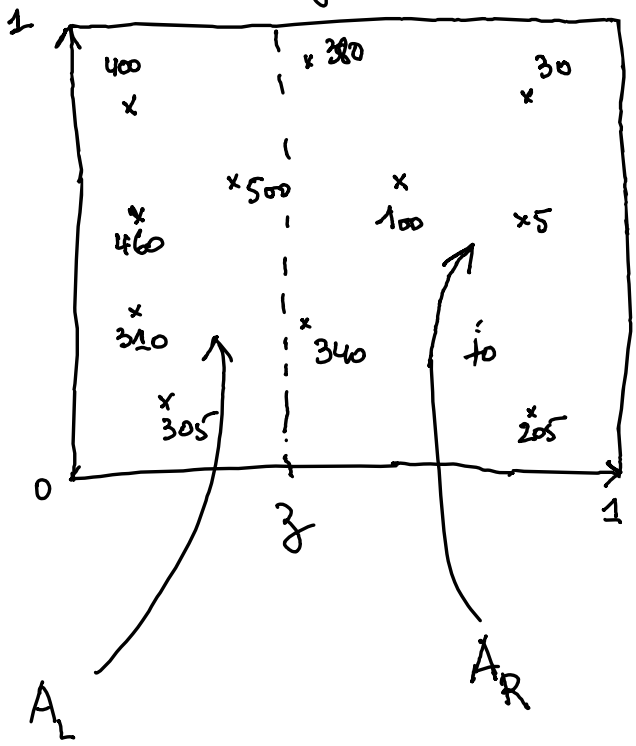
Arbre de régression : (construction)



Critère :

$$L_n(j,z) = \frac{1}{N_n(A)} \sum_{i=1}^{n} \left(Y_i - \bar{Y}_{A_L} \mathbb{1}_{\{X_i^{(j)} < z\}} - \bar{Y}_{A_R} \mathbb{1}_{\{X_i^{(j)} \geq z\}} \right)^2$$

- $A_L = \{ x \in A : x^{(j)} < s \}$
- $A_R = \{ x \in R : x^{(j)} \geq s \}$
- $\bar{Y}_A$ est la moyenne de Y_i qui se trouvent dans A.
- $N_n(A)$ le nombre d'observations dans A.

Le jeu de données $\mathcal{D}_n = \{(x_1, y_1), \dots, (x_n, y_n)\}$

$x \in \mathbb{R}^d$ et $y \in \mathbb{R}$ (régression).

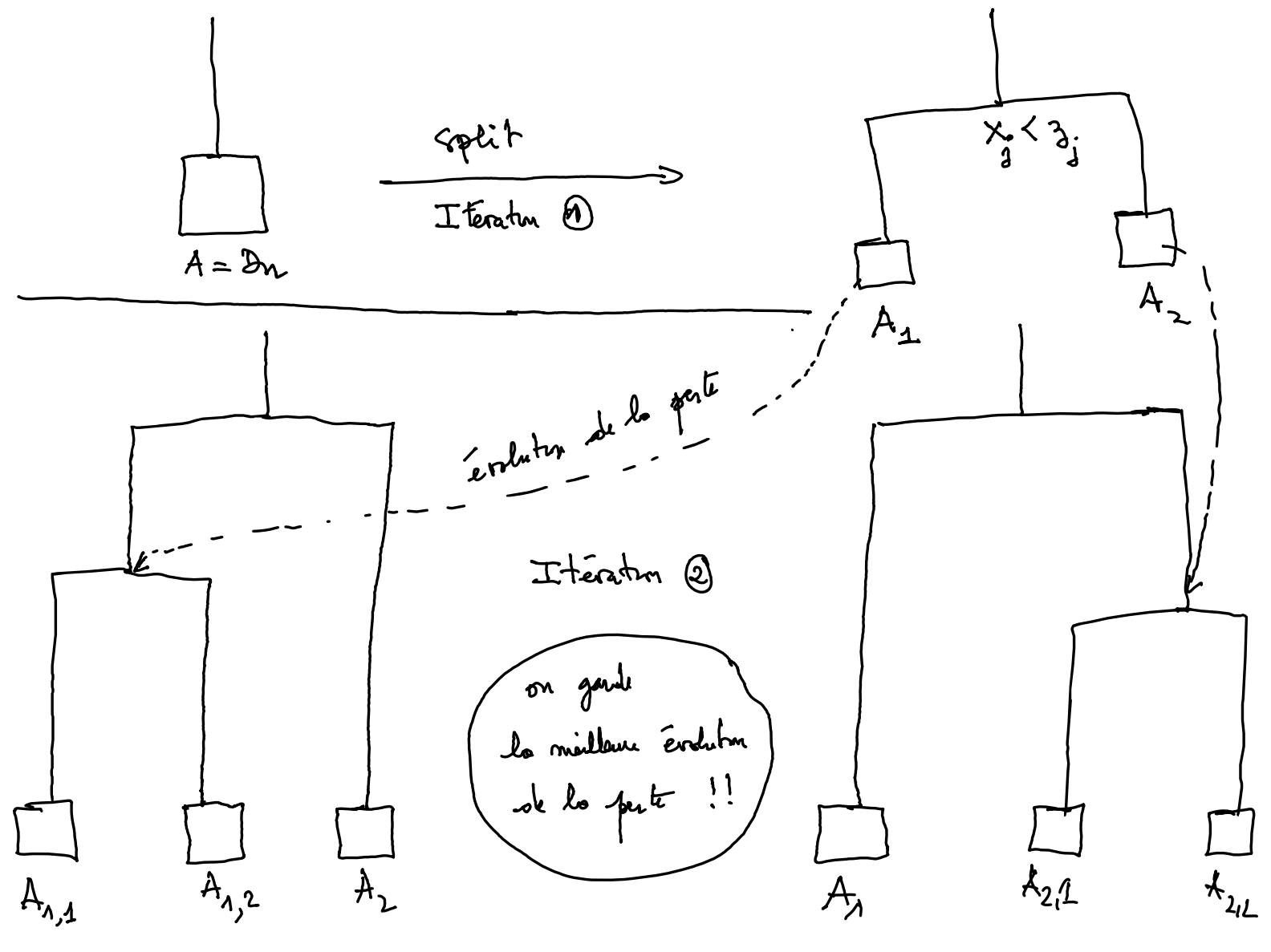
Split d'un noeud A : pour toute variable j

Calculer $L_A(j, z_j)$ comme suit

j :	seuil : z_j	$L_A(j, z_j)$
1	z_1	$L_A(1, z_1)$
2	z_2	$L_A(1, z_2)$
⋮	⋮	⋮
d	z_d	$L_A(d, z_{d1})$

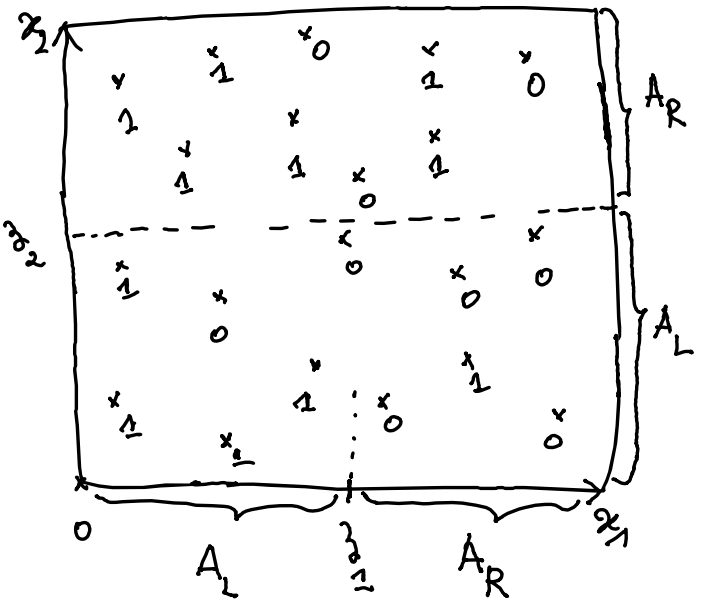
on retient le split qui correspond à

$$\text{argmin}_{j \in \{1, \dots, d\}} L_A(j, z_j).$$


split
Itération ①
$A = D_N$
$X_j < z_j$
A_1
A_2
évolution de la perte
Itération ②
on garde
la meilleure évolution
de la perte !!
$A_{1,1}$
$A_{1,2}$
A_2
A_1
$A_{2,1}$
$A_{2,2}$

En classification : le jeu de données $\mathcal{D}_n = \{(x_1, y_1), \dots, (x_n, y_n)\}$

$x \in \mathbb{R}^d$ et $y \in \{0, 1\}$ (classification).



Indice ou mesure d'impureté de Gini :

$k = 0$ ou 1

$$\hat{\phi}_A(j,k) = \frac{\sum_{i=1}^{n} \mathbb{1}\{y_i = k\} \times \mathbb{1}\{x_i^{(j)} \in A\}}{\sum_{i=1}^{n} \mathbb{1}\{x_i^{(j)} \in A\}}$$

Condition sur $x_i^{(j)}$

$$L_A(j,s) = \underbrace{\left(1 - \sum_{k=0}^{1} \left[\hat{\phi}_{A_L}(j,k)\right]^2\right)}_{\gamma_{A_L}} + \underbrace{\left(1 - \sum_{k=0}^{1} \left[\hat{\phi}_{A_R}(j,k)\right]^2\right)}_{\gamma_{A_R}}$$

$A = A_L \cup A_R$

→ construction de l'arbre identique à la régression en acceptant l'erreur. Interprété au lieu de la perte quadratique moyenne.

Élagage d'un arbre profond :

Proposition : Le paramètre de complexité α ($\equiv$ cp dans rpart dans R)

$$\alpha = \frac{L(t) - L(T_t)}{|\tilde{T}| - 1}$$

T : un arbre

$\tilde{T}$: les feuilles de l'arbre T.

Rappel :

$$C_\alpha(T) = \underbrace{L(T)}_{\text{perte ou erreur}} + \alpha \underbrace{|\tilde{T}|}_{\text{pénalité}}$$

T_t : est la branche de l'arbre qui part du noeud "t".

pour un noeud $t \in T$, nous avons:

$$C_\alpha(t) = L(t) + \alpha \quad \text{Car un seul noeud terminal "}t\text{".}$$

De manière similaire, pour toute branche $T_t \subset T$, nous avons

$$C_\alpha(T_t) = L(T_t) + \alpha |\tilde{T}_t| .$$

Lorsque $\alpha = 0$, $C_0(t) = L(t) > L(T_t) = C_0(T_t)$.

Cette inégalité est vraie parce que la première étape d'élagage consiste à retirer tous les noeuds terminaux tels que $L(t) = L(t_L) + L(t_R)$.
Pour les noeuds restants, nous avons: $L(t) > L(t_L) + L(t_R)$.
Si on augmente α, $C_\alpha(T_t)$ va augmenter plus vite que $C_\alpha(t)$ parce que $|\tilde{T}_t| > 1$.

Autrement dit, pour un certain α, nous allons avoir $C_\alpha(T_t) = C_\alpha(t)$ — (1)

Donc, on obtient $L(t) + \alpha = L(T_t) + \alpha |\widetilde{T}_t|$

$$\Rightarrow \left(|\widetilde{T}_t| - 1\right)\alpha = L(t) - L(T_t)$$

$$\Rightarrow \alpha = \frac{L(t) - L(T_t)}{|\widetilde{T}| - 1}.$$

Ensuite, le procédé devient itératif pour obtenir une suite $\alpha_1 < \alpha_2 < \alpha_3 < \cdots < \cdots$.