Fouille de données

Cours 4 - Exploration des données multidimensionnelles - Apprentissage supervisé

NGUYỄN Thị Minh Huyền
Email: huyenntm@vnu.edu.vn
VNU University of Science, Hanoi

Plan

Plan

Plan

Plan

Plan

Plan

Plan

Plan

Plan

Références

- J. Han and M. Kamber. `https://hanj.cs.illinois.edu/bk3/`
- Charu Aggarwal, Data Mining, Springer 2015.
- Tan, Steinbach, Karpatne and Kumar, Introduction to Data Mining, 2nd ed. `https://www-users.cs.umn.edu/~kumar001/dmbook/index.php`

Plan

Introduction

- Analyse non supervisée
- Analyse supervisée :
  - Exploitation des données du passé ou bien de données connues pour prévoir ou expliquer le futur
  - Résultat : modèle prédictif, évalué au moment de sa production et en cours d'utilisation
  - Qualité d'un modèle :
    - Précision
    - Robustesse en généralisation

Introduction

- Analyse non supervisée
- Analyse supervisée :
  - Exploitation des données du passé ou bien de données connues pour prévoir ou expliquer le futur
  - Résultat : modèle prédictif, évalué au moment de sa production et en cours d'utilisation
  - Qualité d'un modèle :
    - Précision
    - Robustesse en généralisation

Introduction

- Analyse non supervisée
- Analyse supervisée :
  - Exploitation des données du passé ou bien de données connues pour prévoir ou expliquer le futur
  - Résultat : modèle prédictif, évalué au moment de sa production et en cours d'utilisation
  - Qualité d'un modèle :
    - Précision
    - Robustesse en généralisation

Introduction

- Analyse non supervisée
- Analyse supervisée :
  - Exploitation des données du passé ou bien de données connues pour prévoir ou expliquer le futur
  - Résultat : modèle prédictif, évalué au moment de sa production et en cours d'utilisation
  - Qualité d'un modèle :
    - Précision
    - Robustesse en généralisation

Introduction

- Analyse non supervisée
- Analyse supervisée :
  - Exploitation des données du passé ou bien de données connues pour prévoir ou expliquer le futur
  - Résultat : modèle prédictif, évalué au moment de sa production et en cours d'utilisation
  - Qualité d'un modèle :
    - Précision
    - Robustesse en généralisation

Classes de modèles prédictifs

- Classification : la cible est nominale :
  - Le plus souvent, variable binaire (deux classes)
- Régression : la cible est continue

Apprentissage supervisé

Apprendre avec un « professeur » qui connaît la réponse

- $(X_1, X_2, \ldots X_p)$ sont des variables explicatives
- Y est une variable à prédire (une cible); ce peut être une classe, le « professeur »

Il faut trouver

- la classe la plus fréquente de Y pour X donné ou
- la valeur moyenne de Y en fonction de X

Buts de l'apprentissage supervisé

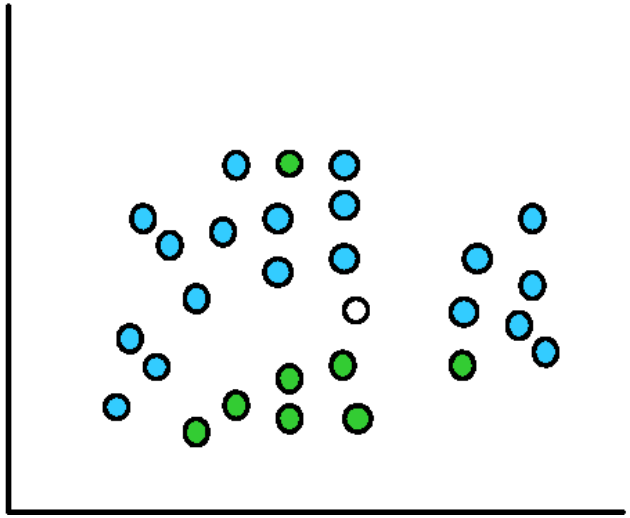
- Prédiction :
  – Trouver un modèle ou une application qui va lier correctement les variables explicatives et les cibles (régression linéaire multiple, régression logistique, etc.)
- Classification :
  – Ajuster un modèle ou un arbre de décision qui associe correctement les variables d'entrée aux classes en sortie (analyse discriminante, arbres d'induction, réseaux neuronaux, SVM, etc.)

Buts de l'apprentissage supervisé

- Prédiction :
  – Trouver un modèle ou une application qui va lier correctement les variables explicatives et les cibles (régression linéaire multiple, régression logistique, etc.)
- Classification :
  – Ajuster un modèle ou un arbre de décision qui associe correctement les variables d'entrée aux classes en sortie (analyse discriminante, arbres d'induction, réseaux neuronaux, SVM, etc.)

Classification

- Apprendre une façon de prédire la classe d'une instance à partir d'un ensemble d'instances déjà classées



Étant donné l'ensemble de points bleus et verts, quelle est la classe du nouveau point o ?

- Approches possibles :
  - Régression
  - Arborescences
  - Bayésien
  - Réseaux neuronaux, ...

Utilisation des méthodes d'apprentissage

- Examiner le problème
- Formuler le problème
- Comprendre les données pour l'apprentissage
- Explorer les données
- Choisir un modèle (paramètres β)
- Apprendre (estimer β)
- Tester les résultats

Description des méthodes

- On dispose d'un ensemble d'apprentissage
- Objectifs :
  - Prédire une variable qualitative Y à k modalités à l'aide de p variables explicatives
  - Les p variables explicatives peuvent être quantitatives ou qualitatives.

Applications

- Organisme financier : juger de la valeur d'un dossier de demande de crédit ;
- Vente par correspondance : sélectionner les clients les plus intéressants ;
- Météorologie : prévision des avalanches à partir de variables liées à l'atmosphère et à la neige ;
- Médecine : aide au diagnostic ;
- etc.

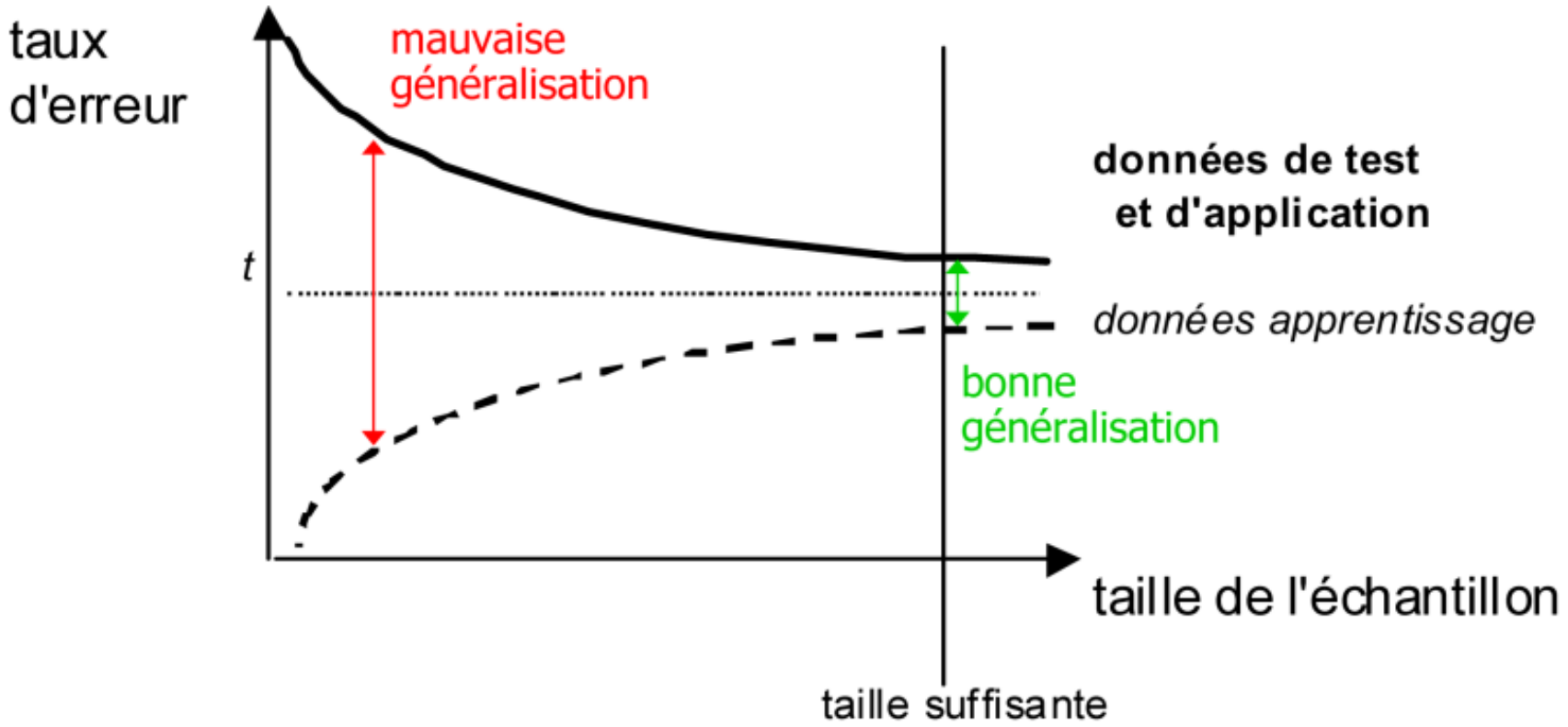
Etapes

- Apprentissage :
  - à partir d'un échantillon d'apprentissage (ensemble de paires entrée-sortie),
    - trouver β qui minimise la fonction d'erreur E exprimée sous forme de paires entrée-sortie
- Test suivi de généralisation :
  - à la population
    - donner une valeur "sortie" à une nouvelle valeur "entrée" étant donné les paramètres β

Etapes

- Apprentissage :
  - à partir d'un échantillon d'apprentissage (ensemble de paires entrée-sortie),
    - trouver β qui minimise la fonction d'erreur E exprimée sous forme de paires entrée-sortie
- Test suivi de généralisation :
  - à la population
    - donner une valeur "sortie" à une nouvelle valeur "entrée" étant donné les paramètres β

Courbes du taux d'erreur en apprentissage et en test



Pièges (1/4)

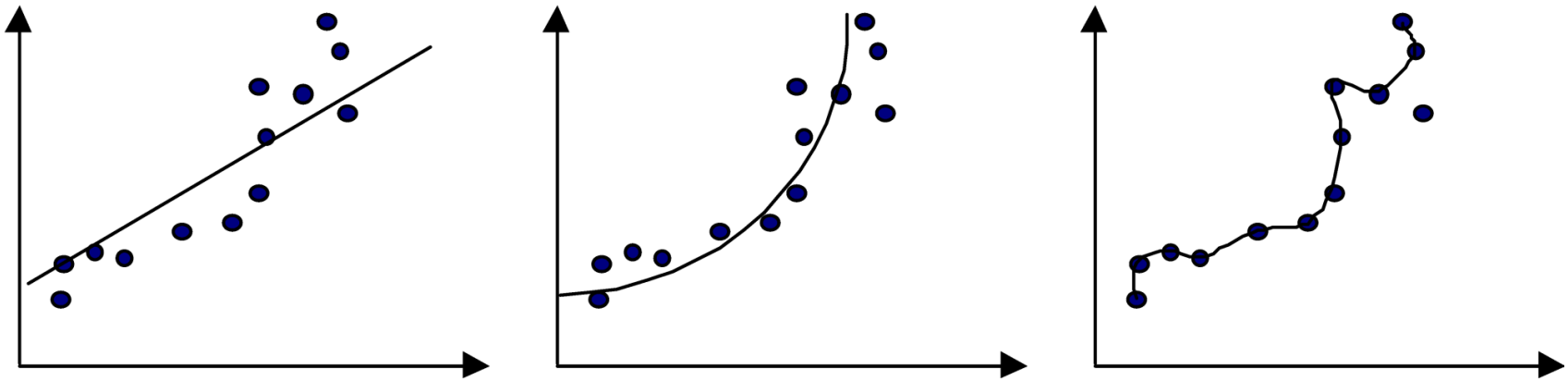
- Sur-ajustement
- Modèle complexe
- La malédiction de la dimension (*curse of dimensionality*)

Pièges (2/4)

Sur-apprentissage

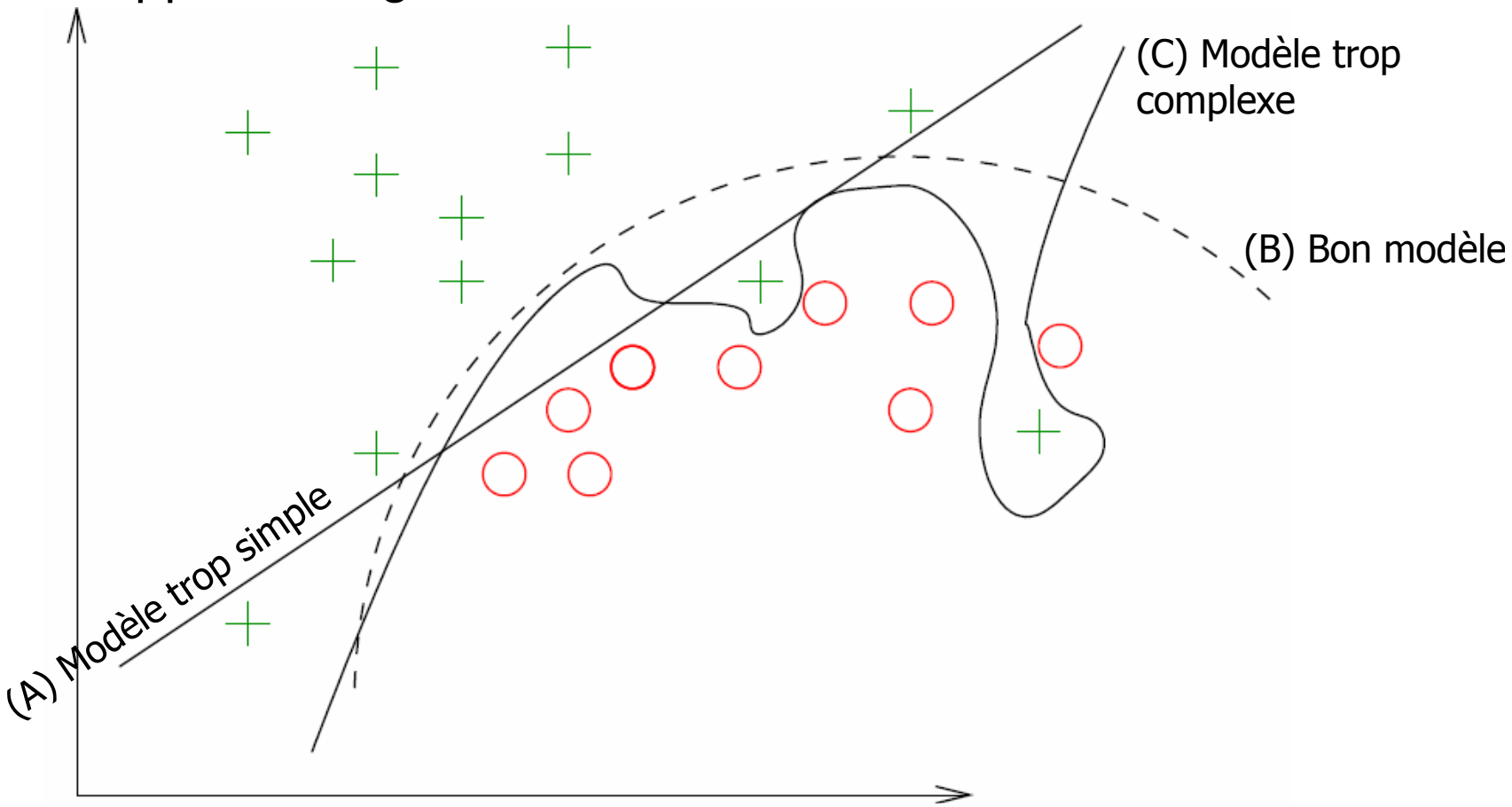
- Le modèle appris s'ajuste parfaitement à l'ensemble d'apprentissage, mais fonctionne mal en généralisation (compromis biais-variance)

Sur-apprentissage en régression



Pièges (3/4)

Sur-apprentissage en classement



Pièges (4/4)

Modèle complexe plutôt que compréhensible

- Les clients (médecins, marketing. . .) n'utiliseront pas un modèle qu 'ils ne comprennent pas.
- Par exemple :
  - Si $\{(\text{durée\_du\_mariage} + 1.35) \times e^{((income)-41.318\$)}\} > 1$ alors envoyer proposition
  - Si {(État = marié) et (revenu > 50.000 $)} alors envoyer proposition

Pièges (4/4)

Modèle complexe plutôt que compréhensible

- Les clients (médecins, marketing. . .) n'utiliseront pas un modèle qu 'ils ne comprennent pas.
- Par exemple :
  - Si {(durée_du_mariage + 1.35) $\times$ $e^{((income)-41.318\$)}$ } >1 alors envoyer proposition
  - Si {(État = marié) et (revenu > 50.000 $)} alors envoyer proposition

Pièges (4/4)

Modèle complexe plutôt que compréhensible

- Les clients (médecins, marketing. . .) n'utiliseront pas un modèle qu 'ils ne comprennent pas.
- Par exemple :
  - Si {(durée_du_mariage + 1.35) $\times$ $e^{((income)-41.318\$)}$ } >1 alors envoyer proposition
  - Si {(État = marié) et (revenu > 50.000 $)} alors envoyer proposition

Techniques de classification supervisée

- Techniques inductives
  - une phase d'apprentissage (phase inductive) pour élaborer un modèle, qui résume les relations entre les variables
  - et qui peut ensuite être appliqué à de nouvelles données pour en déduire un classement ou une prédiction (phase déductive)
- Techniques transductives
  - ne comprennent qu'une seule étape (éventuellement réitérée), au cours de laquelle chaque individu est directement classé (ou objet d'une prédiction) par référence aux autres individus déjà classés
  - Il n'y a pas élaboration d'un modèle

Techniques de classification supervisée

- Techniques inductives
  - une phase d'apprentissage (phase inductive) pour élaborer un modèle, qui résume les relations entre les variables
  - et qui peut ensuite être appliqué à de nouvelles données pour en déduire un classement ou une prédiction (phase déductive)
- Techniques transductives
  - ne comprennent qu'une seule étape (éventuellement réitérée), au cours de laquelle chaque individu est directement classé (ou objet d'une prédiction) par référence aux autres individus déjà classés
  - Il n'y a pas élaboration d'un modèle

k-plus proches voisins (*k-NN*)

- La plus connue des techniques transductives
- Le classement (prédiction) de chaque individu s'opère en regardant, parmi les individus déjà classés, la classe des *k* individus qui sont les plus proches voisins (ou en calculant la moyenne dans le voisinage de la variable à prédire)
- La valeur de *k* sera choisie en sorte d'obtenir le meilleur classement (prédiction) possible :
  - ce choix est la principale difficulté de cet algorithme !

Limites des méthodes transductives

- Une technique inductive résume dans un modèle l'information contenue dans les données
  - ce qui permet d 'appliquer rapidement ce modèle à de nouvelles données
- Une technique transductive manipule l'ensemble des individus déjà classés, pour tout nouveau classement
  - ce qui nécessite donc une grande puissance de stockage et de calcul
- **On utilise surtout les techniques inductives**
- Une méthode transductive, comme les *k*-NN, peut être utilisée dans une étape préalable de détection et de mise à l'écart des individus hors norme (des "outliers")

Méthodes inductives de classification supervisée

- Modèle de Bayes
- Segmentation (arbre de décision)
- SVM
- Réseaux de neurones
- Analyse linéaire discriminante
- Régression logistique

Plan

La classification de Bayes

- Méthode simple de classification supervisée, basée sur l'utilisation du Théorème de Bayes

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)}$$

où H est l'hypothèse à tester, et E est l'évidence associée à l'hypothèse

  - $P(E|H)$ et $P(H)$ sont facilement calculables
  - $P(H)$ est une probabilité a priori : la probabilité de H avant la présentation de l'évidence
  - Pas nécessaire de calculer $P(E)$

- Etant donnée l'évidence E, chercher $h = \arg\max_H P(H|E)$ $\Rightarrow$ chercher $h = \arg\max_H P(E|H)P(H)$

Exemple - Données

Day	Outlook	Temp.	Humidity	Wind	Play
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rainy	Mild	High	Weak	Yes
5	Rainy	Cool	Normal	Weak	No
6	Rainy	Cool	Normal	Strong	Yes
7	Overcast	Cool	Normal	Weak	No
8	Sunny	Mild	High	Weak	Yes
9	Sunny	Cool	Normal	Weak	Yes
10	Rainy	Mild	Normal	Strong	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rainy	Mild	High	Strong	No

Exemple

- Données tennis.xls
- Evidence E : *Sunny, Cool, High, Strong*.
  Hypothèse H : *Yes, No* ?
- $P(Yes|E) = P(Outlook = Sunny|Yes) \times$
  $P(Temp = Cool|Yes) \times P(Humidity = High|Yes) \times$
  $P(Wind = Strong|Yes) \times \frac{P(Yes)}{P(E)}$
  $= \frac{3/9 \times 2/9 \times 4/9 \times 4/9 \times 9/14}{P(E)}$
  $= 0.0094/P(E)$
- $P(No|E) = 0.0137/P(E)$
- $\Rightarrow H = No$

Cas d'un numérateur égal à 0

- Probabilité correspondante égale à 0
- On ajoute une constante k à chaque valeur au numérateur et au dénominateur
  - Un rapport n/d se transforme en $(n + kp)/(d + k)$, où p est une fraction du nombre total des valeurs possibles de l'attribut (par ex. pour l'attribut *Temp* ayant 3 valeurs *Hot*, *Cool* et *Mild* on a $p = 1/3$)
  - k est entre 0 et 1
  - Estimation de Laplace : $k = 1$.

Exemple $P(No|E)$

- $k = 1$
- $P(No|E) = P(Outlook = Sunny|No) \times$
  $P(Temp = Cool|No) \times P(Humidity = High|No) \times$
  $P(Wind = Strong|No) \times \frac{P(No)}{P(E)}$
  $= (2 + 1/3)/(5 + 1) \times (2 + 1/3)/(5 + 1) \times (3 + 1/2)/(5 + 1)$
  $\times (2 + 1/2)/(5 + 1) \times 5/14/P(E)$
  $= 0.0131/P(E)$

Données manquantes

- Les valeurs manquantes de l'évidence sont ignorées, en considérant une probabilité de 1
- Exemple : Evidence E : *?, Cool, High, Strong*.
  Hypothèse H : *Yes, No* ?
- $P(No|E) = 1 \times P(Temp = Cool|No) \times$
  $P(Humidity = High|No) \times P(Wind = Strong|No) \times \frac{P(No)}{P(E)}$

Données numériques

- Une fonction de densité des probabilité $f(x)$ représente la distribution normale $N(\mu; \sigma)$ des données de l'attribut numérique x

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \qquad \mu = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad \sigma = \frac{1}{n-1}\sum_{i=1}^{n}(x_i-\mu)^2$$

- Les valeurs manquantes ne sont pas incluses dans les calculs des moyennes et des déviations standards

Exemple de calcul de $f(x)$

- Supposons l'attribut "Température" est numérique, et à partir des valeurs de température correspondant à la valeur "Yes" de l'attribut "Play", on obtient $\mu = 73$ et $\sigma = 6.2$. Alors pour une évidence de température soit 66, on calcule :

$$f(Temp = 66|Yes) = \frac{1}{\sqrt{2\pi}6.2} e^{-\frac{(66-73)^2}{2*6.2^2}} = 0.0340$$

Critiques de la méthode

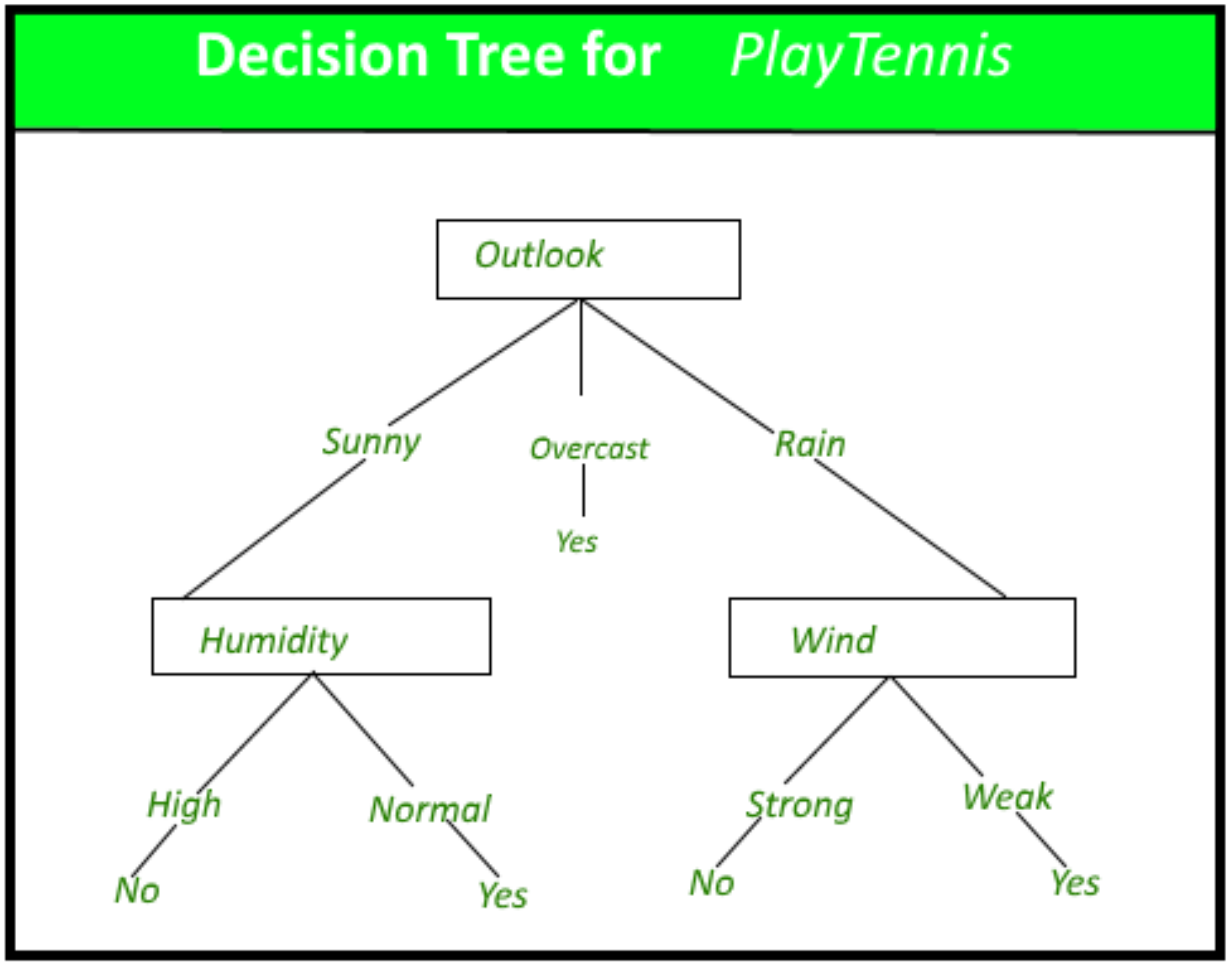
- Méthode efficace
- La classification ne demande pas des estimations exactes des probabilités, mais seulement que la probabilité maximum soit donnée à la bonne classe
- Les numériques ne sont pas toujours distribués normalement, on a donc besoin d'autres estimations (Ex. *Kernel density estimator*)

Plan

Arbres de décision

- Une structure de données utilisée comme modèle pour la classification [Quinlan]
- Méthode récursive basée sur diviser-pour-régner pour créer des sous-groupes (plus) purs (un sous-groupe est pur lorsque tous les éléments du sous-groupe appartiennent à la même classe)
- Construction du plus petit arbre de décision possible
- Nœud = Test sur un attribut
- Une branche pour chaque valeur d'un attribut
- Les feuilles désignent la classe de l'objet à classer
- Taux d'erreur: La proportion des instances qui n'appartiennent pas à la classe majoritaire de la branche
- Problèmes: Choix de l'attribut, terminaison

Arbre de décision : exemple



Principes de base du processus de segmentation

- Itératif
- Descendant
- Dichotomique

Soit une partition existant *a priori* sur la population, définie par la variable Y notée aussi souvent C
$\Rightarrow$ Le principe de la segmentation consiste, à partir de la population totale, à effectuer des dichotomies successives, de façon à obtenir des sous-populations le plus homogènes possibles vis-à-vis des groupes de la partition *a priori*.

Dénominations

- Plusieurs termes utilisés pour cette approche
  - Segmentation
  - Partitionnement récursif (*Recursive Partition*)
  - Discrimination ou régression par arbre (*Classification Tree, Regression Tree*)
  - Arbres de décision (*Decision Tree*)

Algorithmes

- Les deux algorithmes les plus connus et les plus utilisés (l'un ou l'autre ou les deux sont présents dans les environnements de fouille de données) sont CART (Classification And Regression Trees [BFOS84]) et C5 (version la plus récente après ID3 et C4.5 [Qui93]).
  - *[BFOS84]* L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. Classification and regression trees. Technical report, Wadsworth International, Monterey, CA, 1984.
  - *[Qui93]* J. R. Quinlan. C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo, CA, 1993.

Pour quels types de données?

- On se restreint d'abord aux données nominales seulement
- Extension aux numériques :
  - Il est possible de traiter les numériques en les transformant en nominaux (ou ordinaux) par discrétisation

Algorithme

Approche récursive pour construire l'arbre

construire-arbre(X)

Si tous les individus de X sont de même classe

créer une feuille associée à cette classe

Sinon

- choisir (selon critère) le meilleur couple (attribut; test) pour créer un nœud
- ce test sépare X en 2 parties X_g et X_d
- construire-arbre(X_g)
- construire-arbre(X_d)

Critère de choix de l'attribut

- Mesure de l'hétérogénéité du nœud candidat :
  Choisir l'attribut avec le plus grand gain d'information (théorie de l'information)
  - Entropie (ID3, C4.5)
  - Indice Gini (CART)
  - Indice d'erreur
- Entropie : $H = -\sum_k p_k \log p_k$
  p_k : probabilité de la classe k (estimée par la proportion N_k/N)
  - min $H = 0$ si une seule classe présente
  - max $H = \log$(nombre de classes) si classes équi-réparties
- Indice Gini : $Gini = 1 - \sum_k p_k^2$
- Indice d'erreur : $Err = 1 - \max_k p_k$

Gain d'homogénéité apporté par un test

- Soit un test T à m alternatives et divisant le nœud N en m sous-nœud N_j
- Soient Info(N_j) les mesures d'hétérogénéité (entropie, Gini, ...) des sous-nœuds, et $p(N_j)$ les proportions des éléments de N dirigés vers N_j par le test T
  - le gain d'homogénéité/information apporté par le test T est

$$\text{Gain}(N, T) = \text{Info}(N) - \sum_j p(N_j)\text{Info}(N_j)$$

- A chaque nœud, choix du test maximisant le gain (ou éventuellement le "rapport des gains" Gain(N, T)/Info(T), utilisé par C4.5 pour éviter le biais vers m grand)

Exemple - Calcul Info(Outlook) pour l'arbre tennis.xlsx

- Calcul le gain d'information si l'on choisit l'attribut Outlook pour créer des sous-arbres de l'arbre original
- Information = Entropie
- Pour l'arbre
  - Outlook = "Sunny", Info("Sunny") = $-p_1 \log p_1 - p_2 \log p_2$, avec $p_1 = p_{yes} = 3/5, p_2 = p_{no} = 2/5$
  - Outlook = "Overcast", Info("Overcast") = $-p_1 \log p_1 - p_2 \log p_2$, avec $p_1 = 3/4, p_2 = 1/4$
  - Outlook = "Rainy", Info("Rainy") = $-p_1 \log p_1 - p_2 \log p_2$, avec $p_1 = 3/5, p_2 = 2/5$
  - Info(Outlook) = (5/14)Info("Sunny")+(4/14)Info("Overcast")+(5/14)Info("Rainy") =0.925
- A la racine : Info(racine) = Info([9,5])=0.940
- Gain d'information pour Outlook : Gain(racine, Outlook) = 0.940 - 0.925 = 0.015

Exemple - Sélection de l'attribut

- Calculer le gain d'information pour Outlook (0.015), Temperature (0.090), Humidity (0.016), Wind (0.001)
- Choisir l'attribut donnant le gain maximum (Temperature pour cet exemple)
- Passe à l'étape suivant : Pour chaque branche, sélection d'un deuxième attribut
- Et on continue ...

Terminaison

- Règles évidentes :
  - Tous les attributs ont été considérés
  - L'hétérogénéité des nœuds ne diminue plus
  - Le maximum de pureté a été atteint
    - Toutes les instances sont dans la même classe
  - L'arbre a atteint une hauteur maximum
- Contrôle des performances de généralisation (sur base de validation indépendante)
- Elagage a posteriori : supprimer des branches peu représentatives et nuisant à généralisation (parcours bottom-up en remontant d'un niveau tant que cela diminue erreur en généralisation)

Extensions de l'algorithme

- Comment traiter les attributs numériques et les valeurs manquantes
- Comment simplifier le modèle pour éviter les bruits?
- Comment tolérer les bruits?
- Comment interpréter les arbres de décision?

Comment traiter les attributs numériques ?

- Les attributs numériques sont transformés en ordinaux/nominaux. Ce processus est appelé discrétisation
- Les valeurs des attributs sont divisées en intervalles
  - Les valeurs des attributs sont triées
  - Des séparations sont placées pour créer des intervalles / classes pures
  - On détermine les valeurs des attributs qui impliquent un changement de classes
- Ce processus est très sensible au bruit
- Le nombre de classes doit être contrôlé
  - Solution : On spécifie un nombre minimum d'éléments par intervalle
  - On combine les intervalles qui définissent la même classe

Exemple: Les températures

- Étape 1: Tri et création des intervalles

64		65		68 69	70		71 72		72		75 75		80		81	83		85
Y		*N*		*Y Y*	*Y*		*N N*		*Y*		*Y Y*		*N*		*Y*	*Y*		*N*

- Étape 2: Les anomalies sont traitées

64		65		68 69	70		71 72	72		75 75		80		81	83		85
Y		*N*		*Y Y*	*Y*		*N N*	*Y*		*Y Y*		*N*		*Y*	*Y*		*N*

- Étape 3: Un minimum de 3 éléments (de la même classe) par intervalle

64 65	68 69	70		71 72	72 75	75		80 81	83 85
Y N	*Y Y*	*Y*		*N N*	*Y Y*	*Y*		*N Y*	*Y N*

- Étape 4: Combiner les intervalles

64 65	68 69	70 71	72 72	75 75		80	81 83	85
Y N	*Y Y*	*Y N*	*N Y*	*Y Y*		*N*	*Y Y*	*N*

- Étape 5: Changement de classe pour une température de 77.5 ((75 + 80) / 2).

Les valeurs manquantes

- Ignorer les instances avec des valeurs manquantes

  Solution trop générale, et les valeurs manquantes peuvent ne pas être importantes

- Ignorer les attributs avec des valeurs manquantes

  Peut-être pas faisable

- Traiter les valeurs manquantes comme des valeurs spéciales

  Les valeurs manquantes ont un sens particulier

- Estimer les valeurs manquantes

  - Donner la valeur de l'attribut la plus répandue à l'attribut considéré

  - Imputation de données en utilisant diverses méthodes (régression)

Critiques de la méthode

- Apprentissage supervisé
- Le résultat est lisible

  Outils de navigation dans l'arbre
- Les valeurs manquantes peuvent être traitées
- Tous les types d'attributs peuvent être pris en compte
- Elle peut être utilisée comme pré-traitement
- La classification d'un exemple est très efficace
- Moins efficace pour un nombre important de classes
- Elle n'est pas incrémentale
- Sensibilité au bruit et points aberrants
- Stratégie d'élagage délicate

Forêts aléatoires

- Algorithme proposé par Breiman & Cutter en 2001
- Principe : l'union fait la force
  - Une forêt = un ensemble d'arbres
  - Forêt aléatoire
    - Apprendre un grand nombre T d'arbres simples (dizaines ou centaines)
    - Utilisation par *vote des arbres* (classe majoritaire, voire probabilités des classes par % des votes) pour le problème de classification, ou *moyenne des arbres* pour le problème de régression.

Apprentissage de forêt aléatoire

But : obtenir des arbres les plus décorélés possibles

- Chaque arbre appris sur un sous-ensemble ($\sim$ 2/3) aléatoire différent des exemples d'apprentissage
- Chaque nœud de chaque arbre choisi comme "split" optimal parmi k variables tirées aléatoirement dans les entrées (avec $k << d$ la dimension des entrées)
- Chaque arbre appris avec algo CART sans élagage
- On limite fortement la profondeur p des arbres ($\sim$ 2 à 5)

Critiques de la méthode

Avantages

- Reconnaissance très rapide
- Multi-classes par nature
- Efficace sur inputs de grande dimension
- Robustesse aux outliers

Inconvénients

- Apprentissage souvent long
- Valeurs extrêmes souvent mal estimées dans le cas de régression

Plan

SVM

- Séparateur à Vaste Marge / Machine à vecteurs de support (Vapnik, 1998)
- Méthodes du noyau
  - Famille d'algorithmes aux particularités utiles à l'analyse de formes
- Objectif : Trouver une séparation généralement non-linéaire en utilisant des noyaux et assurer que la séparation a de bonnes propriétés statistiques pour éviter le sur-apprentissage

Etapes de la méthode SVM

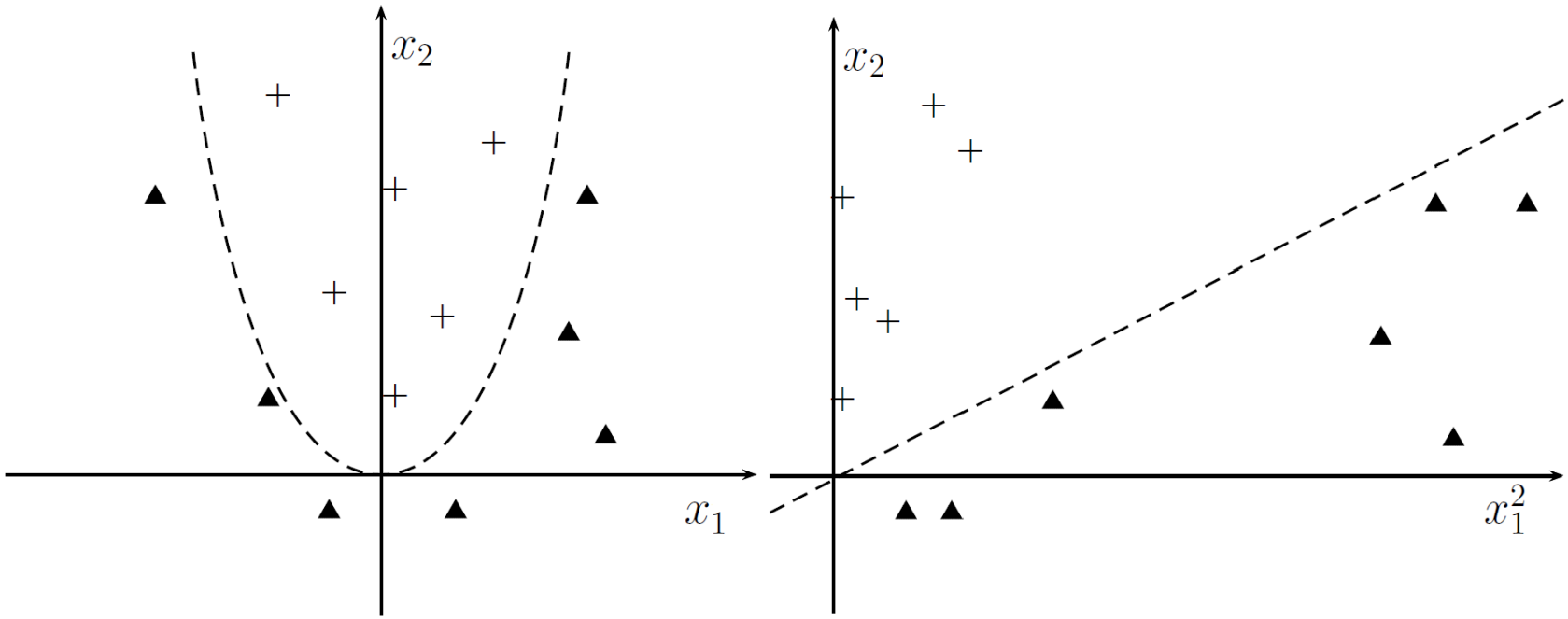
Deux étapes

- Transformation non linéaire Φ pour passer dans un espace de dimension plus grande que l'espace d'origine, mais doté d'un produit scalaire.
- Dans cet espace, on cherche un séparateur linéaire $f(x) = ax + b$ (par ex. : fonction discriminante de Fisher), qui est un hyperplan optimal
  - séparant bien les groupes (précision du modèle) $f(x) > 0 \Rightarrow$ classe A ; $f(x) \leq 0 \Rightarrow$ classe B
  - le plus loin possible de tous les cas
- On exprime $f(\Phi(x))$ sans faire intervenir explicitement Φ

Exemple de transformation

Observer l'augmentation de la dimension

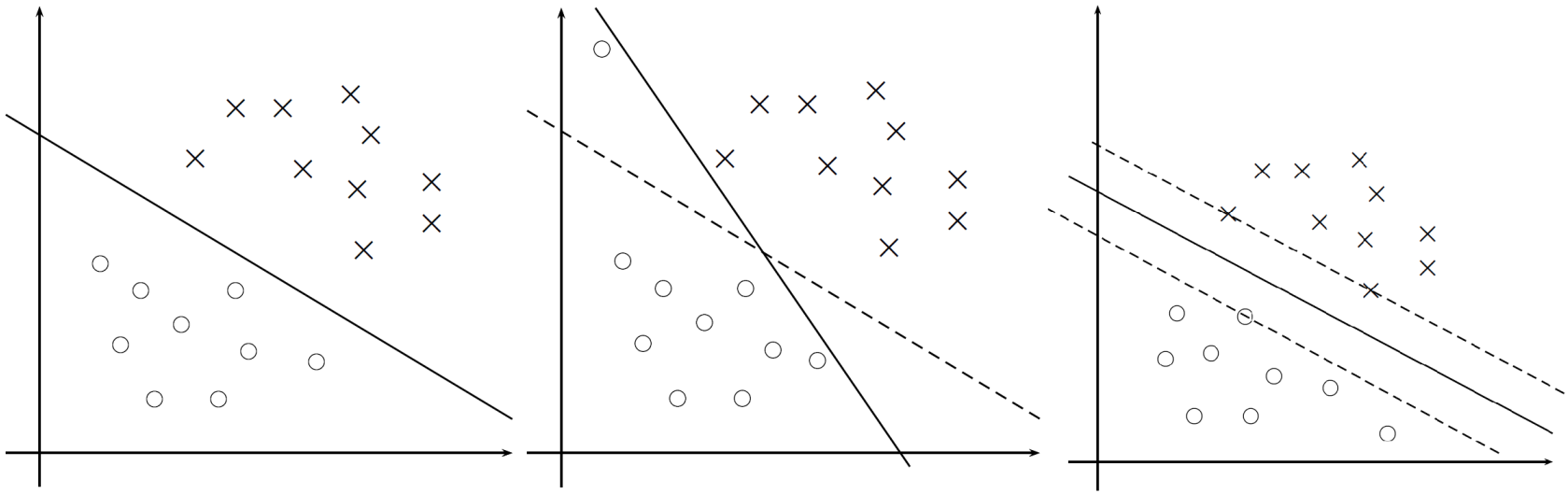
- Espace naturel (*Input Space*) $\vec{x} = (x_1, x_2)$ (2 attributs)
- Espace artificiel (*Feature Space*)
  $\Phi(\vec{x}) = (x_1^2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1x_2, 1)$ (6 attributs)



Maximisation de la marge

- Distance d'un point x à l'hyperplan

$$\frac{|a.x + b|}{||a||}$$



Forme de la solution

- Etant donnés les points (x_i, y_i), avec $y_i = 1$ si $x_i \in A$ et $y_i = 0$ si $x_i \in B$, trouver le séparateur linéaire $f(x) = a.x + b \Leftrightarrow$ trouver (a, b) satisfaisant :
  - $\forall i,\ y_i(a.x_i + b) \geq 1$ (bonne séparation)
  - $||a||^2$ est minimum (marge maximale)

  $\Rightarrow$ La solution $f(x)$ s'exprime en fonction de produits scalaires $x.x'$
- Après transformation Φ, la solution s'exprime en fonction de produits scalaires $\Phi(x).\Phi(x')$
- La quantité $k(x, x') = \Phi(x).\Phi(x')$ est appelé **noyau** (*kernel*)
- C'est k et non Φ que l'on choisit : on peut calculer $k(x, x')$ sans faire apparaître Φ
- Les calculs sont alors faits dans l'espace de départ, et sont beaucoup plus simples et plus rapides.

Exemples de noyaux

- Linéaire $k(x, x') = x.x'$
- Polynomial $k(x, x') = (x.x')^d$
- Gaussien (RBF)

$$k(x, x') = e^{-\frac{||x-x'||^2}{2\sigma^2}}$$

- Sigmoïdal

$$k(x, x') = \tanh(a(x.x' - b))$$

Avantages de SVM

- Intérêt des SVM : précision des prédictions
- Robuste pour un grand nombre de variables et petit nombre d'exemples
- Capable d'apprendre des modèles de classification complexes ou simples
- Utiliser des principles mathématiques sophistiqués pour éviter le sur-apprentissage

Plan

Analyse discriminante (AD)

- Aspect géométrique :
  - Réduction de dimension, axes et variables discriminantes
  - Méthodes géométriques de classement

Représentation des données

- n points dans $\mathcal{R}^p$ appartenant à k groupes.

	1	2	$\cdots$	k
1	0	1	$\cdots$	0
2	1	0	$\cdots$	0
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
i	0	0	$\cdots$	1
n	1	0	$\cdots$	0

indicatrices des groupes

1	2	$\cdots$	j	$\cdots$	p
X_1^1	X_1^2	$\cdots$	X_1^j	$\cdots$	X_1^p
X_i^1	X_i^2	$\cdots$	X_i^j	$\cdots$	X_i^p
X_n^1	X_n^2	$\cdots$	X_n^j	$\cdots$	X_n^p

variables explicatives

Réduction de dimension. Recherche d'axes et de variables discriminantes

- Les n individus forment un nuage de n points dans $\mathcal{R}^p$, formé des k sous-nuages G_i à différencier
- Variance inter-classe = variance des barycentres g_i des classes G_i

$$B = \frac{1}{n}\sum n_i(g_i - g)(g_i - g)' \quad \text{matrice de covariance inter-groupe}$$

- Variance intra-classe = moyenne des variances V_i des classes G_i

$$W = \frac{1}{n}\sum n_i V_i \qquad \text{matrice de covariance intra-groupe}$$

- $W + B = V$ variance totale
- Impossible de trouver un axe u qui simultalnément :
  - maximise la variance interclasse sur u : $\max u'Bu$
  - minimise la variance intraclasse sur u : $\min u'Wu$

Réduction de dimension. Recherche d'axes et de variables discriminantes

- Les n individus forment un nuage de n points dans $\mathcal{R}^p$, formé des k sous-nuages G_i à différencier
- Variance inter-classe = variance des barycentres g_i des classes G_i

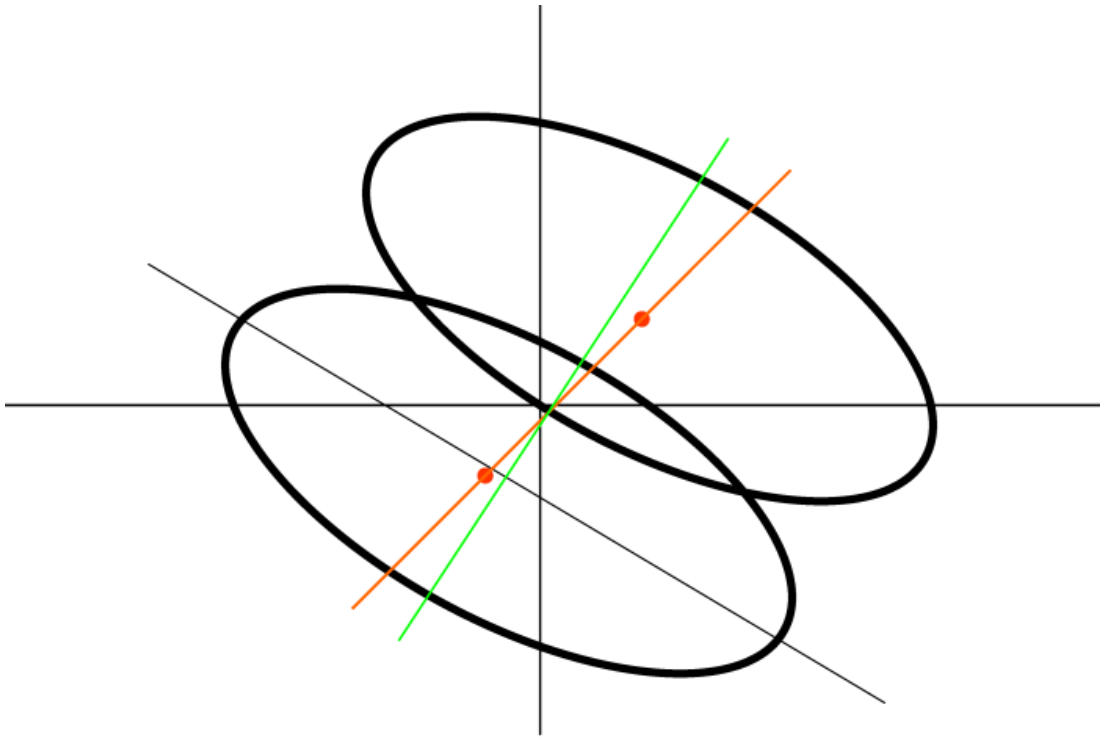
$$B = \frac{1}{n}\sum n_i(g_i - g)(g_i - g)' \quad \text{matrice de covariance inter-groupe}$$

- Variance intra-classe = moyenne des variances V_i des classes G_i

$$W = \frac{1}{n}\sum n_i V_i \qquad \text{matrice de covariance intra-groupe}$$

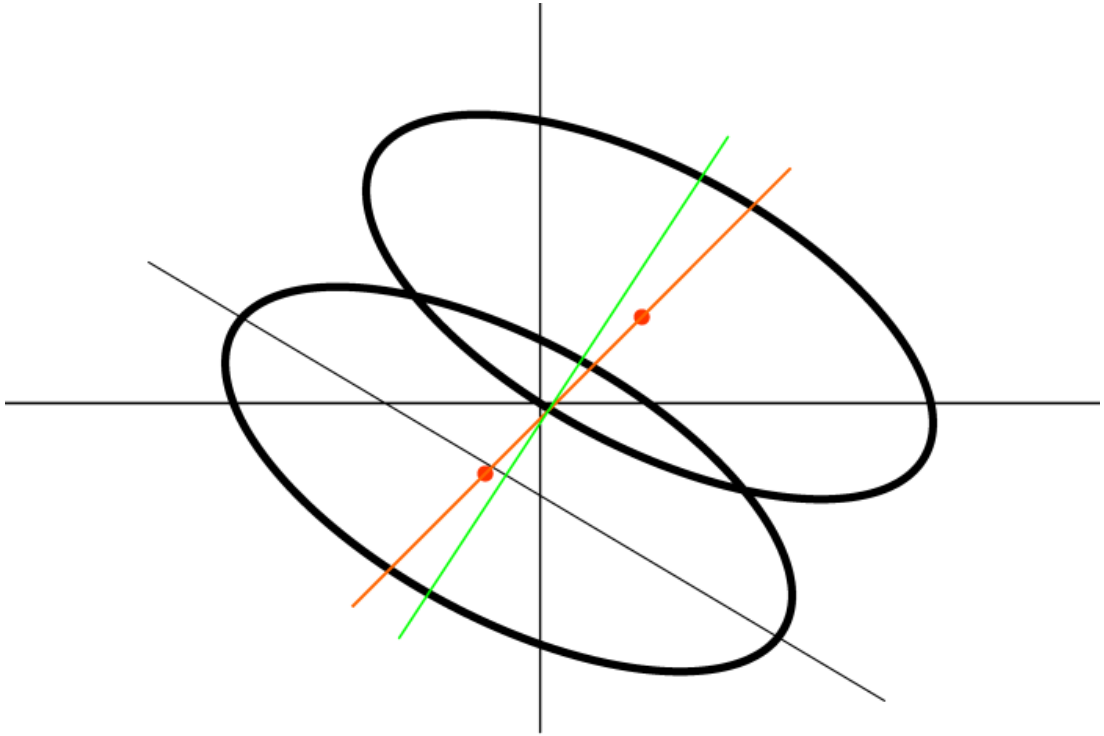
- $W + B = V$ variance totale
- Impossible de trouver un axe u qui simultalnément :
  - maximise la variance interclasse sur u : $\max u'Bu$
  - minimise la variance intraclasse sur u : $\min u'Wu$

Visualisation du double objectif



- maximum de dispersion interclasse : u parallèle au segment joignant les centroïdes
- minimum de dispersion intraclasse : u perpendiculaire à l'axe principal des ellipses (on suppose l'homo-scédasticité)

Visualisation du double objectif



- maximum de dispersion interclasse : u parallèle au segment joignant les centroïdes
- minimum de dispersion intraclasse : u perpendiculaire à l'axe principal des ellipses (on suppose l'homo-scédasticité)

Compromis entre les deux objectifs

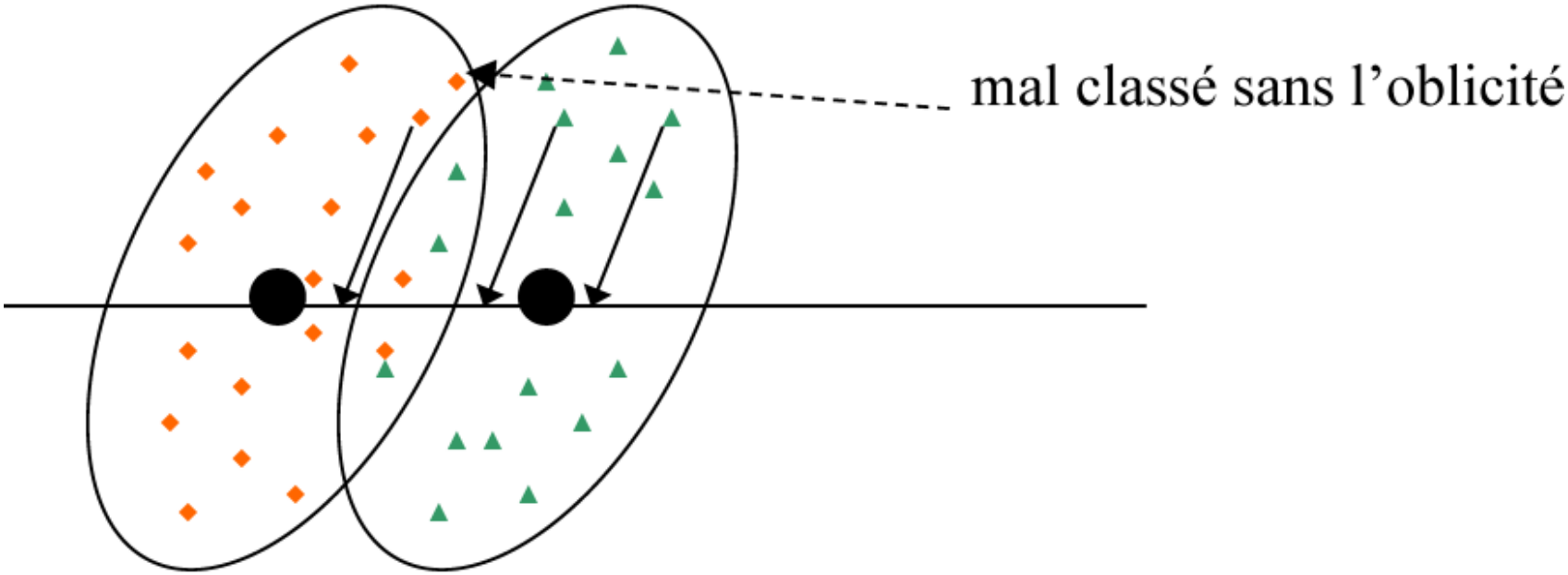
- On reformule l'objectif : au lieu de maximiser $u'Bu$ ou minimiser $u'Wu$, on maximise $u'Bu/u'Wu$

$$\Leftrightarrow \text{ maximiser } u'Bu/u'Vu$$

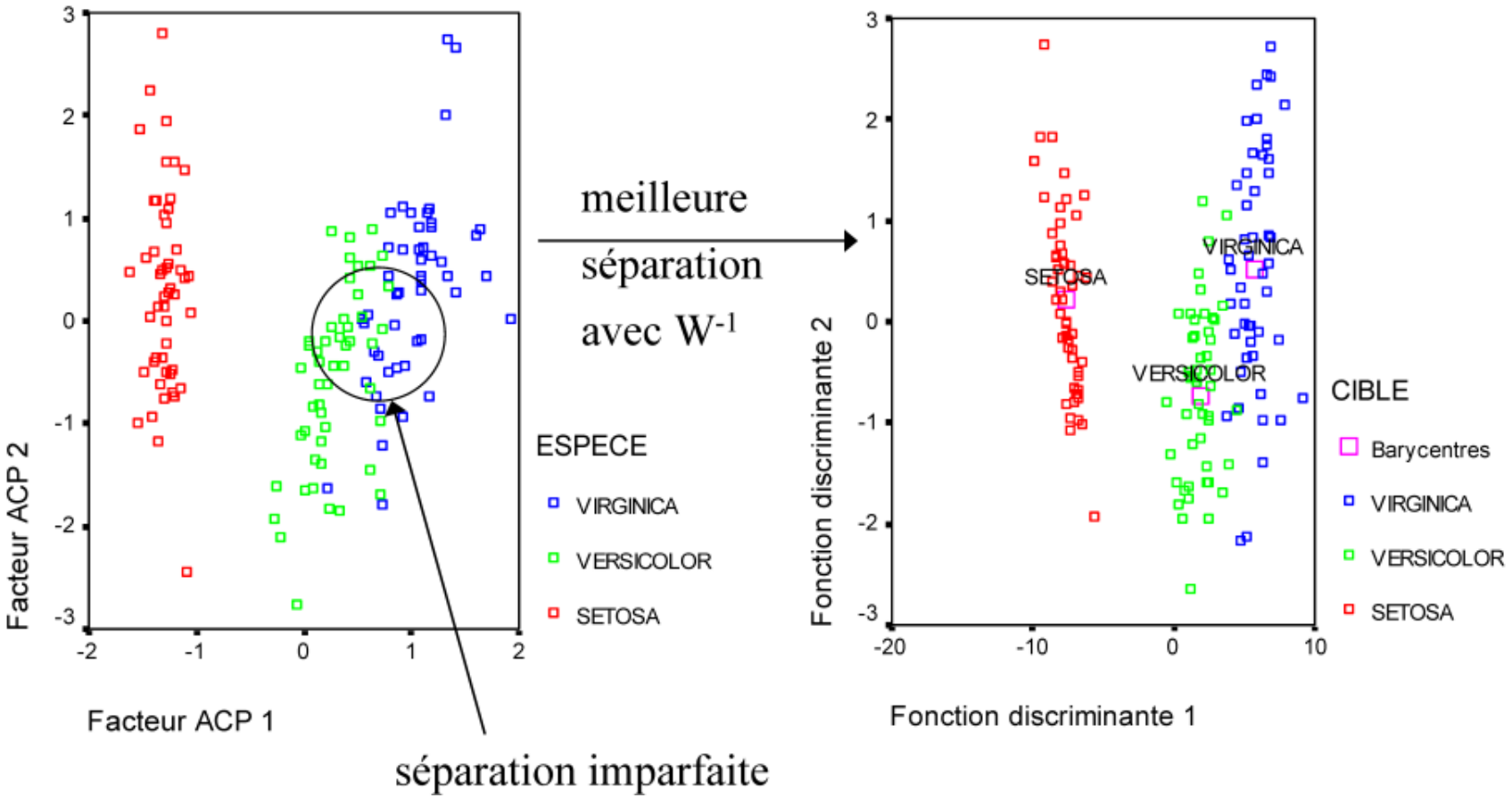
- On a:
  - a) $V^{-1}Bu = \lambda u \quad \Leftrightarrow \quad Bu = \lambda Vu \quad \Leftrightarrow \quad Bu = \lambda(W+B)u$
    $\Rightarrow (1-\lambda)Bu = \lambda Wu$
  - b) $W^{-1}Bu = \frac{\lambda}{1-\lambda}u$
    - la solution u est le vecteur propre de $V^{-1}B$ associé à λ la plus grande valeur propre de $V^{-1}B$
    - u vecteur propre de $V^{-1}B \Leftrightarrow u$ vecteur propre de $W^{-1}B$, de valeur propre $\lambda/(1-\lambda)$
- On dit que les métriques V^{-1} et W^{-1} (de Mahalanobis) sont équivalentes
- Distance de 2 points x et y : $d^2(x,y) = (x-y)'W^{-1}(x-y)$

Autre formulation de la solution

- ACP du nuage des centroïdes g_i avec :
  - métrique V^{-1}
  - ou métrique W^{-1} équivalente
- Ces métriques correspondent à une projection oblique
- Sans cette oblicité, il s'agirait d'une simple ACP mais les groupes seraient mal séparés



ACP avec métrique usuelle et avec W^{-1}

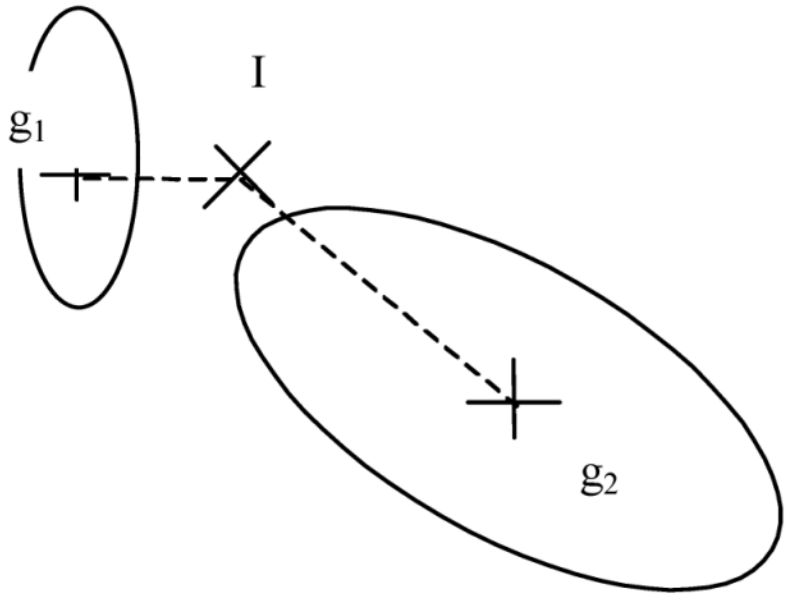


Les différents cas selon λ_1

- $\lambda_1 = 0$: Aucune séparation linéaire n'est possible, groupes concentriques
- $\lambda_1 = 1$: Séparation parfaite
- $0 < \lambda_1 < 1$: Séparation possible avec groupes non recouvrants

Limite de la règle géométrique d'affectation

- Règle géométrique : affecter chaque individu au groupe dont il est le plus proche (distance de l'individu au centroïde du groupe)
  - ce n'est pas trivial car il faut prendre la métrique W^{-1} (faire une projection oblique de x sur l'axe discriminant)
- A éviter si les 2 groupes ont des probabilités *a priori* ou des variances différentes



$\Rightarrow$ analyse discriminante quadratique

Plan

Régression linéaire

- $X = (X_1, \cdots, X_k)$ - variables explicatives continues ou binaires
- Variable à prédire Y est continue
- Les valeurs à prédire peuvent être représentées par une fonction linéaire, donc un hyperplan

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_k X_k + \epsilon$$

où ϵ est un terme aléatoire de loi $N(0, \sigma)$

- Méthode des moindres carrés : recherche $\hat{\beta}_0, \hat{\beta}_1, \cdots, \hat{\beta}_k$ minimisant $\sum_{i=1}^{n} (y_i - \hat{y}_i)^2$, n étant le nombre d'observations.

Régression logistique

- $X = (X_1, \cdots, X_k)$ - variables explicatives
- Variable à prédire Y est binaire (0 ou 1)
- Au lieu de prédire la valeur de Y, on prédit $P(Y = 1|X)$ ou $P(Y = 0|X)$.
- Les probabilités décrivent une sigmoïde (courbe en forme de S) entre 0 et 1

Le modèle de régression logistique

$$P(Y = 1|x_1, x_2, \cdots, x_k) = \frac{e^{\beta_0 + \sum_{i=1}^{k} \beta_i x_i}}{1 + e^{\beta_0 + \sum_{i=1}^{k} \beta_i x_i}}$$

- β_i à estimer par des programmes (utilisant des méthodes comme MLE - Maximum Likelihood Estimate ou Newton-Raphson)
- $\beta_i = 0$: pas d'effet sur la chance de succès, $\beta_i > 0$: augmente la chance, $\beta_i < 0$: décroît la chance

Fonctions de lien

- On écrit : $\pi(x) = P(Y = 1|x_1, x_2, \cdots, x_k)$

$$\pi(x) = \frac{e^{\beta_0 + \sum_{i=1}^{k} \beta_i x_i}}{1 + e^{\beta_0 + \sum_{i=1}^{k} \beta_i x_i}}$$

$$\Leftrightarrow \quad \underbrace{\log\left(\frac{\pi(x)}{1 - \pi(x)}\right)}_{\text{fonction de lien : logit}} = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k$$

- D'autres fonctions de lien (assez similaires)
  - Fonction normit ou probit $g(\mu) = \Phi^{-1}(\mu)$, où Φ est la fonction de répartition d'une loi normale centrée réduite

$$s(t) = \int_{-\infty}^{t} \frac{e^{-z^2/2}}{\sqrt{2\pi}} dz$$

  - Fonction log-log $g(\mu) = \log(-\log(1 - \mu))$

$$s(t) = 1 - \exp[-\exp(t)]$$

Fonctions de lien

- On écrit : $\pi(x) = P(Y = 1|x_1, x_2, \cdots, x_k)$

$$\pi(x) = \frac{e^{\beta_0 + \sum_{i=1}^{k} \beta_i x_i}}{1 + e^{\beta_0 + \sum_{i=1}^{k} \beta_i x_i}}$$

$$\Leftrightarrow \qquad \underbrace{\log\left(\frac{\pi(x)}{1 - \pi(x)}\right)}_{\text{fonction de lien : logit}} = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k$$

- D'autres fonctions de lien (assez similaires)
  - Fonction normit ou probit $g(\mu) = \Phi^{-1}(\mu)$, où Φ est la fonction de répartition d'une loi normale centrée réduite

$$s(t) = \int_{-\infty}^{t} \frac{e^{-z^2/2}}{\sqrt{2\pi}} dz$$

  - Fonction log-log $g(\mu) = \log(-\log(1 - \mu))$

$$s(t) = 1 - \exp[-\exp(t)]$$

Odds-ratio (OR) d'un régresseur X_i

- Mesure l'évolution du rapport de chance d'apparition de l'événement $Y = 1$ contre $Y = 0$ lorsque X_i passe de x à $x + 1$.
  - logit$(\pi(x))$ augmente du coefficient β_i de X_i
    $\Rightarrow$ la cote $\pi(x)/(1 - \pi(x))$ est multipliée par $\exp(\beta_i)$
- Formule générale :

$$OR = \frac{\pi(x+1)/(1-\pi(x+1))}{\pi(x)/(1-\pi(x))} = \exp^{\beta_i}$$

- Si X_i est binaire 0/1, la formule devient :

$$OR = \frac{P(Y=1/X_i=1)/P(Y=0/X_i=1)}{P(Y=1/X_i=0)/P(Y=0/X_i=0)} = \exp^{\beta_i}$$

Critiques de la méthode

- Méthode efficace et facile à comprendre
- Les prédictions sont faciles à réaliser
- Le bruit peut avoir un effet significatif sur la méthode
- Besoin de plusieurs mesures pour évaluer le modèle

Plan

Introduction

Réseaux neuronaux artificiels

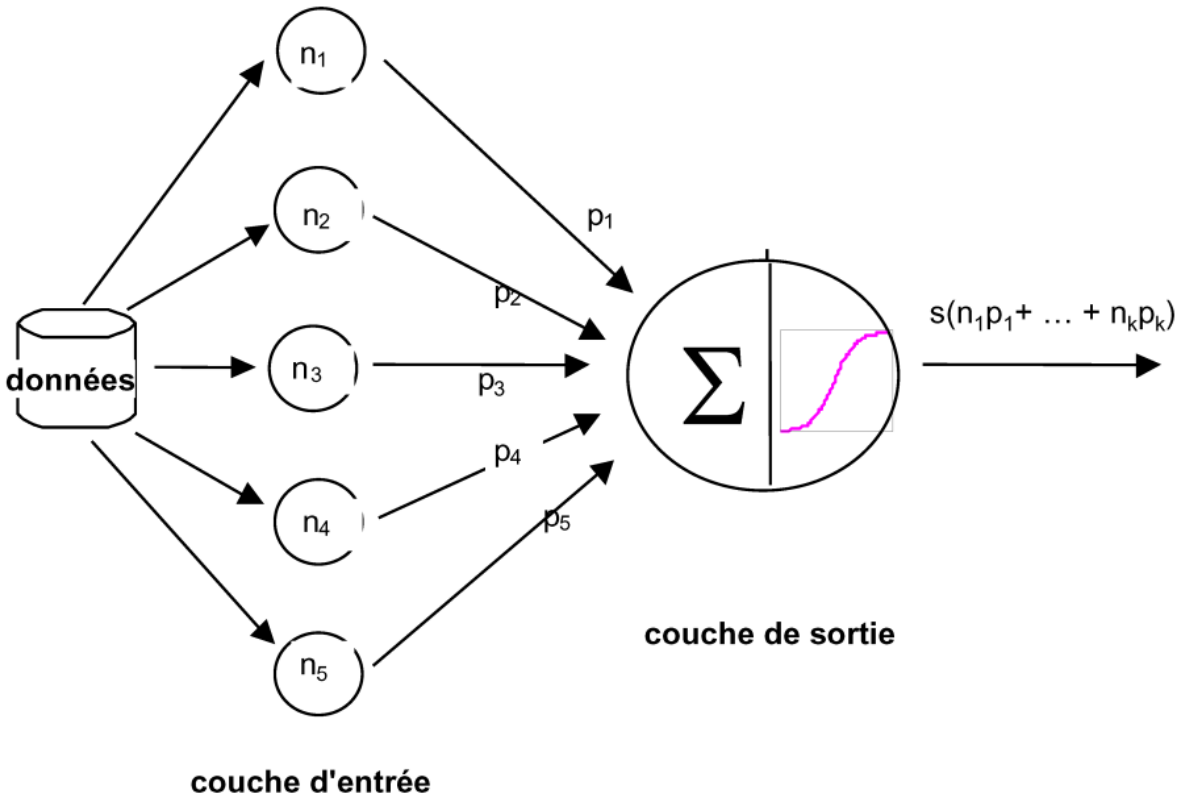
- Inspirés par la biologie : modelé sur le cerveau humain
- Apprentissage, adaptation à la situation, solutions différentes

Classification avec réseaux neuronaux

- Peut choisir des régions plus compliquées
- Peut être plus précis
- Peut sur-ajuster : trouver des motifs dans du bruit

Les réseaux de neurones

- Un réseau de neurones : ensemble de nœuds connectés entre eux, chaque variable correspondant à un nœud
- Le plus courant est le perceptron :



Principe du perceptron multicouches

- Lors de l'**apprentissage du réseau de neurones**, pour chaque exemple présenté en entrée, la valeur renvoyée (" rétropropagée ") par le nœud de sortie est comparée à la valeur réelle, et les poids p_i sont ajustés.
- La fonction de *combinaison* $\sum_i n_i p_i$ (produit scalaire) est suivie d'une fonction de *transfert* : souvent la sigmoïde $s(x) = 1/[1 + exp(-x)]$
- L'échantillon d'apprentissage est parcouru plusieurs fois. L'apprentissage s'achève lorsque
  - une solution optimale a été trouvée ou
  - un nombre fixé d'itérations a été atteint.
- L'apprentissage se fait en ajustant 1 à 1 chaque poids (rétropropagation), ou par modification aléatoire des poids suivie d'un mécanisme de sélection (algorithme génétique)

Principe du perceptron multicouches

- Lors de l'**apprentissage du réseau de neurones**, pour chaque exemple présenté en entrée, la valeur renvoyée (" rétropropagée ") par le nœud de sortie est comparée à la valeur réelle, et les poids p_i sont ajustés.
- La fonction de *combinaison* $\sum_i n_i p_i$ (produit scalaire) est suivie d'une fonction de *transfert* : souvent la sigmoïde $s(x) = 1/[1 + exp(-x)]$
- L'échantillon d'apprentissage est parcouru plusieurs fois. L'apprentissage s'achève lorsque
  - une solution optimale a été trouvée ou
  - un nombre fixé d'itérations a été atteint.
- L'apprentissage se fait en ajustant 1 à 1 chaque poids (rétropropagation), ou par modification aléatoire des poids suivie d'un mécanisme de sélection (algorithme génétique)

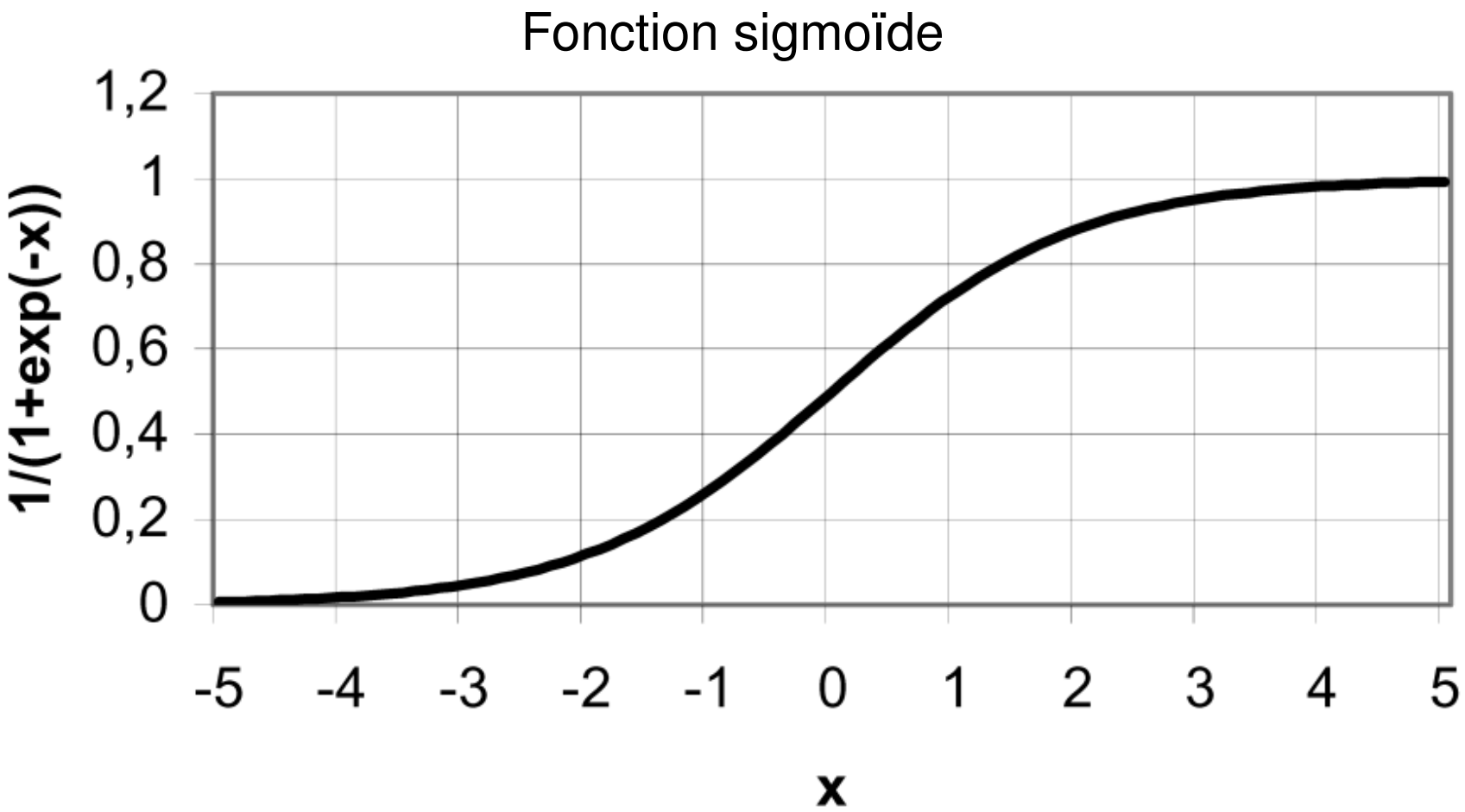
Principe du perceptron multicouches

- Lors de l'**apprentissage du réseau de neurones**, pour chaque exemple présenté en entrée, la valeur renvoyée (" rétropropagée ") par le nœud de sortie est comparée à la valeur réelle, et les poids p_i sont ajustés.
- La fonction de *combinaison* $\sum_i n_i p_i$ (produit scalaire) est suivie d'une fonction de *transfert* : souvent la sigmoïde $s(x) = 1/[1 + exp(-x)]$
- L'échantillon d'apprentissage est parcouru plusieurs fois. L'apprentissage s'achève lorsque
  - une solution optimale a été trouvée ou
  - un nombre fixé d'itérations a été atteint.
- L'apprentissage se fait en ajustant 1 à 1 chaque poids (rétropropagation), ou par modification aléatoire des poids suivie d'un mécanisme de sélection (algorithme génétique)

Principe du perceptron multicouches

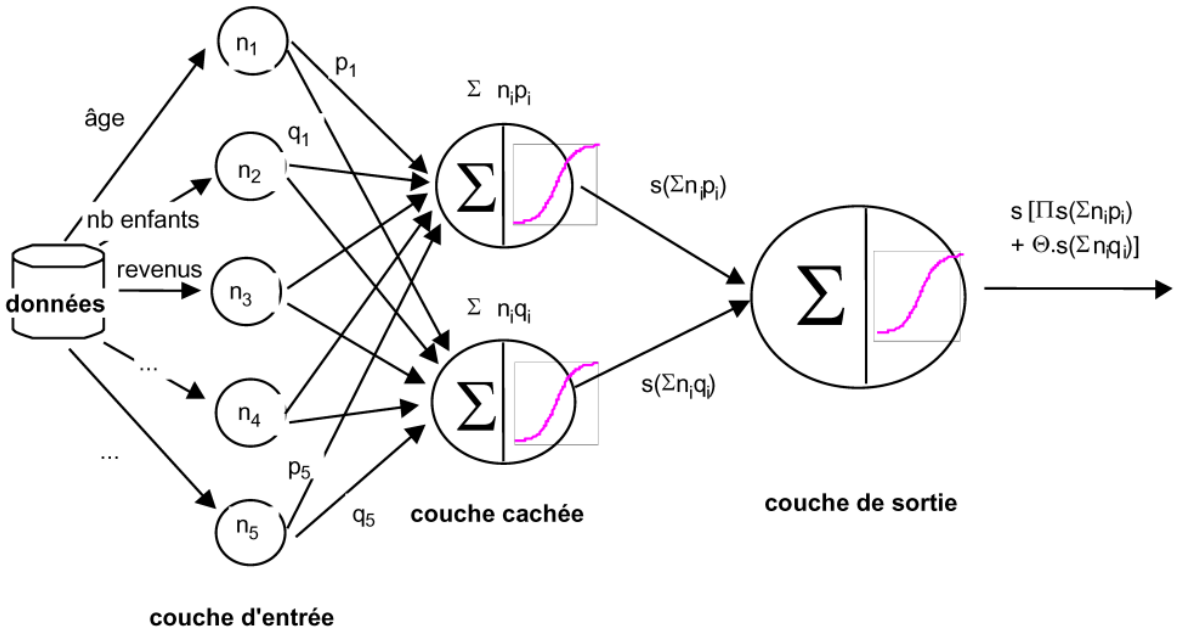
- Lors de l'**apprentissage du réseau de neurones**, pour chaque exemple présenté en entrée, la valeur renvoyée (" rétropropagée ") par le nœud de sortie est comparée à la valeur réelle, et les poids p_i sont ajustés.
- La fonction de *combinaison* $\sum_i n_i p_i$ (produit scalaire) est suivie d'une fonction de *transfert* : souvent la sigmoïde $s(x) = 1/[1 + exp(-x)]$
- L'échantillon d'apprentissage est parcouru plusieurs fois. L'apprentissage s'achève lorsque
  - une solution optimale a été trouvée ou
  - un nombre fixé d'itérations a été atteint.
- L'apprentissage se fait en ajustant 1 à 1 chaque poids (rétropropagation), ou par modification aléatoire des poids suivie d'un mécanisme de sélection (algorithme génétique)

La fonction logistique



Les réseaux à couches cachées (1/2)

- On augmente le pouvoir de prédiction en ajoutant une ou plusieurs couches cachées entre les couches d'entrée et de sortie



Les réseaux à couches cachées (2/2)

- Le pouvoir de prédiction augmente avec le nombre de nœuds des couches cachée
  - le nb de couches cachées est très généralement 1 ou 2
  - lorsque ce nombre = 0, le réseau effectue une régression linéaire ou logistique (selon la fonction de transfert)
- Mais ce dernier doit néanmoins être limité pour que le réseau de neurones ne se contente pas de mémoriser l'ensemble d'apprentissage mais puisse le généraliser
  - sinon, il y a sur-apprentissage
- Le fait que toutes les valeurs soient comprises entre 0 et 1 permet de prendre en entrée d'un nœud la sortie d'un nœud précédent
- Autre but de la normalisation des valeurs : éviter que les données avec de grandes valeurs « écrasent » les autres.

Les réseaux à couches cachées (2/2)

- Le pouvoir de prédiction augmente avec le nombre de nœuds des couches cachée
  - le nb de couches cachées est très généralement 1 ou 2
  - lorsque ce nombre = 0, le réseau effectue une régression linéaire ou logistique (selon la fonction de transfert)
- Mais ce dernier doit néanmoins être limité pour que le réseau de neurones ne se contente pas de mémoriser l'ensemble d'apprentissage mais puisse le généraliser
  - sinon, il y a sur-apprentissage
- Le fait que toutes les valeurs soient comprises entre 0 et 1 permet de prendre en entrée d'un nœud la sortie d'un nœud précédent
- Autre but de la normalisation des valeurs : éviter que les données avec de grandes valeurs « écrasent » les autres.

Les réseaux à couches cachées (2/2)

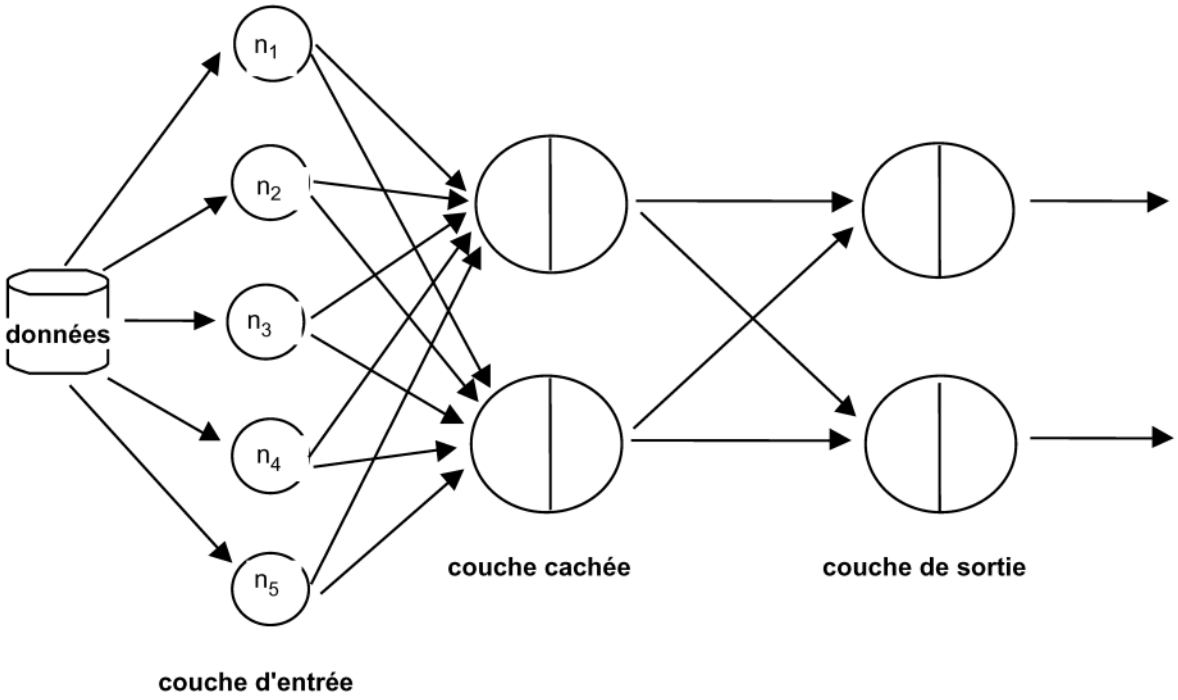
- Le pouvoir de prédiction augmente avec le nombre de nœuds des couches cachée
  - le nb de couches cachées est très généralement 1 ou 2
  - lorsque ce nombre = 0, le réseau effectue une régression linéaire ou logistique (selon la fonction de transfert)
- Mais ce dernier doit néanmoins être limité pour que le réseau de neurones ne se contente pas de mémoriser l'ensemble d'apprentissage mais puisse le généraliser
  - sinon, il y a sur-apprentissage
- Le fait que toutes les valeurs soient comprises entre 0 et 1 permet de prendre en entrée d'un nœud la sortie d'un nœud précédent
- Autre but de la normalisation des valeurs : éviter que les données avec de grandes valeurs « écrasent » les autres.

Les réseaux à couches cachées (2/2)

- Le pouvoir de prédiction augmente avec le nombre de nœuds des couches cachée
  - le nb de couches cachées est très généralement 1 ou 2
  - lorsque ce nombre = 0, le réseau effectue une régression linéaire ou logistique (selon la fonction de transfert)
- Mais ce dernier doit néanmoins être limité pour que le réseau de neurones ne se contente pas de mémoriser l'ensemble d'apprentissage mais puisse le généraliser
  - sinon, il y a sur-apprentissage
- Le fait que toutes les valeurs soient comprises entre 0 et 1 permet de prendre en entrée d'un nœud la sortie d'un nœud précédent
- Autre but de la normalisation des valeurs : éviter que les données avec de grandes valeurs « écrasent » les autres.

Les réseaux à plusieurs sorties

- La couche de sortie du réseau peut parfois avoir plusieurs nœuds, lorsqu'il y a plusieurs valeurs à prédire.



Différents réseaux de neurones

- Le **Perceptron multicouches** (PMC) est utilisé pour prédire une variable cible continue ou discrète
- Le **réseau à fonction radiale de base** (« radial basis function » RBF) est aussi utilisé pour prédire une variable cible continue ou discrète
- Le **réseau de Kohonen** effectue les analyses typologiques (clustering, recherche de segments)
- Réseaux par **estimation de densité de probabilité** de Speckt (1990)
  - PNN (*Probabilistic Neural Networks*) : classement
  - GRNN (*General Regression Neural Networks*) : régression
- DNN (*Deep Neural Netwokrs*): plusieurs couches cachées
  - RNN (*Recurrent Neural Networks*): traitement des langues
  - CNN (*Convolutional Neural Networks*): traitement de paroles, traitement d'images

Mise en œuvre d'un réseau

- Les étapes dans la mise en œuvre d'un réseau de neurones pour la prédiction ou le classement sont :
  - identification des données en entrée et en sortie
  - normalisation de ces données
  - constitution d'un réseau avec une topologie adaptée
  - apprentissage du réseau
  - test du réseau
  - application du modèle généré par l'apprentissage
  - dénormalisation des données en sortie.

Quelques règles empiriques

- Il faut 5 à 10 individus pour ajuster chaque poids
- On recommande d'avoir 1 couche cachée (réseau RBF) ou 1 à 2 (réseau PMC)
- Un réseau à n unités d'entrée, 1 couche cachée, m unités dans la couche cachée et 1 unité de sortie a $n.m + m$ poids
  - Il faut donc un échantillon d'au moins $5(n.m + m)$ individus pour l'apprentissage
- La valeur de m est généralement comprise en $n/2$ et $2n$
- On a l'intérêt à diminuer n (en utilisant l'ACP par exemple)
- Pour un classement, $m \geq$ nombre de classes
- L'échantillon d'apprentissage ne doit pas être trié selon un ordre significatif, qui pourrait induire le réseau en erreur
- L'échantillon d'apprentissage doit couvrir tous les cas.

Avantages des réseaux de neurones

- Aptitude à modéliser des structures complexes et des données irrégulières
  - prise en compte des relations non linéaires (interactions) entre les variables
- Assez bonne résistance aux données bruitées
- Aptitude à modéliser des problèmes très variés

Problèmes modélisés par les réseaux de neurones

- Analyse typologique
- Prédiction - classement
- Séries temporelles (prévision de cours boursiers)
- Reconnaissance de caractères optiques et de l'écriture manuscrite (sur des chèques, lettres, signatures)
- Reconnaissance/synthèse de la parole
- Jeu d'échecs (Deep Blue vainqueur de Kasparov en 1997)
- Reconnaissance des formes (prévention des pannes de machines par l'analyse de leurs vibrations)
- Reconnaissance des visages
- Analyse d'images (détecter si une gare est bondée)
- Traitement du signal

Inconvénients des réseaux de neurones

- Résultats totalement non explicites
  - rédhibitoire pour le diagnostic médical ou les pilotes automatiques d'avion
- Sensibilité aux individus hors norme
- Sensibilité à un trop grand nombre de variables non discriminantes
- Convergence vers la meilleure solution globale pas toujours assurée
- Difficulté d'utilisation correcte - paramètres nombreux et délicats à régler (nb et tailles des couches cachées, taux d'apprentissage, moment...)
- Ne s'appliquent naturellement qu'aux variables continues dans l'intervalle [0,1]
  - multiplication des nœuds pour les variables catégorielles

Quelques références pour l'apprentissage profond

http://eric.univ-lyon2.fr/ricco/cours/slides/reseaux_neurones_perceptron.pdf
https://medium.com/@CharlesCrouspeyre/comment-les-r%C3%A9seaux-de-neurones-%C3%A0-convolution-fonctionnent-b288519dbcf8
https://medium.com/@CharlesCrouspeyre/comment-les-r%C3%A9seaux-de-neurones-%C3%A0-convolution-fonctionnent-c25041d45921

Plan

Validation

Mesure des performances en généralisation

- La question de la validité des résultats se pose toujours que l'analyse soit menée à des fins explicatives ou à des fins décisionnelles
- Les calculs se font sur un échantillon, dit échantillon d'apprentissage
- La performance de l'outil discriminant est donnée par les taux de bien classés reposant sur les résultats de l'analyse, mas des taux de bien classés sur l'échantillon d'apprentissage ne suffisent pas pour conclure sur la qualité de la classification lorsque la règle sera appliquée à de nouveaux cas.

Validation croisée (rappel)

- La méthode des échantillons-tests consiste à effectuer l'analyse discriminante sur une partie de l'échantillon d'apprentissage (par ex. 80%) et de tester les règles de discrimination sur la partie restante
- La validation croisée consiste à effectuer n discriminations en exculant à chaque fois une observation, à affecter l'observation exclue, et à calculer le taux d'erreur moyen.
- La minimisation du taux d'erreur moyen peut être utilisée pour comparer plusieurs modèles de discrimination.

Qu'est-ce qu'un résultat valide (1/2)

- En structuration des données (classification non supervisée, ACP, AFC, etc.) :
  Validité = Stabilité du résultat
- Avec un autre échantillon de la population cible, la même méthode donne à peu près les mêmes résultats :
  - mêmes frontières et mêmes caractéristiques de classes,
  - mêmes axes factoriels, etc.

Qu'est-ce qu'un résultat valide (2/2)

- Pour un modèle de prédiction d'une variable catégorielle ou numérique (classification supervisée, régression, etc.) : Validité = Exactitude du résultat en généralisation
  - c'est-à-dire sur un autre échantillon de la population-cible.

Deux problèmes distincts

- Validation d'une méthode d'apprentissage (= d'une classe de modèles) pour un type de problèmes
  - Quelle est la meilleure méthode pour ce type de données ?
- Validation d'un modèle précis (y compris les estimiations des paramètres)
  - Quelle est la qualité de ce modèle pour ce type de données ?

Comment valider ?

Avec un grand ensemble de données

- Partition aléatoire de l'ensemble des données disponibles en
  - échantillon d'apprentissage (*training set*)
  - échantillon de validation (*validation set*)
  - échantillon-test (*test set*)

1) On estime les paramètres du modèle en l'aujustant sur l'échantillon d'apprentissage
2) On évalue l'erreur de prédiction sur l'échantillon de validation, et on modifie éventuellement un paramètre a priori $\rightarrow$ étape 1)
3) On teste le modèle final sur l'échantillon-test.

Validation pour un plus petit ensemble de données

- Validation croisée : quelle est la qualité de ctte méthode pour ce type de données
- L'échantillon disponible est divisé aléatoirement en K sous-échantillons de tailles à peu près égales
- On apprend sur l'ensemble de $(K-1)$ sous-échantillons, teste sur le dernier
  $\Rightarrow$ On obtient une évaluation E_i de l'erreur
- Recommence en changeant à chaque fois le sous-échantillon test
- L'erreur est estimé par $(1/K)\sum E_i$

Rééchantillonage

- Dans tous les cas, il faut ré-échantillonner en respectant la méthode d'échantillonnage (souvent impliciite) pour consituter la base de données disponible.
- Par exemple : il faut conserver la structure en grappes

Autres variantes de la validation croisée

- Leave-one-out : si les données disponibles sont peu nombreuses
- Bootstrap : On tire aléatoirement avec remise un échantillon d'apprentissage dans l'ensemble d'apprentissage. On effectue l'apprentissage sur l'échantillon et on teste sur les exemples non utilisés ; on obtient une valeur P_1 de l'erreur et on teste sur l'ensemble des exemples pour obtenir P_2. On répète ceci K fois.
  - L'estimation du taux d'erreur est donné par 0,636 * moyenne des P_1 + 0,368 * moyenne des P_2.

Plan

Qualités attendues d'une méthode (1/2)

- La précision
  - le taux d'erreur doit être le plus bas possible, et l'aire sous la courbe ROC la plus proche possible de 1
- La robustesse
  - être le moins sensible possible aux fluctuations aléatoires de certaines variables et aux valeurs manquantes
  - ne pas dépendre de l'échantillon d'apprentissage utilisé et bien se généraliser à d'autres échantillons
- La concision
  - les règles du modèle doivent être les plus simples et les moins nombreuses possible

Qualités attendues d'une méthode (2/2)

- Des résultats explicites
  - les règles du modèle doivent être accessibles et compréhensibles
- La diversité des types de données manipulées
  - tous les algorithmes ne sont pas aptes à manipuler les données qualitatives, discrètes, continues et manquantes
- La rapidité de calcul du modèle
  - un apprentissage trop long limite le nombre d'essais possibles
- Les possibilités de paramétrage
  - dans un classement, il est parfois intéressant de pouvoir pondérer les erreurs de classement, pour signifier, par exemple, qu'il est plus grave de classer un patient malade en « non-malade » que l'inverse

Choix d'une méthode : nature des données

- La régression linéaire traite les variables continues
- L'analyse discriminante traite les variables à expliquer nominales et les variables explicatives continues
- L'analyse discriminante DISQUAL traite les variables à expliquer nominales et les variables explicatives qualitatives
- La régression logistique traite les variables à expliquer qualitatives (nominales ou ordinales) et les variables explicatives continues ou qualitatives
- Les réseaux de neurones traitent les variables continues dans [0,1] et transforment les autres
- Certains arbres de décision (CHAID) traitent nativement les variables discrètes et qualitatives (et transforment les autres)
- CART, C5.0 peuvent aussi traiter les variables continues

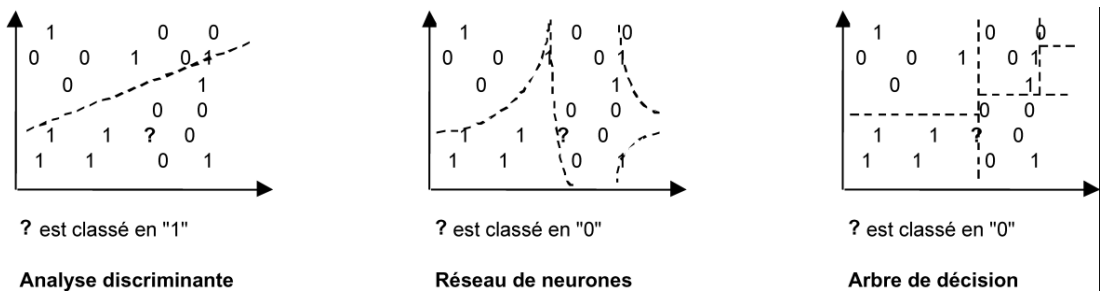
Choix d'une méthode : précision, robustesse, concision, lisibilité

- Précision : privilégier la régression linéaire, l'analyse discriminante et la régression logistique, et parfois les réseaux de neurones en prenant garde au surapprentissage (ne pas avoir trop de neurones dans la ou les couches cachées)
- Robustesse : éviter les arbres de décision et se méfier des réseaux de neurones, préférer une régression robuste à une régression linéaire par les moindres carrés
- Concision : privilégier la régression linéaire, l'analyse discriminante et la régression logistique, ainsi que les arbres sans trop de feuilles
- Lisibilité : préférer les arbres de décision et prohiber les réseaux de neurones. La régression logistique, DISQUAL, l'analyse discriminante linéaire et la régression linéaire fournissent aussi des modèles faciles à interpréter

Choix d'une méthode : autres critères

- Peu de données : éviter les arbres de décision et les réseaux de neurones
- Données avec des valeurs manquantes : essayer de recourir à un arbre, à une régression PLS, ou à une régression logistique en codant les valeurs manquantes comme une classe particulière
- Les valeurs extrêmes de variables continues n'affectent pas les arbres de décision, ni la régression logistique et DISQUAL quand les variables continues sont découpées en classes et les extrêmes placés dans 1 ou 2 classes
- Variables explicatives très nombreuses ou très corrélées : utiliser les arbres de décision ou la régression PLS
- Mauvaise compréhension de la structure des données : réseaux de neurones (sinon exploiter la compréhension des données par d'autres types de modèles)

Choix d'une méthode : topographie des classes à discriminer



- Toutes les méthodes inductives de classement découpent l'espace des variables en régions, dont chacune est associée à une des classes
- La forme de ces régions dépend de la méthode employée