x^1	x^2	y
1.2	1	a
1.4	0.3	a
0.5	-0.2	b
0.3	0	b

How to perform the classification tree?

Remember in regression we do the computation for one division δ & $\Delta R(t, \delta)$

For the classification we have to compute

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

? what is imp

$$imp(t) = \underset{\uparrow}{h}(p(1|t), \ldots, p(I,t))$$

???

In general, the imp f^{ct} is the Gini function defined by

$$imp(t) = \sum_{i \neq j} p(i|t) \times p(j|t)$$

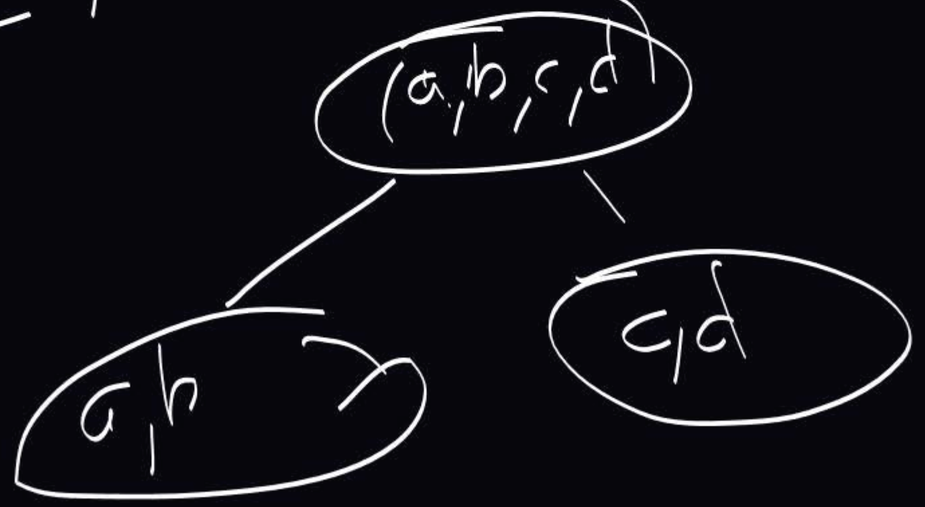
$$\Rightarrow h(p_1, \ldots, p_I) = \sum_{i \neq j} p_i p_j$$

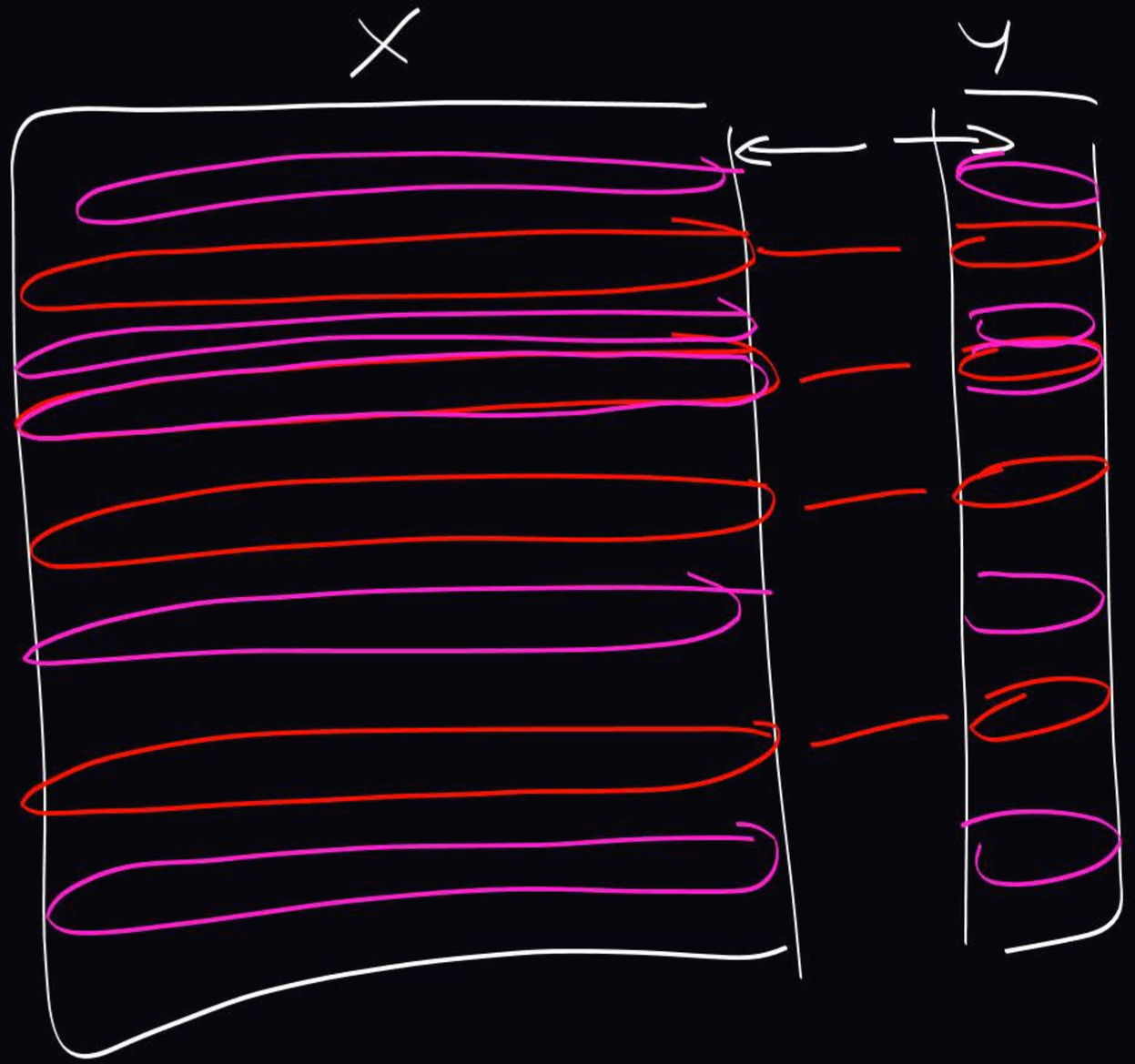
$$imp(t) = \sum_{i=1}^{I} \left(p(i|t) \times \sum_{j \neq i} p(j|t) \right)$$

$$= \sum_{i=1}^{I} p(i|t) \times \left(1 - p(i|t)\right)$$

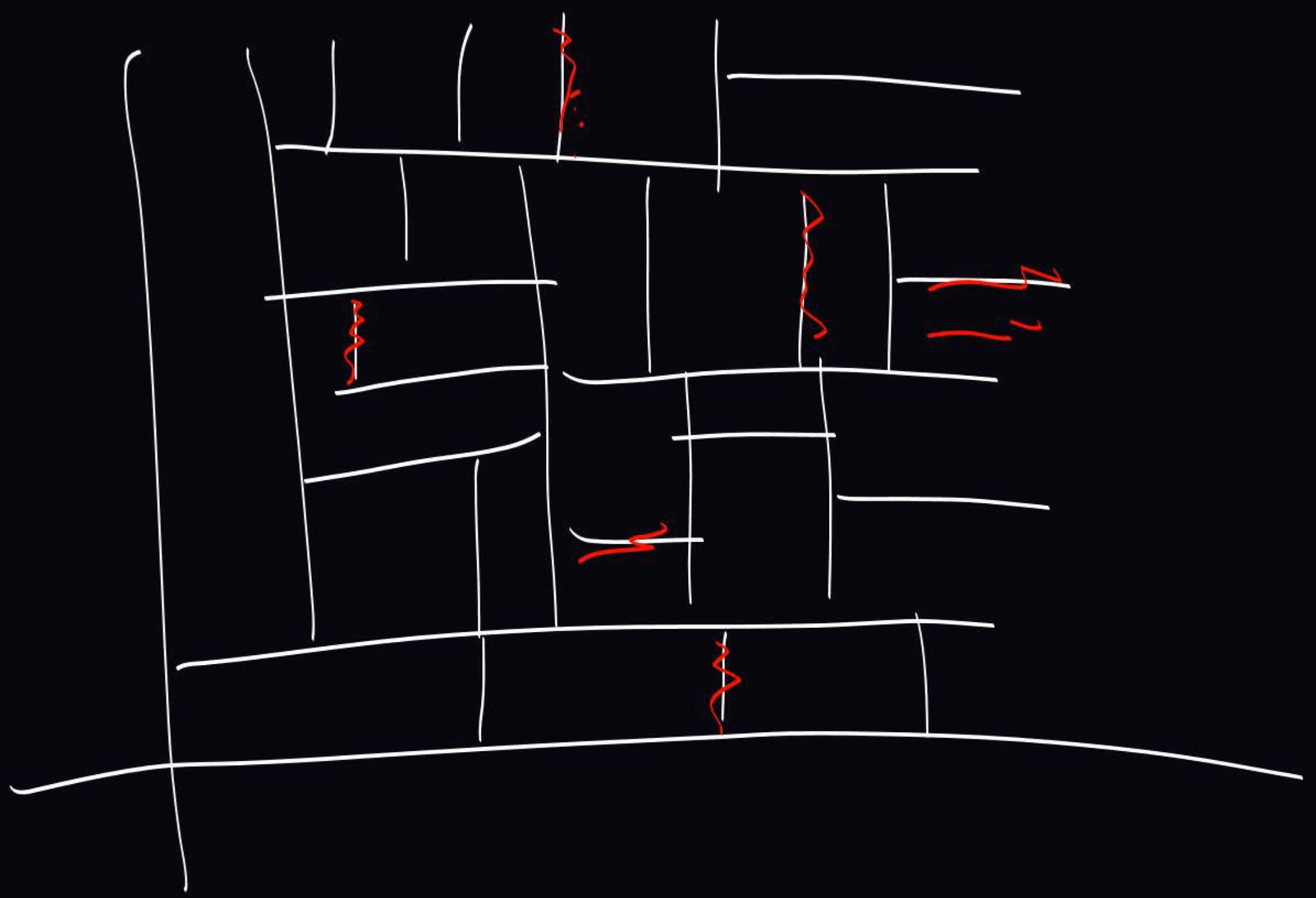
Shanon

$$\sum_{i=1}^{I} -p_i \log p_i$$




X
Y

At the end of the explained construction of the maximal tree we get a tree that cannot be used in practice because it is too complex and it overfits the data.



The idea is thanks to the maximal tree, we want to create a smaller tree!

a criterion: $crit_\alpha(T) = \underbrace{error(T)}_{\text{goodness-of-fit}} + \underbrace{\frac{\alpha}{n}|\tilde{T}|}_{\text{complexity}}$

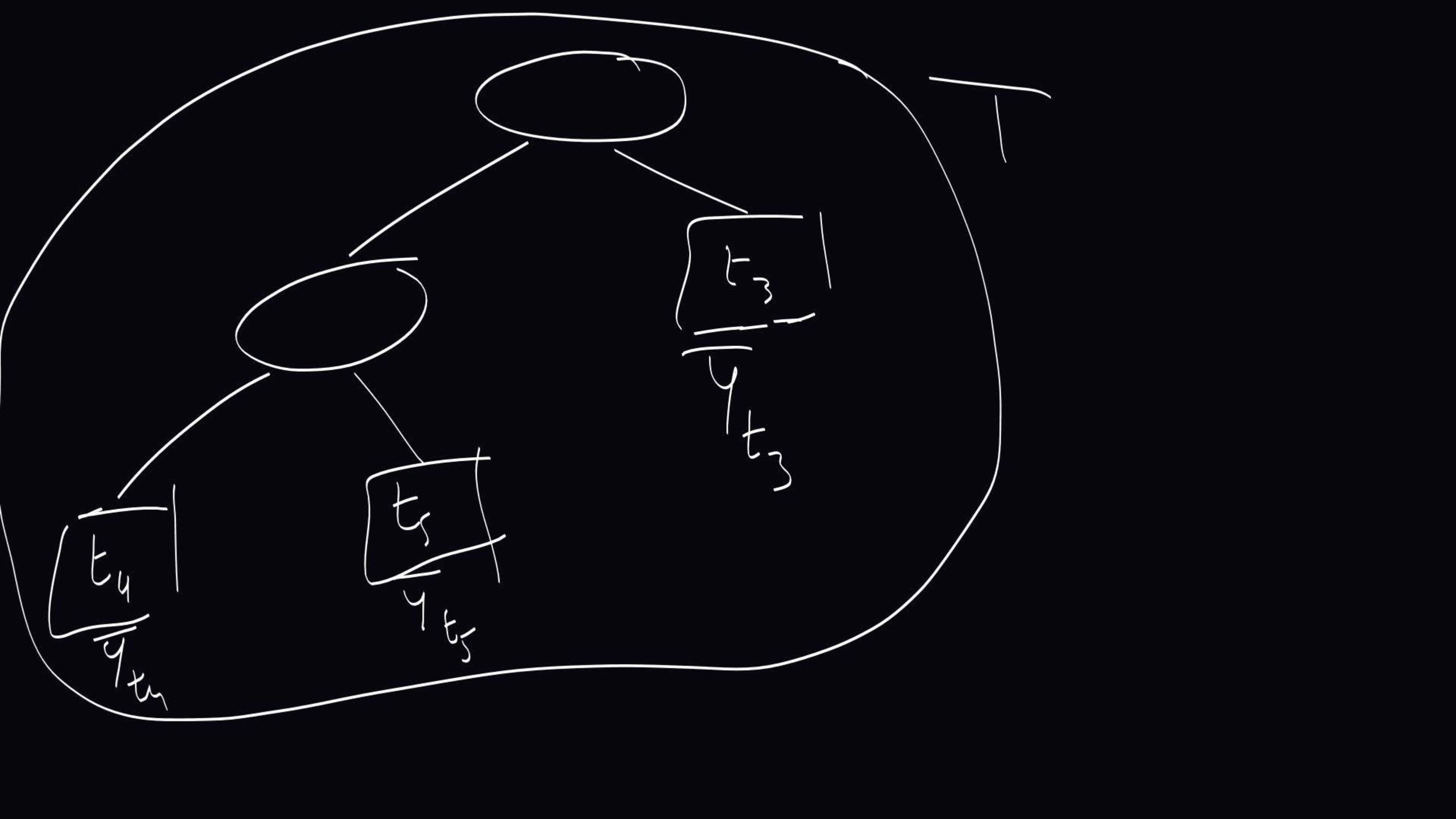
with $\alpha \geq 0$, $|\tilde{T}|$ the number of leaves of the tree T.

α

in R, this is proportional to CP

$$error(T) = \begin{cases} \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_{T,i})^2 \\ = \frac{1}{n}\sum_{t \in T_n}\sum_{x_i \in t}(y_i - \overline{y}_t)^2 & \text{in regression} \\ \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{y_i \neq \hat{y}_{T,i}} & \text{in classification} \end{cases}$$

where $\hat{y}_{T,i}$: is the prediction thanks to the T for the individual i


T
t_3
γ_{t_3}
t_5
γ_{t_5}
t_4
γ_{t_4}

The best tree associated to α is defined by:

$$T_\alpha = \begin{cases} \underset{T' \preceq T}{\operatorname{argmin}}\ crit_\alpha(T') \\ \text{if } crit_\alpha(T_\alpha) = crit_\alpha(T') \\ \text{then } T_\alpha \preceq T' \end{cases}$$

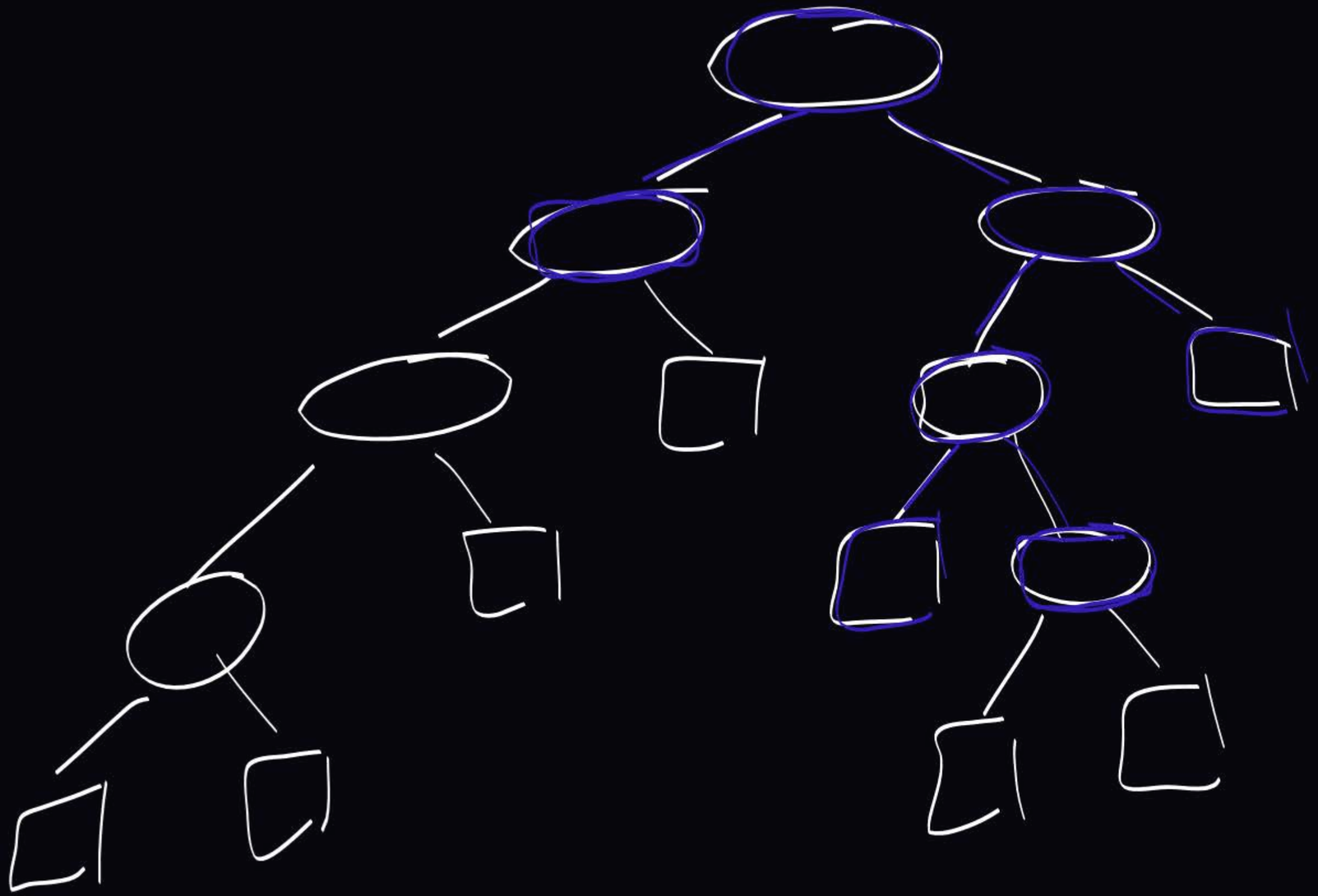
$$crit_\alpha(T_\alpha) = \min_{T' \preceq T} crit_\alpha(T')$$

and T_α is the smallest sub-tree of T that minimizes $crit_\alpha$.

where $T' \lhd T$ means

T' is a sub-tree of T

T' sub-tree of T means that T' is obtained from T by suppressing nodes of T.



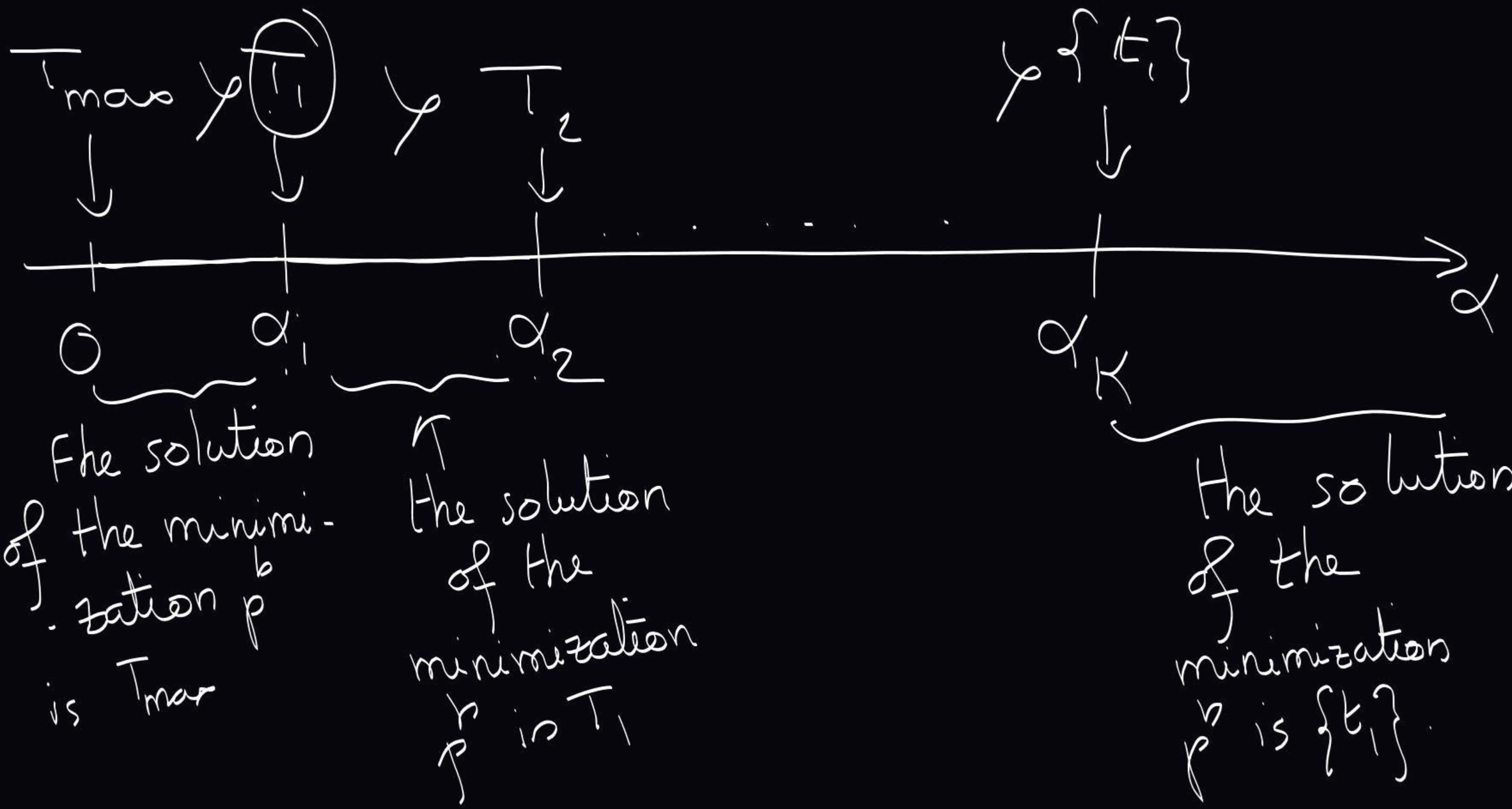
Rk: for $\alpha = 0$, T_α is the maximal tree.

Then we increase α. Thus the complexity of T_α becomes smaller.

Theorem:

There exists a sequence $(\alpha_k)_{k \in \{0,\dots,K\}}$ with is data-driven such that:

- $\alpha_0 = 0$ and $\overline{T_0} = \overline{T}_{max}$
- $\alpha_1 < \alpha_2 < \dots < \alpha_K$
- $\overline{T_0} \supset \overline{T_1} \supset \dots \supset \overline{T_K} = \{t_1\}$ with $\overline{T_k} = \overline{T_{\alpha_k}}$
- $\forall k \in \{0,\dots,K-1\}$, $\forall \alpha \in [\alpha_k, \alpha_{k+1}[$, $\overline{T_\alpha} = \overline{T_k}$


T_{max}
T_1
T_2
$\{t_1\}$
0
α_1
α_2
α_K
α
The solution of the minimization p^b is T_{max}
The solution of the minimization p^b is T_1
The solution of the minimization p^b is $\{t_1\}$.

Rk:

- The sequence of sub-tree that is created is a sequence of nested sub-trees!
- If $|\tilde{T}_{max}| = m$, your sequence of sub-trees induced by the minimization problem does not necessary contains sub-tree for all the complexity between 1 and m.

⇒ the creation of the sequence of sub-trees is the second step of the CART algorithm. The name of this second step is the pruning step !

p^b: at the end of the second step, we have $(k+1)$ estimators.

How to chose one?

This is the goal of step 3 which is called final selection.

Final selection: 2 ways

1) test sample
2) Cross-validation

test sample: We assume that we have individuals that have not been used in the construction of T_{max}, and of $T_1, \dots, T_K$.

We compute, for an individual j of the test sample, the prediction induced by $\overline{T}_{max}, \overline{T}_1, \ldots, \overline{T}_K$.

$\Rightarrow$ for the individual j we have $(K+1)$ predictions

We denote them $\hat{y}_{j,k}$ where $\hat{y}_{j,k}$ is the prediction for individual j by $\overline{T}_k$.

We compute the error in test for all the sub-trees $T_0, \dots, T_K$ by

$$\text{terror}_k = \begin{cases} \frac{1}{m} \sum_{j=1}^{m} \left(y_j - \hat{y}_{j,k} \right)^2 & \text{regression} \\ \frac{1}{m} \sum_{j=1}^{m} \mathbb{1}_{y_j \neq \hat{y}_{j,k}} & \text{classification} \end{cases}$$

(m : nb of individuals in the test sample)

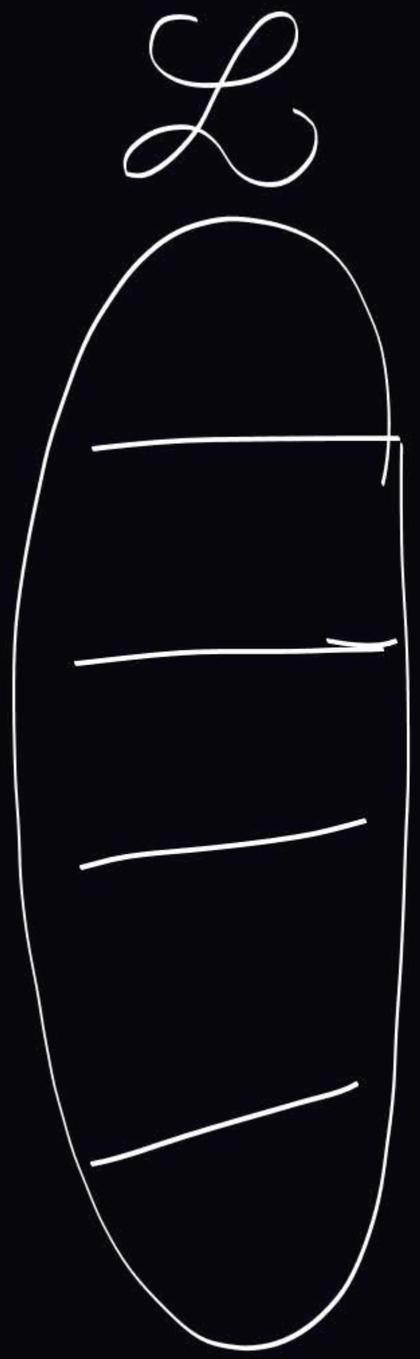
The final selection is the sub-tree associated to the smallest test error.

Cross-Validation

$\mathcal{L} = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ training sample

We apply a method A to construct a decision rule $\hat{f}$ to estimate f.

$$\mathcal{L} \xrightarrow{A} \hat{f}$$



→ a partition of $\mathcal{L}$ in V elements, created at random.

$\mathcal{L}_1$
$\mathcal{L}_2$
⋮
$\mathcal{L}_V$

Each element of the partition has quite the same number of observations $\left(\simeq \frac{n}{V}\right)$

Step 1:

Consider $\mathcal{L}_2 \cup \dots \cup \mathcal{L}_V = \mathcal{L}_{-1}$

$\frac{R\mathcal{L}}{\#\mathcal{L}_{-1}} \approx \frac{(V-1)}{V} n$

to construct an estimator $\hat{f}_{-1}$

by application of the method A.

We compute the test error of $\hat{f}_{-1}$ thanks to $\mathcal{L}_1$:

err_{-1}

Example:

Linear regression

$$X = \begin{pmatrix} 1 & 1.2 & 2.4 \\ 1 & 1.1 & 3.7 \\ 1 & 0.5 & 1.4 \\ 1 & 2.3 & 0.7 \\ 1 & 0.7 & -0.1 \end{pmatrix}$$

$$Y = \begin{pmatrix} 0.5 \\ 1.2 \\ -0.1 \\ 0.3 \\ 1.5 \end{pmatrix}$$

$$\hat{\beta} = (X^T X)^{-1} X^T Y$$

$$\hat{f}(T) = T\hat{\beta}$$

$$\hat{f}(x_1, x_2) = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2$$

$$= (1 \; x_1 \; x_2) \times \begin{pmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \hat{\beta}_2 \end{pmatrix}$$

$V = 5$

$\mathcal{L}_1 : \quad x_1 = (1 \;\; 1.2 \;\; 2.4) \quad y_1 = 0.5$

$$X_{-1} = \begin{pmatrix} 1 & 1.1 & 3.7 \\ 1 & 0.5 & 1.4 \\ 1 & 2.3 & 0.7 \\ 1 & 0.7 & -0.1 \end{pmatrix}$$

$$\underline{y}_{-1} = \begin{pmatrix} 1.2 \\ -0.1 \\ 0.3 \\ 1.5 \end{pmatrix}$$

$$\hat{\beta}_{-1}(\tau) = \tau \, \hat{\beta}_{-1}$$

with

$$\hat{\beta}_{-1} = \left(\tau X_{-1}^{T} X_{-1} \right)^{-1} \tau X_{-1}^{T} y_{-1}$$

$$\text{error}_{-1} = (X_1 \hat{\beta}_{-1} - y_1)^2$$

Step 2:

We use $\mathcal{L}_{-2} = \bigcup_{v \in \{1,\dots,V\}, v \neq 2} \mathcal{L}_v$

to construct thanks to the method A an estimator $\hat{f}_{-2}$.

We compute the test error (err_{-2}) of $\hat{f}_{-2}$ by using $\mathcal{L}_2$.

We continue by Step 3, Step 4, ...,
until Step V.

At the end, we have

$$err_1, err_2, \dots, err_V$$

The cross validation error of $\hat{f}$ is given by:

$$err_{CV} = \frac{1}{n} \sum_{i=1}^{V} n_i \, err_i \quad \text{with } n_i = \#\mathcal{L}_i$$

Rk: if $V = n$ this is the leave-one-out cross validation.

For the application to CART algorithm:

$\mathcal{L} \xrightarrow{\text{CART}} T_{max}: T_1, \ldots, T_K \quad (\alpha_0 = 0, \alpha_1, \ldots, \alpha_K)$

$\mathcal{L} \rightarrow$ partition $\mathcal{L}_1, \ldots, \mathcal{L}_V$

Step 1:

$$\mathcal{L}_{-1} = \bigcup_{i \in \{2, \dots, N\}} \mathcal{L}_i$$

we use $\mathcal{L}_{-1}$ to construct $\overline{T}_{max,-1}, \overline{T}_{1,-1}, \dots, \overline{T}_{K_{(-1)},-1}$

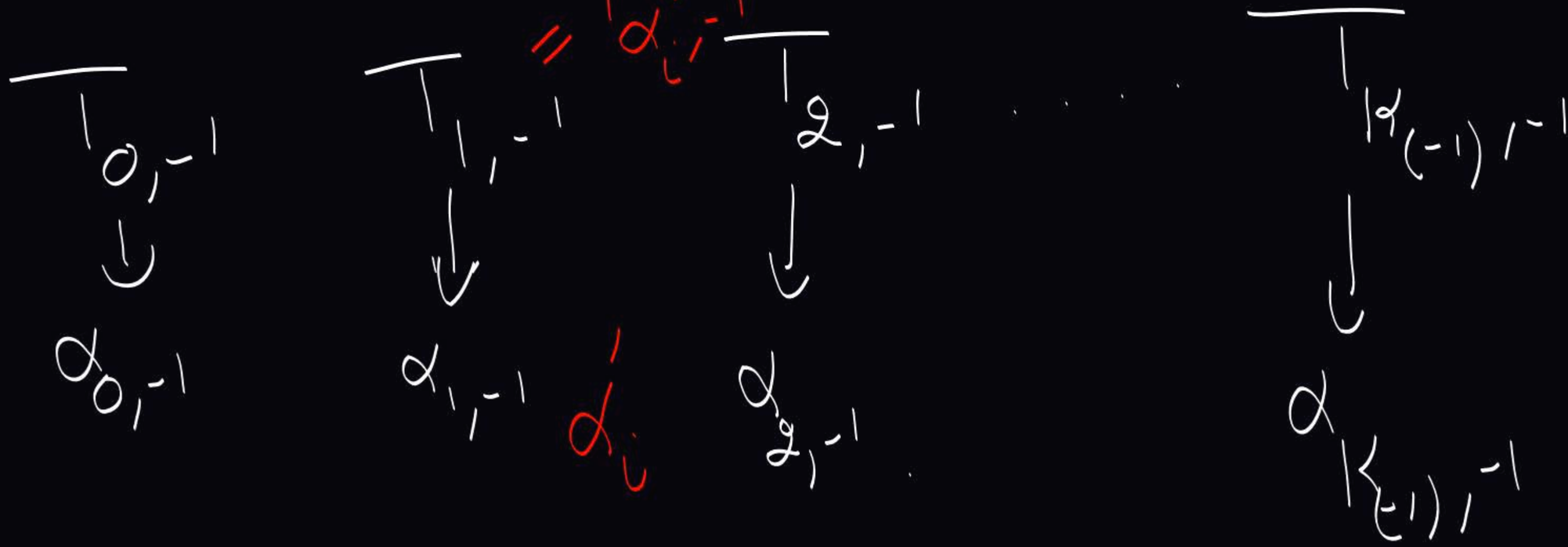
we use $\mathcal{L}_1$ to compute the test error associated

to this sequence:

$$\alpha_0' = \sqrt{\alpha_0 \alpha_1}, \quad \alpha_1' = \sqrt{\alpha_1 \alpha_2}, \quad \alpha_2' = \sqrt{\alpha_2 \alpha_3}, \dots, \quad \alpha_K' = \sqrt{\alpha_K (\alpha_{K+1})}$$

for $i \in \{0, \ldots, k\}$

$err_{i,-1}$ the test error using $\mathcal{L}_1$ of $T_{\alpha'_{i,-1}}$



Example:

$\alpha_0' = 0 \quad \alpha_1' = 0.2 \quad \alpha_2' = 0.3$

Step 1 $\overline{T}_{0,-1}$ $\overline{T}_{1,-1}$ $\overline{T}_{2,-1}$ $\overline{T}_{3,-1}$

$\alpha_{0,-1} = 0 \quad \alpha_{1,-1} = 0.05 \quad \alpha_{2,-1} = 0.1, \quad \alpha_{3,-1} = 0.4$

$err_{1,-1}$ = test error of $\overline{T}_{2,-1}$

$err_{2,-1} =$ _____

We repeat the same idea for step 2 until step V.

step 1 : $err_{0,-1}$ $err_{1,-1}$ ... $err_{K_1,-1}$

step 2 : $err_{0,-2}$ $err_{1,-2}$ ... $err_{K_1,-2}$

step V : $err_{0,-V}$ $err_{1,-V}$ ... $err_{K_1,-V}$

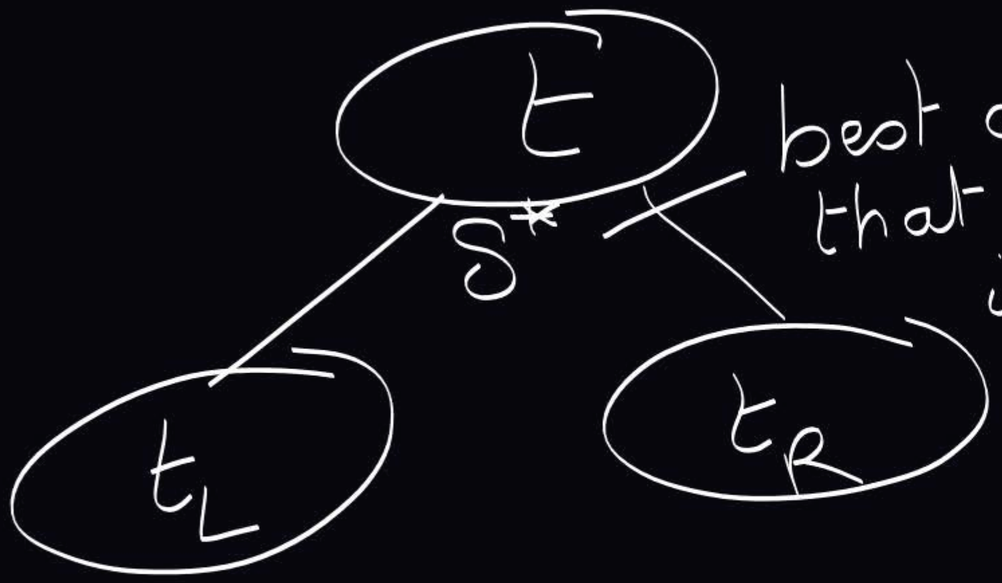
cross validation error of T_i is given by:

$$r_{cv}(T_i) = \sum_{j=1}^{V} err_{\mathcal{L}_j} \times \frac{n_j}{n}$$

with $n_j = \#\mathcal{L}_j$

⇒ To select the best sub-tree, we keep the one associated to the minimal value of the cross-validation error!

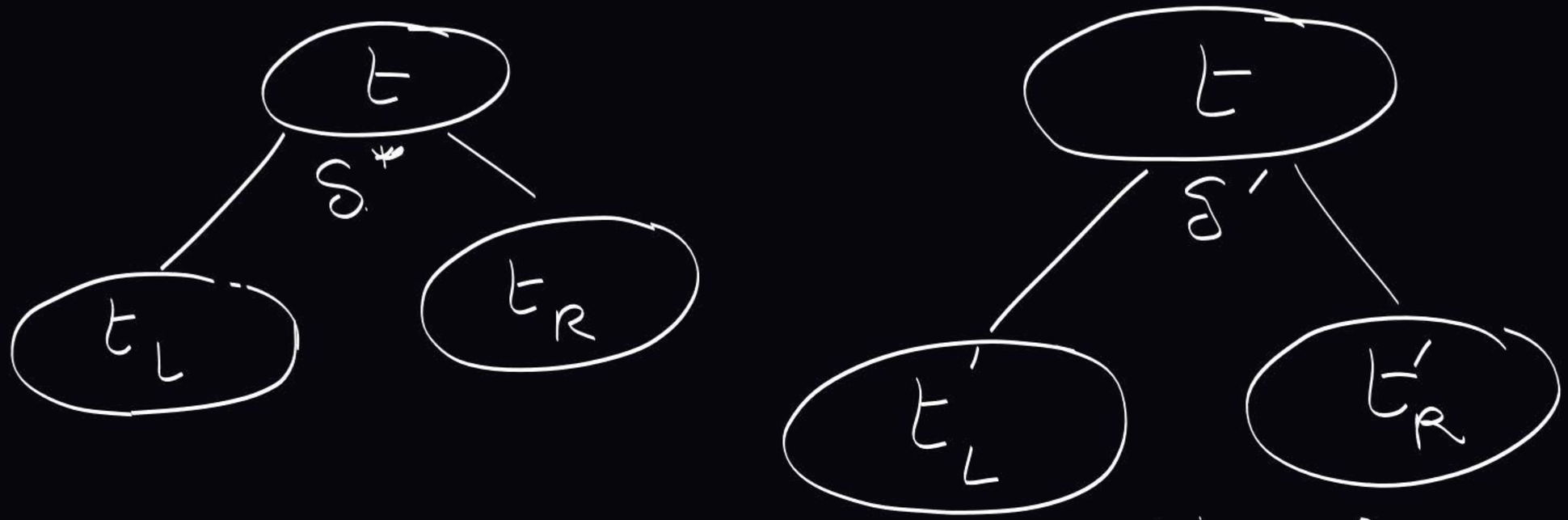
Surrogate splits



best division that used for instance the variable X^1.

We are going to define the best division or X^2 or X^3 or X^4... or X^p which will be "similar" to s^*.

"Similar"



$p_{LL'}$ = the proportion of observations of t that are in t_L and in $t_{L'}$

$p_{RR'}$ also

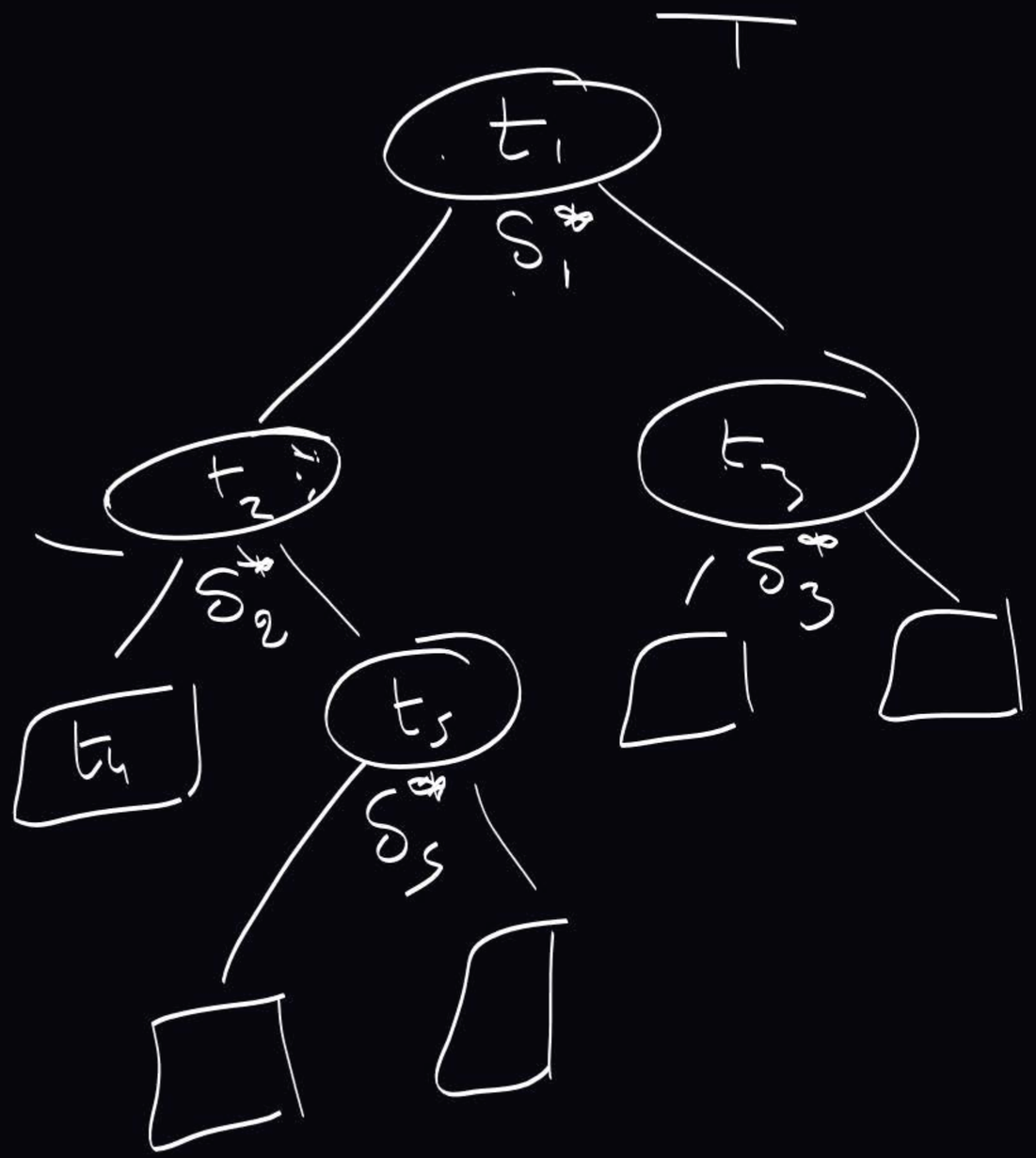
We would like to maximize

$$P_{LL'} + P_{RR'}$$

Variable importance

$$VI(X^i) = \begin{cases} \sum_{t \in T - \tilde{T}} \Delta' imp(t) & \text{classi-fication} \\ \sum_{t \in \bar{T} - \tilde{T}} \Delta' R(t) & \text{regression} \end{cases}$$

$\bar{T} - \tilde{T}$: all the nodes of T that are not a leaf



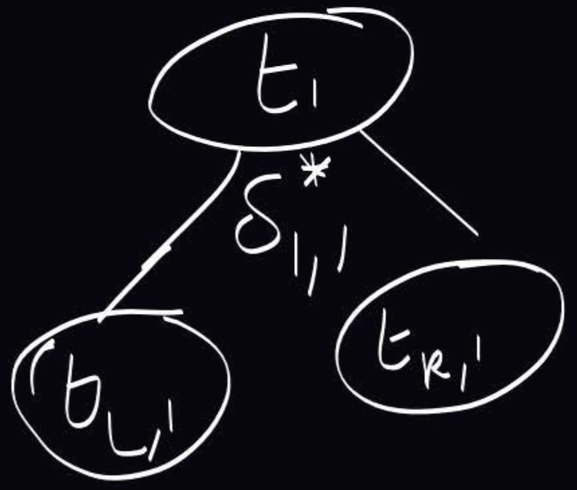
Assume that we want to compute the variable importance of X^l.

If s_1^* uses X^l we are just compute $\Delta' imp(t_1)$

$= \Delta imp(t_1)$

If s_1^* does not use X'.

We consider $s_{1,1}^*$ which is the best surrogate split of t_1 that uses X'.



$$\Delta imp(t_1) = imp(t_1) - p_R\, imp(t_{R,1}) - p_L\, imp(t_{L,1})$$

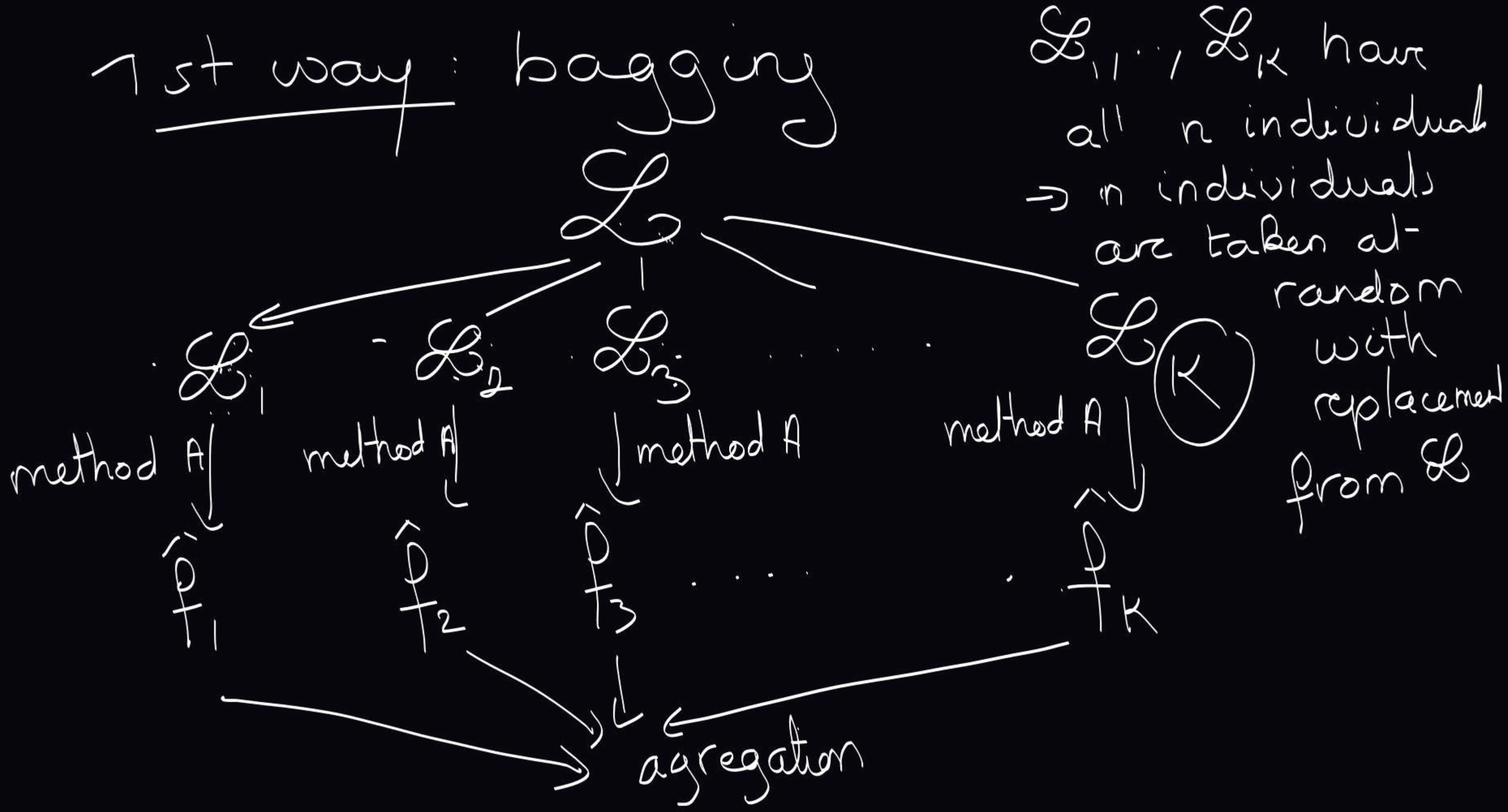
often:

$$VI(x^i) = 100 \times \frac{VI(x^i) - \min_i VI(x^i)}{\max_i (x^i)}$$

P^b: A CART tree is unstable.
So if we change just a little the data, we may have something different

→ stabilization

1st way: bagging



$\mathcal{L}_1, \ldots, \mathcal{L}_K$ have all n individuals $\Rightarrow$ n individuals are taken at-random with replacement from $\mathcal{L}$

aggregation of $\hat{f}_1, \dots, \hat{f}_K$:

- in regression : $\hat{f}(x) = \frac{1}{K} \sum_{i=1}^{K} \hat{f}_i(x)$

- in classification :

the response variable takes values in $\{1, \dots, I\}$.

Let x be an observation

$$\forall i \in \{1, \dots, I\}, \quad n_i(x) = \#\left\{ j \in \{1, \dots, K\} \text{ s.t. } \hat{f}_j(x) = i \right\}$$

$$\hat{f}(x) = \underset{i \in \{1, \dots, I\}}{\operatorname{argmax}} \; n_i(x)$$

CART $\mathcal{L}$: all your data

1) $\downarrow T_{max}$

2) $T_0, T_1, \ldots, T_K$

3) $\bar{T}_i$ by cross validation