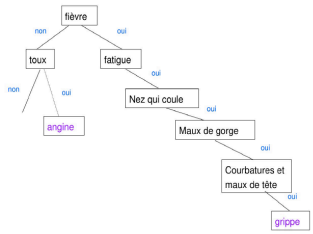
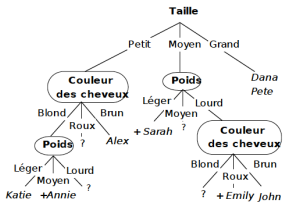
Apprentissage statistique:

Arbre de décision binaire et Random Forest





Plan

- 1. Introduction
- 2. Les arbres de décision binaire
- 3. Application à l'apprentissage supervisé
- 4. Forêt Aléatoire (Random Forest)

Plan

- 1. **Introduction**
- 2. Les arbres de décision binaire
- 3. Application à l'apprentissage supervisé
- 4. Forêt Aléatoire (Random Forest)

Introduction

- **Data Mining**
  - Flot de données en constante augmentation
  - Nécessité de gérer ce flot pour en tirer les informations les plus discriminantes.
  - **Outils :** SVM, Boosting, Réseau de neurones, Arbre de décision, etc.

Introduction

- **Apprentissage statistique**
  - Phénomène physique, biologique, etc. trop complexe ou trop bruité.
  - Pas de description analytique permettant d'en déduire un modèle déterministe.
  - **Solution :** Description du comportement au moyen d'une série d'observation

Introduction

- **Apprentissage statistique**
  - ***Domaines d'application***
    - Reconnaissance de « patterns » (parole, caractères manuscrits, détection/reconnaissance)
    - L'imagerie médicale ou satellitaire,
    - Prévision d'une grandeur climatique ou économique,
    - Comportement d'un client...

Introduction

- **Apprentissage statistique**
  - ***Les données***
    - ***n*** objets, individus ou unités statistiques
    - ***P*** observations : $X = (X^1, \ldots, X^p)$
  - ***Le traitement de ces données***
    - **Etape 1:** Exploration statistique (allures des distributions, corrélation, transformation, réduction de dimensionnement
    - **Etape 2:** Apprentissage (modélisation)

Introduction

- **Apprentissage statistique**
  - ***Les étapes:***
    1. **Extraction** des données avec ou sans échantillonnage
    2. **Exploration** des données pour la détection de valeurs aberrantes ou seulement atypiques
    3. **Partition** aléatoire de l'échantillon (apprentissage, validation, test)
    4. **Mise en œuvre** de l'apprentissage (méthodes).
    5. **Comparaison** des méthodes
    6. **Choix et exploitation** sur la base complète

Introduction

- **Apprentissage statistique**
  - ***Les méthodes:***
    - *Modèle linéaire général* (gaussien, binomial ou poissonien),
    - *Discrimination paramétrique* (linéaire ou quadratique) ou *non paramétrique*,
    - *k plus proches voisins*,
    - *Arbre de décision*,
    - *Réseau de neurones* (perceptron),
    - *Support vecteur machine*,
    - *Combinaison de modèles* (bagging, boosting).

Introduction

- **Apprentissage statistique**
  - ***Objectif de ce cours***

On s'intéresse aux méthodes ayant pour objectif la construction d'arbres binaires de décision, modélisant une discrimination ou une régression

Introduction

- **Apprentissage statistique**
  - ***Objectif de ce cours***

Ces méthodes ne sont efficaces que pour des tailles d'échantillons importantes et elles sont très calculatoires.

Néanmoins leur mode de représentation les rend très accessibles même au néophyte.

Plan

- 1. Introduction
- 2. **Les arbres de décision binaire**
- 3. Application à l'apprentissage supervisé
- 4. Forêt Aléatoire (Random Forest)

Arbre de décision binaire

- **Exemple Matlab**

```
>>load fisheriris;
```


Iris Setosa


Iris Versicolor


Iris Virginica

Exemple Matlab

- **Arbre de décision binaire**

```
>>load fisheriris;

>>t = classregtree(meas,species,...
                  'names',{'SL' 'SW' 'PL' 'PW'})

>>view(t)
```


Iris Setosa


Iris Versicolor


Iris Virginica

Exemple Matlab

- **Arbre de décision binaire**

```
>>load fisheriris;

>>t = classregtree(meas,species,...
                  'names',{'SL' 'SW' 'PL' 'PW'})

>>view(t)
```

SL : Sepal Length
SW : Sepal Width
PL : Petal Length
PW : Petal Width


Iris Setosa


Iris Versicolor


Iris Virginica

Exemple Matlab

- **Arbre**

```
>>load
>>t = cl
>>view(
```

l Length
al Width
Length
l Width

Iris S
ca

Exemple Matlab

- **Arbre de décision binaire**

Fisher's *Iris* Data

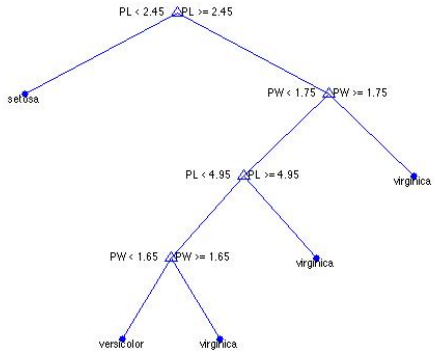
SL	SW	PL	PW	Species
5.1	3.5	1.4	0.2	*I. setosa*
4.9	3.0	1.4	0.2	*I. setosa*
4.7	3.2	1.3	0.2	*I. setosa*
…	…	…	…	
7.0	3.2	4.7	1.4	*I. versicolor*
6.4	3.2	4.5	1.5	*I. versicolor*
6.9	3.1	4.9	1.5	*I. versicolor*
…	…	…	…	
6.3	3.3	6.0	2.5	*I. virginica*
5.8	2.7	5.1	1.9	*I. virginica*
7.1	3.0	5.9	2.1	*I. virginica*
…	…	…	…	



Exemple Matlab



- **Arbre de décision binaire**

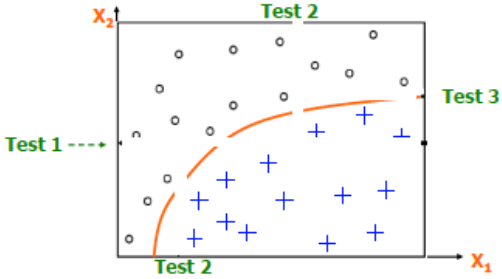


Les arbres de décision binaire

- **Principe**
  - Les données sont constituées de l'observation de variables X^j
  - Et d'une variable à expliquer Y à m modalités T_l observées sur un échantillon de n individus.
  - ***Objectif :*** Construire un arbre de discrimination binaire par identification d'une séquence de *noeuds*

Les arbres de décision binaire

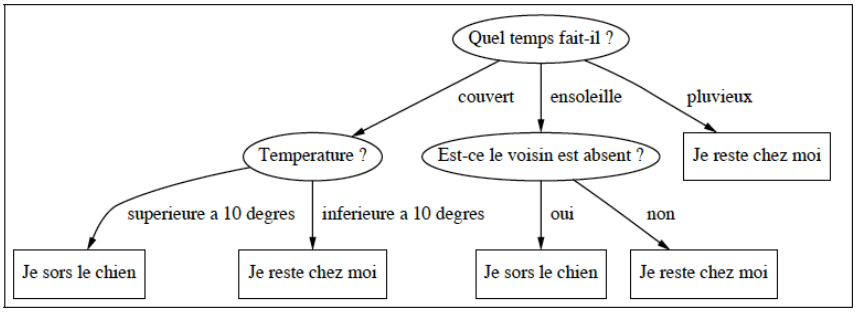
- **Illustration**



Construire une suite de tests permettant une partition de l'espace des données en sous-régions homogènes en terme de groupe

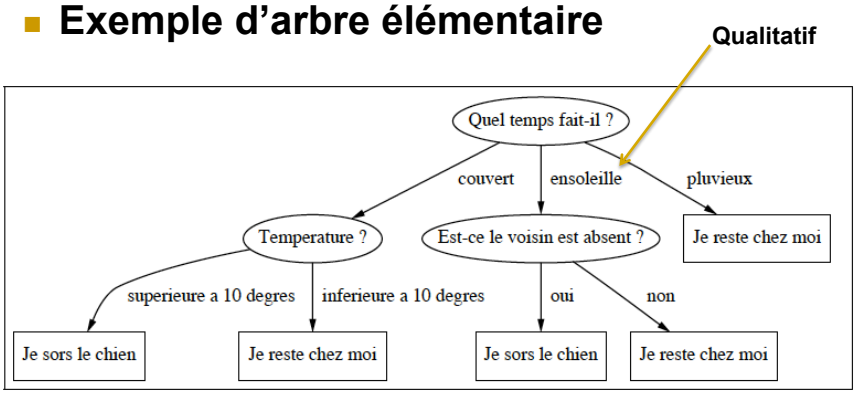
Les arbres de décision binaire

- **Exemple d'arbre élémentaire**



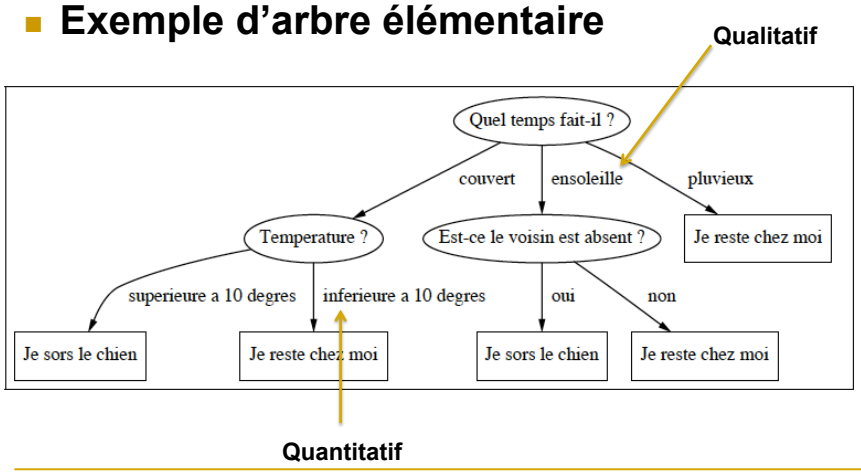
Les arbres de décision binaire

- **Exemple d'arbre élémentaire**



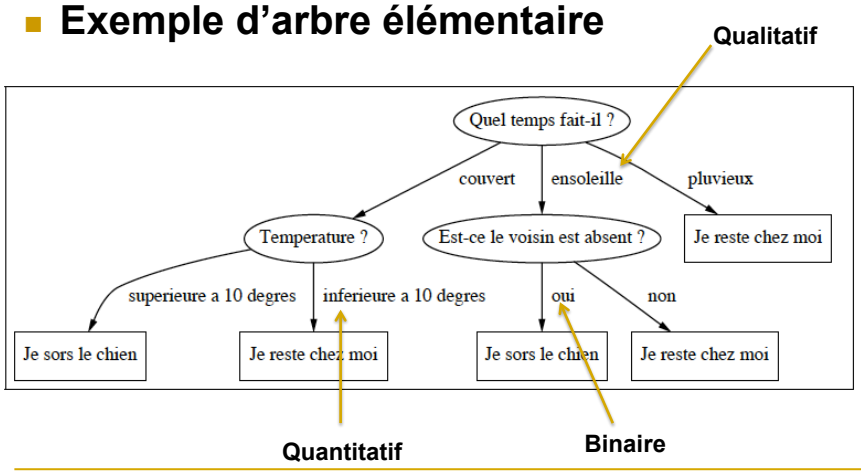
Les arbres de décision binaire

- **Exemple d'arbre élémentaire**



Les arbres de décision binaire

- **Exemple d'arbre élémentaire**

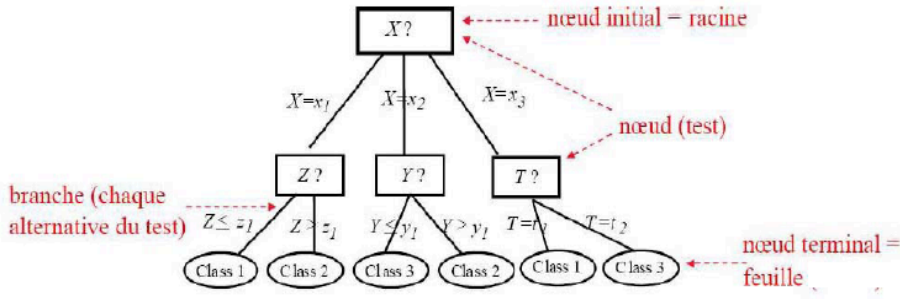


Les arbres de décision binaire

- **Définitions**
  - Un **noeud** est défini par le choix conjoint d'une variable parmi les explicatives et d'une division qui induit une partition en deux classes.
  - A chaque noeud correspond donc un sous-ensemble de l'échantillon auquel est appliquée **une dichotomie**.
  - Une **division** est elle-même définie par une valeur seuil de la variable quantitative sélectionnée ou un partage en deux (ou plus) groupes des modalités si la variable est qualitative.

Les arbres de décision binaire

- **Vocabulaire**



Les arbres de décision binaire

- **Algorithme : prérequis**

1. la **définition d'un critère** permettant de sélectionner la "meilleure" division parmi toutes celles admissibles pour les différentes variables ;

Les arbres de décision binaire

- **Algorithme : prérequis**

2. Une **règle** permettant de décider qu'un noeud est terminal : il devient ainsi *une feuille* ;

Les arbres de décision binaire

- **Algorithme : prérequis**

3. L'**affectation** de chaque *feuille* à l'une des classes ou à une valeur de la variable à expliquer.

Les arbres de décision binaire

- **Remarques**
  - Le point 2 est le plus délicat.
  - Un arbre trop détaillé, associé à une *surparamétrisation*, est instable et donc probablement plus défaillant pour la prévision d'autres observations.
  - Recherche donc d'un ***modèle parcimonieux.***

Les arbres de décision binaire

- **Algorithme générale**

Entrée : échantillon *S*
Initialiser l'arbre courant à l'arbre vide ;
(la racine est le nœud courant)
répéter
Décider si le nœud courant est terminal
Si le nœud est terminal **alors**
Lui affecter une classe
sinon
Sélectionner un test et créer autant de nouveaux nœuds fils qu'il y a de réponses possibles au test
FinSi
Passer au nœud suivant non exploré s'il en existe
Jusqu'à obtenir un arbre de décision *A*
Sortie : *A*

Les arbres de décision binaire

- **Remarques**
  - *Un noeud est **terminal** lorsque (presque) tous les exemples correspondant à ce noeud sont dans la même classe,*
  - *ou encore, s'il n'y a plus d'attributs non utilisés dans la branche correspondante. . .*

Les arbres de décision binaire

- **Remarques**
  - *On attribue à un noeud terminal la classe majoritaire*
  - *Lorsque plusieurs classes sont en concurrence, on peut choisir la classe la plus représentée dans l'ensemble de l'échantillon, ou en choisir une au hasard,*

Les arbres de décision binaire

- **Critères de construction**
  - **CART :** utilise l'indice de GINI
  - **ID3 et C4.5 :** utilise l'entropie
  - **CHAID :** un test du Khi2

Les arbres de décision binaire

- **Critère de GINI**
  - Soit ***S*** un échantillon de données et $S_1,\ldots, S_k$ un partitionnement de S en classe.
  - Le critère de ***GINI*** se définit alors par :

$$GINI(S) = \sum_{i=1}^{k} \frac{|S_i|(1-|S_i|)}{|S|} = \sum_{i=1}^{k} p_i(1-p_i)$$

Avec p_i *la probabilité d'appartenir à la classe i*

Les arbres de décision binaire

- **Critère de C4.5 (et IDE)**
  - Soit ***S*** un échantillon de données et $S_1,\ldots, S_k$ un partitionnement de S en classe.
  - Le critère de ***C4.5*** se définit alors par :

$$H(S) = -\sum_{i=1}^{k} \frac{|S_i|}{|S|} \log\left(\frac{|S_i|}{|S|}\right) = -\sum_{i=1}^{k} p_i \log(p_i)$$

Avec p_i *la probabilité d'appartenir à la classe i*

Les arbres de décision binaire

- **Remarques**
  - Les deux critères sont à valeur dans [0,1]
  - Sont nuls pour p_i=*0 et 1*
  - Sont maximums pour p_i=*0.5*
  - *Il s'agit donc bien d'une mesure du désordre de S.*

Les arbres de décision binaire

- **Notion de Gain**
  - Considérons un ensemble d'attributs binaires
  - Soit p_{pos} la position courante dans l'arbre et *T* un test :
    - *f* désigne le critère de Gini ou entropique
    - On définit alors le **Gain** associé à p_{pos} et *T* :

$$G_f(P_{pos},T) = f(S_{pos}) - \sum_{j=1}^{2} p_j . f(S_{pos_j})$$

Les arbres de décision binaire

- **Notion de Gain**

$$G_f(P_{pos},T) = f(S_{pos}) - \sum_{j=1}^{2} p_j . f(S_{pos_j})$$

  - Avec :
    - S_{pos} l'échantillon associé à P_{pos},
    - S_{posj} l'ensemble des éléments de S_{pos} qui satisfont la j^e branche de T,
    - p_j la proportion des éléments de S_{pos} qui satisfont la j^e branche de T.

Les arbres de décision binaire

- **Stratégie « gloutonne » de construction**

$$G_f(P_{pos},T) = f(S_{pos}) - \sum_{j=1}^{2} p_j . f(S_{pos_j})$$

  - Le 1er terme ne dépend pas de T, maximiser le gain revient donc à minimiser

$$\sum_{j=1}^{2} p_j . f(S_{pos_j})$$

  - Le gain est donc maximal quand l'attribut considéré permet d'annuler ce terme : tout est bien classé !

Les arbres de décision binaire

- **Stratégie « gloutonne » de construction**

$$G_f(P_{pos}, T) = f(S_{pos}) - \sum_{j=1}^{2} p_j \cdot f(S_{pos_j})$$

  - **Sélectionner l'attribut dont le gain est maximum correspond à une stratégie gloutonne : rechercher le test faisant le plus progresser la classification.**

Plan

- 1. Introduction
- 2. Les arbres de décision binaire
- 3. **Application à l'apprentissage supervisé**
- 4. Forêt Aléatoire (Random Forest)

Les arbres de décision binaire

- **Exemple**

Match à domicile ?	Balance positive ?	Mauvaises cond. climatiques ?	Match précédent gagné ?	Match gagné
V	V	F	F	V
F	F	V	V	V
V	V	V	F	V
V	V	F	V	V
F	V	V	V	F
F	F	V	F	F
V	F	F	V	F
V	F	V	F	F

Les arbres de décision binaire

Match à domicile ?	Balance positive ?	Mauvaises cond. climatiques ?	Match précédent gagné ?	Match gagné
V	V	F	F	V
F	F	V	V	V
V	V	V	F	V
V	V	F	V	V
F	V	V	V	F
F	F	V	F	F
V	F	F	V	F
V	F	V	F	F

- **Départ : « Quel attribut ? »**
  - **Critère de Gini:**

$$G_{GINI}(P_{pos},T) = GINI(S_{pos}) - \sum_{j=1}^{2} p_j . GINI(S_{pos_j})$$

  - **Match à domicile :**

$$\sum_{j=1}^{2} p_j . GINI(S_{pos_j}) = \frac{5}{8} GINI(S_1) + \frac{3}{8} GINI(S_2)$$

$$\sum_{j=1}^{2} p_j . GINI(S_{pos_j}) = \frac{5}{8}\frac{2}{5}\frac{3}{5} + \frac{3}{8}\frac{1}{3}\frac{2}{3} = \frac{7}{15}$$

Les arbres de décision binaire

Match à domicile ?	Balance positive ?	Mauvaises cond. climatiques ?	Match précédent gagné ?	Match gagné
V	V	F	F	V
F	F	V	V	V
V	V	V	F	V
V	V	F	V	V
F	V	V	V	F
F	F	V	F	F
V	F	F	V	F
V	F	V	F	F

- **Départ : « Quel attribut ? »**
  - **Critère de Gini:**

$$G_{GINI}(P_{pos},T) = GINI(S_{pos}) - \sum_{j=1}^{2} p_j.GINI(S_{pos_j})$$

  - **Balance Positive :**

$$\sum_{j=1}^{2} p_j.GINI(S_{pos_j}) = \frac{4}{8}GINI(S_1) + \frac{4}{8}GINI(S_2)$$

$$\sum_{j=1}^{2} p_j.GINI(S_{pos_j}) = \frac{4}{8}\frac{3}{4}\frac{1}{4} + \frac{4}{8}\frac{1}{4}\frac{3}{4} = \frac{3}{8}$$

Les arbres de décision binaire

Match à domicile ?	Balance positive ?	Mauvaises cond. climatiques ?	Match précédent gagné ?	Match gagné
V	V	F	F	V
F	F	V	V	V
V	V	V	F	V
V	V	F	V	V
F	V	V	V	F
F	F	V	F	F
V	F	F	V	F
V	F	V	F	F

- **Départ : « Quel attribut ? »**
  - **Critère de Gini:**

$$G_{GINI}(P_{pos},T) = GINI(S_{pos}) - \sum_{j=1}^{2} p_j.GINI(S_{pos_j})$$

  - **Match à domicile :** 7/15
  - **Balance Positive :** 3/8
  - **MCC :** 7/15
  - **Match Préc. Gagné :** 1/2

Les arbres de décision binaire

Match à domicile ?	Balance positive ?	Mauvaises cond. climatiques ?	Match précédent gagné ?	Match gagné
V	V	F	F	V
F	F	V	V	V
V	V	V	F	V
V	V	F	V	V
F	V	V	V	F
F	F	V	F	F
V	F	F	V	F
V	F	V	F	F

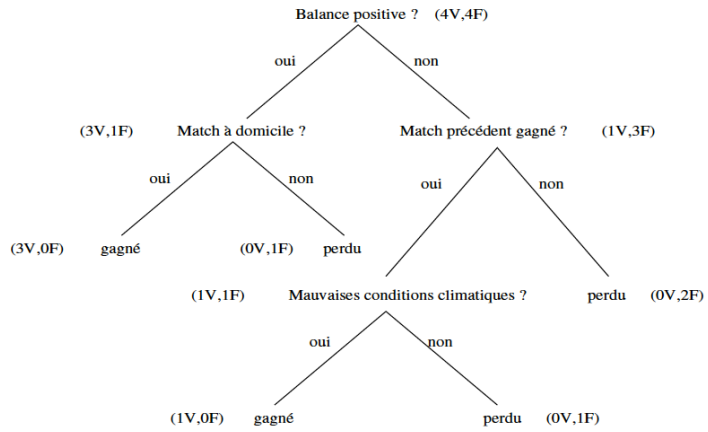
- **Départ : « Quel attribut ? »**
  - **Critère de Gini:**

$$G_{GINI}(P_{pos}, T) = GINI(S_{pos}) - \sum_{j=1}^{2} p_j . GINI(S_{pos_j})$$

  - **Match à domicile :** 7/15 (2)
  - **Balance Positive :** 3/8 (1)
  - **MCC :** 7/15 (2)
  - **Match Préc. Gagné :** 1/2 (3)

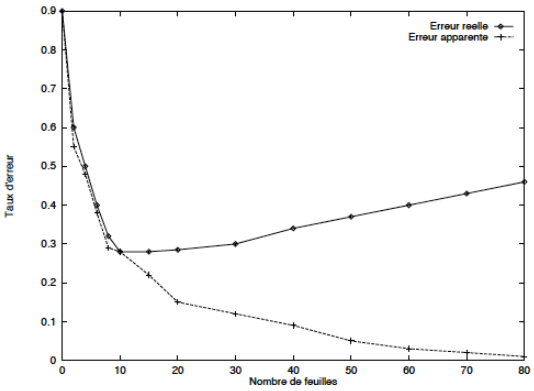
Les arbres de décision binaire

- **Arbre obtenu**



Les arbres de décision binaire

- **Limite de cette approche**



Un arbre peut avoir une erreur apparente nulle mais une erreur réelle importante, c'est-à-dire être bien adaptée à l'échantillon mais avoir un pouvoir de généralisation faible.

Notion de sur-apprentissage (overfitting).

Les arbres de décision binaire

- **Méthode de Breiman et al. (1984)**

1. *Construire l'arbre maximal* $\mathbf{A_{max}}$
2. *Ordonner les sous-arbres selon une séquence emboîtée suivant la décroissance d'un critère pénalisée de déviance ou de taux de mal-classés*
3. *Sélectionner le sous-arbre optimal ; c'est la procédure d'**élagage**.*

Les arbres de décision binaire

- **Elagage (*pruning*)**
  - **L'enjeu de la recherche de la taille optimale consiste à :**
    - Soit stopper la croissance de l'arbre sur une branche : pré-élagage
    - Soit à réduire l'arbre complet : post-élagage
  - **Objectif :** obtenir un arbre de taille correspondant au « coude » de la courbe du taux d'erreur sur l'échantillon test, quand le taux d'erreur commence à stagner ou diverger.

Les arbres de décision binaire

- **Pré-élagage (*pruning*)**
  - ***Méthode CHAID:***
    - Chi-Squared Automatic Interaction Detection
    - On accepte la segmentation si le Khi-2 calculé sur un noeud est significativement supérieur à un seuil théorique lié à un risque α fixé.

Les arbres de décision binaire

- **Pré-élagage (*pruning*)**
  - ***Méthode CHAID:***
    - Chi-Squared Automatic Interaction Detection
    - On accepte la segmentation si le Khi-2 calculé sur un noeud est significativement supérieur à un seuil théorique lié à un risque α fixé.

Questions:
Comment choisir
α ?

Les arbres de décision binaire

- **Post-élagage (*pruning*)**
  - ***Méthode CART*** *(arbre binaire)*
    - Le principe est de construire l'arbre en deux temps *:*
      - ***1re phase d'expansion*** *: produire des arbres les plus purs possibles*
      - ***2e phase*** *: réduire l'arbre en utilisant un autre critère pour comparer des arbres de tailles différentes.*
    - Le temps de construction de l'arbre est plus élevé
    - L'objectif est d'obtenir un arbre plus performant en classement et limitant *l'overfitting*

Les arbres de décision binaire

- **Post-élagage (*pruning*)**
  - ***Méthode CART*** *(arbre binaire)*
    - **1.** Soit A_{max} l'arbre obtenu à partir de l'ensemble d'apprentissage
    - **2.** Construire une suite d'arbres $\{A_{max}, A_1, A_2, ..., A_n\}$ en partant des feuilles et en remontant vers la racine en transformant un nœud en feuille à chaque étape.
    - **3.** Comparer le coût du nouvel arbre à celui du précédent et arrêter l'élagage si le coût est supérieur

Les arbres de décision binaire

- **Post-élagage (*pruning*)**
  - ***Méthode CART*** *(arbre binaire)*
    - *Le critère du coût en complexité utilisé par CART pour un arbre A est :*

$$Err(A) + a\left|L(A)\right|$$

      - avec *Err(A)* la fraction des observations de validation mal classées par l'arbre A
      - *L(T)* le nombre de feuilles dans l'arbre A,
      - *a,* le coût de pénalité par nœud variant de 0 à l'infini.

Les arbres de décision binaire

- **Post-élagage (*pruning*)**
  - ***Méthode CART*** *(arbre binaire)*
    - *Le critère du coût en complexité utilisé par CART pour un arbre A est :*

$$Err(A) + a\left|L(A)\right|$$

  - **Si *a* = *0* pas de pénalité de noeud; le meilleur arbre = l'arbre complet non élagué**
  - **Si *a* devient très grand, le coût de la pénalité de nœud submerge l'erreur de mauvaise classification ; le meilleur arbre = l'arbre avec un nœud**

Les arbres de décision binaire

- **Post-élagage (*pruning*)**
  - ***Exemple (CART)***

Match à domicile ?	Balance positive ?	Mauvaises cond. climatiques ?	Match précédent gagné ?	Match gagné
V	V	F	F	V
F	F	V	V	V
V	V	V	F	V
V	V	F	V	V
F	V	V	V	F
F	F	V	F	F
V	F	F	V	F
V	F	V	F	F

Les arbres de décision binaire

- **Post-élagage (*pruning*)**
  - ***Exemple (CART)***

L'arbre **T0** est l'arbre construit précédemment.

T1 est l'arbre obtenu à partir de la position 2.

T2 est obtenu en élaguant à partir de la position 1.

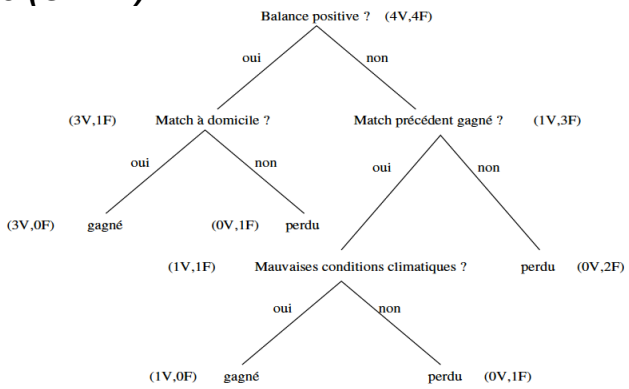
T3 est réduit à une feuille, portant par exemple la classe gagné.

L'algorithme d'élagage retournera alors l'arbre T2

Les arbres de décision binaire

- **Post-élagage (*pruning*)**
  - ***Exemple (CART)***

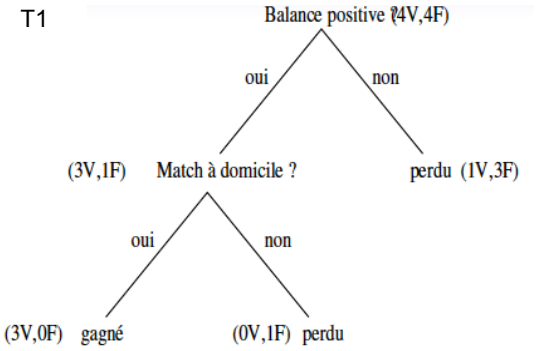
T0



Les arbres de décision binaire

- **Post-élagage (*pruning*)**
  - ***Exemple (CART)***

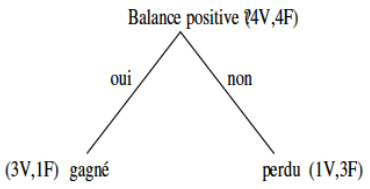
T1



Les arbres de décision binaire

- **Post-élagage (*pruning*)**
  - ***Exemple (CART)***

T2

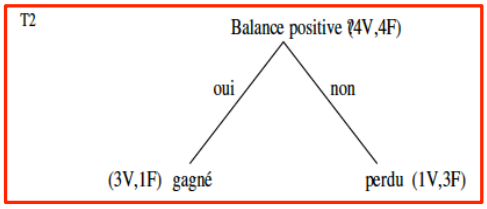


T3

Gagné (4V,4F)

Les arbres de décision binaire

- **Post-élagage (*pruning*)**
  - ***Exemple (CART)***



C'est l'arbre qui minimise l'erreur tout en gardant un pouvoir de généralisation à d'autres données

Les arbres de décision binaire

- **Avantages / Inconvénients**
  - ***Avantages***
    - *Méthode non paramétrique : aucune hypothèse a priori sur la distribution des données*
    - *Tout type de données : variables quantitatives ou qualitatives*
    - *Pas de paramétrage difficile pour la construction de l'arbre*
    - *Efficacité et disponibilité (présent dans tous les logiciels de Data Mining)*
    - *Lisibilité du résultat*

Les arbres de décision binaire

- **Avantages / Inconvénients**
  - ***Avantages***
    - *Un arbre de décision peut être lu et interprété directement : on peut le traduire en règles sans perte d'information.*
    - *Des règles de classification compréhensibles (si les arbres ne sont pas trop grands)*
    - *Des règles facilement explicables à des non-statisticiens comparées aux sorties d'autres classifieurs tels que les fonctions discriminantes.*

Les arbres de décision binaire

- **Avantages / Inconvénients**
  - ***Avantages***
    - ***Remarque :*** *L'appropriation de l'outil par les experts du domaine assure une meilleure interprétation et compréhensibilité des résultats.*

Les arbres de décision binaire

- **Avantages / Inconvénients**
  - ***Limites***
    - *Test d'un seul attribut à la fois: coupes parallèles aux axes*
    - *Echantillon d'apprentissage de grande taille*
    - *Non incrémental : recommencer la construction de l'arbre si on veut intégrer de nouvelles données*
    - *Sensible à de petites variations dans les données*
    - *Instabilité*

Les arbres de décision binaire

- **Avantages / Inconvénients**
  - ***Solutions***
    - *le **Bagging** (pour Bootstrap Aggregating)*
    - *les **Random Forests (RF)***

Plan

- 1. Introduction
- 2. Les arbres de décision binaire
- 3. Application à l'apprentissage supervisé
- 4. **Forêt Aléatoire (Random Forest)**

Forêt aléatoire (RF)

- **Introduction**
  - Les arbres ont connu un net regain d'intérêt lorsque les méthodes d'agrégation des classifieurs tels que le ***boosting***, le ***bagging***, ***Random Forests*** (forêts d'arbres) ont été développées et popularisées dans la communauté de l'apprentissage automatique.
  - Leur variabilité très marquée, leur permettent de tirer parti des avantages de la combinaison des prédicteurs.

Forêt aléatoire (RF)

- **Bootstrap**
  - Un ensemble de données « bootstrap » est un ensemble de données obtenu en sélectionnant **au hasard avec remise** *n* observations parmi les *n* observations de l'ensemble d'entraînement.

Forêt aléatoire (RF)

- **Bootstrap**
  - Certaines observations sont dupliquées tandis que d'autres sont absentes ; ce qui introduit une part d'aléatoire.
  - L'intérêt de telle méthode est de répéter la procédure et d'utiliser chaque ensemble de données pour construire un modèle et le tester.
  - On dispose alors de différentes réalisations de la statistique estimée (ou du modèle).

Forêt aléatoire (RF)

- **Bagging (Breiman et al 2001)**
  - *Bagging* contraction de « Bootstrap aggregation »
  - Le but premier du *bagging* est d'atténuer l'instabilité inhérente à certaines méthodes de discrimination.
  - Une méthode de discrimination est dite peu robuste si un changement mineur dans les données provoque un changement assez important de modèle. (Ex : arbres de décision).

Forêt aléatoire (RF)

- **Bagging : En pratique**
  - On va construire un grand nombre d'arbres de décision différents pour un même problème : **une forêt**
  - Pour construire une forêt, on injecte **de l'aléatoire** avant ou pendant la construction d'un arbre
  - On construit **plusieurs arbres** « randomisés »
  - On **agrège** l'ensemble des arbres obtenus : Pour classer un individu on prend la décision finale par un vote majoritaire

Forêt aléatoire (RF)

- **Bagging : En pratique**
  - Construction :
    - On tire à chaque *noeud* de l'arbre ***m*** variables uniformément et on cherche la « meilleure » coupure uniquement parmi celles-ci.
    - Les arbres de décision sont complets : construits automatiquement sans *pre*- ou *post*-pruning

Forêt aléatoire (RF)

- **Bagging : En pratique**
  - **Construction :**
    - Normalement pour chaque échantillon « *bootstrap* » 2/3[1] des exemples sont sélectionnés en moyenne, le reste étant des doublons.
    - Donc pour chaque sous base 1/3 des exemples de *D* ne sont pas sélectionnés et sont considérés comme « *oob* » *(out of bag)*.
    - Ils serviront à : *(i)* l'évaluation interne d'une forêt (estimation de l'erreur de classification après l'ajout d'un arbre à la forêt), *(ii)* estimer l'importance des variables.

63,2% en fait ☺

Forêt aléatoire (RF)

- **Performances**
  - La sélection d'un sous-ensemble de variables explicatives (input) parmi un grand nombre, permet généralement :
    - **De réduire de beaucoup les temps de calcul.**
    - **D'obtenir une plus grande variété de modèles**.
  - L'aggrégation des valeurs ou classes prédites (vote majoritaire) par tous les modèles générés devrait alors donner un classifieur plus robuste et plus précis**.**

Forêt aléatoire (RF)

- **Erreur de généralisation**
  - Pour chaque élément *x* et pour chaque arbre *k* (parmi les K arbres construits) pour lequel *x* a été sélectionné comme *oob*,
    - Prévoir la classe de *x* selon *k*.
    - Déterminer la classe majoritaire *j* de *x* pour les cas trouvés.
    - Pour tous les points *x,* calculer la moyenne de fois que *j* est différente de sa vrai classe : C'est l'erreur de classification sur l'*oob*.

Forêt aléatoire (RF)

- **Erreur de généralisation**

Breiman démontre que lorsque le nombre d'arbres impliqués dans la forêt de prédiction augmente, le taux d'erreur en généralisation converge vers une valeur limite.