Fouille de Données et Media Sociaux

Cours 1 – Méthodologie du traitement de données

Olivier Schwander <olivier.schwander@lip6.fr> (et Ludovic Denoyer avant)

Master DAC Data Science
UPMC - LIP6

Lundi 18 septembre 2017

Informations

Objectifs

- Appronfondir votre connaissance de la méthodologie “data-driven” pour le traitement de données
- Vous donner des notions de traitement de données “complexes” (graphes, réseaux sociaux)

Intervenants

- Olivier Schwander: Méthodologie
- Sylvain Lamprier: Media Sociaux et Graphes
- Laure Soulier: Conception de Modèles
- Industriels: Dataiku + Talend

Organisation du cours

1. **Méthodologie du traitement de données**
   - Sélection de modèles, de caractéristiques
   - Méthodes d'ensemble
   - Introduction à la visualisation
   - *Projet machine learning*: des données à la prédiction
2. **Présentation de la plateforme Dataiku**
3. **Etude de cas concret: la recommendation**
   - Modèles de filtrage collaboratif / Factorisation matricielle
   - Modèles mixtes de recommendation - texte + ratings
4. **Présentation de la plateforme Talend**
5. **Graphes et Media Sociaux**
   - Recommendation sociale, Classification Collective, Modèles publicitaires, Parallélisation de modèles
6. **Développons votre créativité...**
   - Bibliographie, création d'un "nouveau" modèle, restitution

Les différentes étapes

Quelles sont les différentes étapes effectuées par un système de fouille de données ?

Conception d'un système

Les questions à se poser en premier

- Quel type de données ?
- Quel type de tâche ?
- Quelle quantité de données ?
- Quelle qualité des données ?
- Quels objectifs ?

Ensuite

- Quel prétraitement des données ?
- Quelles méthodes ?
- Comment choisir les paramètres ?
- Comment les évaluer ?
- Comment présenter les résultats ?
- Comment les interpréter ?

Données et tâches

Types de données

- Vectorielles
- Temporelles
- Graphes
- Texte

Différentes tâches

- Classification
- Régression
- Détection d'évènements
- Segmentation
- Recherche d'information
- Recommandation

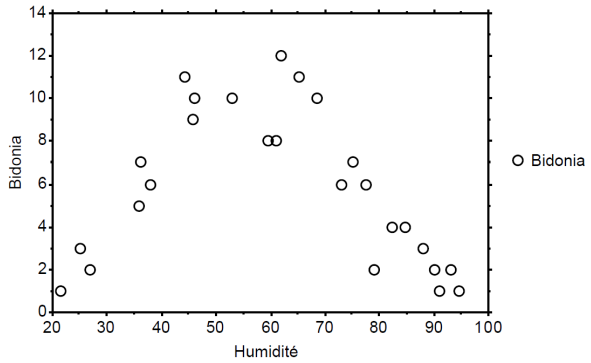
Chaîne de traitement des données

1. **Données**
   - Charger
   - Analyser
   - Transformer
2. **Méthodes**
   - Choisir
   - Paramétrer
   - Apprendre
3. **Évaluation**
   - Mesurer
   - Présenter
   - Interpréter

Exemple: la régression polynomiale

Collecte des données

- Acquisition de données **vectorielles** à l'aide de capteurs, wrappers, etc...
- Inputs= $\mathcal{X} = (x_1, ..., x_n) \in \mathbb{R}$
- Acquisition d'une **vérité terrain** - valeurs à prédire - sur $\mathcal{X}$
- $y_1, ..., y_n$ avec $y_i \in \mathbb{R}$



N.B: La collecte de données implique un nettoyage important des données, ainsi qu'une sélection/engineering des caractéristiques

Exemple: la régression polynomiale

Choix du modèle

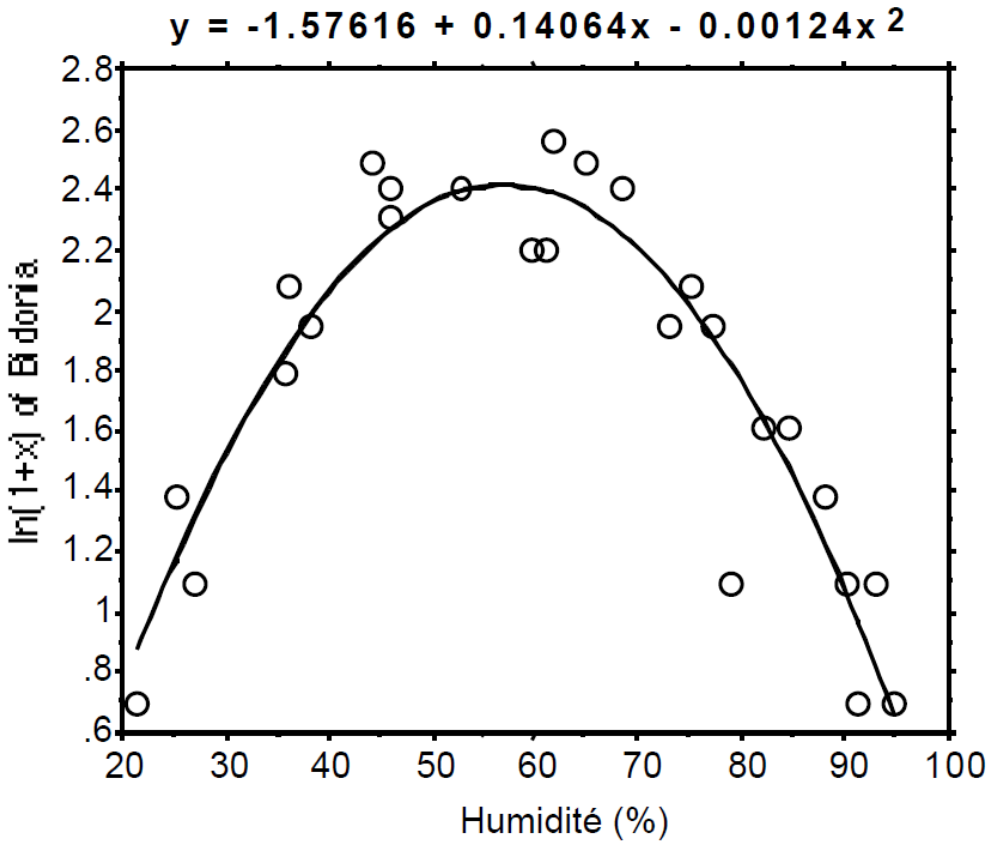
- Choisir un modèle paramétrique $f_\theta(x) \rightarrow y$
  - Catalogue: régression, arbres, forêts, réseaux de neurones, SVM, CRFs, HMMs, réseaux bayésiens, machine de Boltzmann, ...
  - Choix de la topologie/hyper-paramètres du modèle
    - Topologie du réseau de neurone, type de régressions, type d'arbres, liens entre variables, ...
- $\theta = (\theta_1, ..., \theta_P)$
- $P =$ degré du polynôme
- Modèle: $y = \sum_{k=1}^{P} \theta_k x^k$

Exemple: la régression polynomiale

Apprentissage du modèle

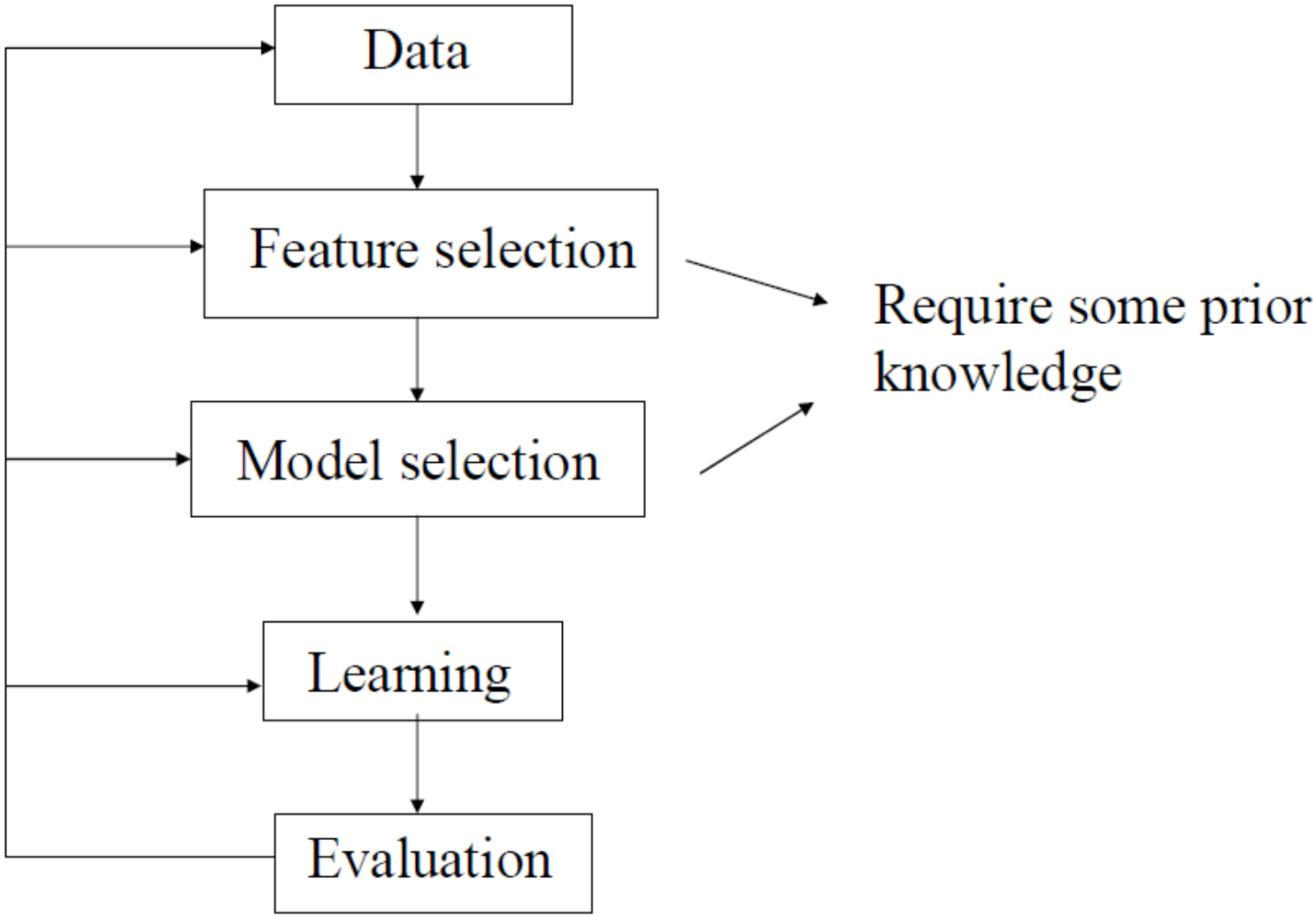
- Choix du critère d'apprentissage
  - Critère des moindres carrés : $\theta^* = argmin_\theta \sum_i^n (x_i - f_\theta(x_i))^2$
- Choix de l'algorithme d'optimisation
  - Descente de gradient, SMO, résolution analytique, ...
  - $\Rightarrow$ choix des paramètres de l'algorithme d'optimisation
- Optimisation effective des paramètres $\Rightarrow \theta^* \Rightarrow f_\theta^*$

Exemple: la régression polynomiale



- Le modèle est prêt à être utilisé sur de nouvelles données
- Mais, beaucoup de choses à choisir. Comment faire ?

Concevoir un modèle



Quelques mots sur le nettoyage de données

Acquisition des données

- A travers des “capteurs”
  - Capteurs réels, logiciels, connecteurs
  - Nécessite une connaissance experte
- A travers l’usage de systèmes d’ETL
- ...

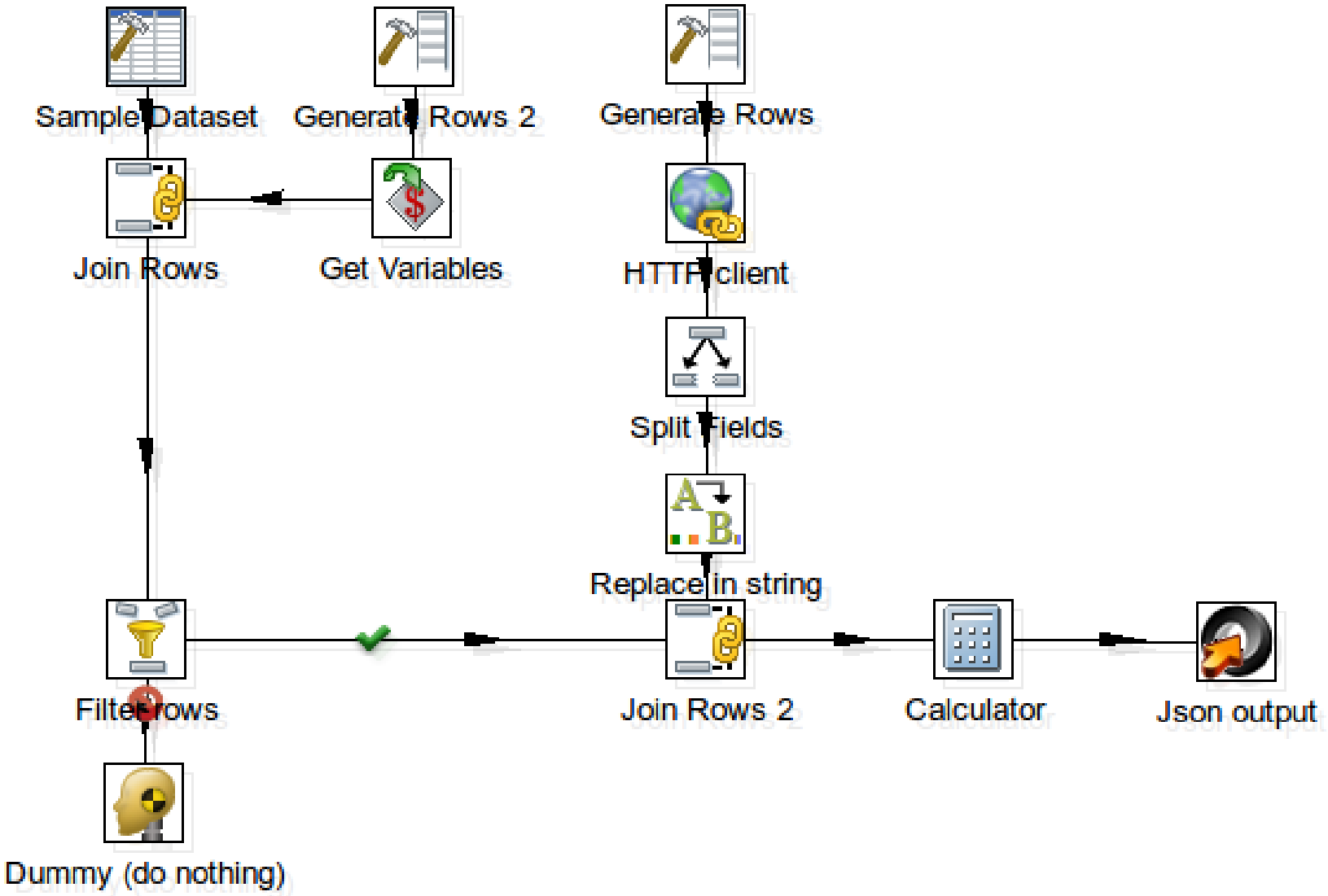
Petit retour en arrière: Intégration de données

Définition

Il existe plusieurs système d'intégration de données :

- La médiation au service de l'intégration de données d'entreprise (EII).
- L'intégration de données via les applications (EAI).
- L'intégration de données via les services Web (ESB, SOA).
- L'intégration de données en nuage (Data Cloud).
- L'ETL (Extract - Transform - Load)

Petit retour en arrière: Intégration de données



Quelques mots sur le nettoyage de données

Acquisition des données

- A travers des “capteurs”
  - Capteurs réels, logiciels, wrappers
  - Nécessite une connaissance experte
- A travers l'usage de systèmes d'ETL
- A travers l'usage de systèmes d'apprentissage
  - **Apprendre à acquérir de l'information**
  - Apprendre à dialoguer
  - Apprendre à voir

Quelques mots sur le nettoyage de données

Acquisition des données ...

Preprocessing

- Renommage
- Normalisation
- Discrétisation
- Abstraction
- Aggrégation
- *Sélection d'attributs - Features sélection*
- Création d'attributs

Quelques mots sur le nettoyage de données

Acquisition des données ...

Preprocessing ...

Biais dans les données

- Nécessité de comprendre la source des données sous peine d'obtenir des résultats inattendus
- Les résultats obtenus à partir de données pré-sélectionnées sont rarement les meilleurs ! Attention à l'intuition !!

Features Selection/Examples Selection

Features Selection

- Classification de texte: un document = un vecteur fréquentiel de mots
- Quel est le problème ?

Features Selection/Examples Selection

Features Selection

- Classification de texte: un document = un vecteur fréquentiel de mots
- Plusieurs millions de termes possibles $\Rightarrow f_\theta : \mathbb{R}^{1000000} \rightarrow \mathbb{R}$
- *Fléau de la dimension*

Features Selection/Examples Selection

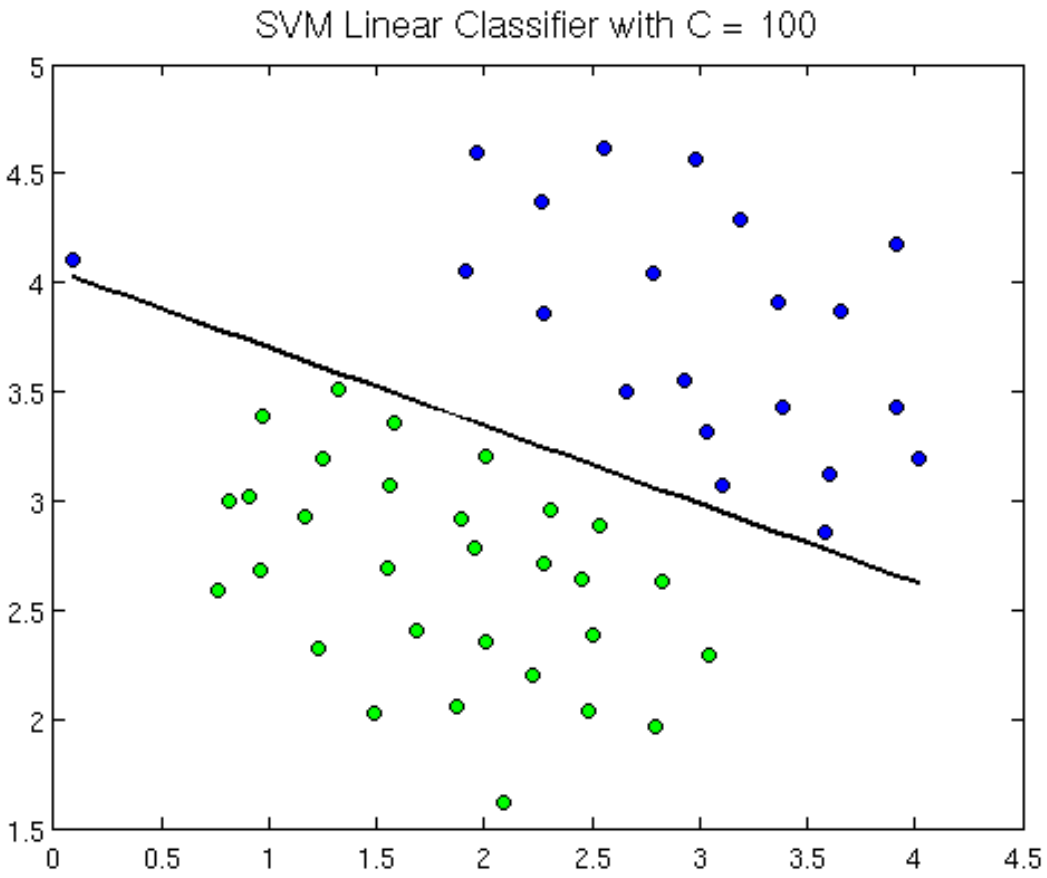
Features Selection

- Classification de texte: un document = un vecteur fréquentiel de mots
- Plusieurs millions de termes possibles $\Rightarrow f_\theta : \mathbb{R}^{1000000} \rightarrow \mathbb{R}$
- *Fléau de la dimension*
- *Mais aussi : plusieurs millions de paramètres à apprendre...*

Solutions

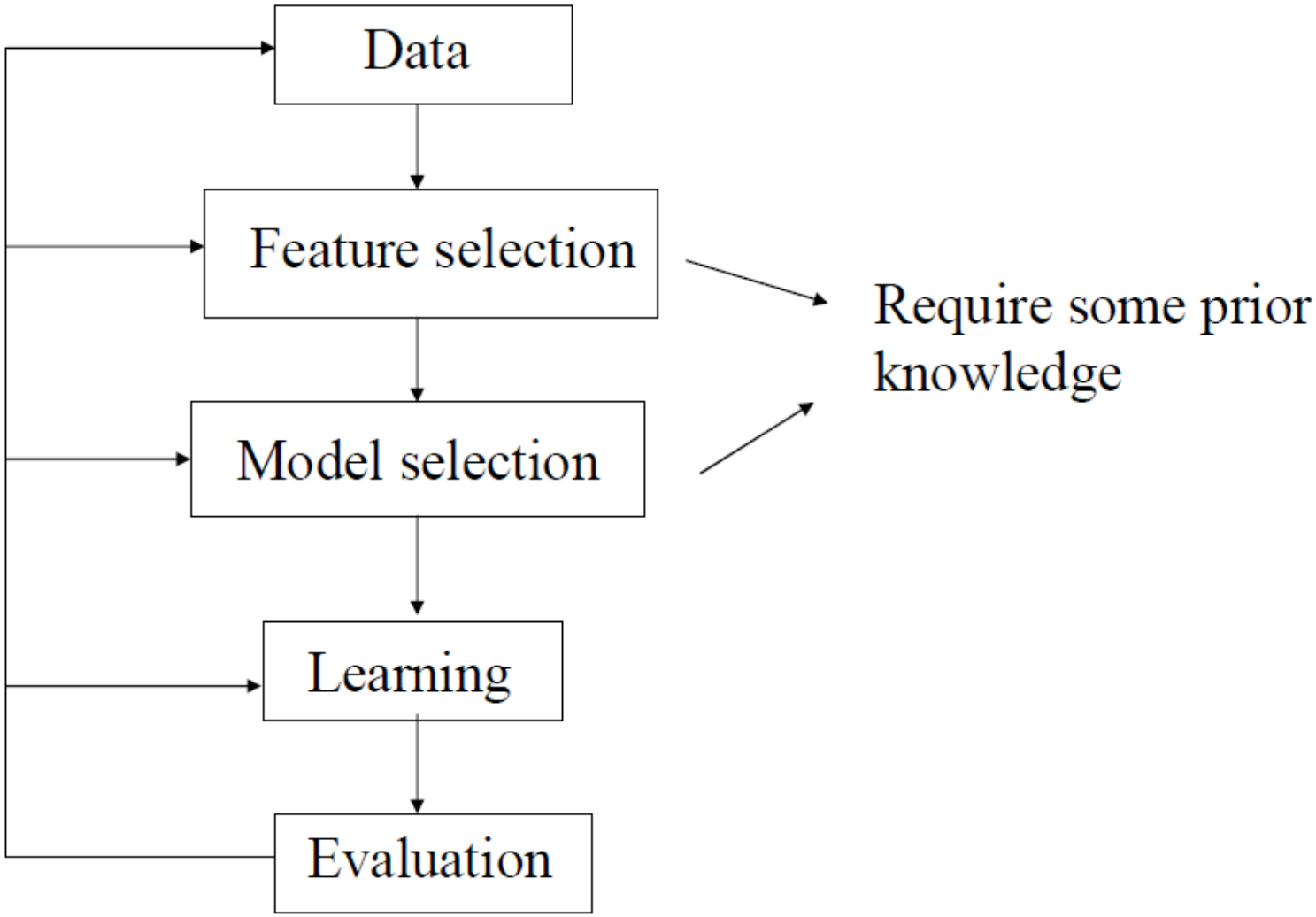
- Réduire la dimension
- Transformation manuelle (expert)
- Apprendre la transformation

Outliers

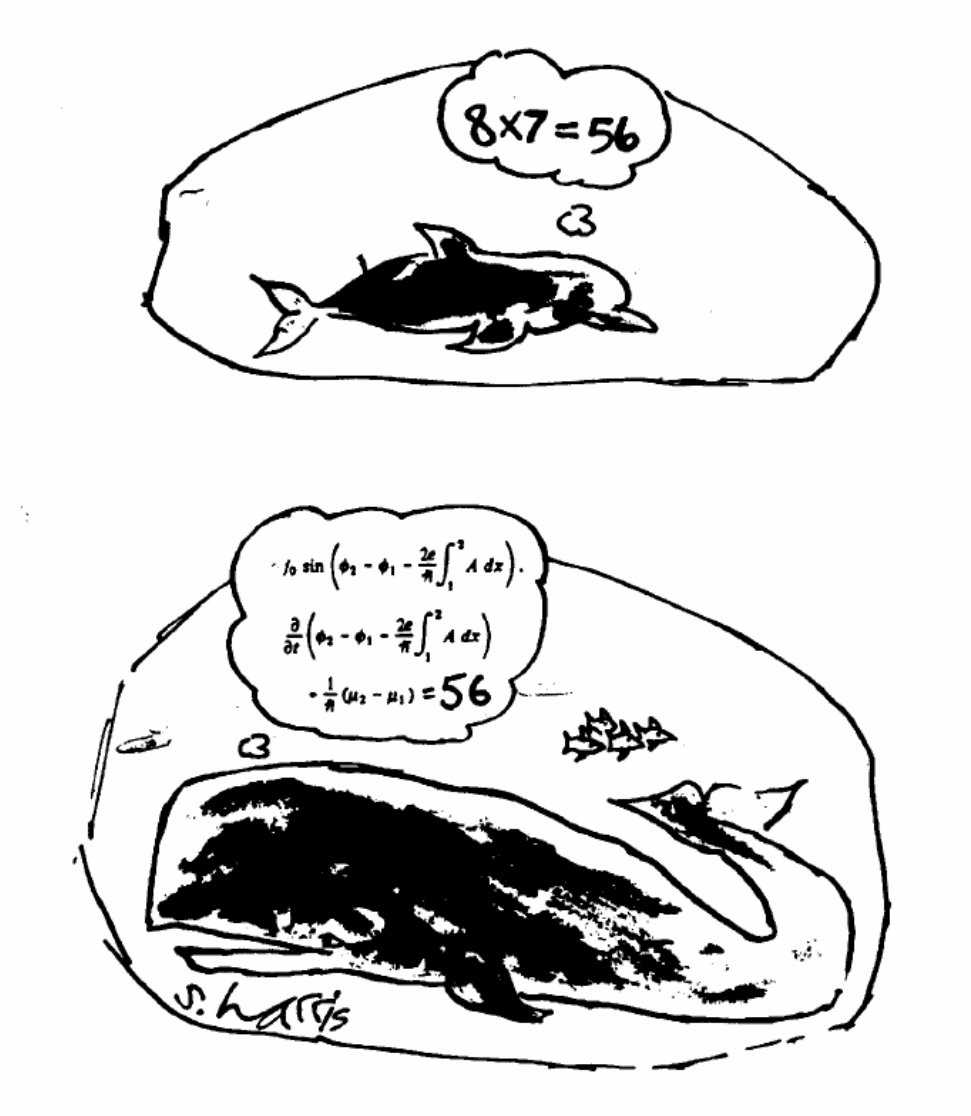


- Il faut supprimer les outliers...
- ...mais ça n'est pas simple
- **Besoin des connaissances d'un expert**

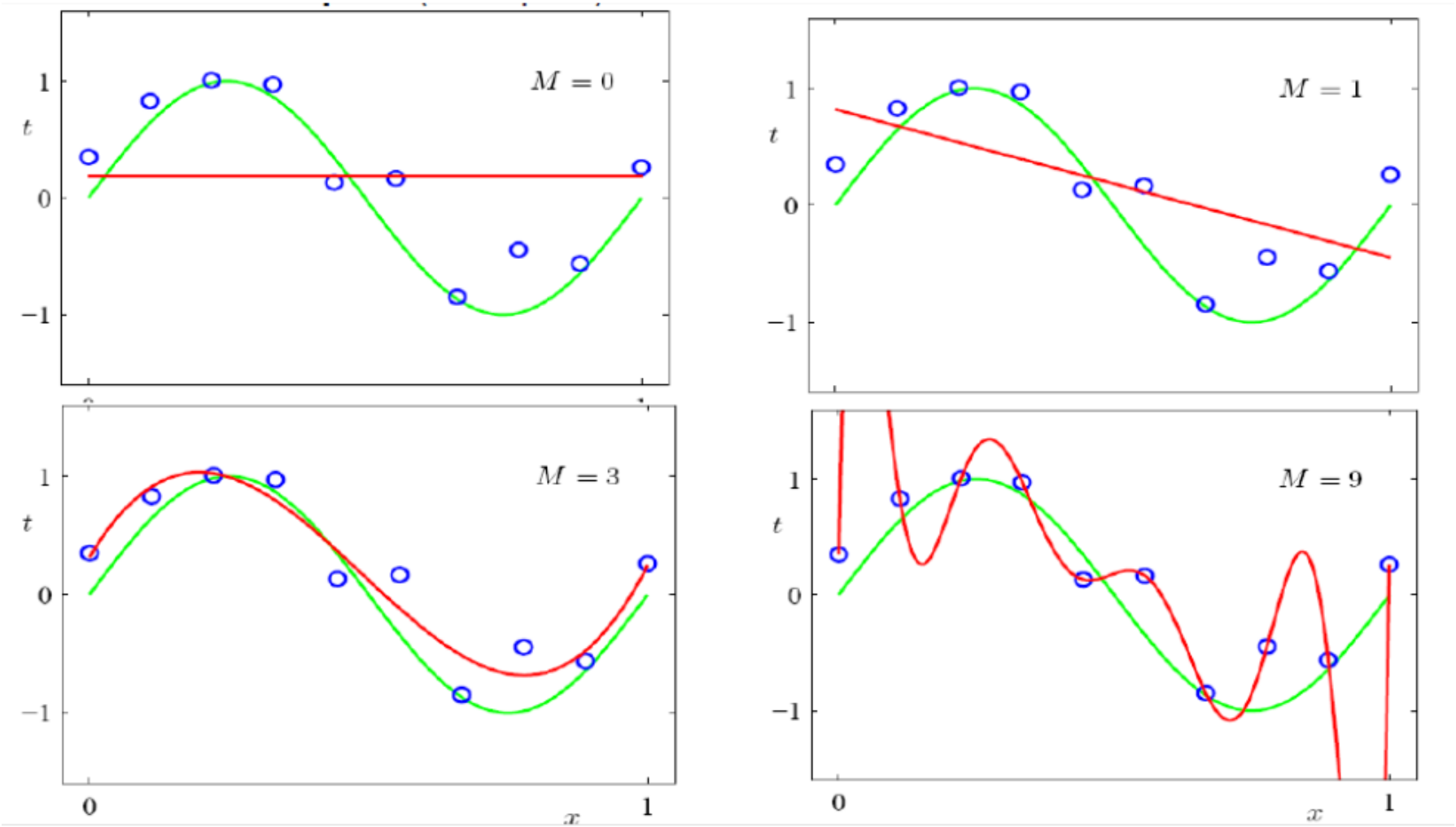
Designer un modèle



Sélection de modèles

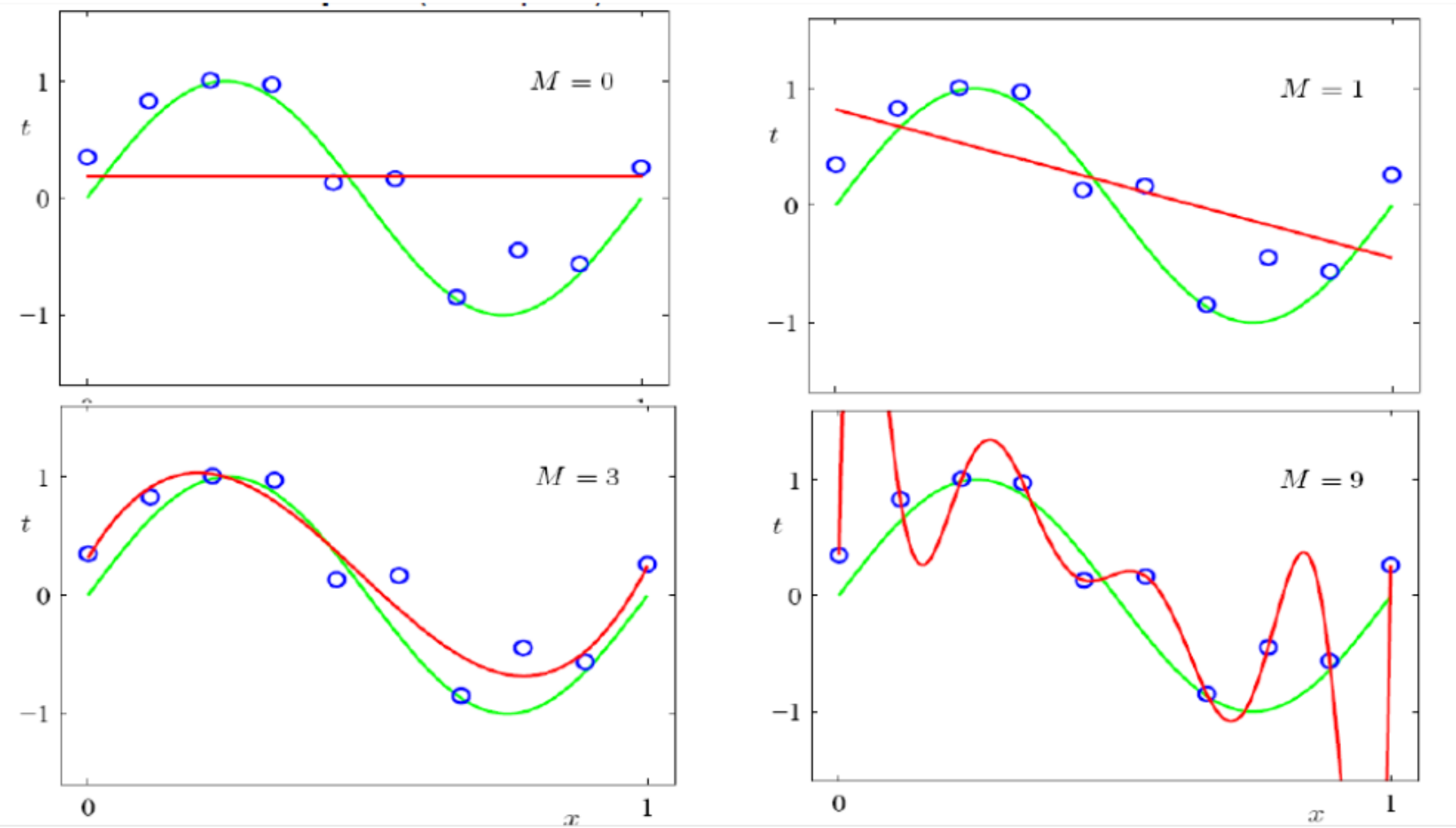


Sélection de modèles



Quel est le meilleur modèle ?

Sélection de modèles



Conclusion: On ne doit pas choisir le modèle qui correspond le mieux aux données, mais celui qui **généralise** le mieux

Sélection de modèles

On cherche des moyens de sélectionner le “meilleur” modèle parmi un ensemble de modèles possibles

Bruit et Régularités **Données** = **Bruit** + **Régularités**

- Bruit: Erreurs dans l'acquisition
- Régularités: Processus de génération sous jacent

Objectif: **Modèle final** = **Capture du bruit** + **Modèle des régularités**

Meilleur modèle:

- Meilleur modèle des régularité
- Meilleure capture du bruit

Sur-apprentissage / Overfitting

Sélection de modèles par échantillonage

Deux grandes familles de méthodes pour se faire une idée de l'erreur de généralisation..

- La loi des grands nombres: l'utilisation de bornes statistiques permettant de borner la différence entre l'erreur empirique et l'erreur théorique (sous certaines hypothèses)

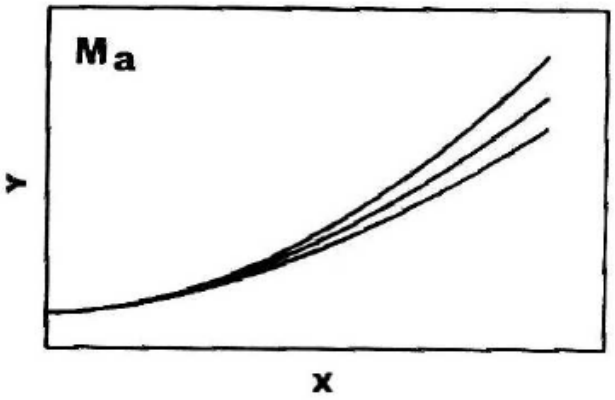
$$\forall f \in \mathcal{F}, \quad \mathcal{R}_P(f) \leqslant \widehat{\mathcal{R}}_n(f) + \frac{1}{\sqrt{2n}} \sqrt{\ln(2) \underbrace{|f|_\pi}_{\text{complexite}} + \ln \frac{1}{\delta}}.$$

- L'utilisation d'échantillons différents pour l'évaluation de l'erreur

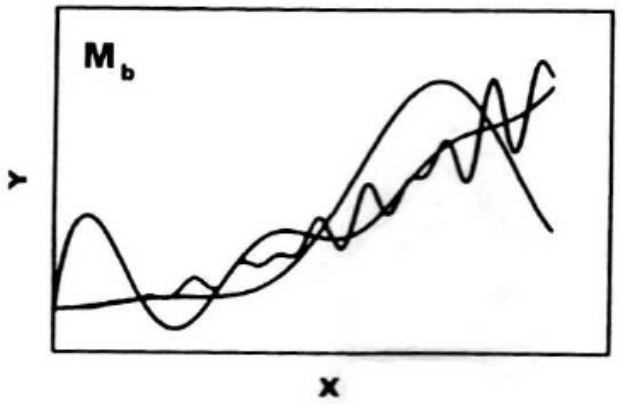
Sur-apprentissage

Quand est-ce qu'un modèle sur-apprend ?

Simple Model



Complex Model

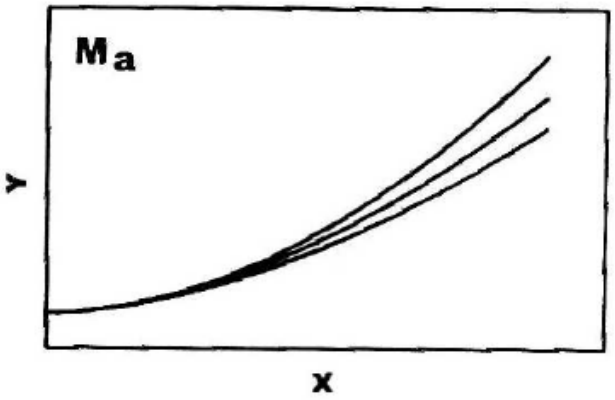


La complexité d'un modèle est liée au nombre de ses paramètres, et à la complexité sous-jacente de la classe de fonction choisie.

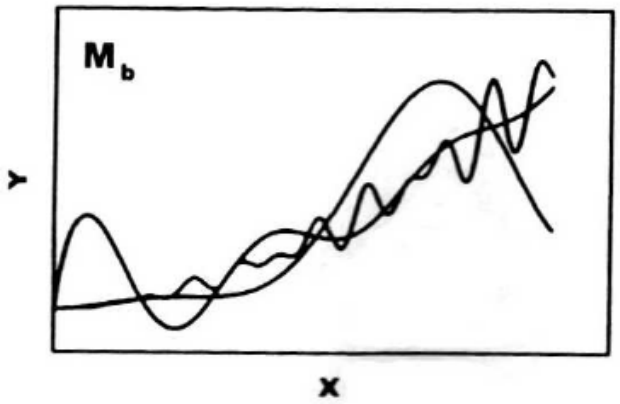
Sur-apprentissage

Quand est-ce qu'un modèle sur-apprend ?

Simple Model



Complex Model



La complexité d'un modèle est liée au nombre de ses paramètres, et à la complexité sous-jacente de la classe de fonction choisie.

Critère d'information d'Akaike - 1973

$$AIC = -2 \ln \hat{L} + 2k$$

- $\hat{L}$ est la vraisemblance du modèle sur les données $= P(x|\theta^*, f)$
- k est le nombre de paramètres du modèle

Méthodologie

- Entraîner plusieurs modèles
- Calculer leur AIC
- Prendre le modèle avec le meilleur AIC (le plus élevé)

Critère d'information d'Akaike - 1973

Divergence de Kullback-Leibler (KL)

- On suppose que les données sont générées par un processus p
- Soit des modèles f_i
- $KL(p||f_i)$ mesure l'information perdue en approchant p par f_i
- Le meilleur modèle est celui qui minimise cette divergence
- **Problème:** on ne connait pas p

Estimateur **asymptotique**

- l'AIC permet de comparer des modèles

Variante pour petits jeux de données:

- $AICc = AIC + \frac{2k(k+1)}{n-k-1}$

Autres critères

- Critère d'information Bayésien - 1978: $BIC = -2 \ln \hat{L} + k \ln n$
- Minumum Description Length - 1978: *learning as data compression*

Principe général à retenir: rasoir d'Occam

- *Pluralitas non est ponenda sine necessitate*
- *Les multiples ne doivent pas être utilisés sans nécessité*
- Sélectionner le modèle le plus simple qui modélise les données *suffisamment* bien

Sélection de modèles par échantillonage

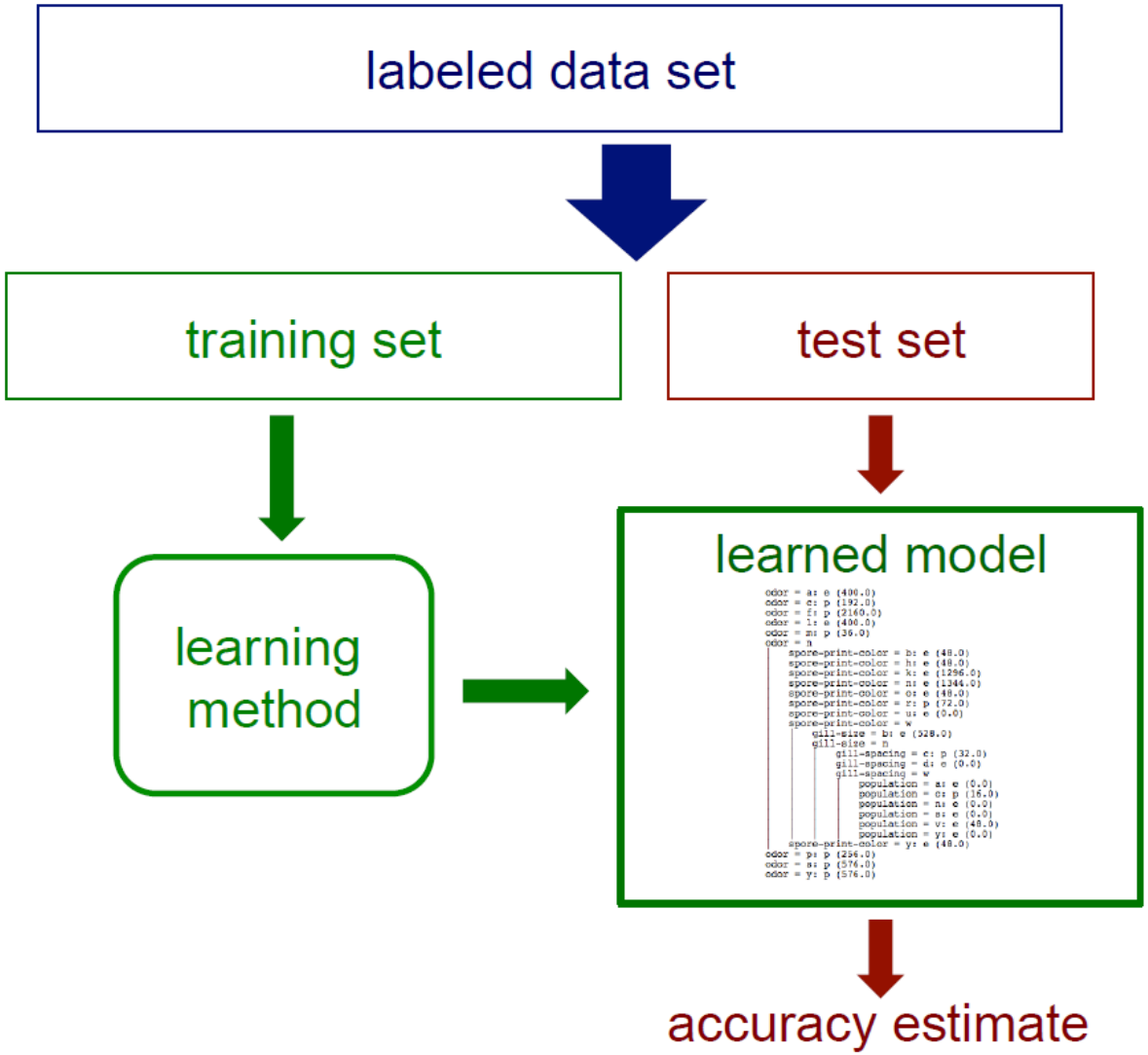
Deux grandes familles de méthodes pour se faire une idée de l'erreur de généralisation..

- La loi des grands nombres: l'utilisation de bornes statistiques permettant de borner la différence entre l'erreur empirique et l'erreur théorique (sous certaines hypothèses)

$$\forall f \in \mathcal{F}, \quad \mathcal{R}_P(f) \leqslant \widehat{\mathcal{R}}_n(f) + \frac{1}{\sqrt{2n}} \sqrt{\ln(2) \underbrace{|f|_\pi}_{\text{complexite}} + \ln \frac{1}{\delta}}.$$

- L'utilisation d'échantillons différents pour l'évaluation de l'erreur

Evaluation en Machine Learning



Sélection de modèles par échantillonage

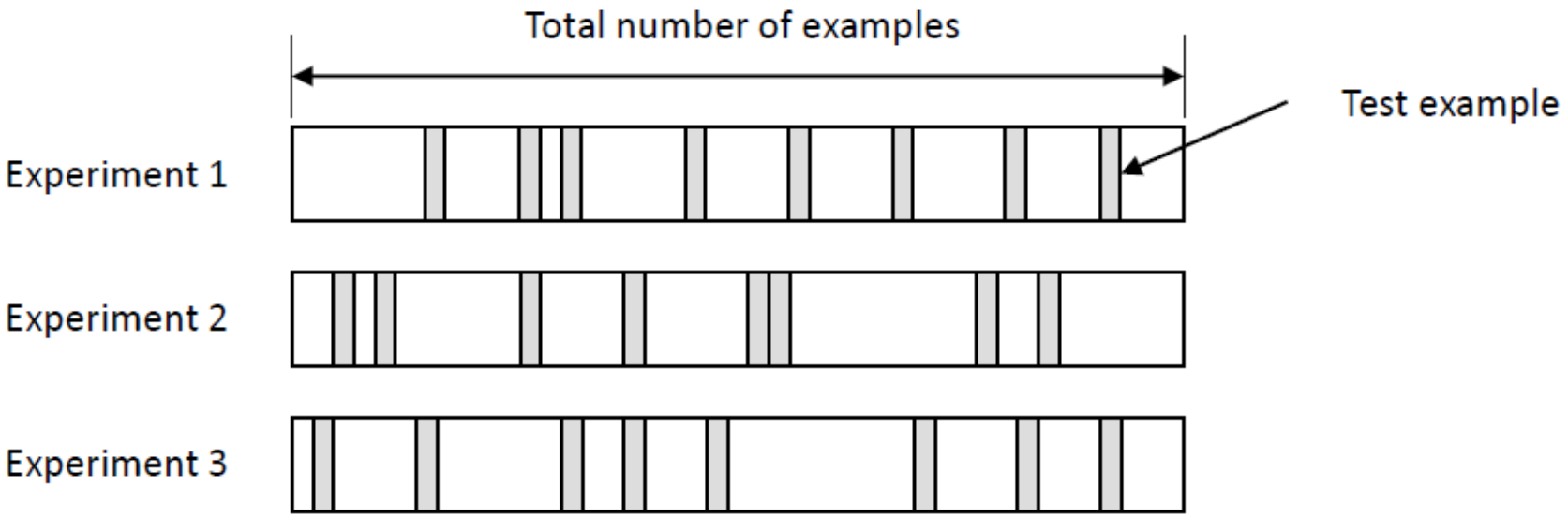
Problèmes

- As-t-on assez de données pour consituter ces différents ensembles ?
- L'utilisation d'un unique ensemble d'apprentissage ne nous permet pas de savoir si le modèle est sensibles aux données d'apprentissage

Plusieurs solutions:

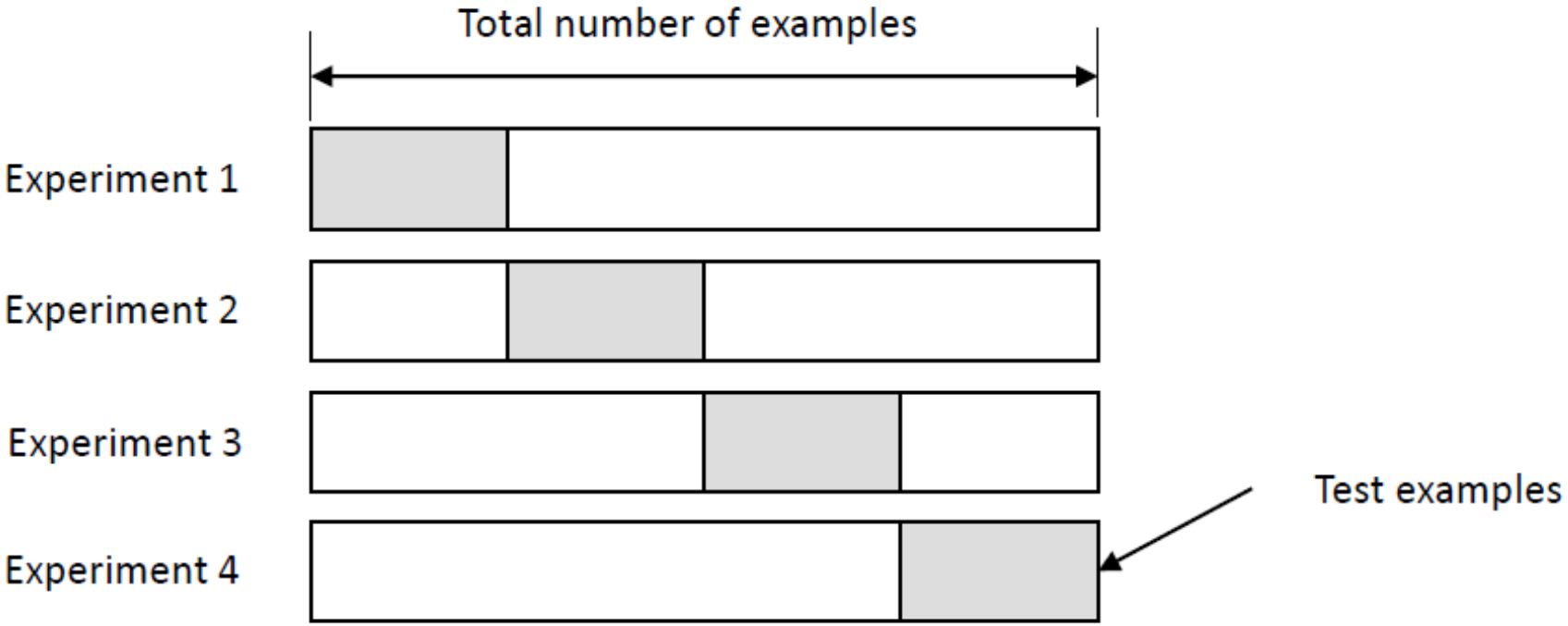
- Rééchantillonage aléatoire
- Cross-Validation

Rééchantillonage aléatoire



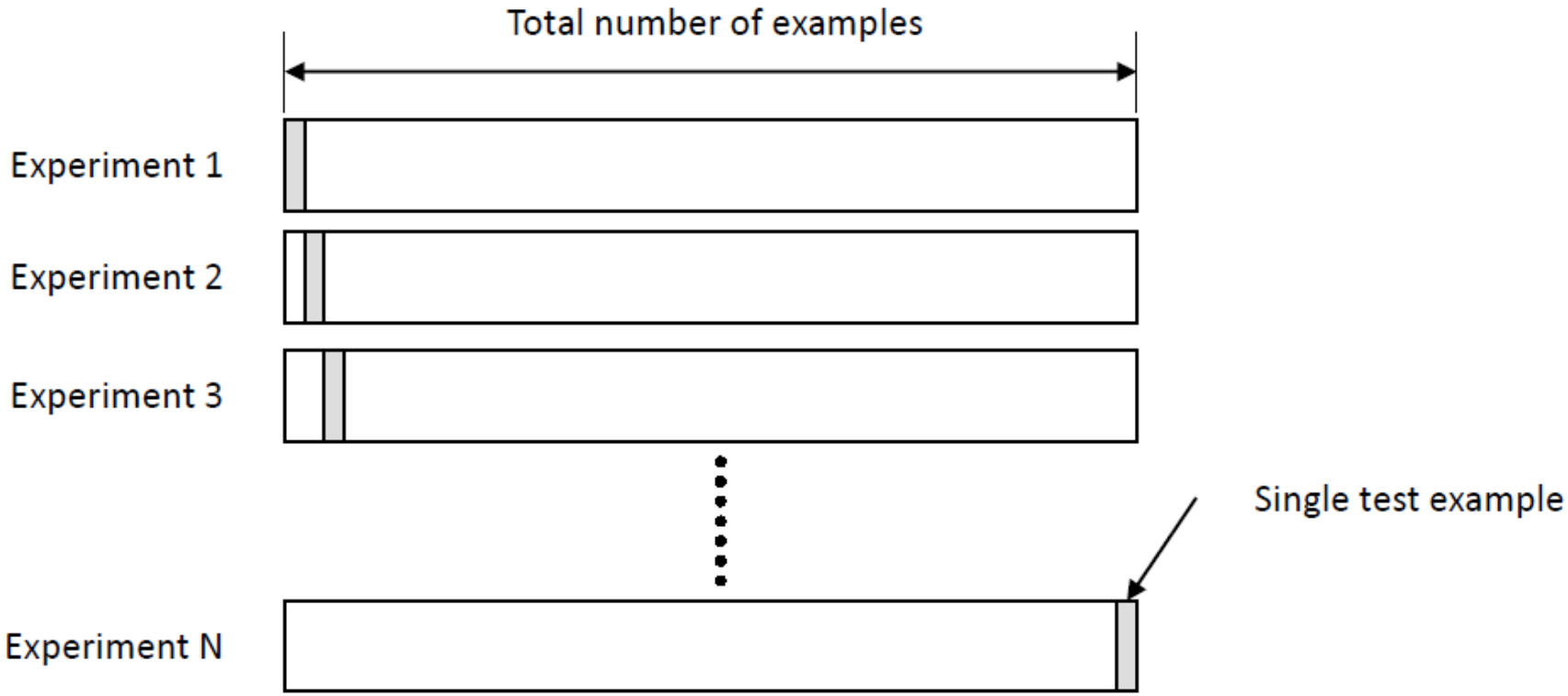
- L'estimation de l'erreur du modèle est obtenue en moyennant les erreurs obtenus sur les différentes expériences
- Cette estimation est significativement meilleure que celle obtenue précédemment, si le nombre d'expériences est suffisant

Cross-Validation



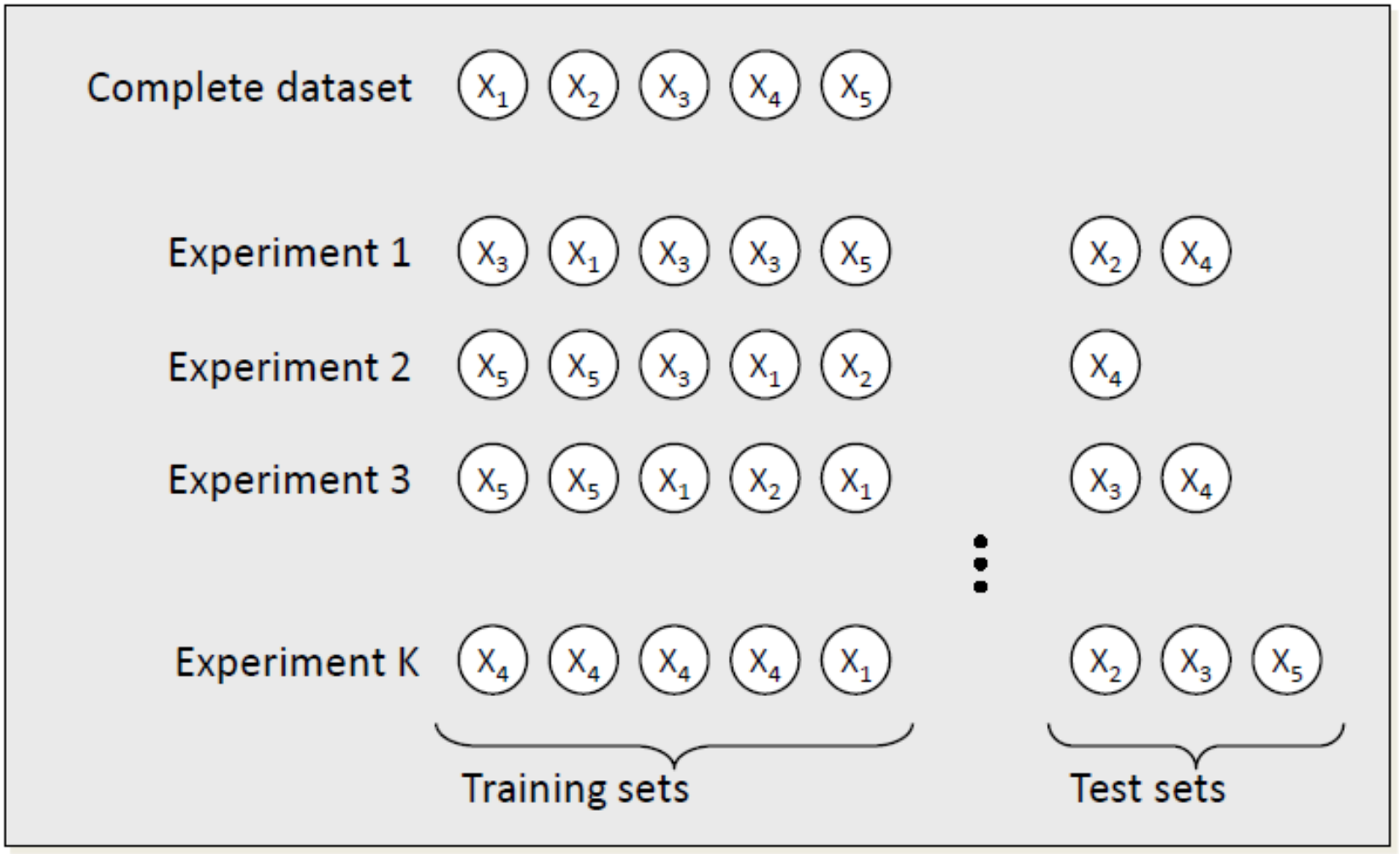
- L'estimation de l'erreur du modèle est obtenue en moyennant les erreurs obtenus sur les différentes expériences
- Tous les exemples sont utilisés pour apprendre au moins un modèle

Leave-one-out



- L'estimation de l'erreur du modèle est obtenue en moyennant les erreurs obtenus sur les différentes expériences
- Cas dégénéré de CV $\rightarrow$ plus robuste, meilleurs pour les petits jeux de données

Bootstrap



- Plus grande variance dans les différents “folds”
- Mais effet désirable car plus réaliste (c.f classification)

Train/Test/Validation

On considère le cas particulier où l'on veut **à la fois** trouver le meilleur modèle **mais aussi** estimer sa performance.

Solution Il faut découper en trois:

- Train set
- Validation set : pour découvrir le meilleur modèle
- Test set : pour évaluer la performance

Courbes d'apprentissage

Conclusion

Protocole expérimental classique:

- Diviser les données en trois ensembles
- Entrainer un modèle sur *train*
- Evaluer le modèle sur *validation*
- Recommencer jusqu'à obtenir le meilleur modèle et les meilleurs hyper-paramètres
- Evaluer la qualité finale du modèle sur l'ensemble de test

Sources:

- CSCE 666 - Ricardo Guttierez-Osuna - CSE@TAMU
- CS2750 - Milos Hauskrecht - University of Pittsburgh
- http://:www.biostat.wisc.edu/~dpage/cs760/