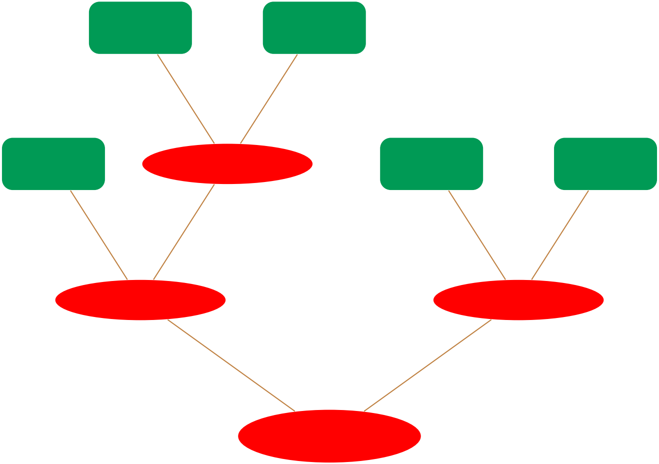
Decision trees

Nicolas Sutton-Charani

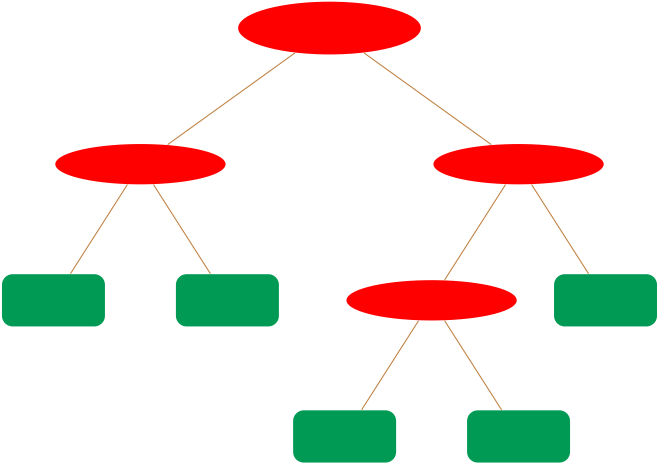
IMT Mines Alès
École Mines-Télécom

Plan

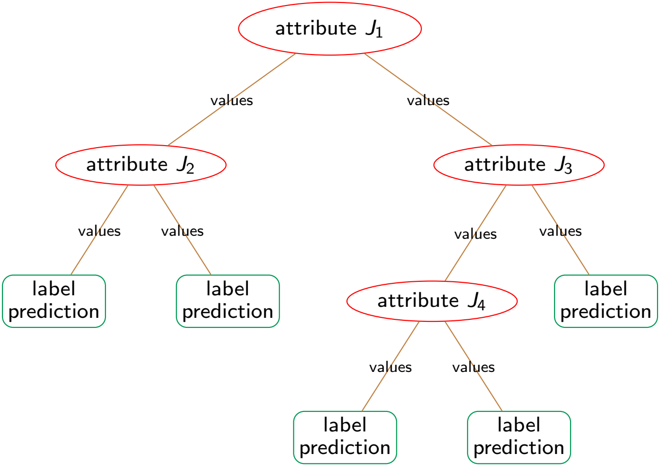
What is a decision tree ?



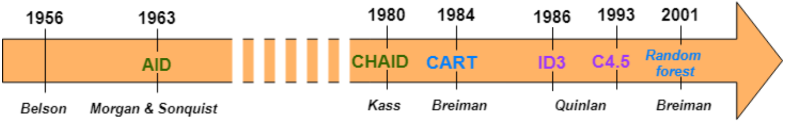
What is a decision tree ?



What is a decision tree ? → **supervised learning**



A little history



⚠ machine learning (or data mining) **decision trees**
$\neq$ decision theory **decision trees**

Types of decision trees

type of class label

- numerical $\rightarrow$ **regression** tree
- nominal $\rightarrow$ **classification** tree

type of algorithm ($\rightarrow$ structure)

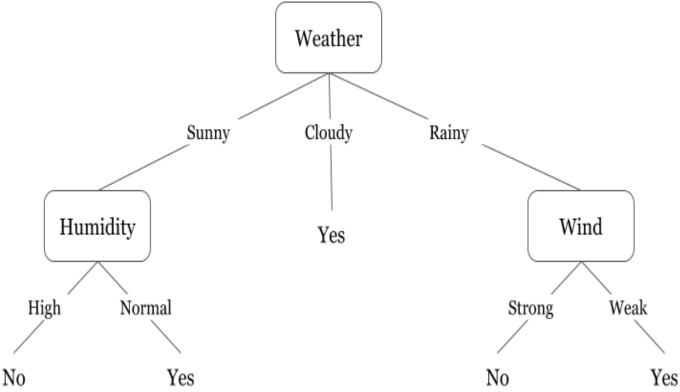
- **CART** : statistics, binary tree
- **C4.5** : computer science, small tree

Plan

Plan

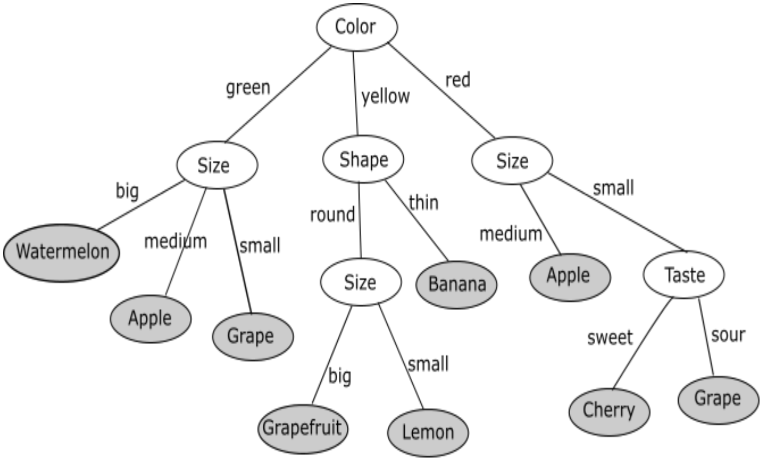
Classification trees

Will the badminton match take place ?



Classification trees

What fruit is it ?

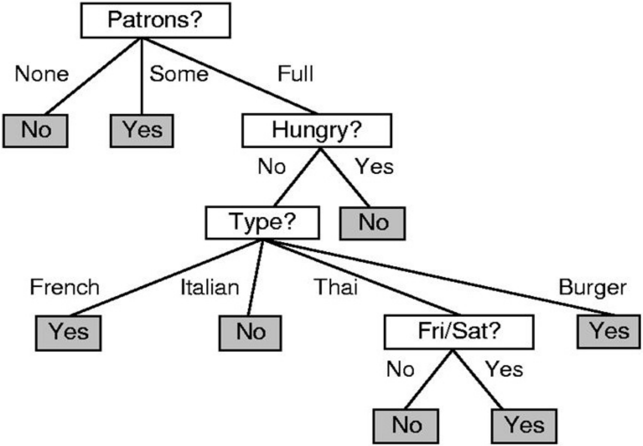


Classification trees

What he/she come to my party ?

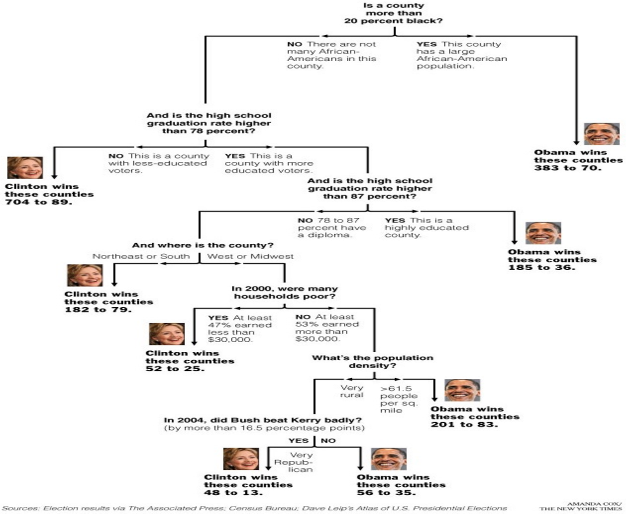
Classification trees

Will they wait ?



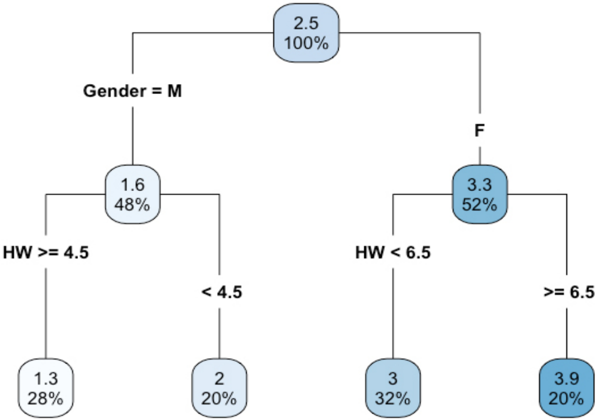
Classification trees

Who will win the US presidential election ?



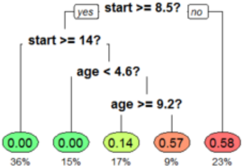
Regression trees

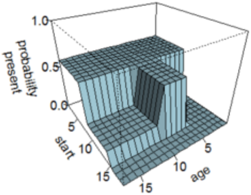
What grade will a student get (given his homework average grade) ?

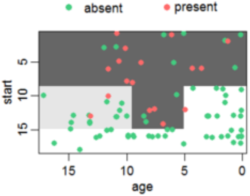


Plan

Data analysis tool







Trees are very interpretable : attributes **spaces partitioning**

→ a tree can be resumed by its leaves which define a **law mixture**

→ wonderful collaboration tool with experts

⚠ **INSTABILITY** ← *overfitting*

Plan

Formalism

Learning dataset (supervised learning)

$$\begin{pmatrix} x_1, y_1 \\ \vdots \\ x_N, y_N \end{pmatrix} = \begin{pmatrix} x_1^1 & \dots & x_1^J & y_1 \\ \vdots & & \vdots & \vdots \\ x_N^1 & \dots & x_N^J & y_N \end{pmatrix}$$

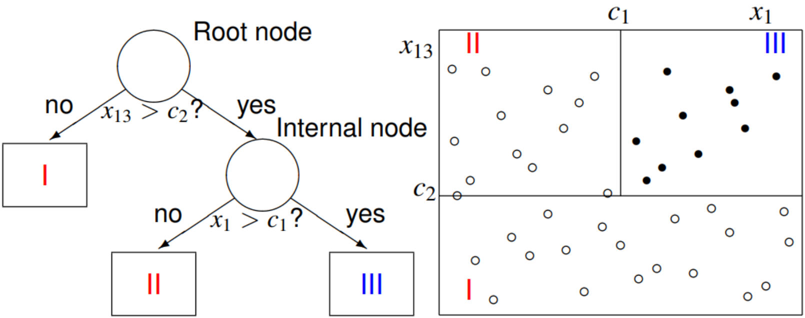
samples are assumed to be i.i.d

- Attributes $X = (X^1, \dots, X^J) \in \mathcal{X} = \mathcal{X}^1 \times \dots \times \mathcal{X}^J$
- Spaces $\mathcal{X}^j$ can be categorical or numerical
- Class label $Y \in \Omega = \{\omega_1, \dots, \omega_K\}$ ($\in \mathbb{R}^K$ for regression)

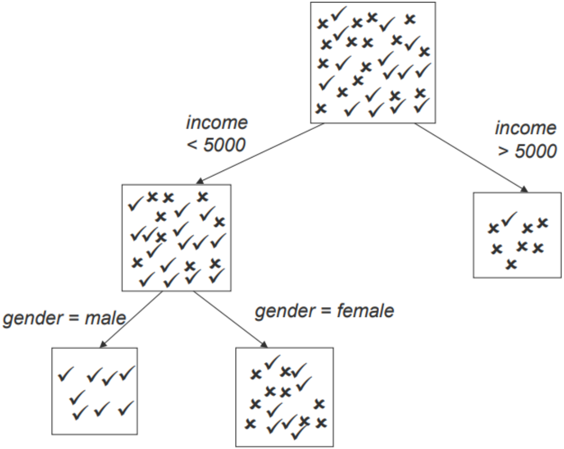
Tree

$\mathcal{P}_H = \{t_1, \dots, t_H\}$ and $\pi_h = P(t_h) \approx \frac{|t_h|}{N}$ with $|t_h| = \#\{i : x_i \in t_h\}$

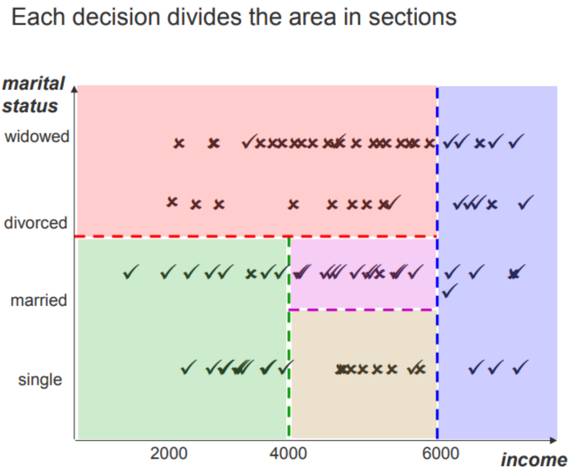
Recursive partitioning



Recursive partitioning



Recursive partitioning



IF income > 6000
THEN accept

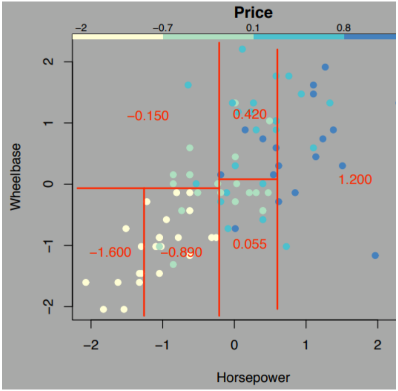
IF income < 6000 and marital status = widowed or marital status = divorced
THEN reject

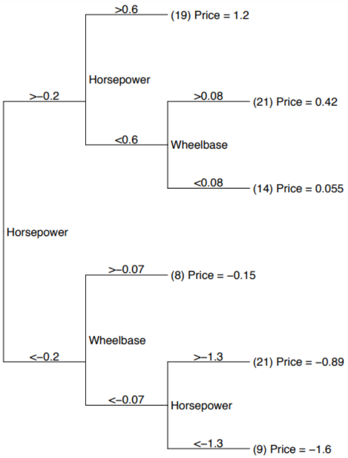
IF income < 4000 and marital status = single or marital status = married
THEN accept

IF income > 4000 and income < 6000 and marital status = married
THEN accept

IF income > 4000 and income < 6000 and marital status = single
THEN reject

Recursive partitioning





Learning principle

- **Start** with all the dataset in the initial node
- **Chose** the best splits (on attributes) in order to get **pure** leaves

Classification trees

purity = *homogeneity* in term of class labels

- **CART** → Gini impurity : $i(t_h) = \sum_{k=1}^{K} p_k(1 - p_k)$
- **ID3, C4.5** → Shanon entropy : $i(t_h) = -\sum_{k=1}^{K} p_k \log_2(p_k)$

whith

$p_k = P(Y = \omega_k | t_h)$

Regression trees

purity = low *variance* of class labels

$$\rightarrow i(t_h) = \widehat{Var}(Y|t_h) = \frac{1}{|t_h|} \sum_{x_i \in t_h} (y_i - \widehat{E}(Y|t_h))^2 \text{ with } \widehat{E}(Y|t_h) = \frac{1}{|t_h|} \sum_{x_i \in t_h} y_i$$

Impurity measures

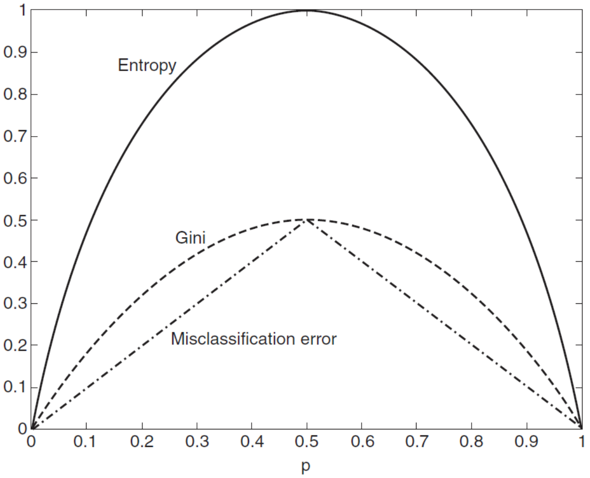
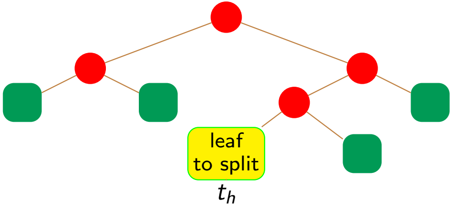


Figure Comparison among the impurity measures for binary classification problems.

Plan

Purity criteria



Impurity measure + tree structure → criteria

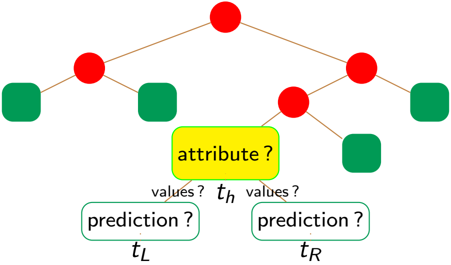
CART, ID3 : purity gain

C4.5 : information gain ratio

Regression trees

CART : Variance minimisation

Purity criteria



Impurity measure + tree structure → criteria

CART, ID3 : purity gain $\rightarrow \Delta i = i(t_h) - \pi_L i(t_L) - \pi_R i(t_R)$

C4.5 : information gain ratio $\rightarrow IGR = \frac{\Delta i}{H(\pi_L, \pi_R)}$

Regression trees

CART : Variance minimisation $\rightarrow \Delta i = i(t_h) - \pi_L i(t_L) - \pi_R i(t_R)$

Plan

Stopping criteria (pre-pruning)

For all leaves $\{t_h\}_{h=1,\dots,H}$ and their potential children :

- leaves **purity** : $\exists k \in \{1, \dots, K\} : p_k = 1$
- leaves and children **sizes** : $|t_h| \leq minLeafSize$
- leaves and children **weights** : $\pi_h = \frac{|t_h|}{t_0} \leq minLeafProba$
- leaves **number** : $H \geq maxNumberLeaves$
- tree **depth** : $depth(\mathcal{P}_H) \geq maxDepth$
- purity **gain** : $\Delta i \leq minPurityGain$

Plan

Learning algorithm

Result: Learnt tree

Start with all the learning data in an initial node (single leaf);

while *Stopping criteria not verified for all leaves* **do**
  for *each splitable leaf* **do**
    compute the purity gains obtained from all possible split;
  end
  SPLIT : select the split achieving the maximum purity gain;
end
prune the obtained tree;

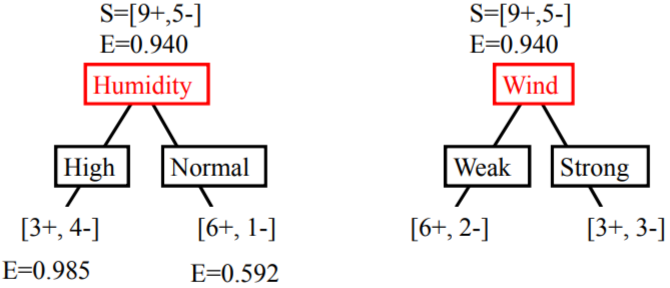
Recursive partitioning

ID3 - Training Examples – [9+,5-]

Day	Outlook	Temp.	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Weak	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cold	Normal	Weak	Yes
D10	Rain	Mild	Normal	Strong	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

ID3 - Selecting Next Attribute

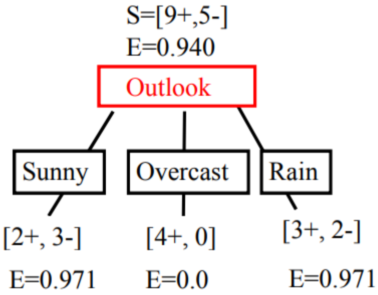
$$\text{Entropy}([9+,5-]) = -(9/14)\log_2(9/14) - (5/14)\log_2(5/14) = 0.940$$



Gain(S,Humidity) =
0.940-(7/14)*0.985-(7/14)*0.592
= **0.151**

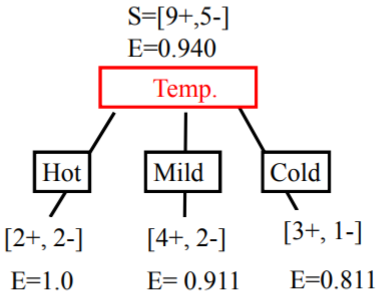
Gain(S,Wind) =
0.940-(8/14)*0.811-(6/14)*1.0
= **0.048**

ID3 - Selecting Next Attribute



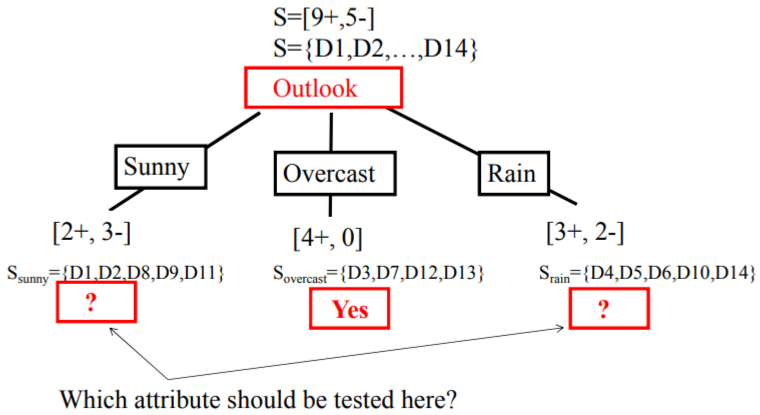
Gain(S,Outlook) =
0.940-(5/14)*0.971 -(4/14)*0.0 -(5/14)*0.0971
= **0.247**

ID3 - Selecting Next Attribute



Gain(S,Outlook) =
0.940-(4/14)*1.0 - (6/14)*0.911 - (4/14)*0.811
= **0.029**

ID3 - Best Attribute - Outlook



ID3 - S_{sunny}

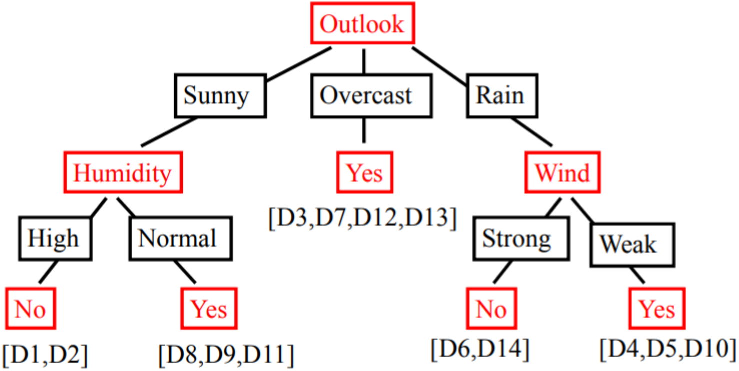
Gain(S_{sunny} , Humidity) = 0.970-(3/5)0.0 – 2/5(0.0) = **0.970**

Gain(S_{sunny} , Temp.) = 0.970-(2/5)0.0 –2/5(1.0)-(1/5)0.0 = 0.570

Gain(S_{sunny} , Wind) = 0.970= -(2/5)1.0 – 3/5(0.918) = 0.019

So, Hummudity will be selected

ID3 - Results


Outlook
Sunny
Overcast
Rain
Humidity
Yes
Wind
[D3,D7,D12,D13]
High
Normal
Strong
Weak
No
Yes
No
Yes
[D1,D2]
[D8,D9,D11]
[D6,D14]
[D4,D5,D10]

Plan

Variables importance weights

- during a tree learning :

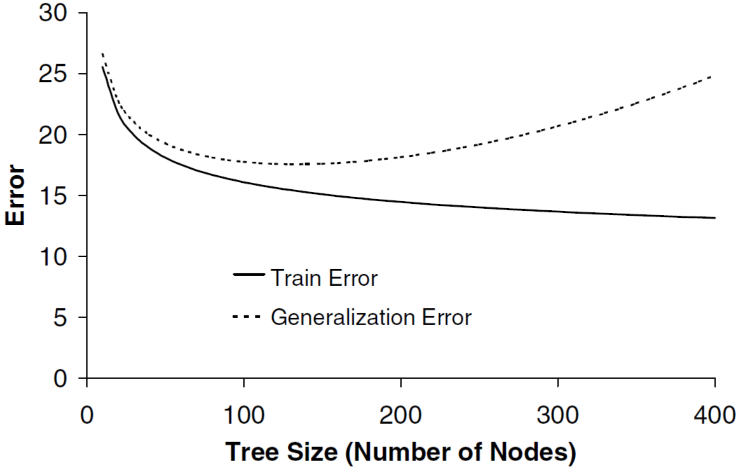
  - all potential split $\rightarrow$ 1 variable $\rightarrow$
    - purity gain
    - accuracy decrease

- after learning :

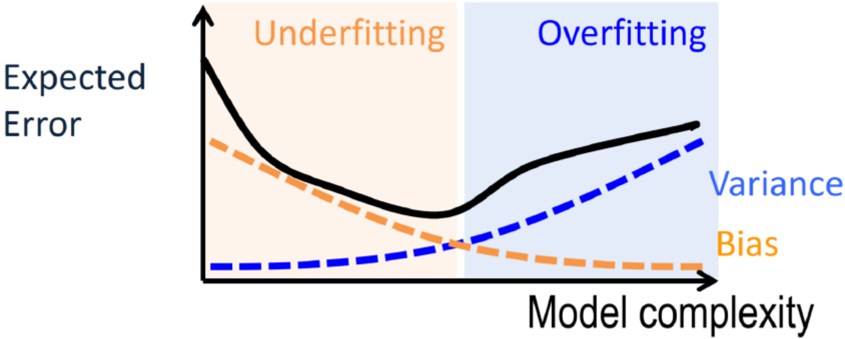
  - these gains, decreases $\rightarrow$ $\sum$
    $\rightarrow$ [0,1]-normalisation
    $\rightarrow$ importance weights

Plan

Overfitting



Overfitting



Remark : decision trees do not need variable selection or dimension reduction (in term of accuracy).

Plan

Cost-Complexity Pruning

The idea

- trade-off between predictive efficiency and complexity
- find a subtree that fulfills this trade-off

Metrics

- 'Err' $\leftarrow$ misclassification rate or MSE
- **Criterion :** $R_\alpha = Err + \alpha H$

Steps

- Find a useful sequence of nested subtrees
- Choose the right subtree

Cost-Complexity Pruning

Sequence of subtrees creation

Result: sequence of trees that are all sub-trees of T_0 :
$T0 \gg T1 \gg T2 \gg T3 \gg \ldots \gg Tk \gg \mathcal{P}_1(initialnode)$
Learn the biggest tree $T_s = T_0 := \mathcal{P}_{H_{max}}$ obtained for $\alpha_0 = 0$ (s=0);
while $T_s \neq \mathcal{P}_1$ **do**

$$T_{s+1} = \underset{t \in subtrees(T_s)}{\text{argmin}} [R_{\alpha_s}(t) - R_{\alpha_s}(T_s)];$$

$$\alpha_{s+1} = R_{\alpha_s}(T_{s+1}) - R_{\alpha_s}(T_s);$$

end

We get 2 bijective sets : $\{T_0, \ldots, T_S\}$ and $\{\alpha_0, \ldots, \alpha_S\}$ (with $T_S = \mathcal{P}_1$)

Selection : $T_{s^*} = \underset{T_s \in \{T_0, \ldots, T_S\}}{\text{argmin}} Err(T_s) \leftarrow$ *pruning set or cross validation*

Cost-Complexity Pruning

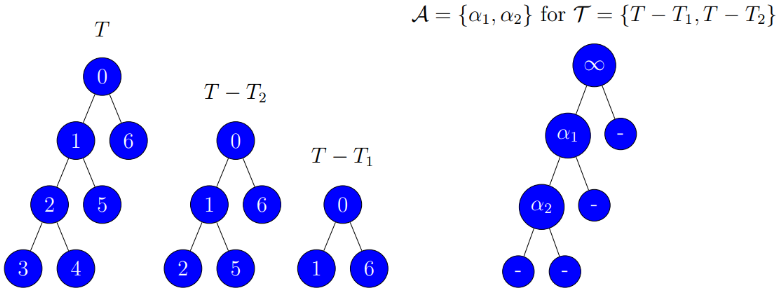


FIGURE – Sequence of nested subtrees

Here, $\alpha_2 < \alpha_1 \implies T - T_1 \subset T - T_2$

Plan

Random forest

Motivation

- trees instability
- bias-variance trade-off

Averaging reduces variance :

$$Var(\overline{X}) = \frac{Var(X)}{N} \quad \text{(for independant predictions)}$$

$\rightarrow$ Average models to reduce model variance

One problem :

- only one training set
- where do multiple models come from ?

Bagging : Bootstrap Aggregation

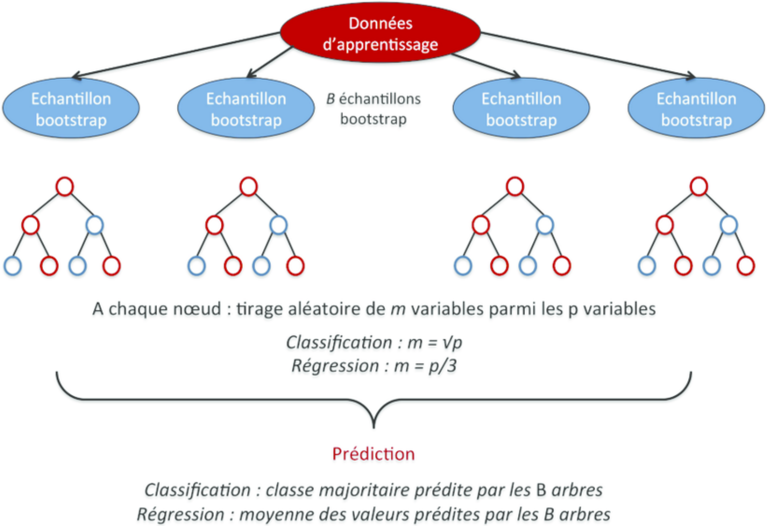
- Tin Kam Ho (1995) $\rightarrow$ Leo Breiman (2001)

- Take repeated bootstrap samples from the training set

- *Bootstrap sampling* : Given a training set D containing N examples, draw N examples at random with replacement from D.

- **Bagging** :
  - create B bootstrap samples $D_1, \ldots, D_B$
  - train distinct classifier on each D_b
  - classify new instance by majority vote / averaging / aggregating predictions

Random forest



References

* L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, Classification And Regression Trees, 1984.

* J. Quinlan, "Induction of decision trees," Machine Learning, vol. 1, pp. 81–106, Oct. 1986

* L. Breiman. Random forests. Statistics, pages 1–33, 2001.

* G. Biau, L. Devroye, and G. Lugosi. Consistency of random forests and other averaging classifiers. J. Mach. Learn. Res., 9 :2015–2033, jun 2008.