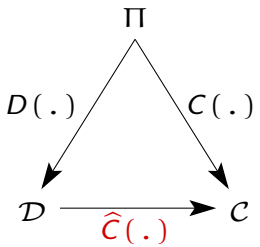
SPLEX
Statistiques pour la classification et fouille de données en génomique

Classification non paramétrique supervisée et non supervisée

Pierre-Henri WUILLEMIN

Decision-axe IA-LIP6
pierre-henri.wuillemin@lip6.fr

Classification non paramétrique



- Classification qui ne dépend pas d'un modèle.

  ... dont il faudrait trouver les paramètres ...

- Il s'agit de trouver une description de la classe $\widehat{C}(X)$ à partir de X uniquement.

  ... avec $X = (x_1, \cdots, x_d)$: méthodes *géométriques* ...

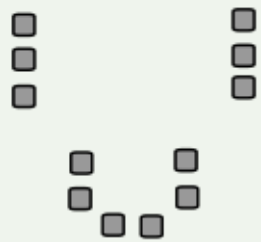
1. non supervisée
   1. K-means
   2. Classification hiérarchique
   3. (Analyse de données : ACP, ...)
2. supervisée
   1. k-PPV et fenêtre de Parzen
   2. Arbres de décision

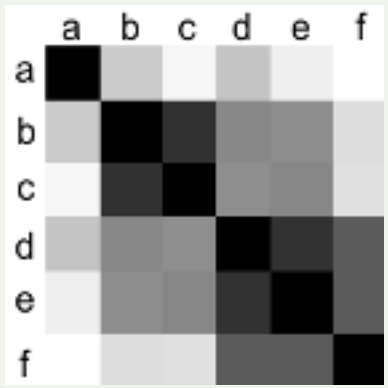
Classification non supervisée

Classification non supervisée – Formalisation

Généralement, le problème de classification non supervisée prendra la forme suivante :

- Soit n individus représentés par des vecteurs réels de dimension d : $X_{n \times d}$
- Soit un tableau de distance (similarités, dissimilarités) inter-individus $D_{n \times n}$.





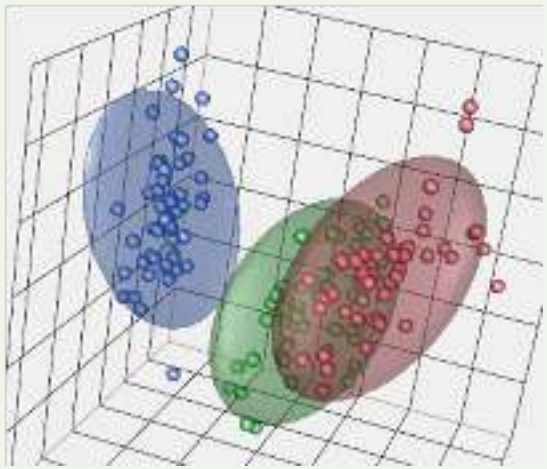
Objectif : Regrouper les individus en un certain nombre de classes homogènes.

- On passe de $X_{n \times d}$ à $D_{n \times n}$ grâce à une fonction de distance entre individus.
- On pourrait passer de $X_{n \times d}$ à $D_{d \times d}$ grâce à une distance entre variables (clustering de variables).

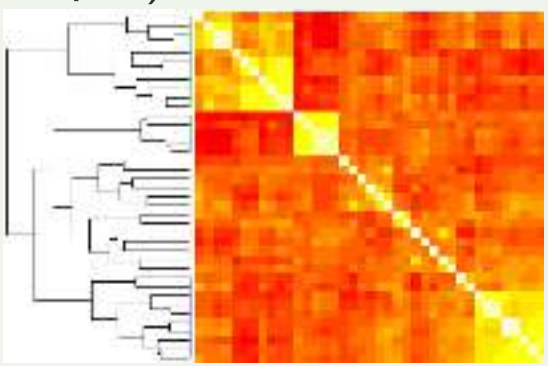
Classification non supervisée (2)

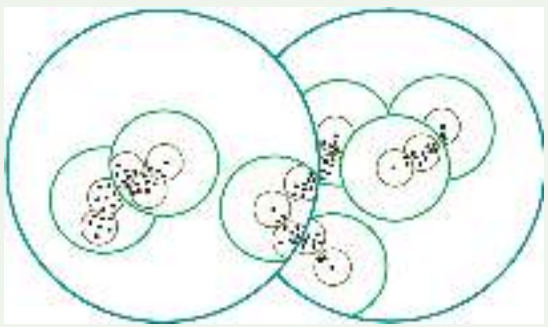
Approches de la classification non supervisée

- Approche non hiérarchique : Partitionnement en k classes (k donné)



- Approche hiérarchique : regroupement (ou division) successif d'individus (ou de groupes).





Classification non supervisée (3) : comparer

Comparaison

1. Entre individus : distance/dissimilarité ou proximité/similarité
2. Entre groupes :
   - Qualité d'un regroupement (homogénéité / différence inter-groupe) : méthode non hiérarchique
   - Critères de fusion ou de scission : méthode hiérarchique

 Le choix de ces distances/critères est crucial et hors apprentissage non supervisé ! Il y a donc supervision :-)

Distance/dissimilarité/similarité

	Distance $d(x,y)$	Dissimilarité $\overline{s}(x,y)$	Similarité $s(x,y)$
Positivité	$d(x,y) \geq 0$	$\overline{s}(x,y) \geq 0$	$s(x,y) \geq 0$
Symétrie	$d(x,y) = d(y,x)$	$\overline{s}(x,y) = \overline{s}(y,x)$	$s(x,y) = s(y,x)$
Identité	$d(x,y) = 0$ $\iff x = y$	$x = y \Longrightarrow$ $\overline{s}(x,y) = 0$	
Inégalité triangulaire	$d(x,z) \leq$ $d(x,y) + d(y,z)$		
Maximalité			$s(x,x) \geq s(x,y)$

- Si $d(.)$ distance alors $\frac{1}{1+d(.)} = s(.)$ similarité
- Si $s(.)$ similarité alors $\sqrt{s(x,x) + s(y,y) - 2 \cdot s(x,y)} = \overline{s}(.)$ dissimilarité
- etc.

Choix de la métrique

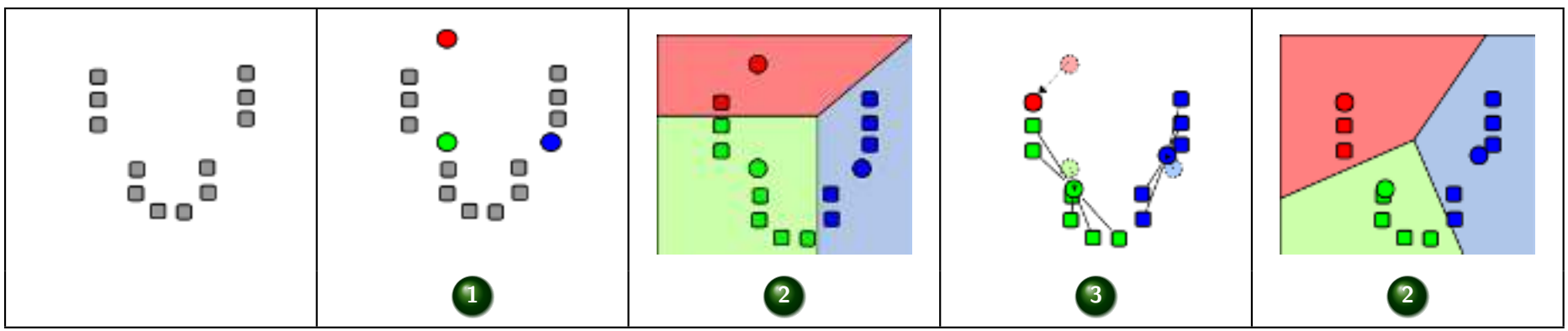
Différentes métriques :

euclidienne	L_2	$d(a,b) = \left(\sum_{i=1}^{d}(a_i - b_i)^2\right)^{\frac{1}{2}}$
euclidienne^2	L_2^2	$d(a,b) = \left(\sum_{i=1}^{d}(a_i - b_i)^2\right)$
Minkowski	L_p	$d(a,b) = \left(\sum_{i=1}^{d}(a_i - b_i)^p\right)^{\frac{1}{p}}$
	L_∞	$d(a,b) = \max_{i=1}^{d}\lvert a_i - b_i\rvert$
Manhattan	L_1	$d(a,b) = \sum_{i=1}^{d}\lvert a_i - b_i\rvert$
...	...	...

K-means : partitionnement non hiérarchique

K-means

1. Choisir K centres de groupe (aléatoirement parmi les individus par exemple),
2. Allouer chaque item au groupe dont le centre est le plus proche,
3. Calculer les centres de gravité des groupes, qui deviennent les nouveaux centres de groupe.
4. Répéter les 2 étapes précédentes jusque stabilisation.



Sensibilité à la valeur de K. Sensibilité aux centres initiaux.

Analyse de K-means : groupes bien concentrés, bien séparés

On note :
- $(x_i)_{i\in\{1,\cdots,n\}}$ les individus, g le centre de gravité global des individus ($g = \frac{1}{n}\sum_i x_i$) (indépendants de K)
- $(G_k)_{k\in\{1,\cdots,K\}}$ les groupes, de centre $(g_k)_{k\in\{1,\cdots,K\}}$.

Inerties (RSS : Residual Sum of Squares)

Inertie de groupe : $\forall k \in \{1,\cdots,K\}, I_k = \sum_{x_i \in G_k} d^2(x_i, g_k)$

fonction objectif – Inertie intra-groupe : $\min I_G = \min \sum_{k=1}^{K} I_k$

Pour un groupe G_k, $I_k = \sum_{x_i \in G_k} d^2(x, g_k)$. On peut chercher son minimum en annulant sa dérivée :

$$\frac{\partial I_k}{\partial g_k} = \sum_{x_i \in G_k} 2(x_i - g_k) = 0 \iff g_k = \frac{1}{\mid G_k \mid} \sum_{x_i} x_i$$

L'étape 3 correspond à la minimisation de I_G étant donné une structure des groupes.

Un second critère, l'**inertie inter-groupe** : $\max I_X = \max \sum_{k=1}^{K} \mid G_k \mid \cdot d^2(g_k, g)$.

K-means : contrôles de l'algorithme

Il y a un nombre fini de partitions en k classes de n éléments + décroissance de I_K. Donc la convergence vers un minimum local est assurée

Critères d'arrêt

- Si les centres de groupe ne bougent plus trop
- Si I_K et I_X quasiment fixes
- Nombre maximum d'étapes

Si K augmente, alors I_K diminue mathématiquement (minimum en $K = n$). Donc minimiser I_K n'est pas un bon critère de sélection de K.
Par contre, à K constant, I_K permet de sélectionner parmi différents regroupements.

valeur de K : IsoData (par exemple) – Fusion/éclatement de groupes

- fusionner 2 groupes k et k' si $d(g_k, g_{k'}) < \epsilon_F$
- éclater un groupe g si $I_k > \epsilon_E$

Comment choisir ϵ_F et ϵ_E !

Classification ascendante hiérarchique (CAS)

Soit un critère d'évaluation $\mathfrak{D}$ de la distance entre groupes, compatible avec la distance entre individus :
$$\mathfrak{D}(\{x\},\{y\}) = d(x,y)$$

Hiérarchie ascendante

1. Soit $\mathfrak{G}_0 = \{\{x_i\}, \forall i \leq n\}$ l'ensemble des individus en singleton
2. Pour i de 1 à $n-1$,
3. $$(G_1, G_2) = \underset{A,B \in \mathfrak{G}_{i-1}}{\arg\min}\, \mathfrak{D}(A,B)$$
4. $$\mathfrak{G}_i = \mathfrak{G}_{i-1} \setminus \{G_1, G_2\} \bigcup \{G_1 \cup G_2\}$$

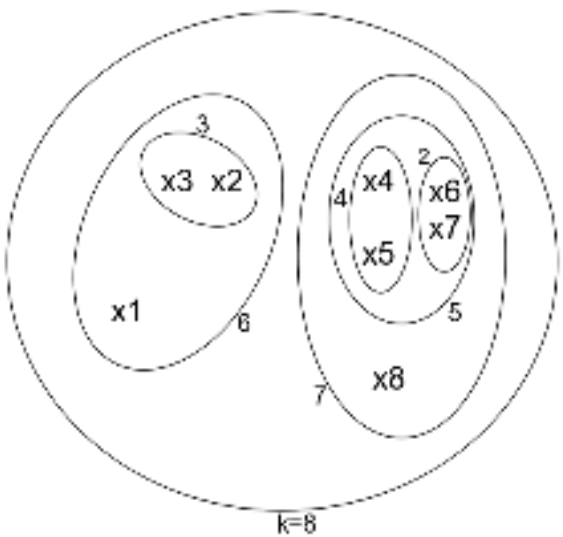
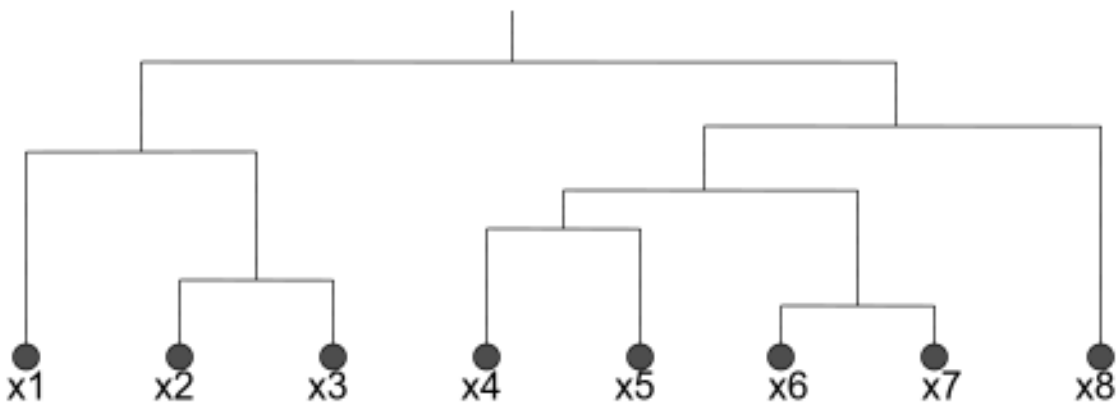


Diagramme de Venn



dendogramme

Distance entre groupes : $\mathfrak{D}$

Pour deux groupes G_1 et G_2, comment calculer $\mathfrak{D}(G_1, G_2)$ compatible avec d entre individus ?

$d(x, y)$ dissimilarité

- Saut minimum, *simple linkage* $\mathfrak{D}(G_1, G_2) = \min_{x \in G_1, y \in G_2} d(x, y)$
- Saut maximum, diamètre, *complete linkage* $\mathfrak{D}(G_1, G_2) = \max_{x \in G_1, y \in G_2} d(x, y)$
- Saut moyen, *group average linkage*

$$\mathfrak{D}(G_1, G_2) = \frac{1}{\mid G_1 \mid \cdot \mid G_2 \mid} \sum_{x \in G_1, y \in G_2} d(x, y)$$

$d(x, y)$ distance (euclidienne)

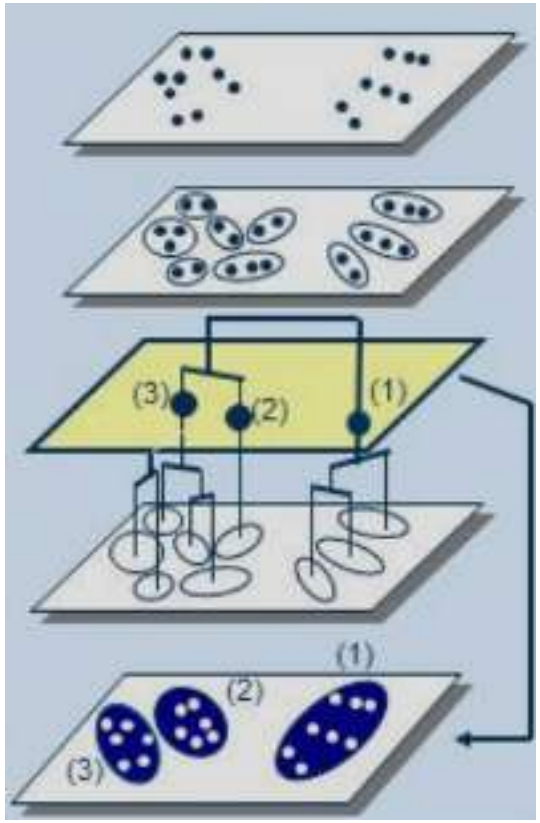
avec g_1 et g_2 les centre de G_1 et G_2,

- Distance des centres $\mathfrak{D}(G_1, G_2) = d(g_1, g_2)$
- Saut de Ward $\mathfrak{D}(G_1, G_2) = \frac{\mid G_1 \mid \cdot \mid G_2 \mid}{n \cdot (\mid G_1 \mid + \mid G_2 \mid)} d(g_1, g_2)$

La stratégie de Ward est la plus courante est revient à miminiser l'inertie intra-groupe et maximiser l'inertie inter-groupe I_X

Méthode mixte (Lebart)

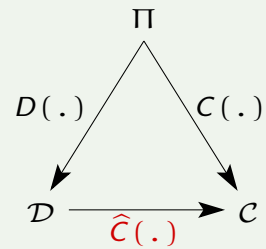
1. Données avant classification,
2. K-means avec K assez grand,
3. CAS sur les K groupes,
4. Coupure du CAS au niveau de similarité (ou nombre de groupes P) fixé,
5. Stabilisation par P-means.



Méthodes supervisées non paramétriques

Tâche de classification supervisée

Trouver $\widehat{C} : \mathcal{D} \to \mathcal{C}$ telle que
$\forall \pi, \widehat{C}\left(D(\pi)\right) \approx C(\pi)$ avec "le moins d'erreur possible"



- On connaît la classe d'un certain nombre de points (Π_a),
- On veut inférer la classe de tout point de $\mathcal{D}$.
-  Toujours pas de modèle, ni de paramètres ! On utilise que la carte des classes sur Π_a

k-PPV ou k-nearest neighbour

Approche par voisinage

- Trouver des points de la base qui soient *similaires* au point testé
- Faire *voter* les points trouvés.

Deux approches :

- nombre de voisins fixes : k-PPV
- voisinage de taille fixe : fenêtre de Parzen, kernel density estimation

Fonction de vote

$$g(x) = \frac{1}{|V(x)|} \sum_{x_i \in V(x)} y_i$$

où $V(x)$ est le voisinage de la description x.

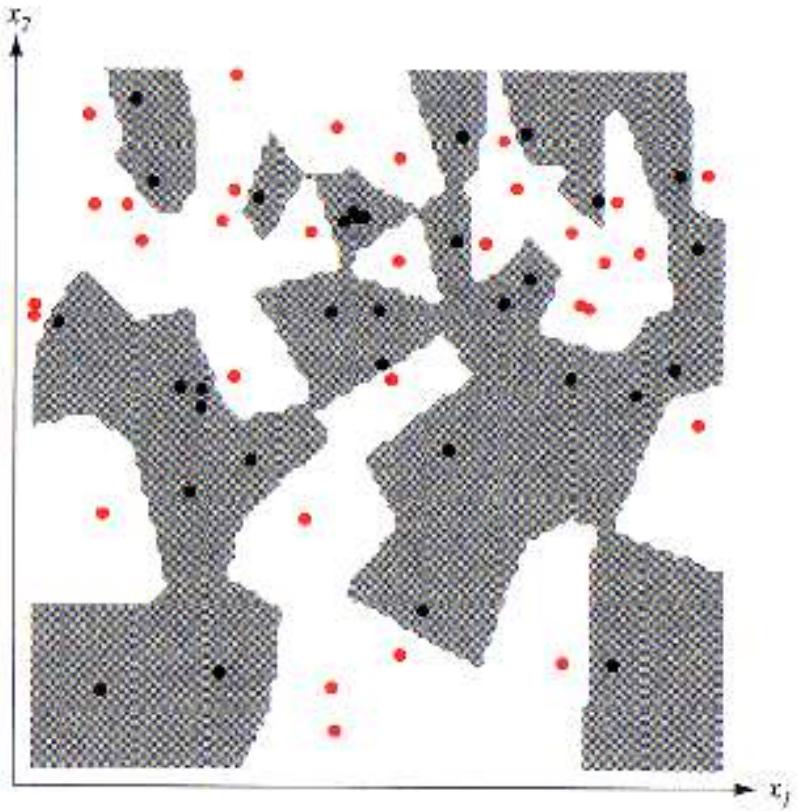
k-PPV : $|V(x)| = k$ (k est le paramètre de k-PPV)

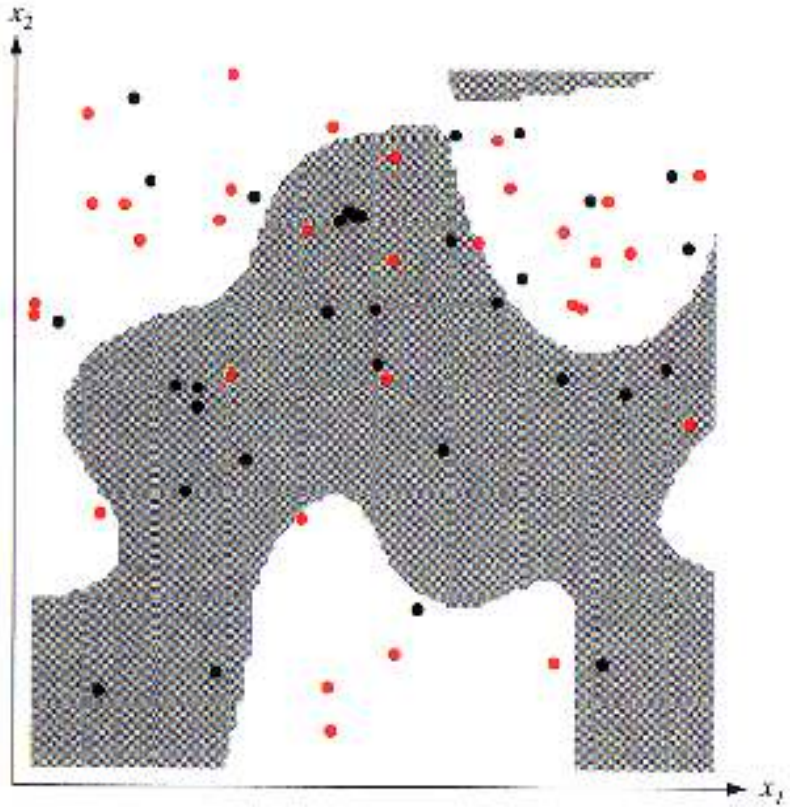
Parzen : $V(x) = \{X_i, d(x, x_i) < h\}$ (h est le paramètre de la fenêtre de Parzen)

Qualité de la classification

Qualité de l'approximation

- k ou h petit : les voisins sont proches (donc représentatifs) mais peu nombreux (donc $\widehat{C}$ est bruitée et peu robuste)
- k ou h grand : les voisins sont nombreux (donc lissage statistique) mais lointains (donc moins fiables)





1-PPV avec $d = 2$: diagramme de Voronoï

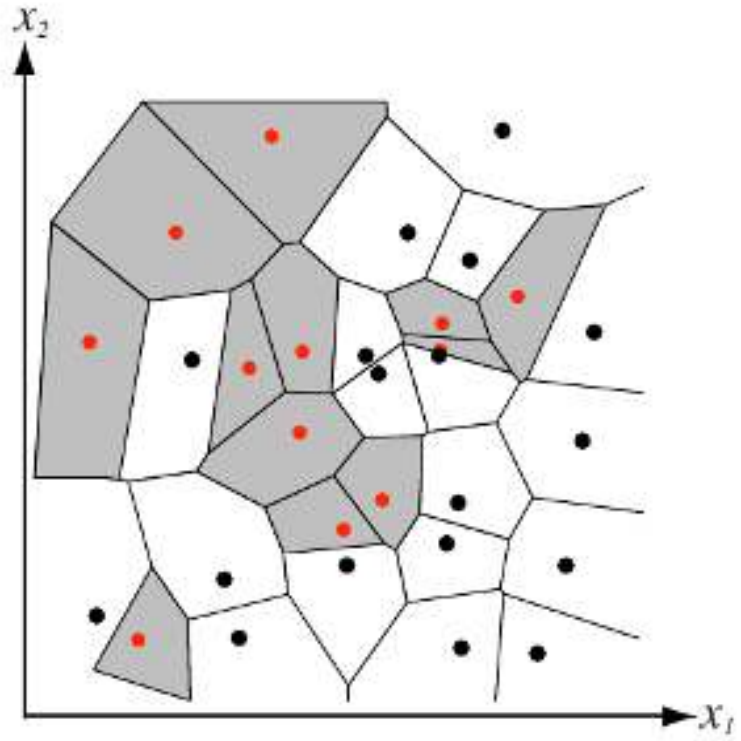


Diagramme de Voronoï [Georgi Fedoseevich Voronoï (1868 - 1908)]

Un diagramme de Voronoï (aussi appelé décomposition de Voronoï, partition de Voronoï ou encore polygones de Voronoï) représente une décomposition particulière d'un espace métrique déterminée par les distances à un ensemble discret d'objets de l'espace, en général un ensemble discret de points.

Efficacité de k-PPV et de Parzen

Avantages

- Facile à implémenter
- Pas de modèle à construire
- Utile pour données hétérogènes, classes irrégulières, etc.
- Version incrémentale aisée

Inconvénients

- Pas de modèle, donc pas d'explication de la classification,
- Classification lente et gourmande (garder les points en mémoire, etc.),
- Sensible à la valeur de d : The Curse of Dimensionality [R.Bellman]
- Sensible à la valeur de h ou k ...
- Pour k-PPV : choix de la métrique !

k-PPV : Fléau de la dimensionalité

Curse of dimensionality

- Les espaces de haute dimension sont vide !
- À densité constante, nombre de points nécessaires : $O(c^d)$.
- Les points les plus proches sont très loin !
- La notion de voisinage devient inutilisable.
- Complexité du calcul de $\widehat{C}$: $O(n^2 \cdot d)$

Kernel Density Estimation (kde)

On rappelle la forme de la fonction de vote pour la fenêtre de Parzen :

$$g(x) = \frac{1}{|V(x)|} \sum_{x_i \in V(x)} y_i \text{ avec } V(x) = \{X_i, d(x, x_i) < h\}$$

On peut réécrire cette fonction de vote en utilisant la fonction indicatrice de $V(x)$: $I_{d(x,x_i)<h} = I_{\frac{d(x,x_i)}{h}<1}$ (I_p vaut 1 si p est vraie et 0 sinon).

$$g(x) \propto \sum_{i=1}^{n} I_{\frac{d(x,x_i)}{h}<1} \cdot y_i$$

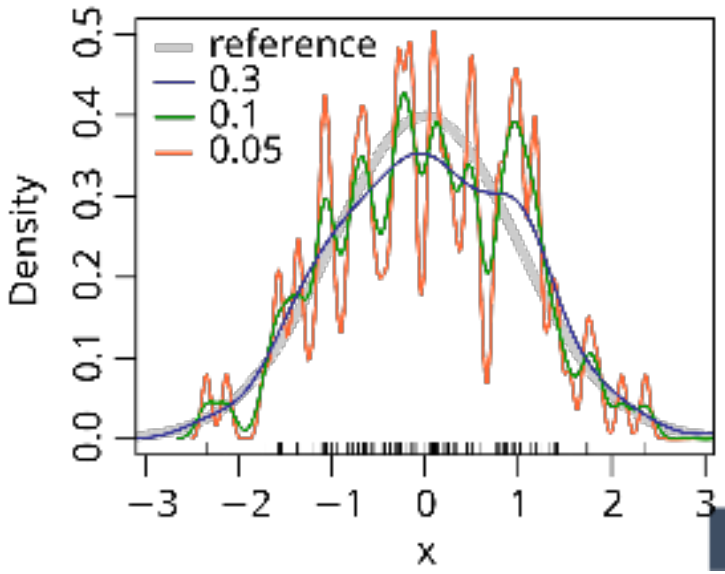
La généralisation consiste à changer la fonction indicatrice par une autre plus "lisse" : un noyau.

Méthode à noyaux (Parzen-Rozenblatt)

$$g(x) = \sum_{i=1}^{n} \Phi(\frac{d(x, x_i)}{h}) \cdot y_i$$

Exemple : noyau Gaussien $\mathcal{N}(0, 1)$:

$$\Phi(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}}$$

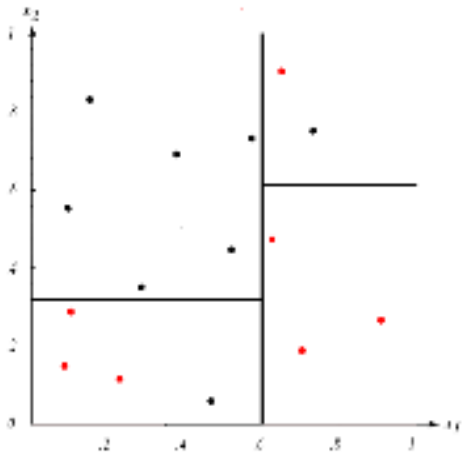


Arbre de décision

Arbre de décision

Un arbre de décision est une division récursive de $\mathcal{D}$ en région de plus en plus petite.

- Un nœud non terminal de l'arbre contient une variable (ou une fonction de plusieurs variables si découpage oblique).
- Un arc contient une valeur possible du nœud parent.
- Une feuille représente une région déterminée par les valeurs des variables de son chemin depuis la racine et y attribue la classe majoritaire dans cette région.



La classification par arbre de décision s'effectue par une série de tests des valeurs de x suivant les nœuds depuis la racine vers une feuille de l'arbre. La classe dans la feuille obtenue est $\widehat{C}(x)$.

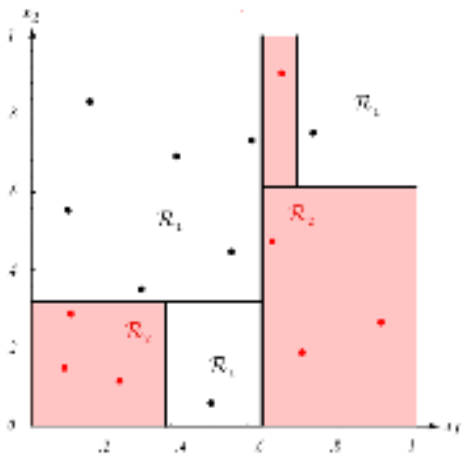
Classifications par arbre de décision

Avantages

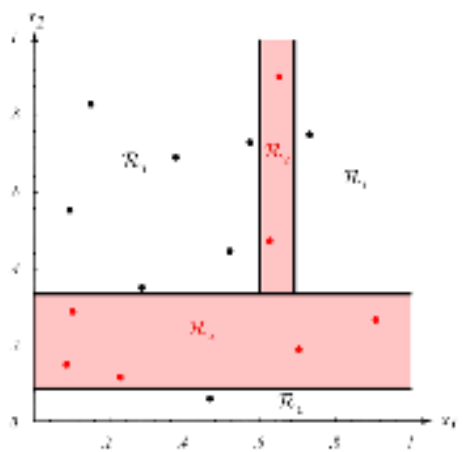
- Classification aisée et rapide
- Interprétation aisée (règles)
- Permet d'hybrider variables qualitatives/quantitatives

Inconvénients

- Principalement : manque de robustesse (sensibilité au bruit) $\Rightarrow$ instabilité !
- Mais aussi : difficulté d'optimisation : quel est le meilleur arbre ?
- Concision vs exactitude : quand arrêter l'arbre ? feuilles représentant des régions non pures ?

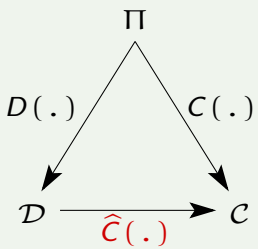


ou

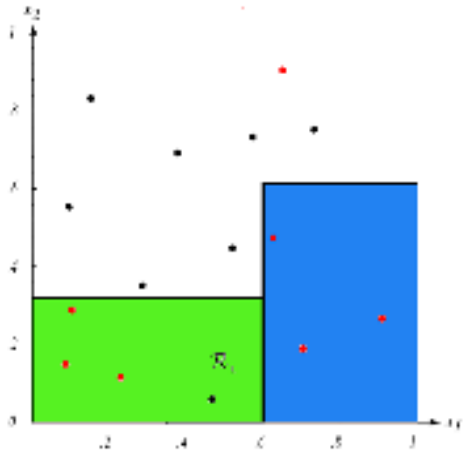


Pureté d'une région

Région, pureté d'une région



- Une région de $\mathcal{D}$ est un $\mathcal{R} \subset \mathcal{D}$.
- Une région $\mathcal{R}$ est dite "pure", de classe c si $C\left[D^{-1}(\mathcal{R})\right] = \{c\}$.
- La pureté d'une région se mesure donc par rapport à l'ensemble $C\left[D^{-1}(\mathcal{R})\right]$



Dans le cadre de l'apprentissage supervisé, on ne connaît que $C\left[D^{-1}(\mathcal{R} \cap \Pi_a)\right]$, donc une *approximation*.

Induction d'arbre de décision

Question : comment apprendre un tel arbre ?
Principe à suivre : Rasoir d'Occam ou principe de Parcimonie.

D'un point de vue général, induire un arbre consiste à proposer récursivement en chaque nœud de l'arbre une séparation de la région représentée par ce nœud.
Beaucoup d'algorithmes différents : ITI, ID3, C4.5 (1993), CART (1984), etc.

Principe général de l'algorithme

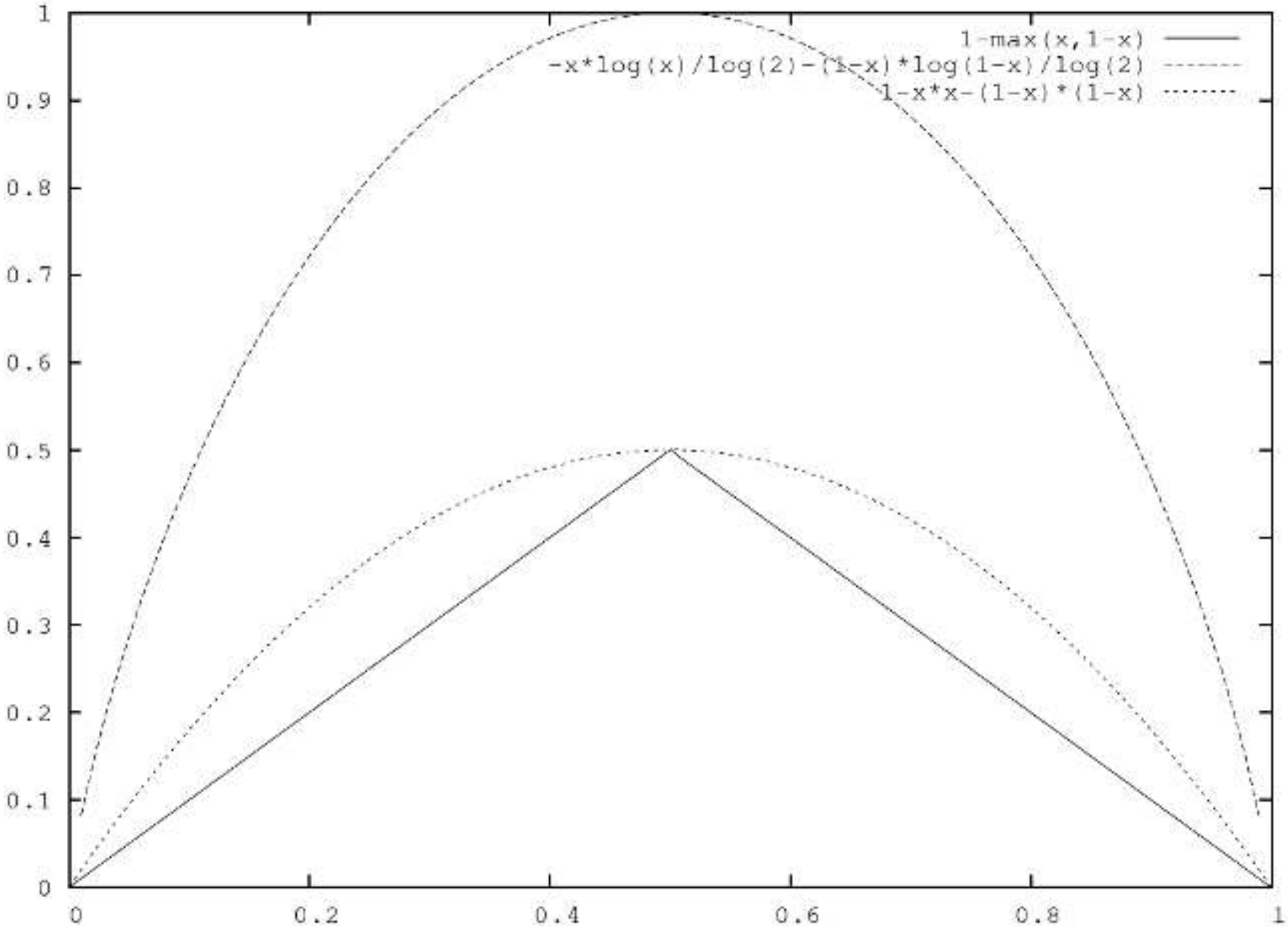
Le critère de séparation d'une région le plus intéressant est celui qui augmentera le plus la pureté des régions obtenues.

Soit un nœud N, $P(c \mid N) \propto \left|C\left[D^{-1}\left(\mathcal{R}_N \cap \Pi_a\right)\right]\right|$

Mesurer l'impureté du nœud N

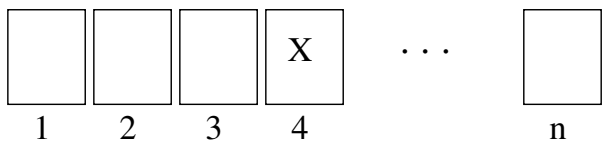
- Nombre de mal classés dans N : $Erreur(N) = 1 - \max_c P(c \mid N)$
- Évaluation de la dispersion : $Gini(N) = 1 - \sum_c P(c \mid N)^2$
- Évaluation de l'information : $Entropy(N) = -\sum_c P(c \mid N) \cdot \log_2 P(c \mid N)$

Comportements des 3 indices d'impureté



Digression culturelle : information et entropie statistique

Soit une expérience de probabilité simple : un gain se trouve dans une des boites numérotées de 1 à n. Il y a équiprobabilité d'occurrence des n positions pour le gain.



Définition (Le nombre d'information H – Hartley, 1928)

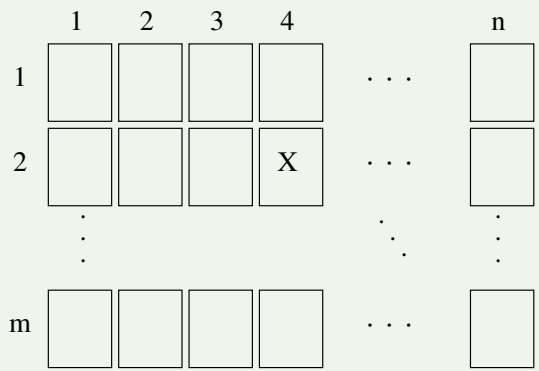
Le nombre d'information $H(n)$ est la quantité d'information reçue en apprenant où se trouve le gain. Elle est équivalente à la quantité d'incertitude *expérimentée au début de l'expérience (sans connaissance).*

$H(n)$ doit nécessairement avoir quelques propriétés.
Par exemple, $H(1) = 0$

Propriétés de H

Quelques propriétés de H

1. $H(1) = 0$
2. Arbitrairement, $H(2) = 1$
3. Monotonie : $H(n) \leq H(n+1)$ (n augmente $\Rightarrow$l'incertitude grandit.)
4. $H(n \cdot m)$?



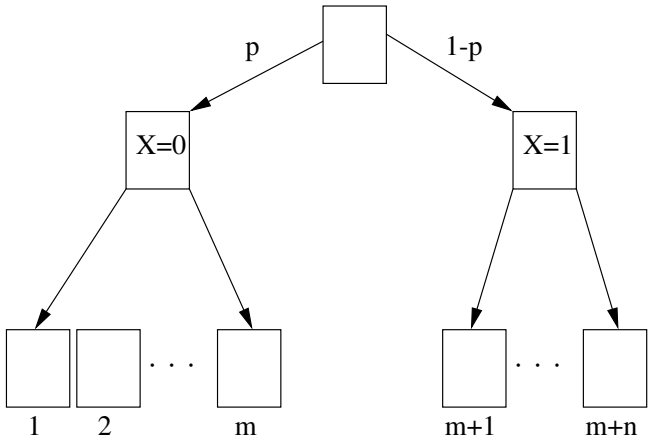
Additivité : $H(n \cdot m) = H(n) + H(m)$

Théorème

$H(n) = \log_2(n) = -\log_2(\frac{1}{n})$ vérifie ces conditions et est la seule si on considère n et m rationnels en 4.

Entropie de Shannon (1948)

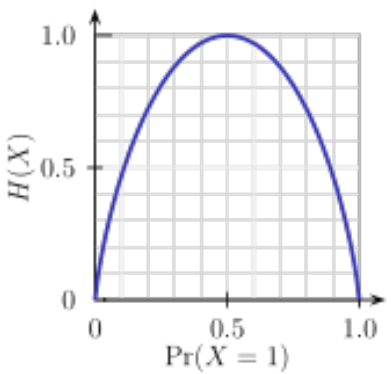
Plutôt que définir l'information apportée par le résultat d'une expérience, il s'agit de mesurer la *quantité moyenne d'information contenue dans une loi de probabilité*.
Soit une v.a. binaire X tel que $P(X = 0) = p$, p rationnel : $p = \frac{m}{m+n}$.



- Quantité d'information par la position du gain parmi $m + n$: $H(m + n) = log_2(m + n)$
- Quantité d'information par $X = 0$: $H(m + n) - H(m)$ (position du gain parmi m superflue)
- Quantité d'information par $X = 1$: $H(m + n) - H(n)$ (position du gain parmi n superflue)
- En moyenne : $p(H(m + n) - H(m)) + (1 - p)(H(m + n) - H(n)) = -p \log_2(p) - (1 - p) \log_2(1 - p)$

Définition (Entropie de Shannon)

$$h(p_1, \cdots, p_n) = -\sum_i p_i \cdot \log_2(p_i)$$



Induction d'arbre

Algorithme général

```
Procedure DeveloppperNoeud(D,N)
  Déterminer la variable V de coupure
          (en fonction de l'indice d'impureté)
  Étiqueter N avec cette variable V
  Créer les fils de N (valeurs de la variable V)
  Pour tout fils F
    Créer DF sous-région de D vérifiant F
    Si critère de terminaison sur F alors
      F est une feuille
      F a pour classe la classe majoritaire dans DF
    Sinon
      DeveloppperNoeud(DF,F)
    FinSi
  FinPour
Fin
```

Déterminer la variable de coupure

On sait calculer l'impureté d'une région dans un nœud : $I(N)$.

Définition (Gain)

Le gain représente la diminution d'impureté entre un nœud et ses fils :

- *soit V la variable coupé en N,*
- *soit N_v le fils de N où V vaut v,*

$$\Delta(N, V) = I(N) - \sum_{v \in V} \frac{|\mathcal{R}_v|}{|\mathcal{R}|} \cdot I(N_v)$$

Si $I(N)$ est l'entropie, on appelle $\Delta_{info} = H(V|N)$ le gain d'information.

Détermination de la variable de coupure

On sélectionne la variable de coupure en cherchant la variable maximisant le gain.

Exemple de calcul de gain

Gain Ratio

Les calculs de gain favorisent évidemment les variables avec beaucoup de modalité.
Il faudrait donc prendre en compte de l'information de branchement :

Définition (Gain Ratio)

$$\Delta_{Ratio}(N, V) = \frac{\Delta(N, V)}{H(V)} = \frac{\Delta(N, V)}{\sum_{v \in V} P(V = v) \log_2 P(V = v)}$$

Critères de terminaison

L'algorithme est arrivé à une feuille si

- La région du nœud est pure (ou seuil)
- La région devient trop petite (avec seuil)
- Pas de gain d'information (ou seuil)
- Toutes les variables ont été instanciées
- La hauteur de l'arbre atteint un seuil
- L'algo. arrive à une taille de description minimale

Définition (MDL : Minimum Description Length)

Dans un arbre de décision T,

$$MDL(T) = \alpha \cdot Taille(T) + \sum_{f\ feuille\ de T} I(f)$$

Élagage d'un arbre

Un autre algorithme possible :

1. Construire l'arbre le plus précis possible (critère d'arrêt minimum)
2. Élaguer : remplacer des branches de l'arbre par une feuille.

L'élagage est un processus itératif, *Bottom-Up* qui se termine quand aucune branche n'est plus élagable.

Quand élaguer ?

En fonction de l'erreur de classification : si la feuille (classe majoritaire) provoque moins d'erreur de classification que les branches.

Estimation de l'erreur de classification

- Validation croisée
- En utilisant Π_v
- Estimation statistique (χ^2, intervalle de confiance, etc.)

Algorithmes principaux

- ID3 (Induction Decision Tree, Quinlan 1979) :
  - uniquement sur des variables discrètes (*arbre de discrimination*),
  - critère de gain : entropie
- C4.5 (Quinlan 1993) :
  - extension d'ID3 à des variables continues (*arbre de régression*),
  - gestion des valeurs manquantes
- CART (Classification And Regression Tree, Breilan et al 1984) :
  - critère de gain : indice de Gini

Variables continues : Arbre multivarié