Master EISIS

Expertise et Ingénierie des Systèmes d'Information en Santé

Unité d'Enseignement :

IME-EDAD

Entrepôts de données et aide à la décision

Thème :

Méthodes de fouille de données

Auteur :

Dr Jean-Charles DUFOUR

jean-charles.dufour@univ-amu.fr

Document mis à jour le 04 mars 2014

Préambule

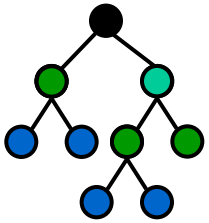
Ce diaporama s'inspire :

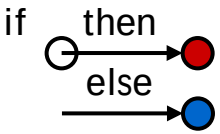
- du polycop « Découverte de connaissances à partir de données » (R.Gilleron et M. Tommasi, Lille).
- du tutorial T17 animé par J.H. Holmes lors de l'AMIA 2007 (Chicago)

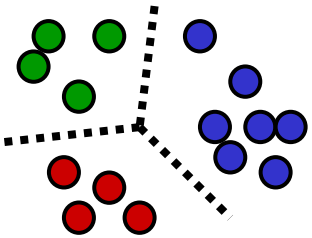
Les autres références utilisées sont mentionnées individuellement dans les diapositives.

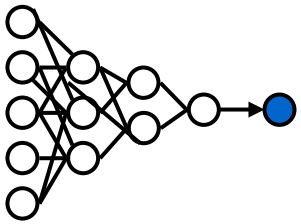
Plan du cours

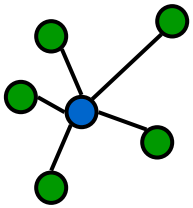
- Introduction
- Aperçue des principales méthodes pour la fouille de données
  - Arbres de décision
  - Règles d'association
  - Algorithme de segmentation
  - Méthode des plus proches voisins
  - Réseaux de neurones



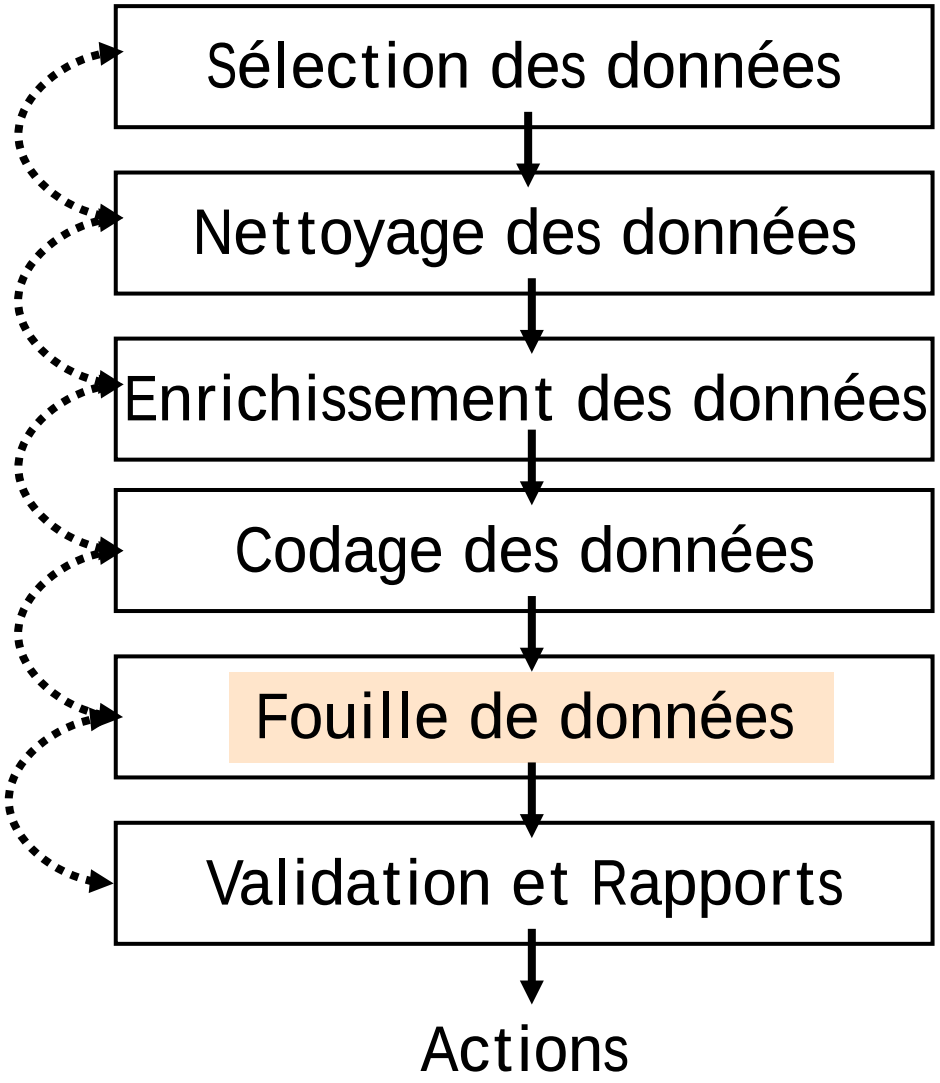








Fouille de données : introduction



Fouille de données : introduction

- Résultats recherchés peuvent s'obtenir sans recours à des techniques de fouille de données
  - Requêtes
  - Outil de reporting, analyses multi-dimensionnelles
  - Visualisations
  - ...
- Approche classique
  1. Regarder, explorer
  2. Établir un modèle ou une hypothèse (la fouille peut être utile sur ce point)
  3. Essayer de contredire ou de le vérifier

Fouille de données : introduction

- Objectif : utiliser les données disponibles dans les bases pour identifier des combinaisons, des structures/arrangements (*patterns*) méconnues, significatifs et utiles
- Orienté vers la génération d'hypothèse mais pas vers la validation d'hypothèse

Fouille de données : introduction

- Le modèle de fouille parfais :
  - rapide à créer
  - rapide à utiliser
  - compréhensible pour l'utilisateur
  - les performances sont bonnes et le modèle est fiable
  - les performances ne se dégradent pas dans le temps
  - Il évolue facilement
- Malheureusement il n’existe pas !
- Il faut faire des compromis selon les besoins et combiner les méthodes

Fouille de données : introduction

- Pré-requis :
  - Disposer des données pertinentes par rapport à la problèmatique explorée
  - Avoir une idée du type de connaissance que l'on souhaite découvrir : règles, similitudes, classes, groupes,…
  - Avoir déjà de solides connaissances sur le domaine exploré (interprétation des résultats)

Fouille de données : introduction

- Schématiquement il existe 3 catégories de méthodes de fouille :

1. Classification, prédiction
   → trouver une classe ou une valeur selon un ensemble de descriptions
2. Association, sequencing
   → trouver des similarités ou des associations
3. Segmentation
   → trouver des groupes homogènes dans une population

Fouilles de données : introduction

- Méthodologies classique pour élaborer et valider un modèle de fouille
  1. Un jeu de donnée pour l'**apprentissage**
  2. Un jeu de donnée de **test**
  3. Un jeu de donnée de **validation**

Fouilles de données : les méthodes

- Algorithme de segmentation
- Règles d'association
- Méthode des plus proches voisins
- Arbres de décision
- Réseaux de neurones

Algorithme de segmentation : principe de la méthode des k-moyennes

- Objectif : diviser la population en groupe
- Principe :
  - Basé sur la notion de similarité entre enregistrements
  - Choix d'un nombre *k* de groupes à constituer
  - Itérations jusqu'à stabilité des groupes

Algorithme des k-moyennes
paramètre : le nombre k de groupes
entrée : un échantillon de m enregistrements $\vec{x_1}, \ldots \vec{x_m}$

1. choisir k centres initiaux $\vec{c_1}, \ldots \vec{c_k}$
2. pour chacun des m enregistrements, l'affecter au groupe i dont le centre $\vec{c_i}$ est le plus proche
3. si aucun élément ne change de groupe alors arrêt et sortir les groupes
4. calculer les nouveaux centres : pour tout i, $\vec{c_i}$ est la moyenne des éléments du groupe i
5. aller en 2

Algorithme de segmentation : principe de la méthode des k-moyennes

- Limites de la méthode
  - Suivant la nature des données à segmenter, la définition de la notion de proximité n'est pas toujours évidente
  - Choix du nombre *k* de groupes
    - recours à un expert
    - itération avec différentes valeurs de *k*

Règles d'association

- Objectif : recherche d'associations entre des faits
- Exemples :
  - analyse du « panier de la ménagère » (quels produits tendent à être achetés ensemble)
  - recherche de complications dues à des associations de médicaments
  - ...
- Principe : établir un tableau de co-occurrence entre les faits et en déduire des règles de la forme « si condition alors résultat » :
  - Si A alors B
  - Si A et B alors C
  - ...

Règles d'association

- Exemple (co-occurrences sur 5 listes d'achats)

	produit A	produit B	produit C	produit D	produit E
produit A	4	1	1	2	1
produit B	1	2	1	1	0
produit C	1	1	1	0	0
produit D	2	1	0	3	1
produit E	1	0	0	1	2

➔ Règles envisagées :

1. Si A alors B (support = 1/5 = 20% ; confiance = 1/4 = 25%)
2. Si A alors D (support= 2/5 = 40% ; confiance = 2/4 = 50%)
3. Si D alors A (support = 2/5 = 40% ; confiance = 2/3 = 67%)

Règles d'association

Les règles sont de la forme
si condition alors résultat

$$\text{Support} = \text{freq}(\text{condition et résultat}) = \frac{d}{m}$$

$$\text{Confiance} = \frac{\text{freq}(\text{condition et résultat})}{\text{freq}(\text{condition})} = \frac{d}{c}$$

$$\text{Amélioration} = \frac{\text{confiance}}{\text{freq}(\text{résultat})}$$

d = nombre d'achats où les articles des parties condition et résultat apparaissent
m = nombre total d'achats
c = nombre d'achat où les articles de la partie condition apparaissent

Règles d'association

- Méthode générique pour la recherche de règles :
  - Une liste de *n* articles étant définie, on considère une liste de *m* achats. On procède comme suit :
    1. on calcule le nombre d'occurrences de chaque article,
    2. on calcule le tableau des co-occurrences pour les paires d'articles,
    3. on détermine les règles de niveau 2 en utilisant les valeurs de support, confiance et amélioration,
    4. on calcule le tableau des co-occurrences pour les triplets d'articles,
    5. on détermine les règles de niveau 3 en utilisant les valeurs de support, confiance et amélioration,
    6. ...

Les plus proches voisins (PPV)

- Objectif : classer (+/- estimer)
- Principe :
  - raisonnement à partir de cas
  - recherchant un ou des cas similaires déjà résolus et mémorisés

Algorithme de classification par k-PPV
paramètre : le nombre k de voisins
donnée : un échantillon de m enregistrements classés $(\vec{x}, c(\vec{x}))$
entrée : un enregistrement $\vec{y}$
1. déterminer les k plus proches enregistrements de $\vec{y}$
2. combiner les classes de ces k exemples en une classe c

sortie : la classe de $\vec{y}$ est $c(\vec{y}) = c$

Les plus proches voisins (PPV)

- Propriété d'une distance :
  - $d(A, A) = 0$
  - $d(A, B) = d(B, A)$
  - $d(A, B) \leq d(A, C) + d(B, C)$
- Un point = un enregistrement de la base de données
- Pour définir la fonction de distance :
  1. on définit d'abord une distance sur chacun des champs
  2. puis on combine ces distances pour définir la distance globale entre enregistrements

Les plus proches voisins (PPV)

Le choix de formule de calcul de la distance entre deux champs dépend du type de champ :

- Champs numériques :
  - $d(x, y) = |x - y|$, ou
  - $d(x, y) = \frac{|x - y|}{d_{max}}$, où d_{max} est la distance maximale

Les plus proches voisins (PPV)

- Champs discrets :
  1. Données binaires (0 ou 1)
     → On choisit d(0,0)=d(1,1)=0 et d(0,1)=d(1,0)=1
  2. Données énumératives
     → La distance vaut 0 si les valeurs sont égales et 1 sinon
  3. Données énumératives ordonnées
     → Soit on les considère comme des valeur énumérative classique (cf. point 2)
     → Soit on utilise la relation d'ordre pour définir une distance
     [ex : si un champ prend les valeurs 1, 2, 3, 4 et 5, on peut définir la distance en considérant 5 points de l'intervalle [0,1] avec une distance de 0,2 entre deux points successifs, on a alors d(1,2)=0,2 ; d(1,3)=0,4 ; … ; d(4,5)=0,2]

Les plus proches voisins (PPV)

- Calcul de la distance globale entre deux enregistrements :
  - Distance euclidienne :

$$d_e(\vec{x},\vec{y})=\sqrt{d_1(x_1, y_1)^2+...+d_n(x_n,y_n)^2}$$

    - ➔ favorise les voisins dont tous les champs sont assez voisins
  - Distance par sommation :

$$d_s(\vec{x},\vec{y})= d_1(x_1, y_1)^2+...+d_n(x_n,y_n)^2$$

    - ➔ permet de tolérer une distance importante sur l'un des champs

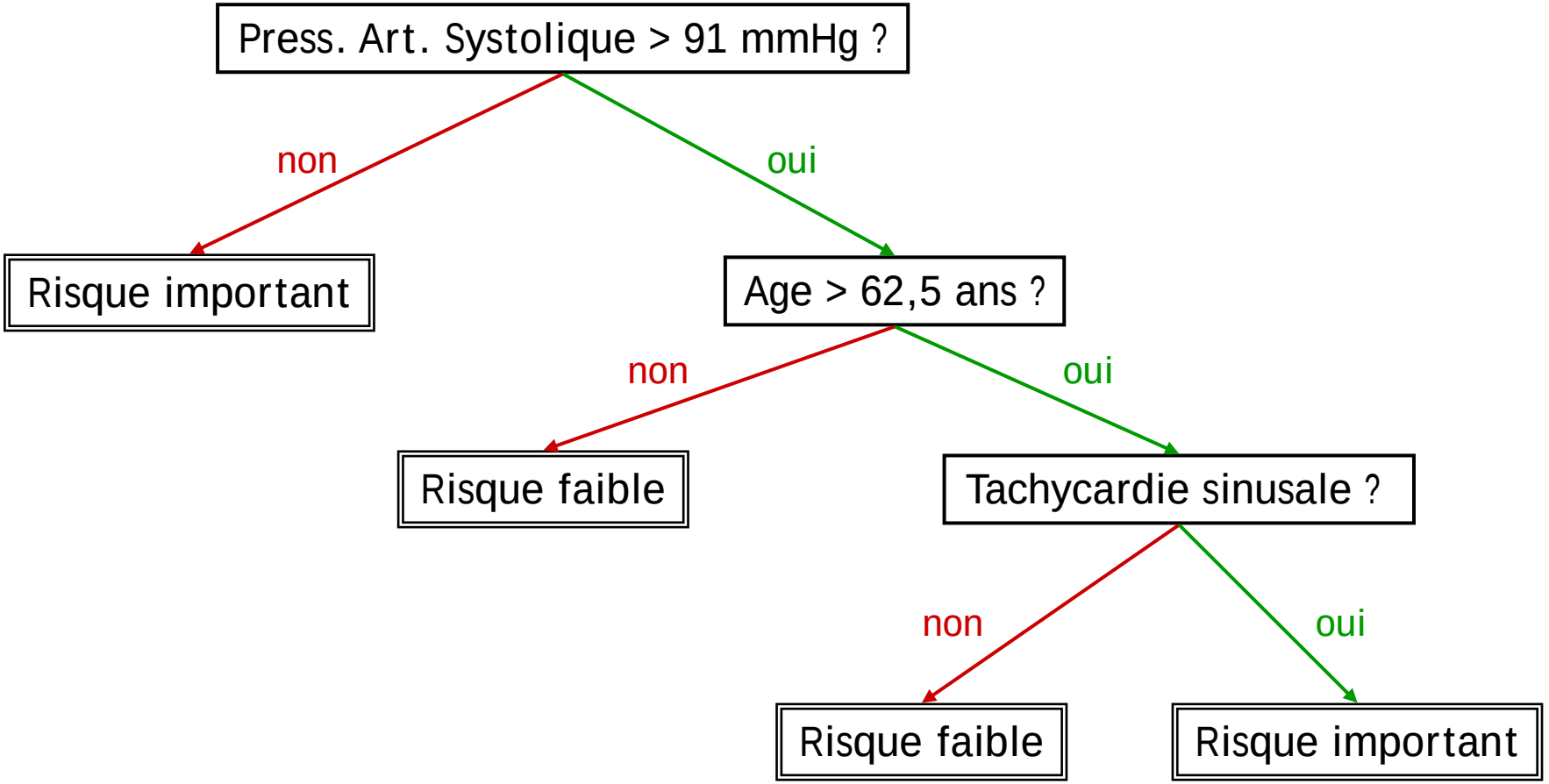
$x=(x_1,...,x_n)$ et $y=(y_1,...,y_n)$ sont deux enregistrement et $d_1,...,d_n$ sont les distances définies sur les différents champs

Les plus proches voisins (PPV)

- Sélection de la classe : on souhaite attribuer à $\vec{y}$ une classe c($\vec{y}$)
  - Méthode 1-PPV
    - $\vec{y}$ prend la même classe que le voisin le plus proche
  - Méthode du vote majoritaire
    - $\vec{y}$ prend la même classe que la majorité des voisins
  - Méthode du vote pondéré
    - On pondère chaque classe des voisins par un facteur inversement proportionnel à la distance

Les arbres de décision

- Arbre de décision = représentation graphique d'une procédure de classification



Arbre de décision permettant de classer les patients présentant un infarctus myocardique (librement adapté de Breiman et al., 1993)

Les arbres de décision

- Algorithme de génération d'arbres à partir de données :
  - CART (Classification And Regression Trees)
  - C5
- Résultats de ces algorithmes = procédures de classification exprimables sous forme de règles

Les arbres de décision

- Algorithme générique de génération d'un arbre :

Algorithme d'apprentissage par arbres de décision
donnée : un échantillon S de m enregistrements classés $(\vec{x}, c(\vec{x}))$
initialisation : arbre vide ; nœud courant : racine ; échantillon courant : S
répéter
  décider si le nœud courant est terminal
  si le nœud courant est terminal **alors**
    étiqueter le nœud courant par une feuille
  sinon
    sélectionner un test et créer le sous-arbre
  finsi
  nœud courant : un nœud non encore étudié
  échantillon courant : échantillon atteignant le nœud courant
jusque production d'un arbre de décision
élaguer l'arbre de décision obtenu
sortie : arbre de décision élagué

Les arbres de décision

Comment décider si un nœud courant est terminal ?

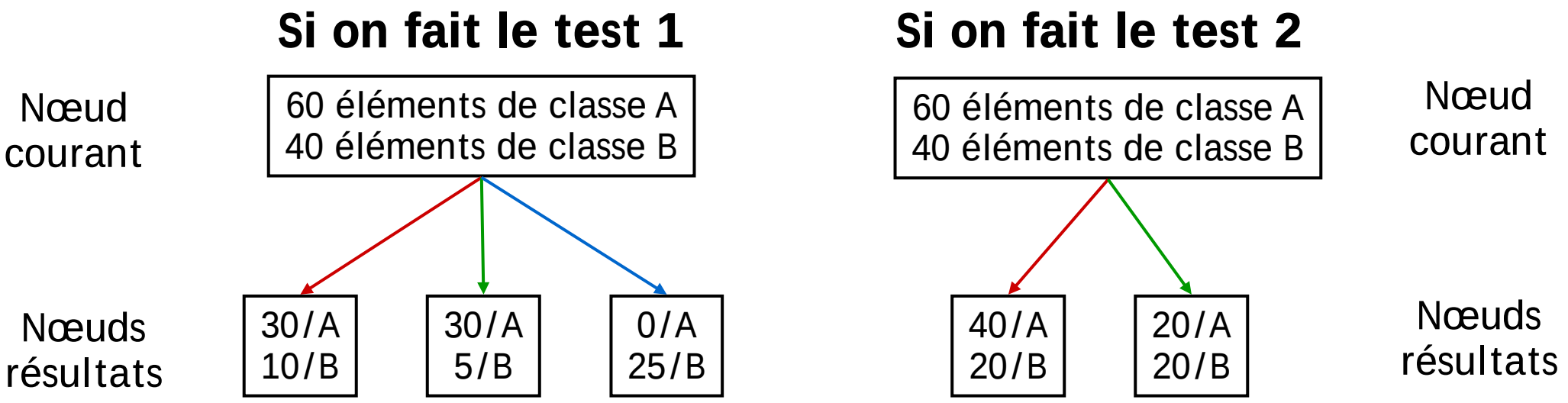
- Critères indiscutables :
  1. Il n'y a plus d'attributs disponibles (i.e. sur le chemin menant de la racine au nœud tous les tests disponibles ont été utilisés)
  2. Tous les exemples de l'échantillon courant sont dans un même classe
- Critères spécifique aux différents algorithmes, par exemples :
  - La proportion enregistrements placés dans une classe est supérieure à un seuil prédéfini (ex: arrêt si une classes contient plus de 95% des enregistrements)
  - S'il n'existe pas de test ayant au moins *k* éléments sur deux branches alors le nœud est terminal (algorithme C5)
  - ...

Les arbres de décision

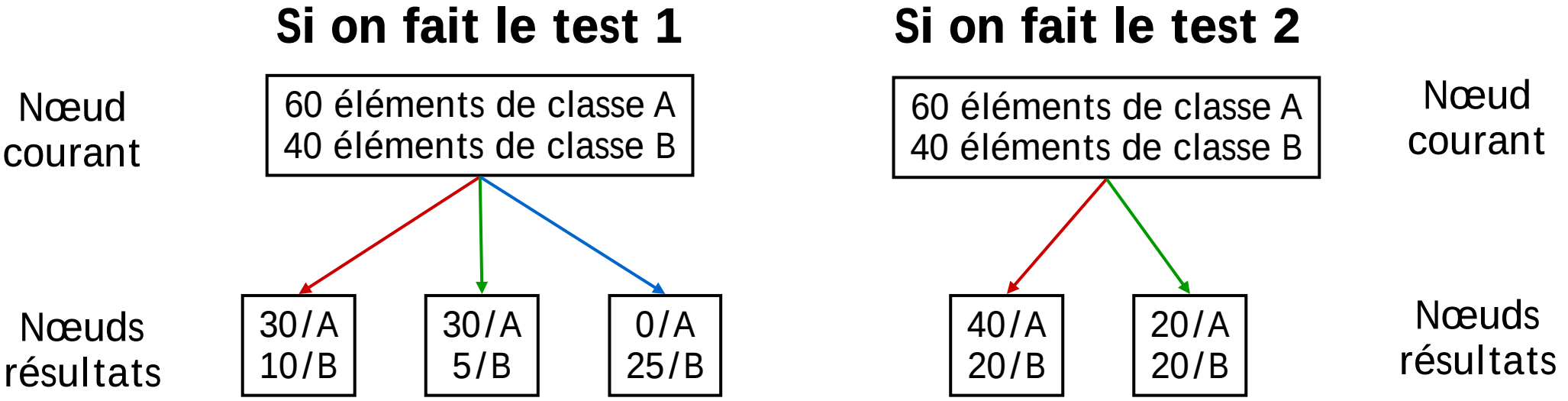
- Comment étiqueter le nœud courant par une feuille ?
  - Généralement, on étiquette le nœud courant par la classe majoritaire
    Par exemple : sur un nœud terminal, il y a 5 éléments de classe A et 20 éléments de classe B → on étiquette par B
  - Il est également possible de définir des « coûts » de mauvaise classification est de choisir la classe en tenant compte des « coûts » engendrés

Les arbres de décision

- Comment sélectionner un test ?
  - Le principe est de sélectionner le test qui, une fois effectué, va minimiser le mélange des classes d'éléments dans les branches sous-jacentes
- Exemple : doit-on choisir le test 1 ou le test 2 ?



Les arbres de décision



- Pour mesurer le « degré de mélange » des 2 classes dans les nœuds résultats on peut utiliser soit :
  - La fonction de Gini : $Gini(x) = 4x\,(1-x)$
  - La fonction entropie : $Entropie(x) = -x \log x - (1-x) \log (1-x)$

(x = proportion d'éléments dans l'une ou l'autre des deux classes)

Les arbres de décision

$$Gini(x) = 4x\,(1-x)$$

$$Entropie(x) = -x \log x - (1-x) \log (1-x)$$

- Ces deux fonctions :
  - prennent leurs valeurs dans l'intervalle réel [0,1]
  - prennent leur minimum pour $x=0$ ou $x=1$
    (c.a.d. quand tous les éléments sont dans une même classe)
  - prennent leur maximum lorsque $x=1/2$
    (c.a.d. quand les éléments sont également répartis entre les deux classes).

Les arbres de décision

test 1

60 éléments de classe A
40 éléments de classe B

30/A 10/B | 30/A 5/B | 0/A 25/B

x=30/40 Gini(x) = 0,75 | x=30/35 Gini(x) = 0,49 | x=0/25 Gini(x) = 0

$Gini(x) = 4x\,(1-x)$

x=60/100
Gini(x) = 4 x (60/100) x (40/100)
Gini(x) = 0,96

test 2

60 éléments de classe A
40 éléments de classe B

40/A 20/B | 20/A 20/B

x=40/60 Gini(x) = 0,89 | x=20/40 Gini(x) = 1

Pour comparer le deux tests, on calcule le « degré de mélange espéré »
(= somme des Ginis pondérés par la proportion des éléments allant sur chacun des fils)

Degré de mélange espéré (DME) =
40/100 x 0,75 + 35/100 x 0,49 + 25/100 x 0 =
0,47
Gain = Gini du nœud courant – DME =
0,96 – 0,47 = **0,49**

Degré de mélange espéré (DME) =
60/100 x 0,89 + 40/100 x 1 =
0,93
Gain = Gini du nœud courant - DME =
0,96 – 0,93 = **0,03**

On choisit le test qui a le DME le plus faible (i.e. le Gain le plus important)

Les arbres de décision

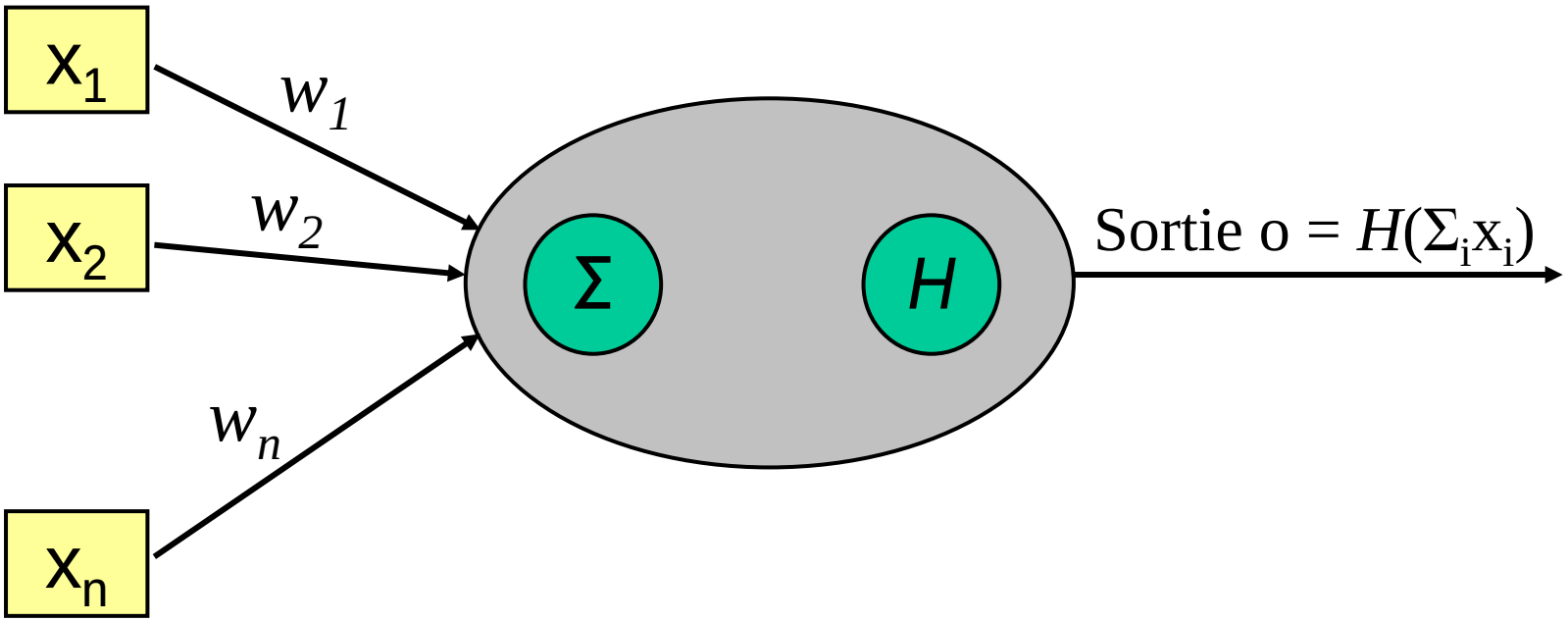
- Élaguer l’arbre obtenu pour améliorer le pouvoir de généralisation :
  - Principe utilisé pour élaguer = comparaison entre erreur de classification de l’arbre élagué versus erreur de classification de l’arbre non élagué

Les réseaux de neurones

- Les réseaux de neurones peuvent être utilisés pour la classification, l'estimation, la prédiction et la segmentation
- Unité élémentaire d'un réseau = le neurone formel *(basé sur le principe de fonctionnement du neurone biologique)*

Les réseaux de neurones

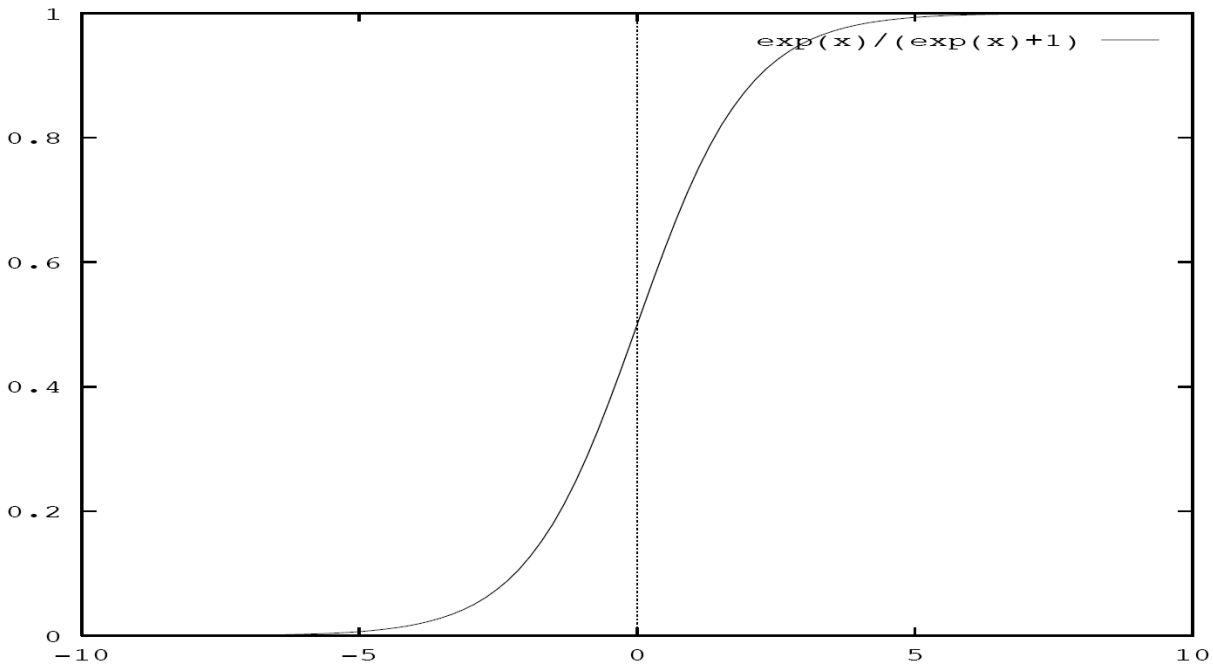
- Neurone formel = une unité de calcul élémentaire



Les réseaux de neurones

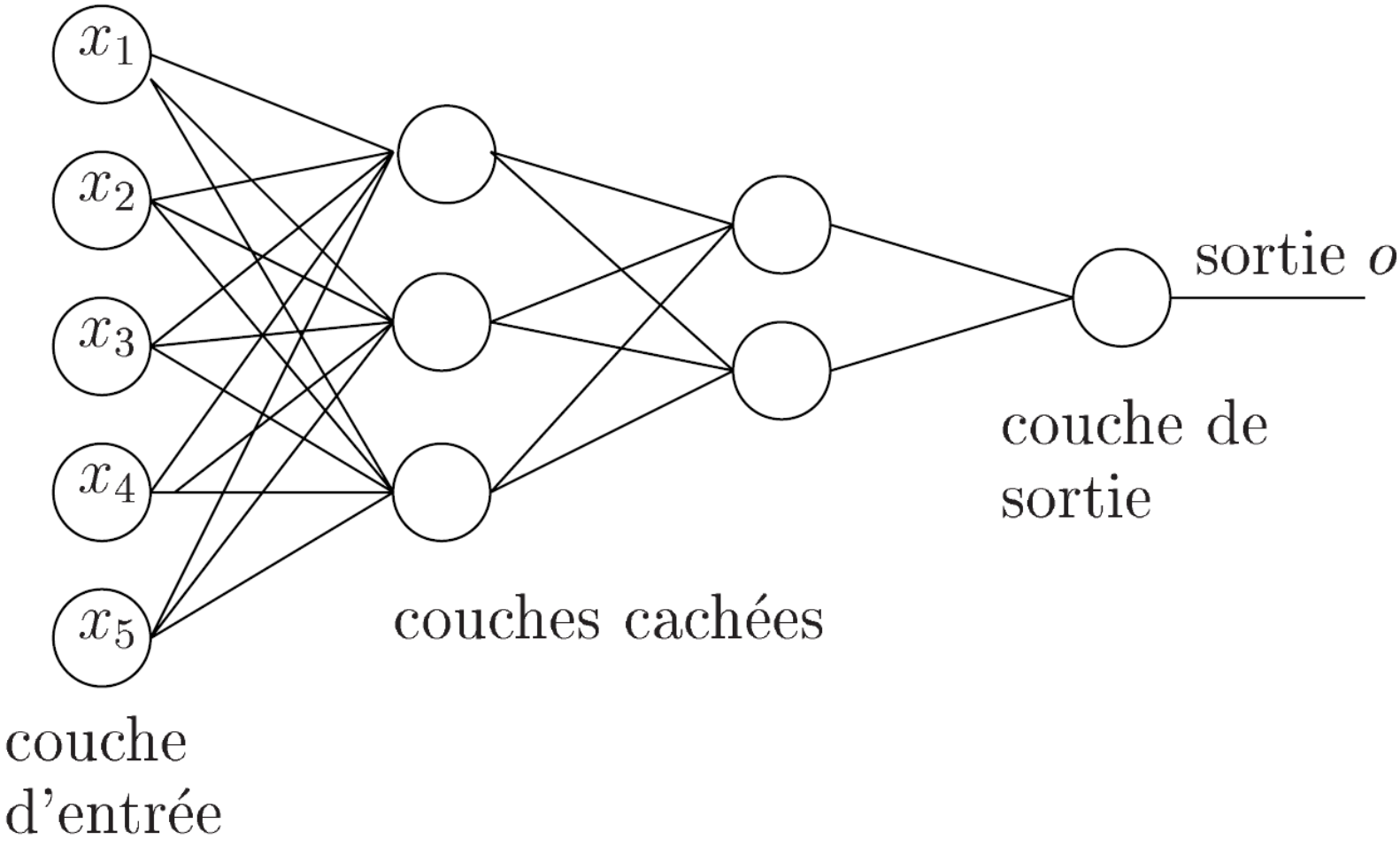
- Approximation de la fonction de Heaviside :

$$\sigma = \frac{e^x}{e^x + 1} = \frac{1}{1 + e^{-x}}$$



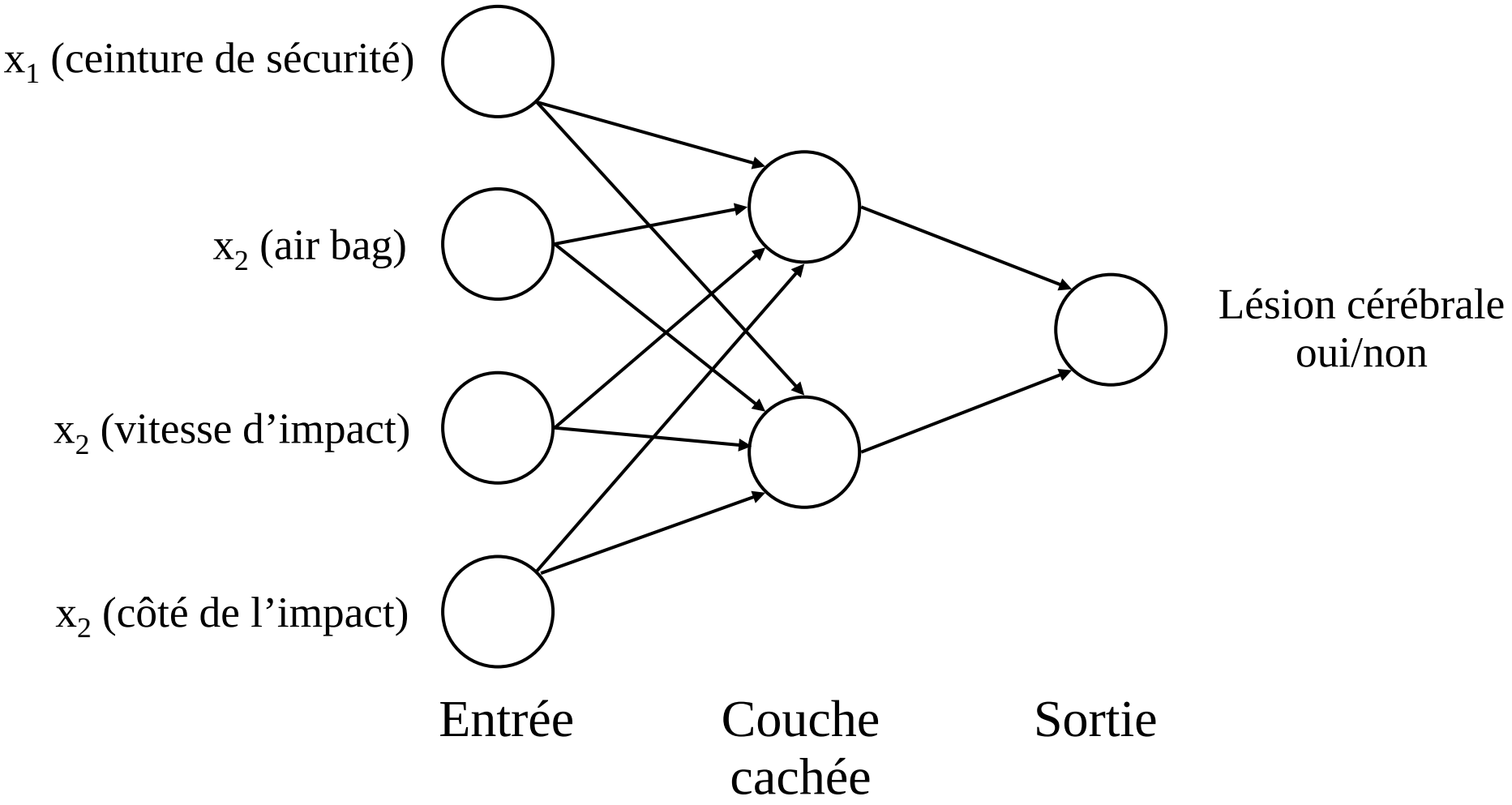
Les réseaux de neurones

- Perceptron Multi-Couches :



Les réseaux de neurones

Exemple :



Les réseaux de neurones

- Principe de l'algorithme d'apprentissage des réseaux de neurones :
  1. Initialisation aléatoire des poids de tous les neurones
  2. Présentation d'un élément en entrée du réseau
  3. Mesure de l'écart entre sortie produite par le réseau et sortie attendue
  4. Modification des poids
  5. Itération (vers étape 2) ou arrêt si écart est minime