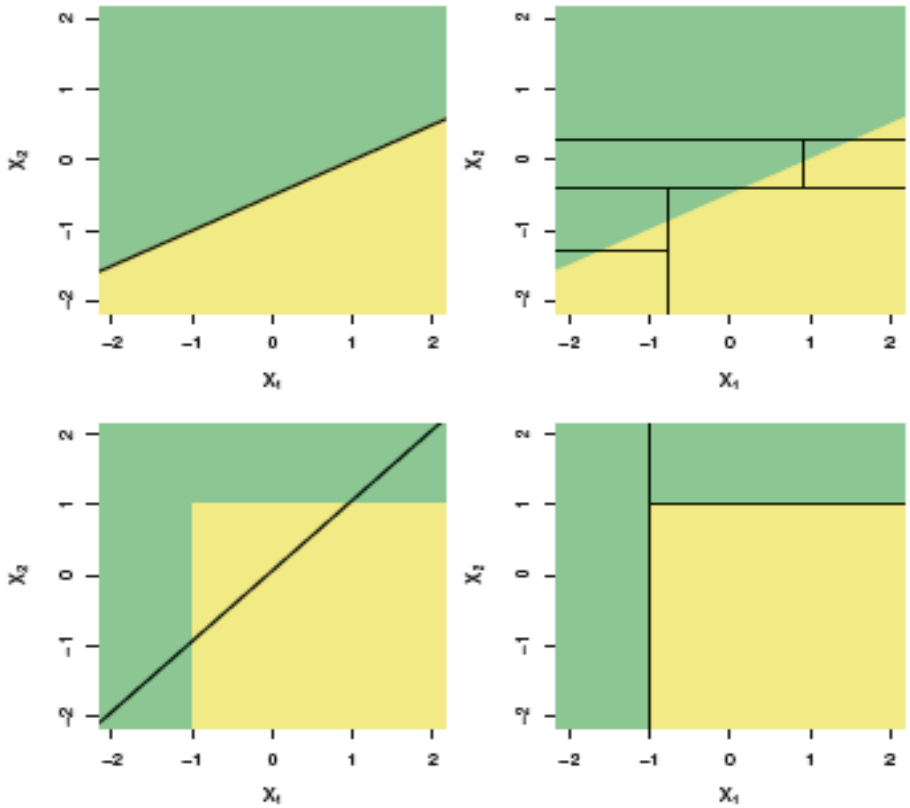


FIGURE 8.7. Top Row: *A two-dimensional classification example in which the true decision boundary is linear, and is indicated by the shaded regions. A classical approach that assumes a linear boundary (left) will outperform a decision tree that performs splits parallel to the axes (right).* Bottom Row: *Here the true decision boundary is non-linear. Here a linear model is unable to capture the true decision boundary (left), whereas a decision tree is successful (right).*

Plan

1. Classification and Regression Trees
2. Pruning algorithm
3. Aggregation Methods
4. Bagging
5. Random Forest
6. Boosting

Aggregation methods

$\mathcal{L}$ the data base and $\widehat{g_1}, \ldots, \widehat{g_M}$ several predictors built over $\mathcal{L}$

$$\widehat{f} = g(\widehat{g_1}, \ldots, \widehat{g_M})$$

Aggregation methods

$\mathcal{L}$ the data base and $\widehat{g_1}, \ldots, \widehat{g_M}$ several predictors built over $\mathcal{L}$

$$\widehat{f} = g(\widehat{g_1}, \ldots, \widehat{g_M})$$

- Homogeneous aggregation methods (sequential and not sequential).
- Not homogeneous aggregation methods (consensus methods).

Aggregation methods

Aggregation methods

$\mathcal{L}$ the data base and $\widehat{g_1}, \ldots, \widehat{g_M}$ several predictors built over $\mathcal{L}$

$$\widehat{f} = g(\widehat{g_1}, \ldots, \widehat{g_M})$$

- Homogeneous aggregation methods (sequential and not sequential).
- Not homogeneous aggregation methods (consensus methods).

Bias-Variance trade-off:

- Several space $\mathcal{H}_1, \mathcal{H}_2, \ldots, \mathcal{H}_M$ for searching predictors of different nature. However if we consider a single family $\mathcal{H}$ and predictors $\widehat{g}_1, \ldots, \widehat{g}_M \in \mathcal{H}$, the aggregated predictor $\widehat{f} = g(\widehat{g}_1, \ldots, \widehat{g}_M)$ is not necessarily a function of $\mathcal{H}$ so it could decrease the bias.
- a mean of $\widehat{g}_1, \ldots, \widehat{g}_M$ reduces the variance of the estimation.

Plan

1. Classification and Regression Trees
2. Pruning algorithm
3. Aggregation Methods
4. Bagging
5. Random Forest
6. Boosting

Bagging, Breiman (1996)

1. $\mathcal{L} = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ where $X_i \in \mathcal{X}$ and $Y_i \in \mathcal{Y}$
2. For $m = 1$ to M:
    1. We consider a bootstrap sample $\mathcal{L}_m^*$ of size n from $\mathcal{L}$.
    2. We build the estimator $g_m : \mathcal{X} \to \mathcal{Y}$ frmo $\mathcal{L}_m^*$.
3. Output: $f_M(x) = \text{Argmax}_y \#\{m : g_m(x) = y\}$ (classification) or $f_M(x) = \frac{1}{M} \sum_{m=1}^{M} g_m(x)$ (regression).

Figura: Bagging

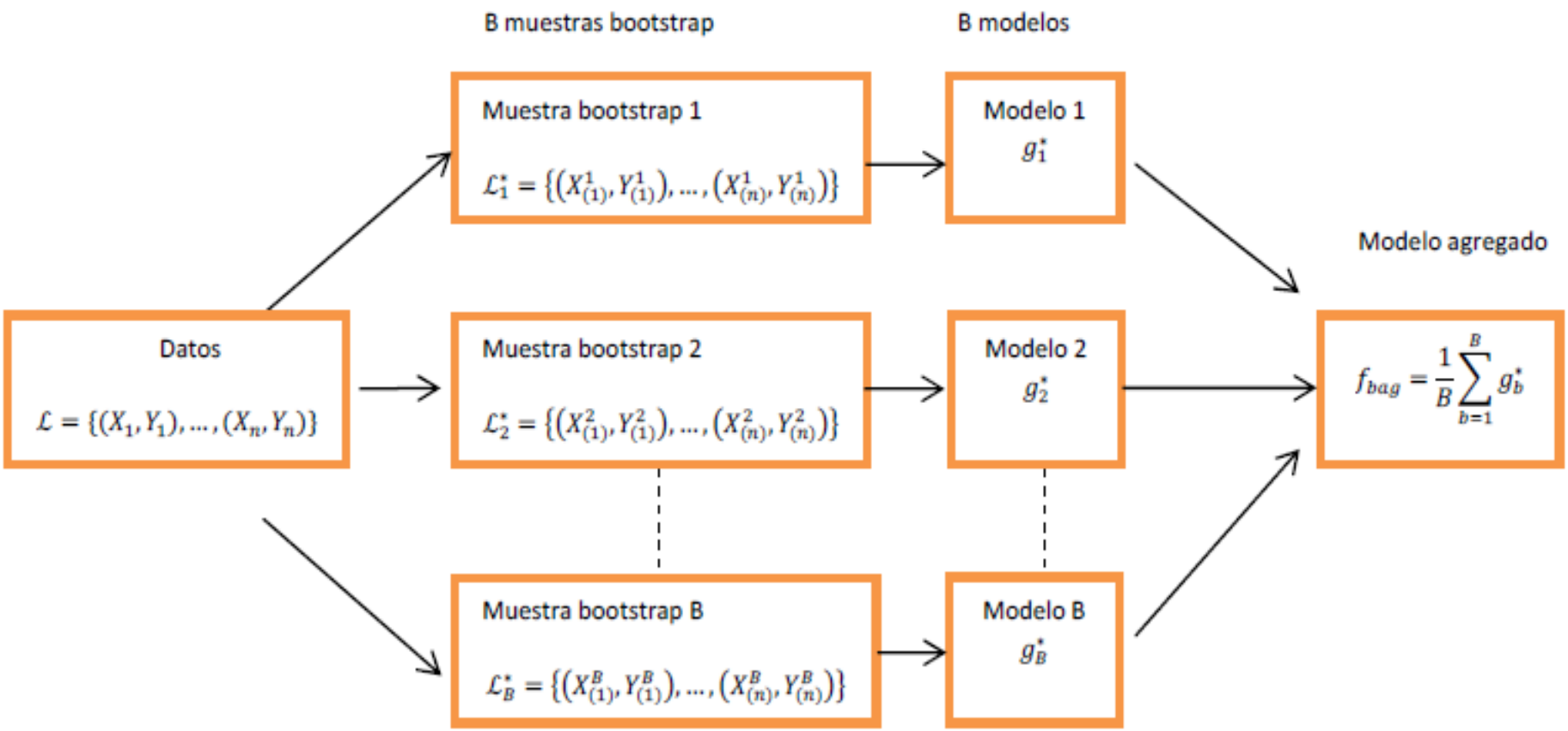
Bagging, Breiman (1996)

1. $\mathcal{L} = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ where $X_i \in \mathcal{X}$ and $Y_i \in \mathcal{Y}$
2. For $m = 1$ to M:
    1. We consider a bootstrap sample $\mathcal{L}_m^*$ of size n from $\mathcal{L}$.
    2. We build the estimator $g_m : \mathcal{X} \to \mathcal{Y}$ frmo $\mathcal{L}_m^*$.
3. Output: $f_M(x) = \text{Argmax}_y \#\{m : g_m(x) = y\}$ (classification) or $f_M(x) = \frac{1}{M} \sum_{m=1}^{M} g_m(x)$ (regression).

Figura: Bagging

- The Bagging estimator generally improves the result of any unstable algorithm. In many cases the reduction of the error is important.
- It loose interpretability.
- The observations that are not drawn in the bootstrap sample are called "out of bag" (OOB).
- OOB error is a good approximation of the test error.

Bagging, Breiman (1996)



Plan

1. Classification and Regression Trees
2. Pruning algorithm
3. Aggregation Methods
4. Bagging
5. Random Forest
6. Boosting

Random Forest, Breiman 2001

1. This method combines the predictions of several trees obtained from bootstrap samples of the data set.
2. In each node, only a small number (e.g $\sqrt{p}$ or $\log(p)$ of the total number p) of randomly chosen variables is taken into account to determine the best partition. This value suggested by Breiman in classification, has been confirmed by several works which showed its optimality in terms of performance of forests on OOB samples.
3. There is no pruning.

Random Forest, Breiman 2001

1. This method combines the predictions of several trees obtained from bootstrap samples of the data set.
2. In each node, only a small number (e.g $\sqrt{p}$ or $\log(p)$ of the total number p) of randomly chosen variables is taken into account to determine the best partition. This value suggested by Breiman in classification, has been confirmed by several works which showed its optimality in terms of performance of forests on OOB samples.
3. There is no pruning.

1. $\mathcal{L} = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ where $X_i \in \mathcal{X}$ and $Y_i \in \mathcal{Y}$
2. For $m = 1$ to M:
   1. We consider a bootstrap sample $\mathcal{L}_m^*$ of size n from $\mathcal{L}$.
   2. We build a maximal tree T_m from $\mathcal{L}_m^*$ (without pruning).
3. Output: $f_M(x) = \text{Argmax}_y \#\{m : T_m(x) = y\}$ (classification) or $f_M(x) = \frac{1}{M} \sum_{m=1}^{M} g_m(x)$ (regression).

Figura: Random Forest

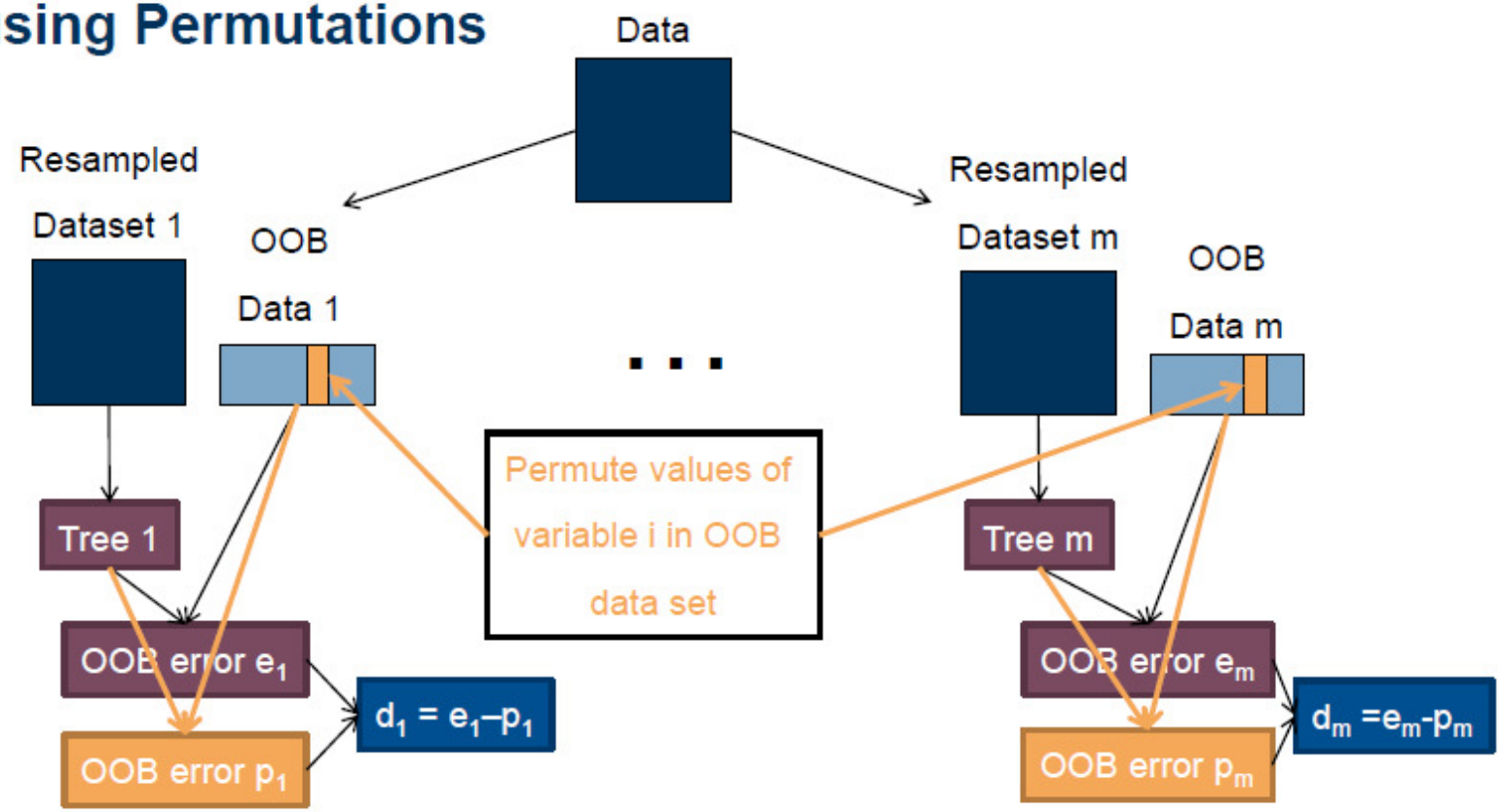
Random Forest, Breiman 2001

RF has a technique to determine the importance of a predictor variable that use the *out of bag* observations.

1. - The OOB error of tree b is:
$$e_b = \underset{i:x_i \text{ is OOB for b}}{\text{mean}} \left(\text{error}\left(x_i, y_i\right)\right)$$
   - The OOB error is defined as
$$\underset{i}{\text{mean}} \left(\underset{b:x_i \text{ is OOB for b}}{\text{mean}} \left(error(x_i)\right) \right)$$
   and is an estimation of the test error.
2. Consider the variable $x^{(j)}$. We permute the value for this variable in the OOB sample of tree b, and recompute :
$$\widehat{e}_b = \underset{i:x_i \text{ is OOB for b}}{\text{mean}} \left(\text{error}\left(x_i, y_i\right)\right)$$
3. The difference between the original OOB error and the latter give an index of the importance of variable j:
$$VI\left(x^{(j)}\right) = \frac{1}{B}\sum_{b=1}^{B}\left(\widehat{e}_b - e_b\right)$$

This index can also be based on the average decrease of another criterion, as example the Gini criterion used in the construction of trees

Variable Importance for variable i using Permutations



$$\left.\begin{array}{l} \overline{d} = \frac{1}{m}\sum_{i=1}^{m} d_i \\ s_d^2 = \frac{1}{m-1}\sum_{i=1}^{m}(d_i - \overline{d})^2 \end{array}\right\} \quad v_i = \frac{\overline{d}}{s_d}$$

Another method that is also used is the *Mean Decrease in Gini coefficient* that gives a measure of how each variable contributes to the homogeneity of the nodes. For variable $x^{(j)}$, we average on all the trees of the forest the change in impurity across all the nodes that are splitted by $x^{(j)}$:

$$VI_G(x_j) = \frac{1}{B}\sum_{b=1}^{B} \sum_{\substack{t \in b \text{ s.t.} \\ v(s_t) = x_j}} p(t)\Delta i(s_t, t)$$

where B is the total number of trees, $p(t)$ is the proportion of observations in node t, $v(s_t)$ is the variable used in split t and $\Delta i(s_t, t)$ is the change in impurity between node t and its two child nodes for the split s_t.

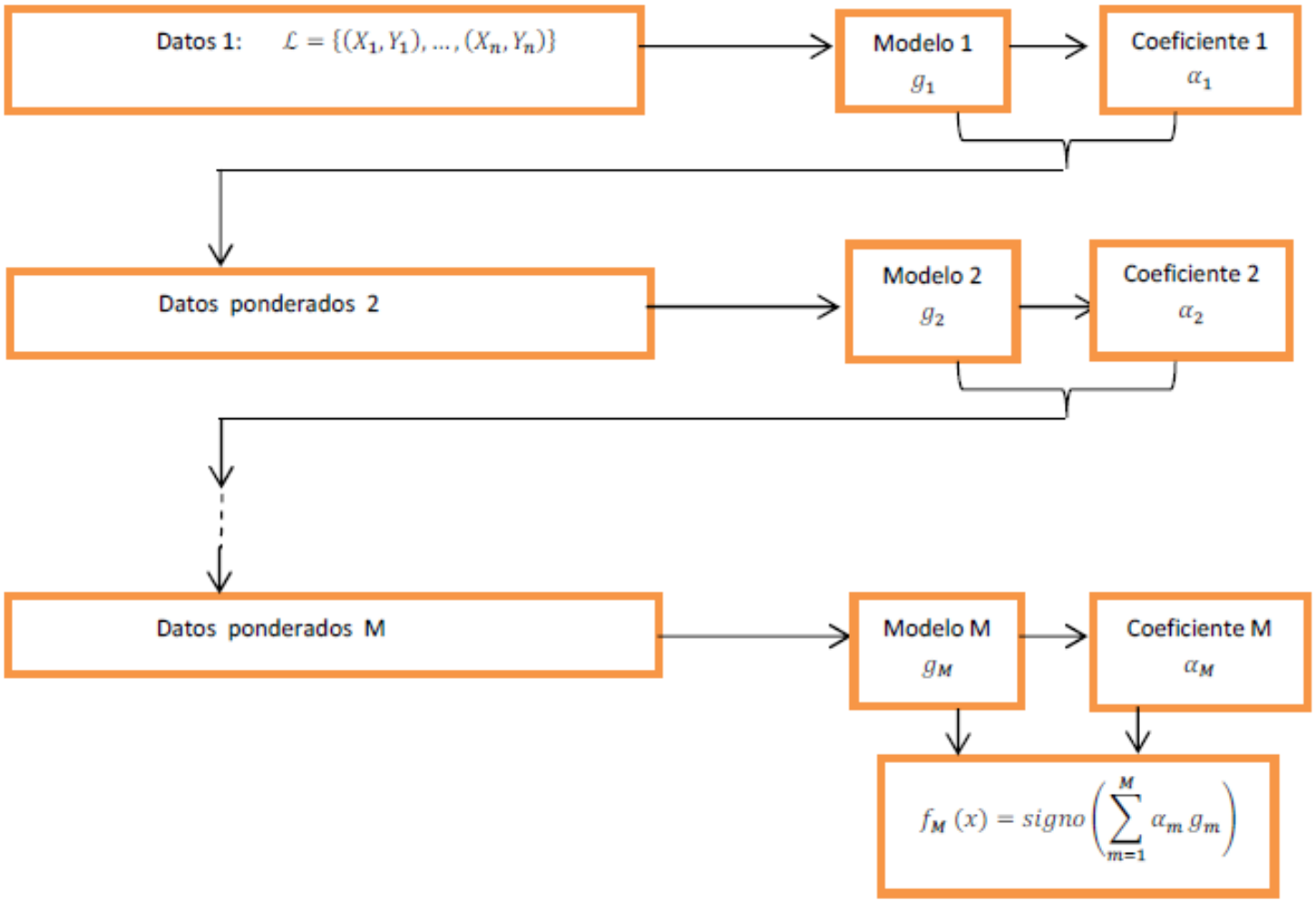
Random Forest, Breiman 2001

1. By using few variables in each partition, overfitting is avoided.
2. In addition, compared to large databases with a high number of variables, the model trains more quickly than for other techniques, such as Bagging or Boosting.
3. As with Bagging, the disadvantage of this method versus CART is the loss of interpretability.
4. But clearly, much is gained in terms of the predictive power of the model.

Plan

1. Classification and Regression Trees
2. Pruning algorithm
3. Aggregation Methods
4. Bagging
5. Random Forest
6. Boosting

Adaboost


Datos 1: $\mathcal{L} = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$
Modelo 1
g_1
Coeficiente 1
α_1
Datos ponderados 2
Modelo 2
g_2
Coeficiente 2
α_2
Datos ponderados M
Modelo M
g_M
Coeficiente M
α_M
$f_M(x) = signo\left(\sum_{m=1}^{M} \alpha_m g_m\right)$

Adaboost (Freund and Schapire, 1997)

1. $\mathcal{L} = \{(X_1, Y_1), \ldots, (X_N, Y_N)\}$ where $X_i \in \mathcal{X}$ and $Y_i \in \{-1, 1\}$
2. Initialization of the weights: $w_1(i) = \frac{1}{N} \qquad i = 1, \ldots, N.$
3. For $t = 1$ to T:
   - From $\mathcal{L}$ and weights $w_t(i)$, we build a predictor $h_t : \mathcal{X} \to \mathcal{Y}$ that minimizes the error $\varepsilon_t = \sum_{i=1}^{N} w_t(i) 1_{\{h_t(X_i) \neq Y_i\}}$
   - Calculate $\alpha_t = \frac{1}{2} \log\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$.
   - Update the weights: $w_{t+1}(i) = \frac{w_t(i)}{Z_t} \exp\left(-\alpha_t Y_i h_t(X_i)\right)$ for all $i = 1, \ldots, N$, where $Z_t = \sum_{i=1}^{n} w_t(i) \exp\left(-\alpha_t Y_i h_t(X_i)\right)$
4. Output: $f_T(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right) = \underset{y \in \{-1,1\}}{\text{Argmax}} \left(\sum_{t=1}^{T} \alpha_t \mathbb{1}_{\{h_t(x)=y\}}\right)$

Figura: Adaboost, Freund and Schapire, 1997

SAMME (Adaboost for the multiclass context)

Let $\mathcal{L} = \{(x_1, y_1), \ldots, (x_N, y_N)\}$ be a training sample where $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y} = \{1, \ldots, K\}$

1. Initialization of the weights: $w_1(i) = \frac{1}{N} \qquad i = 1, \ldots, N.$
2. For $t = 1$ to T:
   - From $\mathcal{L}$ and weights $w_t(i)$, we build a predictor $h_t : \mathcal{X} \to \mathcal{Y}$ that minimizes misclassification the error
   $$\varepsilon_t = \sum_{i=1}^{N} w_t(i) \mathbb{1}_{\{h_t(x_i) \neq y_i\}}$$
   If $\varepsilon_t \geq 1 - \frac{1}{K}$, stop the algorithm.
   - Calculate $\alpha_t = \frac{1}{2} \log \left(\frac{1 - \varepsilon_t}{\varepsilon_t} (K - 1) \right)$.
   - Update the weights: $w_{t+1}(i) = \frac{w_t(i) \exp\left(\alpha_t \mathbb{1}_{\{h_t(x_i) \neq y_i\}}\right)}{Z_t}$ for all $i = 1, \ldots, N$ where Z_t is a normalization factor.
3. Output. The final classifier is: $H_T(x) = \underset{y \in \{1, \ldots, K\}}{\text{Argmax}} \left(\sum_{t=1}^{T} \alpha_t \mathbb{1}_{\{h_t(x) = y\}} \right)$

Figura: *Stagewise Additive Modelling using Multiclass Exponentional loss function* (SAMME), Zhu *et al.*, 2009.

Toy Example Schapire

	Yi	W1	h1	W2	h2	W3	h3
X1	+	0,1	0	0,07	0	0,05	1
X2	+	0,1	0	0,07	0	0,05	1
X3	-	0,1	0	0,07	1	0,17	0
X4	-	0,1	0	0,07	1	0,17	0
X5	+	0,1	1	0,166	0	0,11	0
X6	+	0,1	1	0,166	0	0,11	0
X7	-	0,1	0	0,07	1	0,17	0
X8	+	0,1	1	0,166	0	0,11	0
X9	-	0,1	0	0,07	0	0,05	0
X10	-	0,1	0	0,07	0	0,05	1

e1=0,3	e2=0,21	e3=0,15
b1=(1-0,3)/0,3=7/3	b2=(1-0,2)/0,2=4	b3=(1-0,15)/0,15
ln(b1)= 0,84	ln(b2)= 1,38	ln(b3)= 1,72

Figura: Freund and Schapire, 1997

Toy Example Schapire

Final classifier is

$$H(x) = sgn\Big(\frac{1}{2}\left(0{,}84 \times h_1(x) + 1{,}38 \times h_2(x) + 1{,}72 \times h_3(x)\right)\Big)$$

$$= \underset{y \in \{-1,1\}}{\text{Argmax}} \left(\frac{1}{2}\left(0{,}84 \times \mathbb{1}_{\{h_1(x)=y\}} + 1{,}38 \times \mathbb{1}_{\{h_2(x)=y\}} + 1{,}72 \times \mathbb{1}_{\{h_3(x)=y\}}\right)\right)$$

Final considerations for Adaboost

- Simple and easy to implement
- Single parameter: number of iterations
- It can be extended to cases in which the output variable Y is multiclass and not only for trees (any unstable algorithm).
- Detection of outliers: observations with higher weights are generally outliers.
- It is proved that the classification error on the training sample decays exponentially with the number of iterations.

Bibliography

1. Breiman, Freidman, Stone. Classification and Regression Trees, Chapman & Hall/CRC. 1984.
2. James, Witten, Hastie, Tibshirani. An introduction to Statistical Learning with application in R, Springer, 2013.
3. Hastie, Tibshirani, Friedman. The Elements of Statistical Learning, Springer, 2003.
4. Schapire, R.E and Freund, Y., *Boosting : Foundations and Algorithms*. Adaptive Computation and Machine Learning Series. Mit Press, 2012.
5. Freund, Y. and Schapire, E., A decision-theoretic generalization of on-line learning and application to boosting, *Journal of Computer and System Sciences*, 55(1): p 119-13, 1997.
6. Breiman, L., *Bagging predictors.*, Machine Learning 24, 123?140, 1996
7. Bourel, M., *Métodos de Agregación de modelos y aplicaciones*, Mémorias de trabajos de difusión científica y técnica, Vol. 10, p. 19-32, 2012.
8. Bourel, M., *Agrégation de modèles en apprentissage statistique pour l'estimation de la densité et la classification multiclasse*, Tesis de doctorado, Université Aix-Marseille, 2013.
9. Diaz, J., apuntes del curso 2013 de Aprendizaje Automático y Aplicaciones, FING, UdelaR.
10. Loh W.-Y. (2014) Fifty Years of Classification and Regression Trees, International Statistical Review, 82, pages 329–348,