

# Data Science

## Decision Tree Induction

---

Séverine Affeldt

Université Paris Cité  
Centre Borelli, UMR 9010, Equipe AI-DSCy  
UFR Sciences Fondamentales et Biomédicales

2023 – 2024

# Objective

- Recursively **partitionning** the training records into successively **purer** subsets
- Representing this partitionning with a decision **tree**

x	Attributes $\{A_i\}$							y
Name	Temperature	Skin Cover	Gives Birth	Aquatic Creature	Aerial Creature	Has Legs	Hibernates	Class Label
human	warm-blooded	hair	yes	no	yes	yes	no	mammal
python	cold-blooded	scales	no	no	no	no	yes	reptile
salmon	cold-blooded	scales	no	yes	no	no	no	fish
whale	warm-blooded	hair	yes	yes	no	no	no	mammal
frog	cold-blooded	none	no	semi	yes	yes	yes	amphibian
komodo	cold-blooded	scales	no	no	yes	yes	no	reptile
bat	warm-blooded	hair	yes	no	yes	yes	yes	mammal
pigeon	warm-blooded	feathers	no	no	yes	yes	no	bird
cat	warm-blooded	fur	yes	no	yes	yes	no	mammal
leopard	cold-blooded	scales	yes	yes	no	no	no	fish
turtle	cold-blooded	scales	no	semi	no	yes	no	reptile
penguin	warm-blooded	feathers	no	semi	no	yes	no	bird
porcupine	warm-blooded	quills	yes	no	no	yes	yes	mammal
eel	cold-blooded	scales	no	yes	no	no	no	fish
salamander	cold-blooded	none	no	semi	no	yes	yes	amphibian

if  $y \in \{\text{mammal, non-mammal}\}$

$\Rightarrow$  split  $\{x\}$  into as much as possible homogenous subsets from the point of view of  $y$

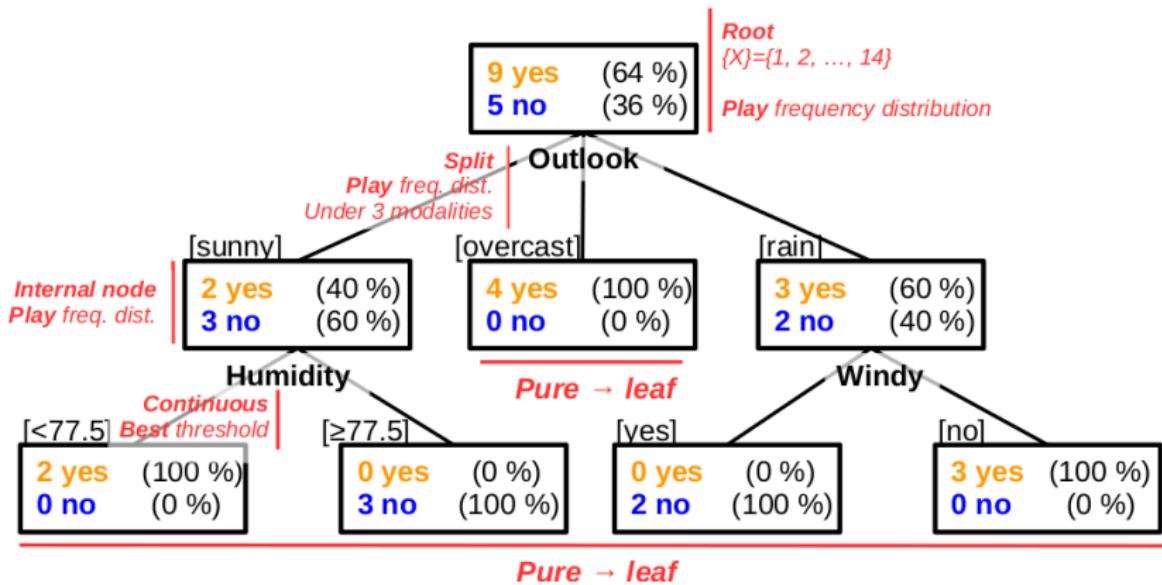
# Objective

Explaining the behaviour of individuals regarding a game ( $y \in \{\text{play, no play}\}$ )

x	Attributes $\{A_i\}$				y
id	Outlook	Temperature (°F)	Humidity (%)	Windy	Play
1	Sunny	75	70	yes	yes
2	Sunny	80	90	yes	no
3	Sunny	85	85	no	no
4	Sunny	72	95	no	no
5	Sunny	69	70	no	yes
6	Overcast	72	90	yes	yes
7	Overcast	83	78	no	yes
8	Overcast	64	65	yes	yes
9	Overcast	81	75	no	yes
10	Rain	71	80	yes	no
11	Rain	65	70	yes	no
12	Rain	75	80	no	yes
13	Rain	68	80	no	yes
14	Rain	70	96	no	yes

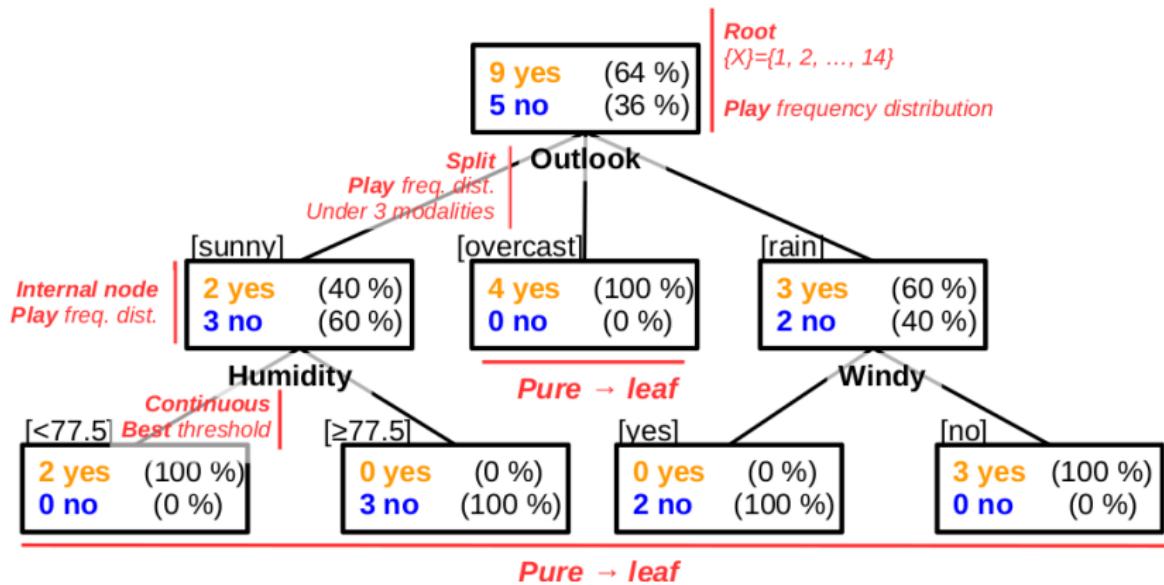
# Objective

Explaining the behaviour of individuals regarding a game ( $y \in \{\text{play, no play}\}$ )



# Objective

Explaining the behaviour of individuals regarding a game ( $y \in \{\text{play, no play}\}$ )



Usage: Inject a new record, and assign the corresponding leaf label

## Objective

Explaining the behaviour of individuals regarding a game ( $y \in \{\text{play, no play}\}$ )

## Issues

- How can we chose the **best split?**
- How do we get the **threshold** for continuous attribute?
- Do we need a **completely pure** partitionning?
- When should the tree-growing process **stop**?
- What is the label when **leaf is not pure**?

## Objective

Explaining the behaviour of individuals regarding a game ( $y \in \{\text{play, no play}\}$ )

## Issues

- How can we chose the **best split?**
- How do we get the **threshold** for continuous attribute?
- Do we need a **completely pure** partitionning?
- When should the tree-growing process **stop?**
- What is the label when **leaf is not pure?**

## Widely used methods

- **CHAID** (Chi-squared Automatic Interaction Detection)
- **CART** (Classification And Regression Trees)

## How can we chose the **best split**?

The **best split** of the data is provided by the attribute  $A$ ; that is the most strongly associated with the target variable  $y$

## CHAID $\Rightarrow \chi^2$ (Chi-squared) test

$(\in [0; +\infty])$

The  $\chi^2$  test measures the rejection of the independence hypothesis  $H_0$

Play	Outlook			Total
	Overcast	Rain	Sunny	
no	0	2	3	5
yes	4	3	2	9
Total	4	5	5	14

Example from the weather data

Y	$\{A_i\}$				Total
	$A_1$	...	$A_l$	...	
$y_1$	...	...	...	...	...
$y_k$	...	...	$n_{kl}$	...	$n_l$
$y_K$	...	...	...	...	...
Total	...	...	$n_k$	...	$n$

Formalism with multiple categories

The **best split** of the data is provided by the attribute  $A$ ; that is the most strongly associated with the target variable  $y$

## CHAID $\Rightarrow \chi^2$ (Chi-squared) test

$(\in [0; +\infty])$

The  $\chi^2$  test measures the rejection of the independence hypothesis  $H_0$

$$\chi^2 = \sum_{k=1}^K \sum_{l=1}^L \frac{(n_{kl} - n_k \cdot \frac{n_l}{n})^2}{n_k \cdot \frac{n_l}{n}}$$

$$\chi^2 = \sum_{k=1}^K \sum_{l=1}^L \frac{(observed - expected)^2}{expected}$$

Y	$\{A_i\}$					Total
	$A_1$	...	$A_l$	...		
$y_1$	...	...	...	...	...	...
$y_k$	...	...	$n_{kl}$	...	...	$n_l$
$y_K$	...	...	...	...	...	...
Total	...	...	$n_k$	...	...	$n$

Formalism with multiple categories

The **best split** of the data is provided by the attribute  $A$ ; that is the most strongly associated with the target variable  $y$

## CHAID $\Rightarrow \chi^2$ (Chi-squared) test

$(\in [0; +\infty])$

The  $\chi^2$  test measures the rejection of the independence hypothesis  $H_0$

$$\chi^2 = \sum_{k=1}^K \sum_{l=1}^L \frac{(n_{kl} - n_{k \cdot} \cdot n_{\cdot l})^2}{n_{k \cdot} \cdot n_{\cdot l}} \quad (\text{test statistic})$$

$$\chi^2_{outlook} \approx 3.5467$$

Play	Outlook			Total
	Overcast	Rain	Sunny	
no	0	2	3	5
yes	4	3	2	9
Total	4	5	5	14

Example from the weather data

The **best split** of the data is provided by the attribute  $A$ ; that is the most strongly associated with the target variable  $y$

## CHAID $\Rightarrow \chi^2$ (Chi-squared) test

$(\in [0; +\infty])$

The  $\chi^2$  test measures the rejection of the independence hypothesis  $H_0$

$$\chi^2 = \sum_{k=1}^K \sum_{l=1}^L \frac{(n_{kl} - n_{k \cdot} \cdot n_{\cdot l} / n)^2}{n_{k \cdot} \cdot n_{\cdot l} / n}$$

$$\chi^2_{outlook} \approx 3.5467$$

$$\chi^2_{wind} \approx 0.9333$$

Play	Wind		Total
	no	yes	
no	2	3	5
yes	6	3	9
Total	8	6	14

Example from the weather data

The **best split** of the data is provided by the attribute  $A$ ; that is the most strongly associated with the target variable  $y$

## CHAID $\Rightarrow \chi^2$ (Chi-squared) test

$(\in [0; +\infty])$

The  $\chi^2$  test measures the rejection of the independence hypothesis  $H_0$

$$\chi^2 = \sum_{k=1}^K \sum_{l=1}^L \frac{(n_{kl} - n_{k \cdot} \cdot n_{\cdot l} / n)^2}{n_{k \cdot} \cdot n_{\cdot l} / n}$$

$$\chi^2_{outlook} \approx 3.5467$$

$$\chi^2_{wind} \approx 0.9333$$

Play	Wind		Total
	no	yes	
no	2	3	5
yes	6	3	9
Total	8	6	14

Example from the weather data

## CHAID $\Rightarrow \chi^2$ (Chi-squared) statistics $\Rightarrow$ Tschuprow's T

$(\in [0; 1])$

Measure of association between two variables. T normalises the  $\chi^2$  statistics to reduce the bias from the number of categories.

The **best split** of the data is provided by the attribute  $A$ ; that is the most strongly associated with the target variable  $y$

## CHAID $\Rightarrow \chi^2$ (Chi-squared) test

( $\in [0; +\infty]$ )

The  $\chi^2$  test measures the rejection of the independence hypothesis  $H_0$

$$\chi^2 = \sum_{k=1}^K \sum_{l=1}^L \frac{(n_{kl} - n_{k \cdot} \cdot n_{\cdot l} / n)^2}{n_{k \cdot} \cdot n_{\cdot l} / n}$$

$$\chi^2_{outlook} \approx 3.5467$$

$$\chi^2_{wind} \approx 0.9333$$

Play	Wind		Total
	no	yes	
no	2	3	5
yes	6	3	9
Total	8	6	14

Example from the weather data

## CHAID $\Rightarrow \chi^2$ (Chi-squared) statistics $\Rightarrow$ Tschuprow's T

( $\in [0; 1]$ )

Measure of association between two variables. T normalises the  $\chi^2$  statistics to reduce the bias from the number of categories.

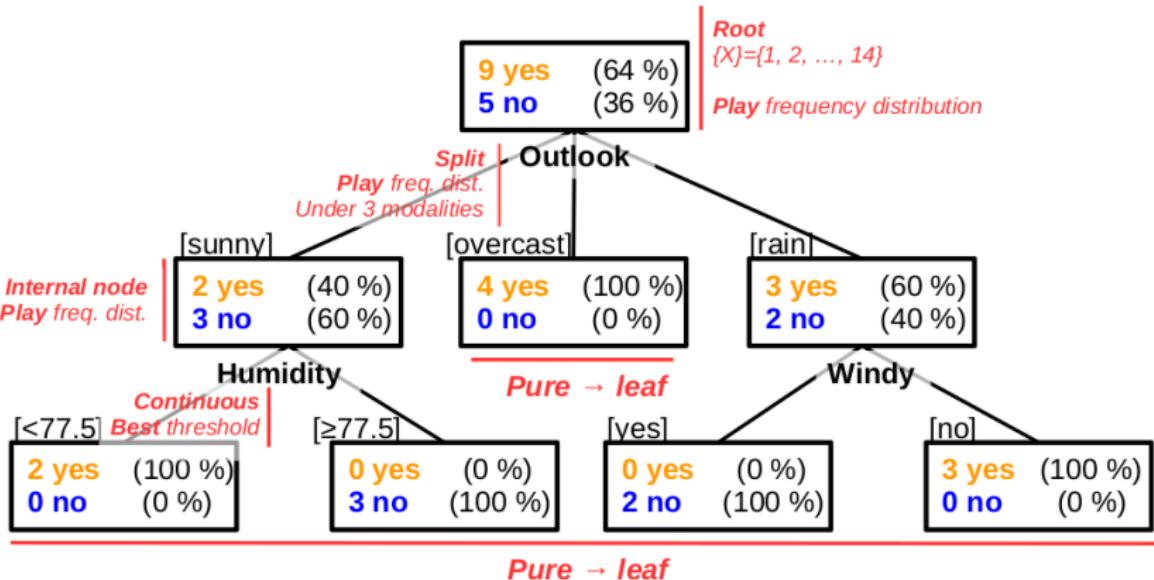
$$T = \sqrt{\frac{\chi^2/n}{\sqrt{(K-1)(L-1)}}}$$

$$T_{outlook} \approx 0.3559$$

$$T_{wind} \approx 0.2582$$

How do we get the **threshold** for continuous attribute?

x	Attributes $\{A_i\}$				y
id	Outlook	Temperature ( $^{\circ}\text{F}$ )	Humidity (%)	Windy	Play
1	Sunny	75	70	yes	yes
2	Sunny	80	90	yes	no
3	Sunny	85	85	no	no
4	Sunny	72	95	no	no
5	Sunny	69	70	no	yes
6	Overcast	72	90	yes	yes
7	Overcast	83	78	no	yes
8	Overcast	64	65	yes	yes
9	Overcast	81	75	no	yes
10	Rain	71	80	yes	no
11	Rain	65	70	yes	no
12	Rain	75	80	no	yes
13	Rain	68	80	no	yes
14	Rain	70	96	no	yes



## CHAID $\Rightarrow \chi^2$ (Chi-squared) statistics $\Rightarrow$ Tschuprow's T

( $\in [0; 1]$ )

Measure of association between two variables. T normalises the  $\chi^2$  statistics to reduce the bias from the number of categories.

Getting the best binary split for a continuous attribute  $A_i$ :

1. Order the values of  $A_i$
2. Create **binary** attributes by splitting at intermediate values
3. Compare all binary attribute candidates with the **Tschuprow's T**

**CHAID**  $\Rightarrow \chi^2$  (Chi-squared) statistics  $\Rightarrow$  Tschuprow's T $(\in [0; 1])$ 

Mesure of association between two variables. T normalises the  $\chi^2$  statistics to reduce the bias from the number of categories.

Getting the best binary split for a continuous attribute  $A_i$ :

1. Order the values of  $A_i$
2. Create **binary** attributes by splitting at intermediate values
3. Compare all binary attribute candidates with the **Tschuprow's T**

Let's  $A_i$  be **Humidity**  $\supset \{70, 85, 90, 95\}$ :

Play	Humidity	
	$< 77.5$	$\geq 77.5$
no	2	0
yes	0	3
T	<b>1.00</b>	

Play	Humidity	
	$< 87.5$	$\geq 87.5$
no	2	0
yes	1	2
T	<b>0.67</b>	

Play	Humidity	
	$< 92.5$	$\geq 92.5$
no	2	0
yes	2	1
T	<b>0.41</b>	

**Best threshold for Humidity is 77.5 with  $T = 1$  (a pure partition here)**

**CHAID**  $\Rightarrow \chi^2$  (Chi-squared) statistics  $\Rightarrow$  Tschuprow's T $(\in [0; 1])$ 

Mesure of association between two variables. T normalises the  $\chi^2$  statistics to reduce the bias from the number of categories.

Getting the best binary split for a continuous attribute  $A_i$ :

1. Order the values of  $A_i$
2. Create **binary** attributes by splitting at intermediate values
3. Compare all binary attribute candidates with the **Tschuprow's T**

Let's  $A_i$  be **Temperature**  $\supset \{64, 65, 68, 69, 70, 71, 72, 75, 80, 81, 83, 85\}$ :

.. / ..

Play	Temperature		T
	< 77.5	$\geq 77.5$	
no	1	2	0.67
yes	2	1	

.. / ..

**Best threshold for Temperature is 77.5 with  $T = 0.67$**

## CHAID $\Rightarrow \chi^2$ (Chi-squared) statistics $\Rightarrow$ Tschuprow's T

( $\in [0; 1]$ )

Mesure of association between two variables. T normalises the  $\chi^2$  statistics to reduce the bias from the number of categories.

Getting the best binary split for a continuous attribute  $A_i$ :

1. Order the values of  $A_i$
2. Create **binary** attributes by splitting at intermediate values
3. Compare all binary attribute candidates with the **Tschuprow's T**

At the [outlook  $\rightarrow$  sunny] intermediary node:

- Best threshold for **Humidity** is 77.5 with  $T = 1$  (*pure partition*)
  - Best threshold for **Temperature** is 77.5 with  $T = 0.67$  here)
- $\Rightarrow$  Choose Humidity !

TP: In fact, the first best split attribute is 'Outlook' :)

## CHAID $\Rightarrow \chi^2$ (Chi-squared) statistics $\Rightarrow$ Tschuprow's T

( $\in [0; 1]$ )

Mesure of association between two variables. T normalises the  $\chi^2$  statistics to reduce the bias from the number of categories.

Getting the best binary split for a continuous attribute  $A_i$ :

1. Order the values of  $A_i$
2. Create **binary** attributes by splitting at intermediate values
3. Compare all binary attribute candidates with the **Tschuprow's T**

For large databases with many continuous attributes:

- Pre-ordering of values for continuous attributes <sup>1</sup>
- Computing the threshold on a subsample <sup>2</sup>
- Avoiding computations for thresholds that won't improve the T <sup>3</sup>

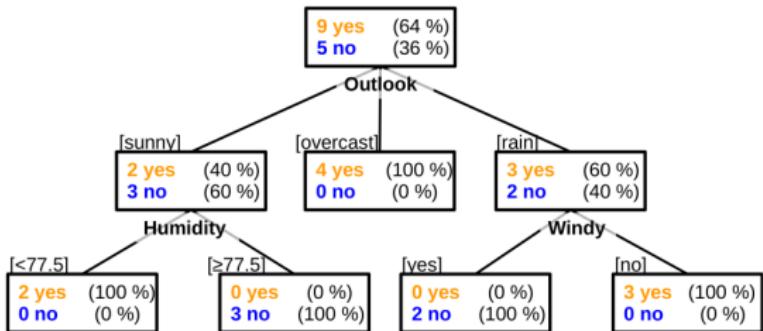
<sup>1</sup> Witten & Franck, 2000

<sup>2</sup> Chauchat & Rakotomalala, 2000

<sup>3</sup> Fayyad & Irani, 1993; Muhlenbach & Rakotomalala, 2005

## Prediction rules

A decision tree represent **a set of rules** along paths that link the leaves to the root node.



1. if [(Outlook = Sunny) & (Humidity < 77.5)] then (Play = yes)
2. if [(Outlook = Sunny) & (Humidity ≥ 77.5)] then (Play = no)
3. if [(Outlook = Overcast)] then (Play = yes)
4. if [(Outlook = Rain) & (Windy = yes)] then (Play = no)
5. if [(Outlook = Rain) & (Windy = no)] then (Play = yes)

## Leaf label

When leaves are not pure, a decision rule (eg., majority) should be applied.

*NB: Leaf frequencies estimate the conditional probability of the label for the new record.*

## CART Overview

It produces a **binary decision tree** by **successively** splitting the data into two subsets.  
**Best split**  $\Leftarrow$  attribute that separates the data into relatively **homogeneous** partitions.

### Recursive Partitioning to find the best way to split $A$ into 2 sub-partitions

1. For each predictor variable  $X_j$ :
  - a. For each value  $s_j$  of  $X_j$ :
    - i. Split the instances in  $A$  with  $X_j$  values  $< s_j$  as one partition, and the remaining records where  $X_j \geq s_j$  as another partition
    - ii. Measure the homogeneity of classes within each subpartition of  $A$
  - b. Select the value of  $s_j$  that produces maximum within-partition homogeneity of class
2. Select the variable  $X_j$  and the split value  $s_j$  that produces maximum within-partition homogeneity of class

### CART cost function at each node

$$J(s_j, X_j) = \frac{m_{left}}{m} G_{left} + \frac{m_{right}}{m} G_{right}$$

where  $= \begin{cases} G_{left/right} & \text{measures the impurity of the left/right subset} \\ m_{left/right} & \text{is the number of instances in the left/right subset} \end{cases}$

## CART Overview

It produces a **binary decision tree** by **successively** splitting the data into two subsets.

**Best split**  $\Leftarrow$  attribute that separates the data into relatively **homogeneous** partitions.

### Recursive Partitioning to find the best way to split $A$ into 2 sub-partitions

1. For each predictor variable  $X_j$ :
  - a. For each value  $s_j$  of  $X_j$ :
    - i. Split the instances in  $A$  with  $X_j$  values  $< s_j$  as one partition, and the remaining records where  $X_j \geq s_j$  as another partition
    - ii. Measure the homogeneity of classes within each subpartition of  $A$
  - b. Select the value of  $s_j$  that produces maximum within-partition homogeneity of class
2. Select the variable  $X_j$  and the split value  $s_j$  that produces maximum within-partition homogeneity of class

### Overall Recursive Algorithm (greedy !)

1.  $A$  is the entire dataset
2. Apply the **recursive partitioning** to split  $A$  into 2 sub-partitions,  $A_1$  and  $A_2$
3. Repeat step 2.on sub-partitions  $A_1$  and  $A_2$
4. **Stop?** when no further partition can be made that sufficiently improves the homogeneity of the partitions

## CART Overview

It produces a **binary decision tree** by **successively** splitting the data into two subsets.

**Best split**  $\Leftarrow$  attribute that separates the data into relatively **homogeneous** partitions.

### Recursive Partitioning to find the best way to split $A$ into 2 sub-partitions

1. For each predictor variable  $X_j$ :
  - a. For each value  $s_j$  of  $X_j$ :
    - i. Split the instances in  $A$  with  $X_j$  values  $< s_j$  as one partition, and the remaining records where  $X_j \geq s_j$  as another partition
    - ii. Measure the homogeneity of classes within each subpartition of  $A$
  - b. Select the value of  $s_j$  that produces maximum within-partition homogeneity of class
2. Select the variable  $X_j$  and the split value  $s_j$  that produces maximum within-partition homogeneity of class

### Overall Recursive Algorithm (greedy !)

1.  $A$  is the entire dataset
2. Apply the **recursive partitioning** to split  $A$  into 2 sub-partitions,  $A_1$  and  $A_2$
3. Repeat step 2.on sub-partitions  $A_1$  and  $A_2$
4. **Stop?** `max_depth`, `min_samples_split`, `min_samples_left`,  
`min_weights_fraction_leaf` or `max_leaf_leaf_nodes`

## CART Overview

The method produces a **binary decision tree** by **successively** splitting the data into two subsets. CART does **not** apply a stopping rule from statistical test. The approach uses a **postpruning** procedure to **select** the best **subtree** from the **whole** decision tree. The postpruning step targets the less informative branches.

## CART Overview

The method produces a **binary decision tree** by **successively** splitting the data into two subsets. CART does **not** apply a stopping rule from statistical test. The approach uses a **postpruning** procedure to **select** the best **subtree** from the **whole** decision tree. The postpruning step targets the less informative branches.

### CART Splitting

The **best split** is provided by the attribute that gives the **purest children** nodes. The rule is strict.

### CART Postpruning

The method uses **holdout** or **cross-validation** methods to pick the best subtree from the whole binary tree. The best subtree **maximizes** the global **purity** criteria.

## CART Overview

The method produces a **binary decision tree** by **successively** splitting the data into two subsets. CART does **not** apply a stopping rule from statistical test. The approach uses a **postpruning** procedure to **select** the best **subtree** from the **whole** decision tree. The postpruning step targets the less informative branches.

## CART algorithm

1. Sort all attribute values (qualitative and quantitative)<sup>a</sup>
2. For **each** attribute, retain the best split (min impurity)
3. From **all** attributes, select the best attribute (min impurity)
4. Do a **binary** split for this attribute
5. Iterate from 2. to 4.
6. Postprune the whole binary tree

---

<sup>a</sup>binary: 1 division; nominal  $k$ :  $2^{k-1} - 1$  divisions; ordinal  $k$ :  $k - 1$  divisions)

## CART splitting criteria

CART uses the **Gini** index  $G$  to evaluate the purity of a node  $i$ .

$$G_i = \sum_c^K P(c|i)(1 - P(c|i)) = 1 - \sum_c^K P(c|i) \times P(c|i)$$

$P(c|i)$  proportion of observations in class  $c$  at node  $i$

### Example

- node  $i$  has  $|i| = 313$  observations
  - class *yes* has 135 observations and class *no* has 178 observations
- $\Rightarrow G_i = \frac{135}{313} * (1 - \frac{135}{313}) + \frac{178}{313} * (1 - \frac{178}{313}) = 1 - ((\frac{135}{313})^2 + (\frac{178}{313})^2) \approx 0.4905$

### Purity

- Pure node: observations only in one class (eg. *yes* or *no*)  $\Rightarrow G_i = 0$
- Mixed node:  $G_i$  grows with impurity

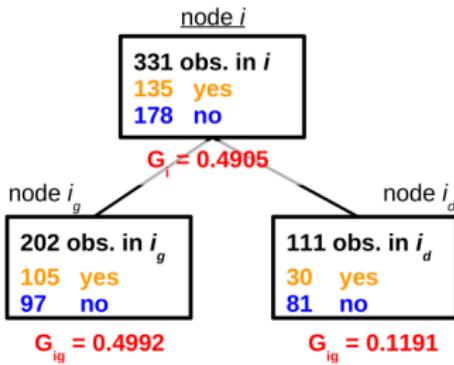
NB:  $G_i$  is the probability of missclassification of an observation at node  $i$

## Impurity reduction

Each split reduces the impurity:

$$\Delta G_i = G_i - |i_g|G_{i_g} - |i_d|G_{i_d}$$

with  $i_g$  and  $i_d$  the proportion of samples in the child as compared to the parent



Each split reduces the impurity:

$$\Delta G_i = \left(1 - \left(\left(\frac{135}{313}\right)^2 + \left(\frac{178}{313}\right)^2\right)\right) - \frac{202}{313} \left(1 - \left(\left(\frac{105}{202}\right)^2 + \left(\frac{97}{202}\right)^2\right)\right) - \frac{111}{313} \left(1 - \left(\left(\frac{30}{111}\right)^2 + \left(\frac{81}{111}\right)^2\right)\right) \approx 0.0285$$

## Impurity reduction for **each** attribute

$$\Delta G_i^* = \max\{\Delta G_i\}$$

## Impurity reduction for **all** attribute

$$\Delta^* = \max\{\Delta G_i^*\}$$

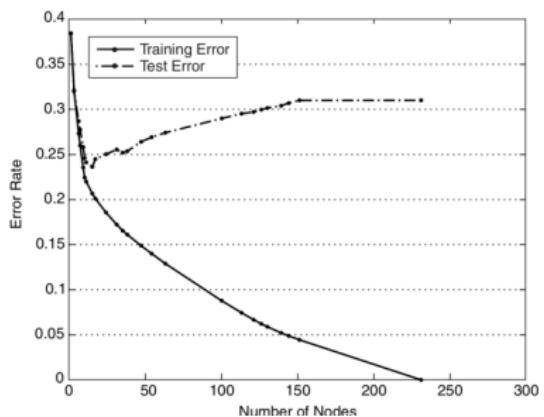
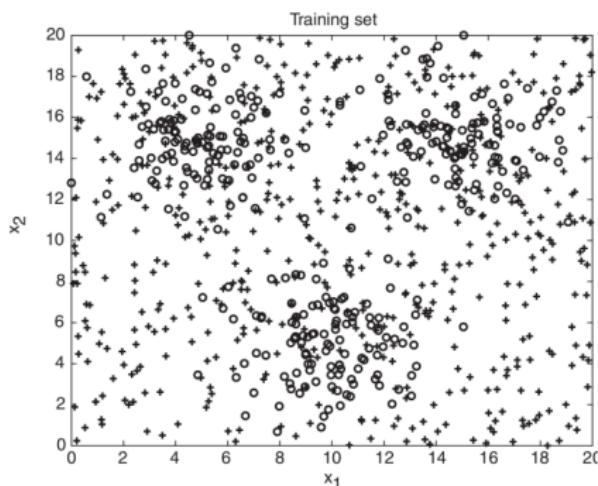
Do we need a **completely pure** partitionning?  
When should the tree-growing process **stop**?

Good classification model

**no underfitting & no overfitting** of the data:



low training & generalization error



Decision tree classifier:

- 1,200, mixture of 3 Gaussian dist.
- + 1,800, uniform distr.

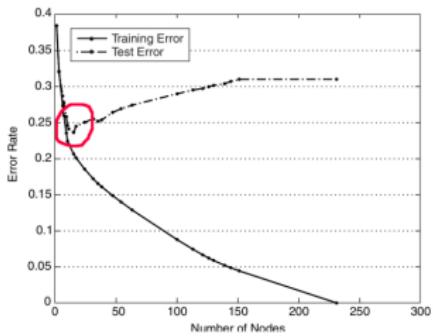
- Reduce *training error* by increasing complexity (ie., nbr. nodes)...
- But then, *test error* increases (ie., poor generalization)...

Do we need a **completely pure** partitionning?  
When should the tree-growing process **stop**?

## Options

- Prepruning (Early stop rule)
- Postpruning

## Objective



## Prepruning principle

Apply a **stop rule** during the tree-growing procedure based on the **amount of information** provided by the **split** to avoid a tree that perfectly fits the training sample.

## Prepruning principle

Apply a **stop rule** during the tree-growing procedure based on the **amount of information** provided by the **split** to avoid a tree that perfectly fits the training sample.

Exp. Split accepted if *significant* (ie.,  $\chi^2$  test of independence rejection )

$A_i = \text{Humidity}$ , with  $T = 1.00$ . The  $\chi^2$  estimated *p-value* is 0.025.

- If risk at 5% ( $2.5 < 5$ ),  $H_0$  can be rejected  $\Rightarrow$  split.
- If risk at 1% ( $!2.5 < 1$ ),  $H_0$  cannot be rejected  $\Rightarrow$  no split.

## Prepruning principle

Apply a **stop rule** during the tree-growing procedure based on the **amount of information** provided by the **split** to avoid a tree that perfectly fits the training sample.

Exp. Split accepted if *significant* (ie.,  $\chi^2$  test of independence rejection )

$A_i = \text{Humidity}$ , with  $T = 1.00$ . The  $\chi^2$  estimated *p-value* is 0.025.

- If risk at 5% ( $2.5 < 5$ ),  $H_0$  can be rejected  $\Rightarrow$  split.
- If risk at 1% ( $!2.5 < 1$ ),  $H_0$  cannot be rejected  $\Rightarrow$  no split.

## Prepruning issues

?? What is the best criteria?

?? How about subsequent splits?

?? How should the multiple tests be corrected?

## Pospruning principle

1. Grow a tree with leaves as pure as possible
2. Choose a tree among trees of different sizes following a criteria  
(eg. unbiased test error estimation with validation set, see CART<sup>a</sup>)

---

<sup>a</sup>Breitman *et al.* 1994

- use the tree to assign a label on new record
- variant: merge intermediate nodes - CART

## CHAID & CART both try to...

... avoid overfitting the data, ie. limit the size of the decision tree

- CHAID Prepruning: Statistical stopping rule
- CART Postpruning: Full tree growth and reduction
  - i. Set holdout dataset (divide original dataset into training and test)
  - ii. Prune until similar performance on both datasets (ie., no overfitting)
- CHAID multiple splits: multiple splits allowed (useful for analysis)
- CART binary splits: only binary splits (useful for prediction)

	<b>CHAID</b>	<b>CART</b>
<i>score</i>	Tschuprow T	Gini index
<b>group</b>	Multiple	Binary
<b>Optimal size</b>	<ul style="list-style-type: none"> <li>- Minimum number of observations for split</li> <li>- Tree depth</li> <li>- Specialization threshold</li> </ul>	
	Prepruning with $\chi^2$ test	Postpruning
<b>Pros</b>	Good for data exploratory phase Suitable for large database	Performance
<b>Cons</b>	Performance	Small dataset Binary split not always suitable

## When do we have a leaf?

A node is a **leaf** if it is **pure** or has **too few observations** for another split.

⇒ The label is obtained from the majority rule.

## How to compute the error rate $E$ for a leaf $i$ ?

If  $s$  is the majority label for the leaf  $i$ , the error rate for the leaf is:

$$E(s|i) = \sum_{r=1}^k p(r|i)$$

where  $p(r|i)$  is the proportion of observations at node  $i$  that are set to class  $s$  but belong to class  $r$  (ie. proportion of misclassified observations).

## How to compute the error rate $E(T)$ for a tree $T$ ?

$$E(T) = \sum_{t \in T} \frac{n_i}{n} E(s|i)$$

where  $n_i$  is the number of observations at node  $i$

## Postpruning using error rate

From a training set (eg. 80%), the method CART

- (i) provides a whole tree  $T_{max}$  from the training set
- (ii) prunes  $T_{max}$  to get subtrees

From a test set (eg. 20%), choose  $T^*$  that minimizes the error  $E_\alpha$  (or the cost  $C_\alpha$ ):

$$E_\alpha(T^*) = E(T) + \alpha|T|$$

where  $\alpha$  penalizes tree with a large number of leaves

In practice:

- i Make  $\alpha$  vary
- ii Pick  $T$  that gives the smallest cross-validation prediction error

## Bibliography

- Rakotomalala, Ricco. "Arbres de décision." Revue Modulad 33 (2005) : 163 – 187.
- Tan, Pang-Ning, Michael Steinbach, and Vipin Kumar. *Classification: basic concepts, decision trees, and model evaluation*. Introduction to data mining 1 (2006) : 145 – 205.
- Gonzales, Pierre-Louis. "Segmentation", 2010
- Zighed, D. A., and R. Rakotomalala. "Graphes d'induction", Hermès. Annexe A (2000).