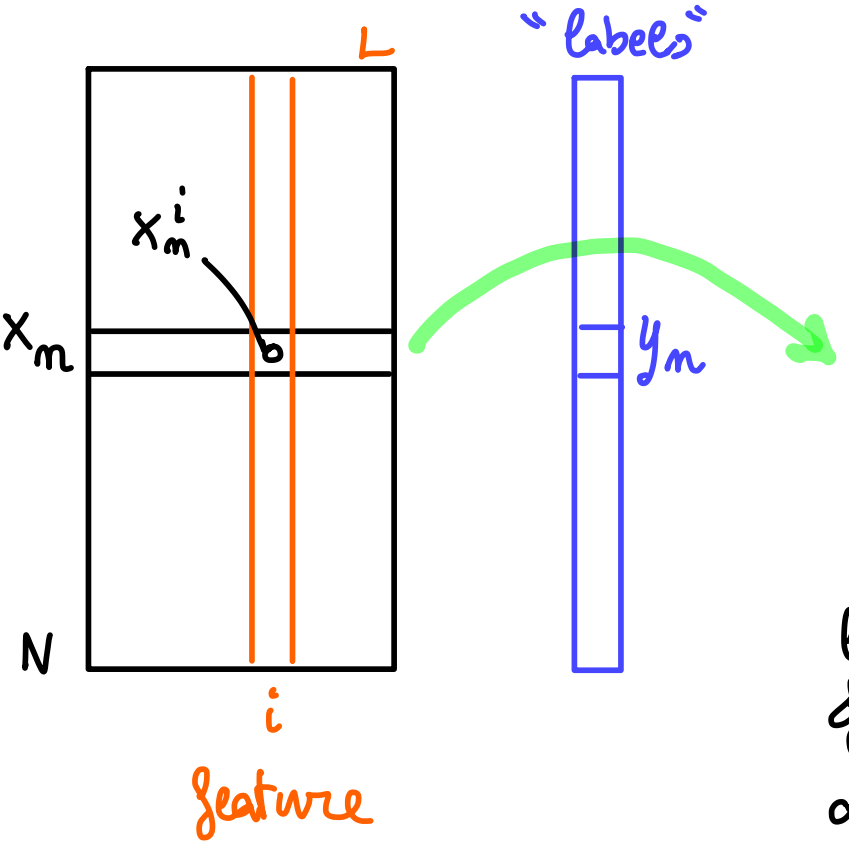
XGBoost

data $\{x_n\}$

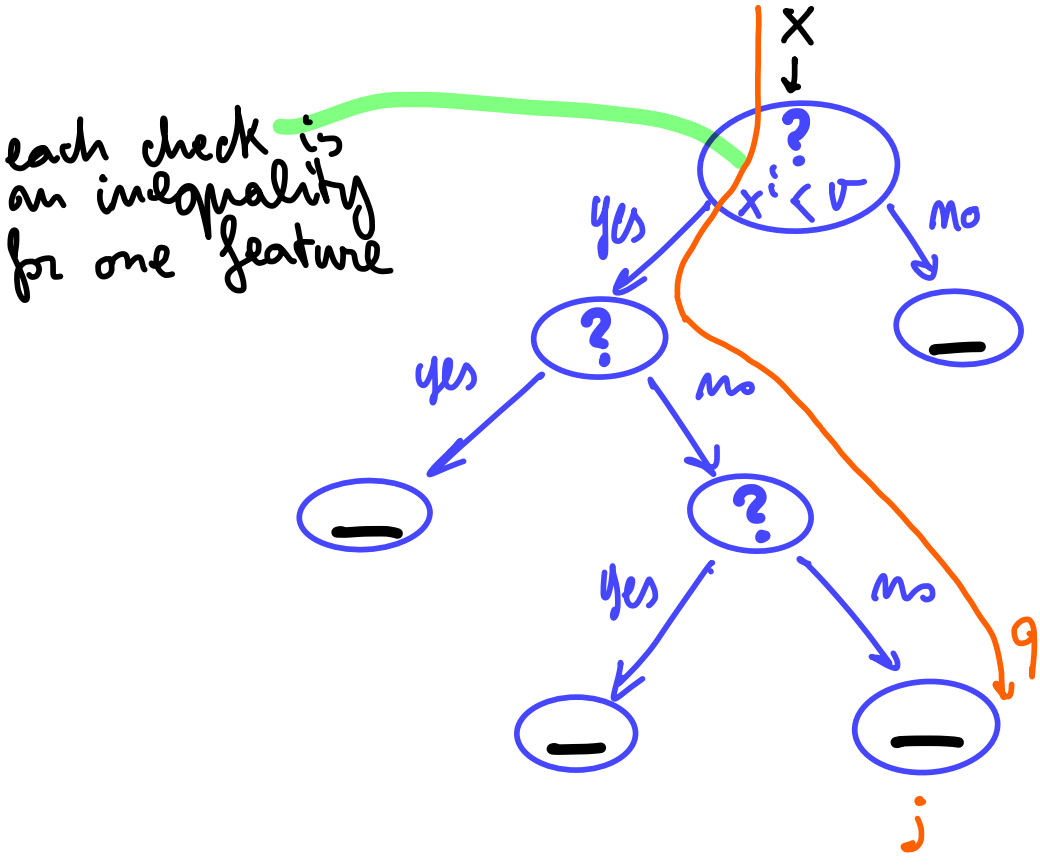


prediction

$$\hat{y}_n = G(x_n) = \sum_{t=1}^{T} g_t(x_n)$$

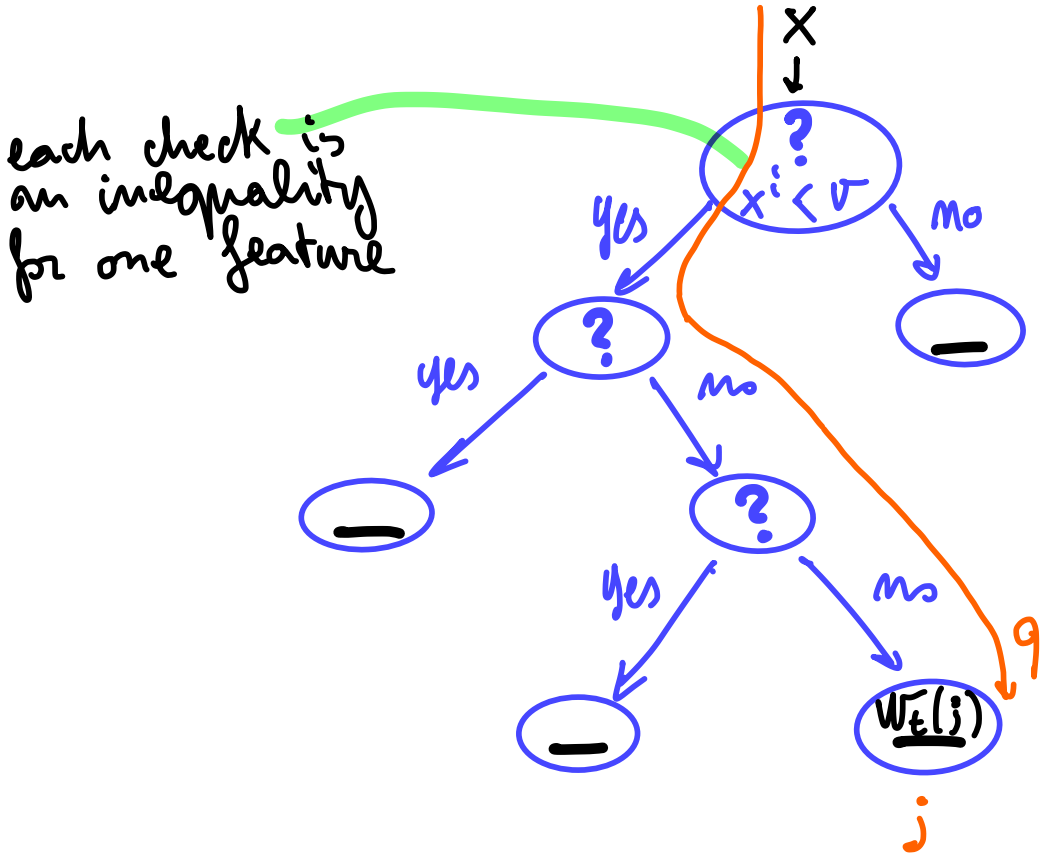
linear combination of $t \leq T$ contrib., one per tree

Prediction of t-th tree



$q(x)$: function mapping sample x to leaf j

Prediction of t-th tree $\Rightarrow$ $g_t(x) = w_t(q(x))$



(q, g)

w

$q(x)$: function mapping sample x to leaf j

Questions

Q_1 • how to find best split $(x^i < v)$ for every node?

Q_2 • when to stop splitting and set a leaf?

Q_3 • why a linear combination of tree's predictions? $\left(G = \sum_{t=1}^{T} g_t \right)$

Loss function

$$C(\{x_n\}, G) = \sum_{n=1}^{N} \ell(y_n, \hat{y}_n) + \sum_{t=1}^{T} \Omega(g_t)$$

convex loss function $\forall$ sample $n \leq N$

e.g.

- square deviation $\frac{1}{2}(y-\hat{y})^2$
- cross entropy for classification $y=(0,1)$

Loss function

$$C(\{x_n\}, G) = \sum_{n=1}^{N} \ell(y_n, \hat{y}_n) + \sum_{t=1}^{T} \Omega(g_t)$$

regularization for every tree $t \leq T$

$$\Omega(g_t) = \gamma \cdot J_t + \frac{\lambda}{2} \|w_t\|^2$$

"cost" of one leaf

J_t = # leaves ($J_t = 4$ in the example)

Ridge on w's

Loss function

$$C(\{x_n\}, G) = \sum_{n=1}^{N} \ell(y_n, \hat{y}_n) + \sum_{t=1}^{T} \Omega(g_t)$$

build C iteratively at every "time" t

$t = 1$

$C_1 \rightsquigarrow C_2 \rightsquigarrow \cdots \rightsquigarrow C_{t-1} \rightsquigarrow C_t \rightsquigarrow \cdots \rightsquigarrow C_T \equiv C$

$t \gg 1$: perturbation expansion, Taylor

Loss function, up to time τ

$$C_\tau(\{x_n\}, G) = \sum_{n=1}^{N} \ell\left(y_n, \hat{y}_n^{(\tau)}\right) + \sum_{t=1}^{\tau} \Omega(g_t)$$

$$\hat{y}_n^{(\tau)} = \hat{y}_n^{(\tau-1)} + g_\tau(x_n)$$

prediction up to tree $\tau-1$

addition to prediction from tree τ

Taylor (answers R_3)

$$\simeq C_{\tau-1} + \Delta C_\tau$$

Contribution to loss function from tree τ

Taylor expansion

$$\Delta C_\tau = \sum_n \left[\partial_z l(y_n, z)\Big|_{z=\hat{y}_n^{(\tau-1)}} g_\tau(x_n) + \frac{1}{2}\partial_z^2 l(y_n, z)\Big|_{z=\hat{y}_n^{(\tau-1)}} g_\tau(x_n)^2 \right] + \Omega(f_\tau)$$

def. a_n, b_n

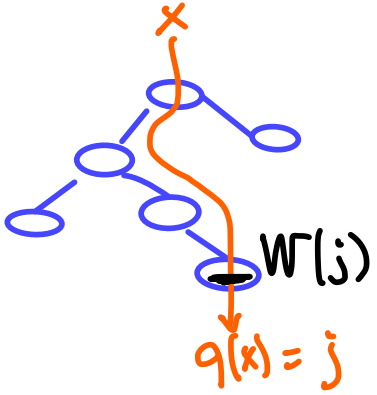
$$\equiv \sum_n \left[a_n g_\tau(x_n) + \frac{1}{2} b_n g_\tau(x_n)^2 \right] + \Omega(f_\tau)$$

⚠ neglecting terms $O(g_t^3)$

Contribution to loss function from tree τ

Taylor expansion

$$\Delta C_\tau = \sum_n \left[a_n \, g_\tau(x_n) + \frac{1}{2} b_n \, g_\tau(x_n)^2 \right] + \Omega(g_\tau)$$

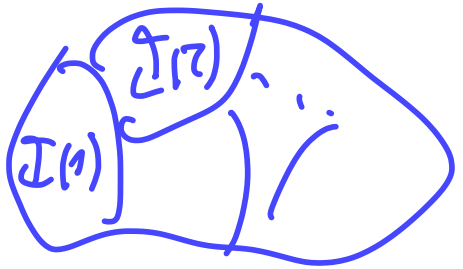


all samples sent by tree τ (via $q_\tau(x_n)$) to the same leaf j will have the same $g_\tau(x_n) = w_\tau(j)$

Contribution to loss function from tree τ

Taylor expansion

$$\Delta C_\tau = \sum_n \left[a_n \, g_\tau(x_n) + \frac{1}{2} b_n \, g_\tau(x_n)^2 \right] + \Omega(g_\tau)$$



all samples sent by tree τ (via $q_\tau(x_n)$) to the same leaf j will have the same $g_\tau(x_n) = W_\tau(j)$

rewrite $\sum_{n \text{ samples}} \dots = \sum_{j=1 \text{ leaves}}^{J_\tau} \sum_{n \in I_\tau(j)} \dots$

Indicator $I_\tau(j) = \{ n \mid q_\tau(x_n) = j \}$

Contribution to loss function from tree τ

Taylor expansion

$$\Delta C_\tau = \sum_{m} \left[a_m \, g_\tau(x_m) + \frac{1}{2} b_m \, g_\tau(x_m)^2 \right] + \Omega(g_\tau)$$

regularization

$$= \sum_{j=1}^{J_\tau} \left[\left(\sum_{m \in I_\tau(j)} a_m \right) w_\tau(j) + \frac{1}{2} \left\{ \left(\sum_{m \in I_\tau(j)} b_m \right) + \lambda \right\} w_\tau(j)^2 \right]$$

$A_\tau(j)$ $\qquad$ $B_\tau(j)$

$$+ \gamma J_\tau$$

Example of A, B, for square deviation loss f.

($\hat{y}_n$ up to previous tree)

$$\ell(y_n, \hat{y}_n) = \frac{1}{2}(\hat{y}_n - y_n)^2$$

$$a_n = \partial_z \ell(y_n, z)\Big|_{z=\hat{y}_n} = \hat{y}_n - y_n \Rightarrow A(j) = \sum_{n \in I(j)} (\hat{y}_n - y_n)$$

sum of residuals in leaf j

$$b_n = \partial_z^2 \ell(y_n, z)\Big|_{z=\hat{y}_n} = 1 \Rightarrow B(j) = \sum_{n \in I(j)} 1$$

samples that $q(x)$ sends to leaf j

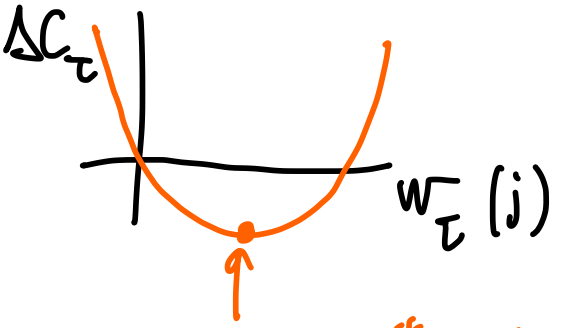
Contribution to loss function from tree τ

$$\Delta C_\tau = \sum_{j=1}^{J_\tau} \left[A_\tau(j)\ W_\tau(j) + \frac{1}{2}\left(B_\tau(j) + \lambda\right) W_\tau(j)^2 \right] + \gamma J_\tau$$

$\forall$ leaf j, parabola

$$\frac{B+\lambda}{2} w^2 + Aw + \gamma J$$

$$B+\lambda > 0$$



best $W_\tau^*(j) = -\dfrac{A_\tau(j)}{B(j)+\lambda}$

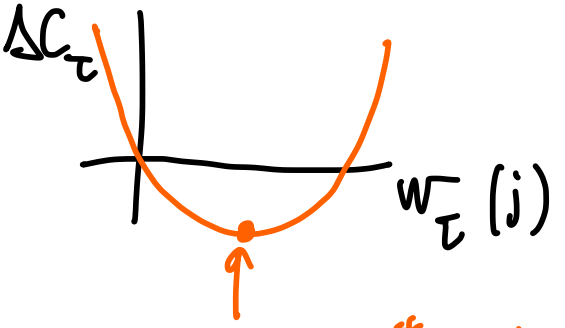
Contribution to loss function from tree τ

$$\Delta C_\tau = -\frac{1}{2} \sum_{j=1}^{J_\tau} \frac{A_\tau(j)^2}{B_\tau(j)+\lambda} + \gamma J_\tau$$

optimized with Newton's method (w^*)

⚠ very quick! no gradient descent

$\forall$ leaf j, parabola



best $w_\tau^*(j) = -\frac{A_\tau(j)}{B(j)+\lambda}$

Questions

Q_1 • how to find best split $(x^i < v)$ for every node?

Q_2 • when to stop splitting and set a leaf?

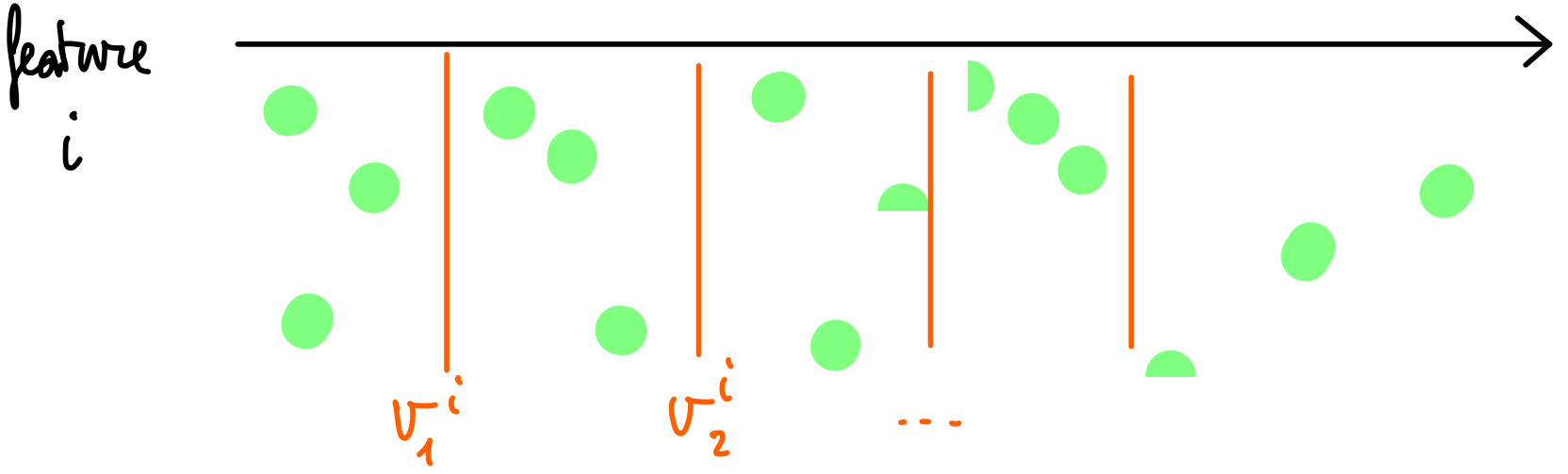
⚠ There is a huge amount of possible combinations (g, q)

weights tree structure

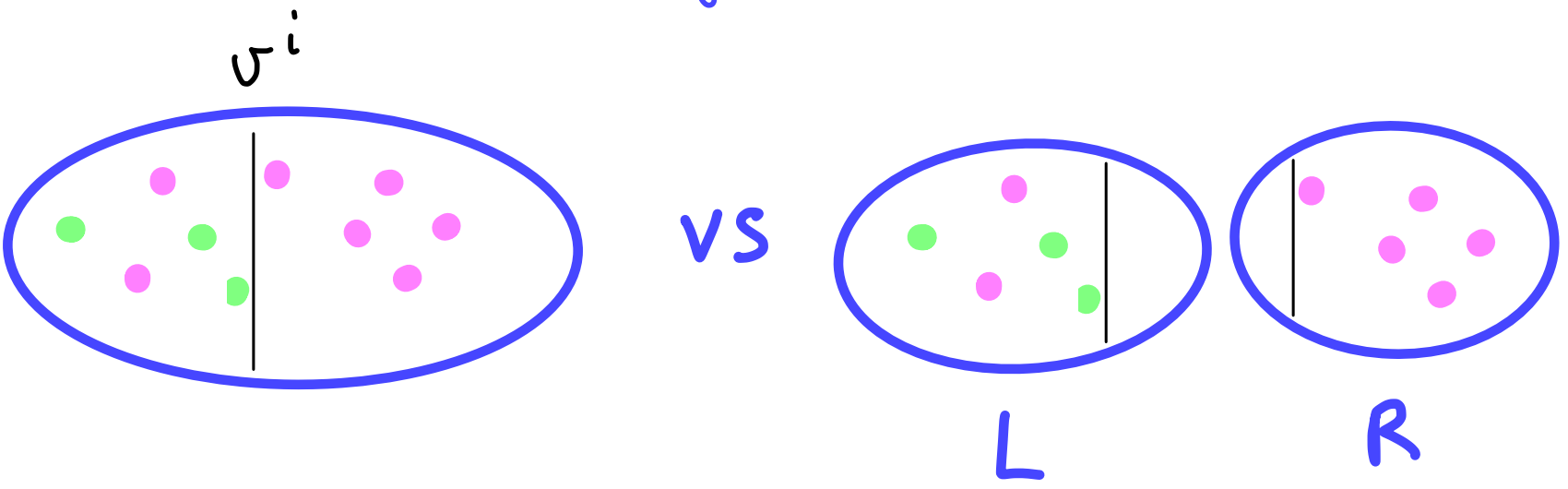
Practical implementation of XGBoost

for many samples ($N \gg 1$) with many features ($L \gg 1$)

- partition each feature in percentiles v^i



Split



Is the original leaf on the left better or worse than the two leaves on the right?

Practical implementation of XGBoost

Greedy algorithm

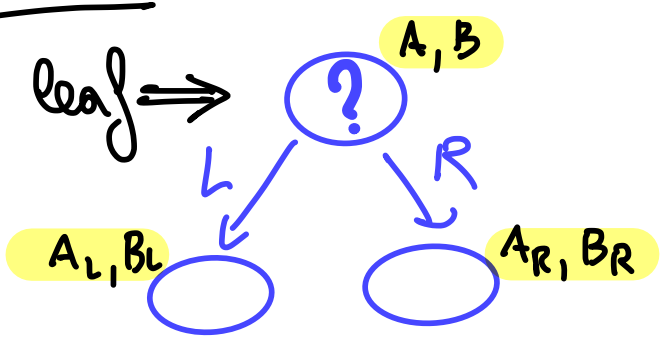
in every current leaf $\Rightarrow$? A, B

try many splits
(∀ percentile)

Practical implementation of XGBoost

Greedy algorithm

in every current leaf ⟹



try many splits (∀ percentile)

choose the one providing smallest ΔC_{SPLIT}

$$\Delta C_{SPLIT} = -\frac{1}{2}\left(\frac{A_L^2}{B_L+\lambda} + \frac{A_R^2}{B_R+\lambda} - \frac{A^2}{B+\lambda}\right) + \gamma$$

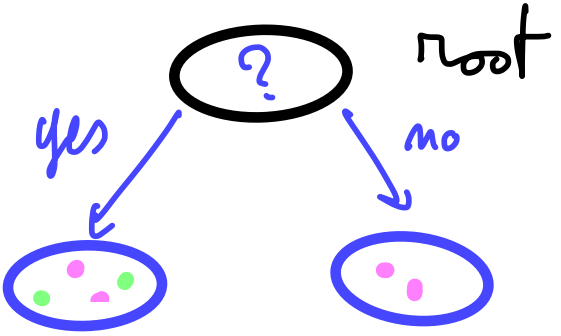
$\frac{A_L^2}{B_L+\lambda} + \frac{A_R^2}{B_R+\lambda}$: L/R split

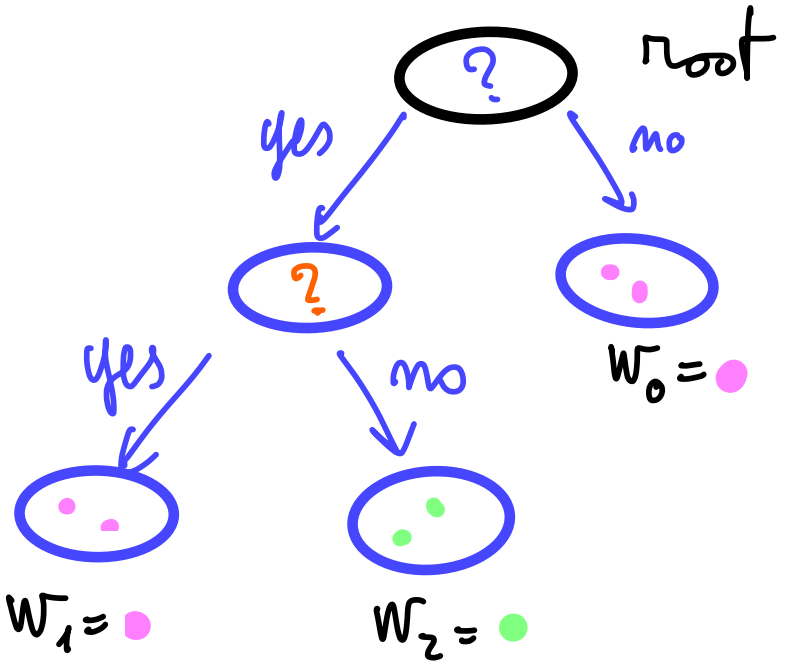
$\frac{A^2}{B+\lambda}$: all together ⟹ Q_1 ✓

- start from a root node
- iterate split for every node until
  a) it is not convenient (all $\Delta C_{SPLIT} > 0$)
  b) conditions max depth, etc., are met

$\Rightarrow Q_2 \checkmark$


root


?
root
yes
no



this was too easy

normally the 1st tree does not solve it all, and the $\hat{y}_m - y_m$ difference is the starting point for the next tree