COURS SUR
L'APPRENTISSAGE ARTIFICIEL

COURS MASTER IFI 2010/2011

JEAN-DANIEL ZUCKER

DR À L'IRD UR GEODES
(MODÉLISATION MATHÉMATIQUES ET INFORMATIQUES DES SYSTÈMES COMPLEXES)
UMMISCO UMI 209

UNIVERSITE PIERRE & MARIE CURIE
SCIENCE A PARIS

IRD
Institut de recherche pour le développement

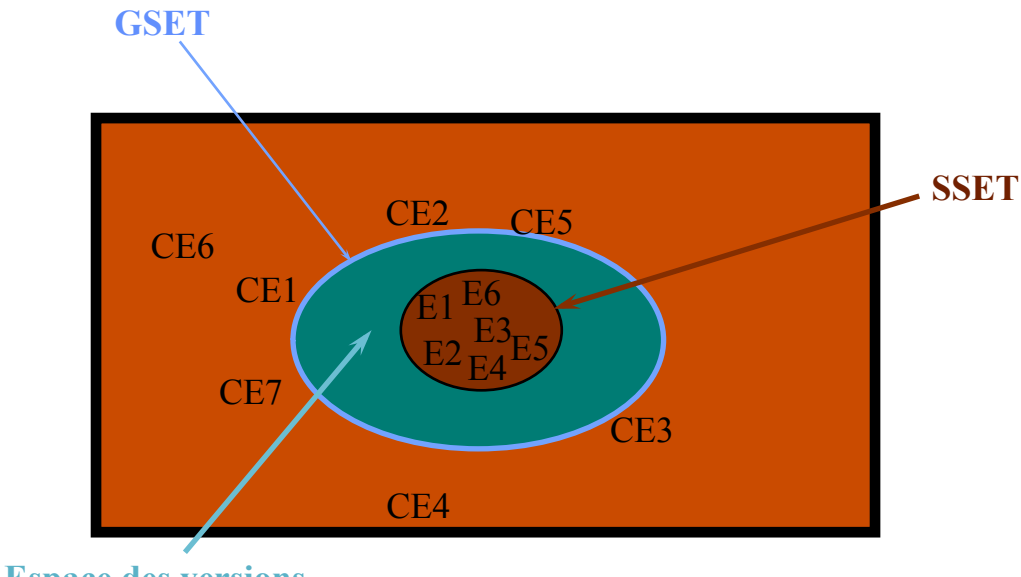
Administratif: 1/2 Module Apprentissage (18ECTS)

- **Séance 1: Jeudi 25 Novembre –** INTRO GÉNÉRALE
  - **Introduction, principe inductif, historique, formulation**
  - **Quelques mots sur l'apprentissage statistique**
  - **Espace des versions et algorithme**
- **Séance 2: Lundi 6 Décembre –** APPRENTISSAGE **SUPERVISÉ**
- **Séance 3: Lundi 13 Décembre –** APPRENTISSAGE **NON-SUPERVISÉ**
- **Séance 4: Mardi 11 Janvier 2011 –** ALGORITHMES **EVOLUTIONNAIRES**
- **Séance 5: Jeudi 14 Janvier 2011 –** ALGORITHMES PAR **RENFORCEMENT**
- **Séance 6: Lundi 17 Janvier 2011 –** MINI-PROJET

I.

Suite et Fin de l'espace des Versions

ESPACE DES VERSIONS



Bornes de l'espace de recherche

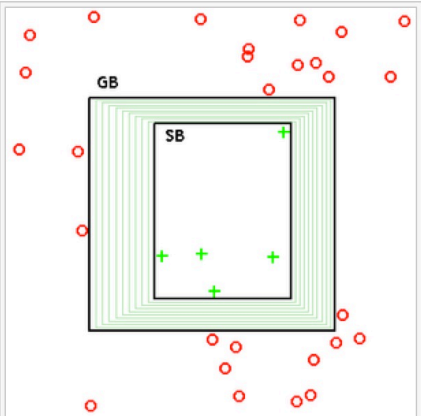
S= {s |s généralisation maximalement spécifique}

s est une généralisation maximalement spécifique si s est cohérente vis-à-vis des observations, et il n'existe pas de généralisations s' ∈ S telle que s' soit plus spécifique que s.

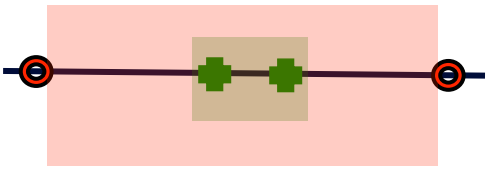
G = {g | g généralisation maximallement générale}

g est une généralisation maximallement générale si g est cohérente vis-à-vis des observations et il n'existe pas de généralisation g' ∈ G qui soit plus générale que g.

Espace des versions (deux illustrations graphiques)



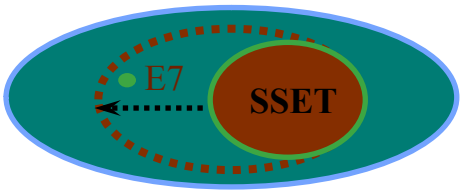
Représentation des hypothèses de rectangles à partir d'exemples exemples positifs (les croix vertes, qui doivent être à l'intérieur du rectangle) et négatifs (les ronds rouges, qui doivent être à l'extérieur du rectangle). Le rectangle GB est l'hypothèses la plus **générale** (en généralisant plus on couvrirait des exemples négatifs), et SB est la plus **spécifique** (en spécialisant plus on ne couvrirait plus certains exemples positifs). Les rectangles verts représentent d'autres hypothèses de l'espace de versions.



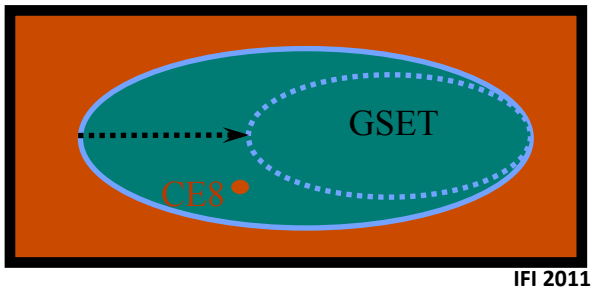
Concept: [a,b]
Hypothèse [x,y]

Algorithme d'élimination des candidats: principe

Ajouter un exemple E7



Ajouter un contre-exemple CE8



Algorithme d'élimination des candidats

- **Initialiser S et G par (respectivement) :**
  - **l'ensemble des hypothèses les plus spécifiques (resp. les plus générales) cohérentes avec le 1er exemple positif connu.**
- **Pour chaque nouvel exemple (positif ou négatif)**
  - **mettre à jour S**
  - **mettre à jour G**
- **Jusqu'à convergence**

  ou jusqu'à ce que S = G = {Ø}

Mise à jour de S

- **x_i est négatif**
  - **Eliminer les hypothèses de *S* couvrant x_i**
- **x_i est positif**
  - **Généraliser les hypothèses de *S* ne couvrant pas x_i juste assez pour qu'elles le couvrent**
  - **Puis éliminer les hypothèses de *S***
    - **couvrant un ou plusieurs exemples négatifs**
    - **plus générales que des hypothèses de *S***

Mise à jour de G

- **x_i est positif**
  - **Eliminer les hypothèses de *G* ne couvrant pas x_i**
- **x_i est négatif**
  - **Spécialiser les hypothèses de *G* couvrant x_i juste assez pour qu'elles ne le couvrent plus**
  - **Puis éliminer les hypothèses de *G***
    - **n'étant pas plus générales qu'au moins un élément de *S***
    - **plus spécifiques qu'au moins une autre hypothèse de *G***

Recherche ascendante en largeur d'abord

- La recherche est monotone (les généralisations de s sont de plus en plus générales).
- Les révisions de S à chaque nouvel exemple sont consistantes avec les exemples positifs passés
  (hyp: pas d'erreurs dans les exemples)
- On ne stocke pas les exemples positifs.
- En revanche, on stocke les exemples négatifs pour vérifier qu'il n'y ait pas de sur-généralisation.

Exemple: Couple d'objets (ordre ne compte pas)

objet : taille {grand petit}

forme-géométrique : {triangle, cercle}

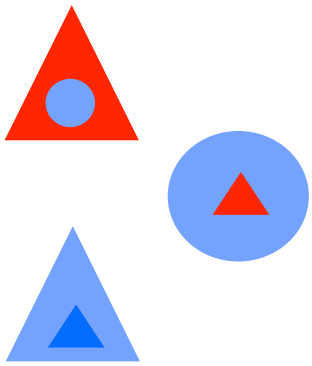
couleur : {rouge, bleu}

S = {∅} G = {∅}

E1 positif: (grand triangle rouge) & (petit cercle bleu)

E2 positif: (petit triangle rouge) & (grand cercle bleu)

E3 négatif: { (grand triangle bleu) (petit triangle bleu)}

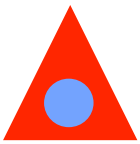


Exemple(1/2)

Exemple positif: {(grand triangle rouge) (petit cercle bleu)}

S1 = {(grand triangle rouge) (petit cercle bleu)}

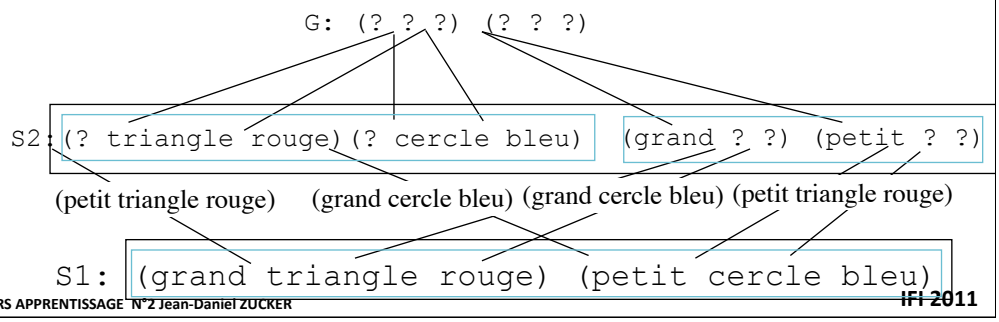
G1 = {(? ? ?) (? ? ?)}



Exemple positif: {(petit triangle rouge) (grand cercle bleu)}

S2 : {(? triangle rouge) (? cercle bleu),
(grand ? ?) (petit ? ?)}

G2 = {(? ? ?) (? ? ?)}

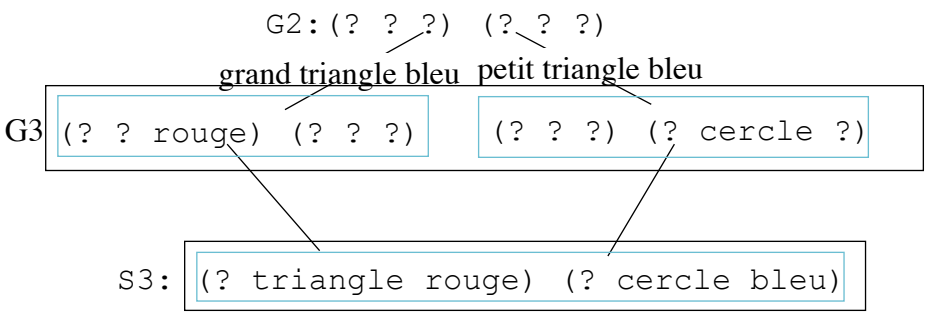


Exemple 2 (2/2)

E3 négatif: { (grand triangle bleu) (petit triangle bleu)}

S3 = { (? triangle rouge) (? cercle bleu) }

G3 = {(? ? rouge) (? ? ?), (? ? ?) (? cercle ?) }



Exercices: calculer S et G (pour le 6 Décembre 2011)

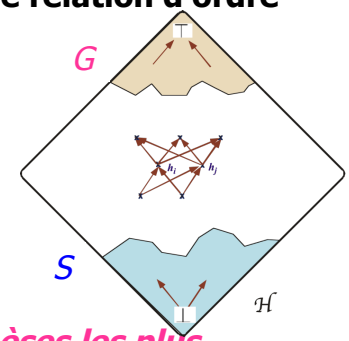
Cas 1 (converge)
- **E1 positif (grand rouge cercle)**
- **E2 négatif (petit rouge triangle)**
- **E3 positif (petit rouge cercle)**
- **E4 négatif (grand bleu cercle)**

Cas 2 (ne converge pas)
- **E1 positif (grand rouge cercle)**
- **E2 négatif (petit bleu triangle)**
- **E3 positif (petit rouge cercle)**
- **E4 négatif (moyen vert carré)**

Cas 3 (langage insuffisant)
- **E1 positif (grand rouge cercle)**
- **E2 négatif (grand bleu cercle)**
- **E4 positif (petit bleu cercle)**

Résumé: l'espace des versions

- **L'espace des versions structuré par une relation d'ordre partiel peut être représenté par :**
  - **sa borne supérieure : *G-set***
  - **sa borne inférieure : *S-set***



- ***G-set = Ensemble de toutes les hypothèses les plus générales cohérentes avec les exemples connus***
- ***S-set = Ensemble de toutes les hypothèses les plus spécifiques cohérentes avec les exemples connus***

II.

Apprentissage supervisé:

Approche symbolique

Algorithmes supervisés

- Recherche d'un **partitionnement** ou d'une **fonction de décision** dans l'espace des exemples
- **Guidée par étiquettes y_i fournies par les experts**
- ***Échantillon d'apprentissage*** **:**

$$\mathcal{S}_m = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), ..., (\boldsymbol{x}_m, y_m)\}$$

  - **Objectif :**
    - Rendre compte des données
    - Prédire l'étiquette de données encore inconnues

Rem: espace des versions pas utilisable en pratique. Il faut choisir.

Apprentissage inductif supervisé

f est cachée !

$$P(\boldsymbol{x}, u)$$
$$u_i = f(\boldsymbol{x}_i)$$

$$\mathcal{S}_m = \{(\boldsymbol{x}_i, u_i)\}_{1 \leq i \leq m}$$

Échantillon d'apprentissage

$$u = h(\boldsymbol{x})$$

- ***Identification*** : ***h*** **« proche de »** ***f***
- ***Prédiction*** : ***h*** **« bonne règle de décision »**

Critère de performance

- **Objectif :** trouver une hypothèse $h \in \mathcal{H}$ minimisant ***le risque réel***

  (*espérance de risque, erreur en généralisation*)

$$R(h) = \int_{X \times Y} l\left(h(\boldsymbol{x}), u\right) \, dP(\boldsymbol{x}, y)$$

Fonction de perte

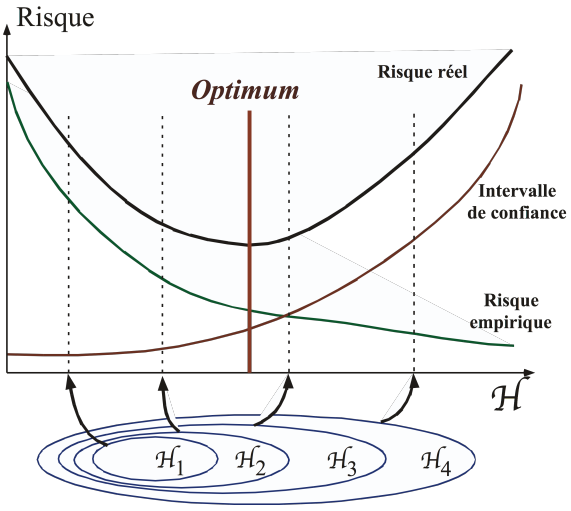
Étiquette prédite

Étiquette vraie (ou désirée)

Loi de probabilité jointe sur $\mathcal{X} \times \mathcal{Y}$

- **PB:** Seul le ***risque empirique*** est mesurable

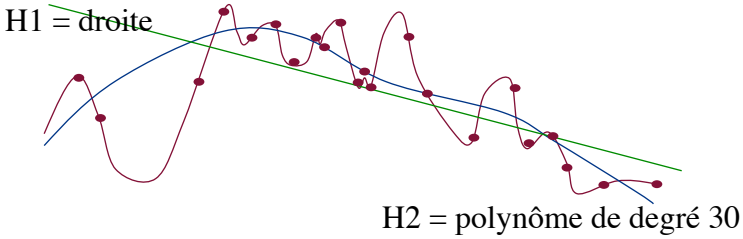
Risque réel, empirique et SRM: *régler le compromis par réglage automatique de l'espace d'hypothèses*



- La procédure s'appuie sur une structure sur $\mathcal{H}$ définie *a priori* -> *compromis*

La théorie de la régularisation

- Issue de l'étude des problèmes ***« mal posés »*** (plusieurs solutions)



- Il faut **imposer des conditions supplémentaires**
  - Contraindre l'espace des paramètres si $\mathcal{H}$ = {fonctions paramétrées}
  - Imposer des conditions de régularité (e.g. dynamique limitée)
  - …

$$R_{Pén.}(h) = R_{Emp}(h) + \lambda\, G(h)$$

Apprentissage supervisé

- **Méthodes**
  - *Plus proches voisins*
  - **Partitionnement**
    - *Induction d'arbres de décision* **(CART, C5.0, J48, …)**
  - **Fonction de décision**
    - *Réseaux de neurones* **(Perceptron multi-couches)**
    - *SVM* **(Séparateurs à Vastes Marges)**
    - **…**
- **Applications**
  - **Détection de fraudes, identification des cibles marketing, prédiction du risque cardio-vasculaire, …**

Induction d'arbres de décision [Quinlan, 1983] [Breiman et al., 1984]

- **Les arbres de décision sont des classeurs pour des instances représentées dans un formalisme attribut/valeur**
  - **Les nœuds de l'arbre testent les attributs**
  - **Il y a une branche pour chaque valeur de l'attribut testé**
  - **Les *feuilles* spécifient les catégories (deux ou plus)**

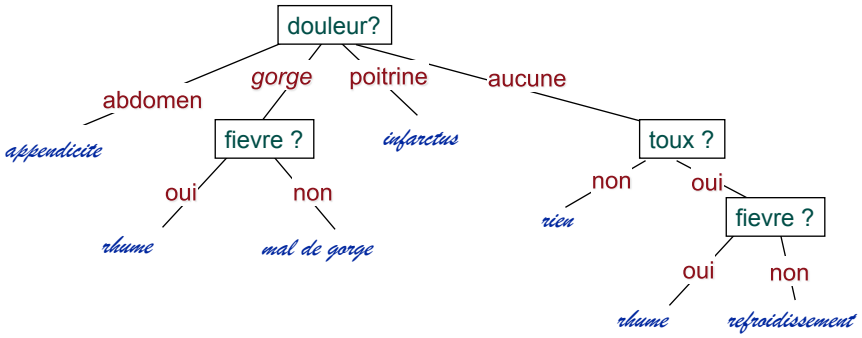


Illustration [Quinlan, 86]

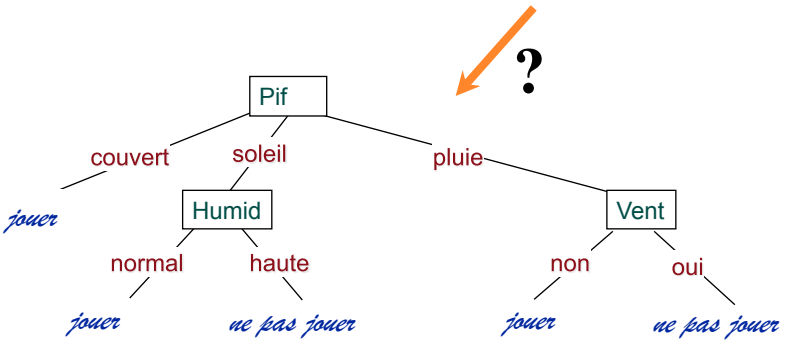
Attributs	Aspect Ciel (Pif)	Temp.	Humid.	Vent
Valeurs possibles	soleil, couvert, pluie	chaud, bon, frais	normale, haute	vrai, faux

N°	Pif	Temp	Humid	Vent	Golf
1	soleil	chaud	haute	faux	NePasJouer
2	soleil	chaud	haute	vrai	NePasJouer
3	couvert	chaud	haute	faux	Jouer
4	pluie	bon	haute	faux	Jouer
5	pluie	frais	normale	faux	Jouer
6	pluie	frais	normale	vrai	NePasJouer
7	couvert	frais	normale	vrai	Jouer
8	soleil	bon	haute	faux	NePasJouer
9	soleil	frais	normale	faux	Jouer
10	pluie	bon	normale	faux	Jouer
11	soleil	bon	normale	vrai	Jouer
12	couvert	bon	haute	vrai	Jouer
13	couvert	chaud	normale	faux	Jouer
14	pluie	bon	haute	vrai	NePasJouer

la classe

Illustration [Quinlan, 86]

N°	Pif	Temp	Humid	Vent	Golf
1	soleil	chaud	haute	faux	NePasJouer
2	soleil	chaud	haute	vrai	NePasJouer
3	couvert	chaud	haute	faux	Jouer
4	pluie	bon	haute	faux	Jouer
5	pluie	frais	normale	faux	Jouer
6	pluie	frais	normale	vrai	NePasJouer
7	couvert	frais	normale	vrai	Jouer
8	soleil	bon	haute	faux	NePasJouer
9	soleil	frais	normale	faux	Jouer
10	pluie	bon	normale	faux	Jouer
11	soleil	bon	normale	vrai	Jouer
12	couvert	bon	haute	vrai	Jouer
13	couvert	chaud	normale	faux	Jouer
14	pluie	bon	haute	vrai	NePasJouer



Induction d'arbres de décision

Quel critère de choix ?

- **L'espace des arbres de décision est gigantesque**
- **Critère à optimiser**
  1- ***Nombre d'erreurs d'étiquetage*** *(risque empirique)* **induit par l'arbre sur l'échantillon d'apprentissage**
     - **Insuffisant**
  2- ***« Simplicité »*** **de l'arbre** (SRM)
     - **Mesure de taille …**
     - **Espérance du nombre d'attributs à tester**

Induction d'arbres de décision

Comment explorer l'espace des arbres ?

- **Recherche exhaustive ?**
  - **Exclue**
- **Recherche par gradient ?**
  - **Pas efficace**
- **… ?**

➥ **Construction itérative de l'arbre**

Algorithme C5.0 (C4.5, ID3, J48, …)

- **Caractéristiques :**
  - **Approche « diviser pour régner »**
  - **Récursive, en profondeur d'abord**
  - **Recherche gloutonne**

Algorithme

Procédure `Construire_arbre`(*S*, *nœud*)

Si tous les exemples dans *S* appartiennent à la même classe

Alors créer une feuille portant le nom de cette classe

Sinon

`Choix_meilleur_attribut`(***X***) **pour créer un nœud test**

Si aucun test acceptable, alors créer une feuille étiquetée par la classe majoritaire dans *S*

Sinon

Pour chaque branche *i* de *X*

Soit S_i l'ensemble des exemples correspondant à cette branche

Créer $noeud_i$;

`Construire_arbre`(S_i, $noeud_i$)

Algorithme

- ***Question*** *: comment choisir « localement » le meilleur attribut à tester pour optimiser la* ***simplicité*** *de l'arbre (mesure globale) ?*
  - **Réponse théorique (semi-fondée)**
    - Mesurer le *gain d'information* (sur la classe à prédire) obtenu par chaque attribut testé
    - Retenir **l'attribut apportant le plus grand gain d'information**
  - **Gain d'information**
    - Mesure d'entropie, Critère Gini, …

Le critère entropique

- **L'entropie de Boltzmann ...**
- **... et de Shannon**
  - **Shannon en 1949 a proposé une mesure d'entropie valable pour les distributions discrètes de probabilité.**
  - **Elle exprime la quantité d'information, c'est à dire le nombre de bits nécessaire pour spécifier la distribution**
  - **L'entropie d'information est:**

$$I = -\sum_{i=1..k} p_i \times \log_2(p_i)$$

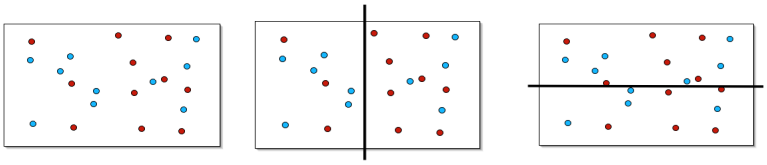
où p_i est la probabilité de la classe C_i.

Le critère entropique

$$I(S) = -\sum_{i=1}^{C} p(c_i) \cdot \log\ p(c_i)$$

$p(c_i)$: *probabilité de la classe* c_i

- Nulle quand il n'y a qu'une classe
- D'autant plus grande que les classes sont équiprobables
- Vaut $log_2(k)$ quand les k classes sont équiprobables



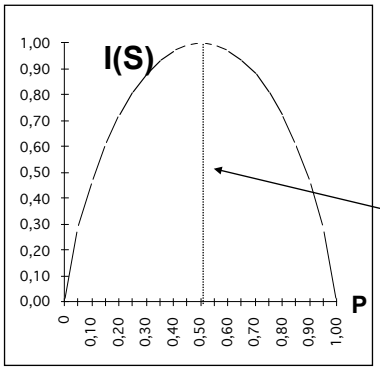
Le critère entropique : le cas de deux classes

- **Pour *C*=2 on a :** $I(S) = -p_+ \times \log_2(p_+) - p_- \times \log_2(p_-)$

D'après l'hypothèse on a $p_+ = p/(p+n)$ et $p_- = n/(p+n)$

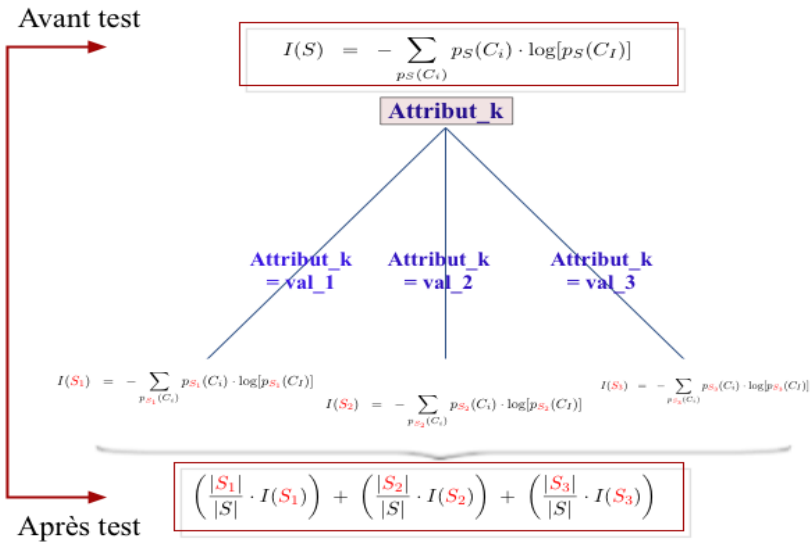
d'où $I(S) = -\frac{p}{(p+n)} \log \left(\frac{p}{(p+n)}\right) - \frac{n}{(p+n)} \log\left(\frac{n}{(p+n)}\right)$

et $I(S) = -P \log P - (1-P) \log(1-P)$



P=p/(p+n)=n/(n+p)=0.5
équiprobable

Mesure du gain d'information



Gain entropique associé à un attribut A

$$Gain(S,A) = I(S) - \sum_{v \in valeurs(A)} \frac{|S_v|}{|S|} \cdot I(S_v)$$

$|S_v|$: taille de la sous-population dans la branche *v* de *A*

En quoi la connaissance de la valeur de l'attribut A m'apporte une information sur la classe d'un exemple

Exemple

- **Entropie de l'ensemble initial d'exemples**

  $I(p,n) = -9/14 \log_2(9/14) - 5/14 \log_2(5/14) = 0.940$

- **Entropie des sous-arbres associés au test sur Pif ?**
  - $p_1 = 4\ n_1 = 0$: $I(p_1,n_1) = 0$
  - $p_2 = 2\ n_2 = 3$: $I(p_2,n_2) = 0.971$
  - $p_3 = 3\ n_3 = 2$: $I(p_3,n_3) = 0.971$

  } **0.694**

- **Entropie des sous-arbres associés au test sur Temp ?**
  - $p_1 = 2\ n_1 = 2$: $I(p_1,n_1) = 1$
  - $p_2 = 4\ n_2 = 2$: $I(p_2,n_2) = 0.918$
  - $p_3 = 3\ n_3 = 1$: $I(p_3,n_3) = 0.811$

  } **0.911**

Exemple

- **Pour les exemples initiaux**

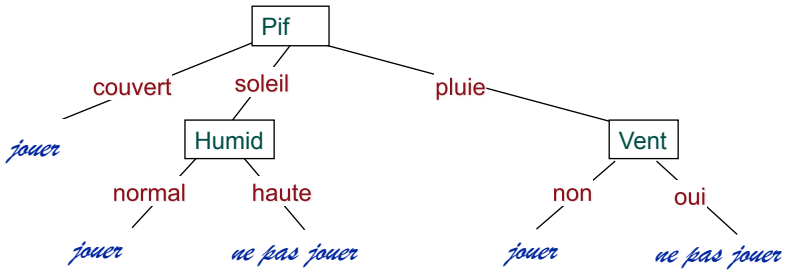
  $I(S) = -9/14 \log_2(9/14) - 5/14 \log_2(5/14)$

- **Entropie de l'arbre associé au test sur Pif ?**
  - $E(Pif) = 4/14\ I(p_1,n_1) + 5/14\ I(p_2,n_2) + 5/14\ I(p_3,n_3)$
  - ➥ $Gain(Pif) = 0.940 - 0.694 = 0.246$ bits
  - $Gain(Temp) = 0.029$ bits
  - $Gain(Humid) = 0.151$ bits
  - $Gain(Vent) = 0.048$ bits

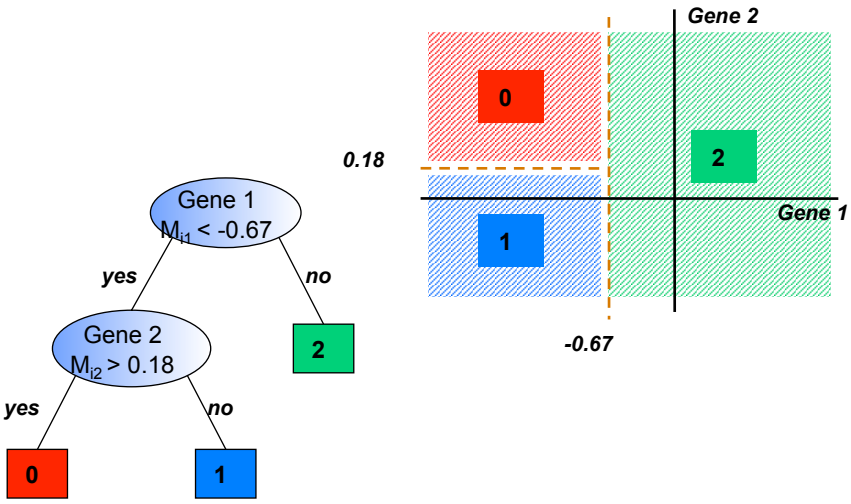
➥ **Choix de l'attribut Pif pour le premier test**

Exemple (suite)

Arbre final obtenu :



Sortie interprétable

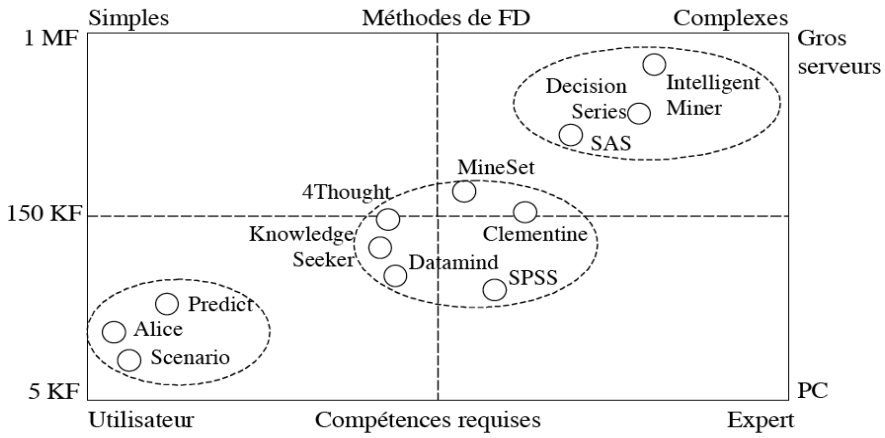


Analyse

- **Complexité** en $O(d\ n\ (\log n)^2)$ (faible)
- **Sortie interprétable**
- **Limites**
  - **Procédure gloutonne**
  - **Que faire si :**
    - **Attributs à valeur continue (dans $\mathcal{R}$)**
    - **Valeurs manquantes**
    - **Bruit dans les données**
    - **…**

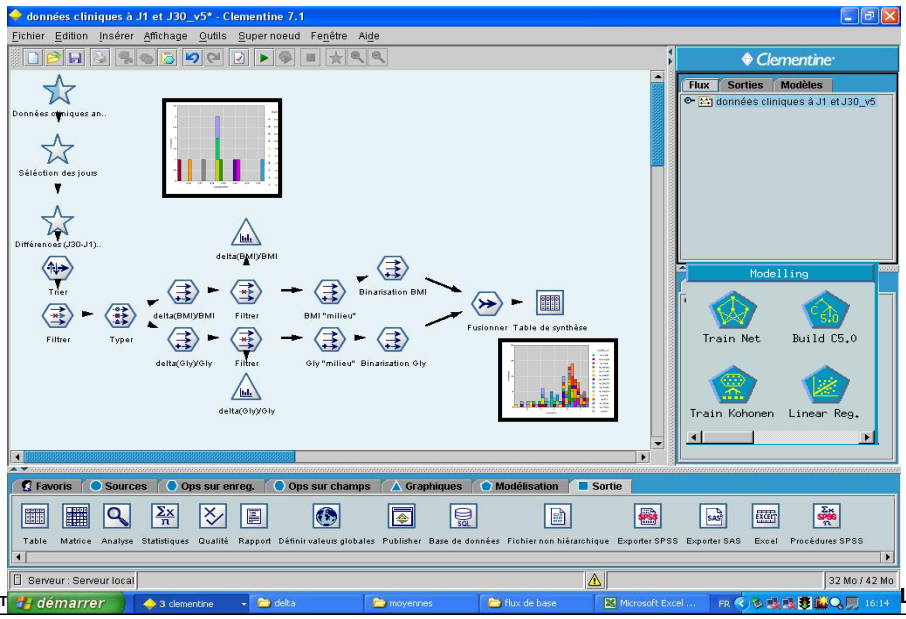
Outils commerciaux

Étude du Gartner Group (1996)



Les outils

Clementine© IBM SPSS



Conclusions

- **Grande variété de tâches et d'outils**
  - **Domaine en développement**
    - **Exploration de nouvelles applications**
    - **Travaux d'analyse et de justification des algorithmes**
    - **Outils logiciels commerciaux (e.g. Clementine)**
    - **Outil logiciels acédémiques (R, Weka, …)**
  - **Point de vue algorithmique**
    - Algorithmes « intuitifs » à très sophistiqués
      - Algorithmes **itératifs** : *k*-moyenne
      - Algorithmes **récursifs** : Arbres de décision
      - Algorithme par **exploration** et **élagage** : motifs fréquents
      - Algorithmes **d'optimisation** : SVM