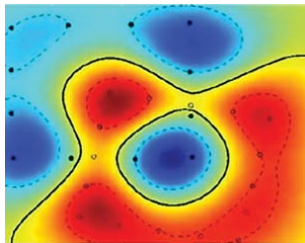
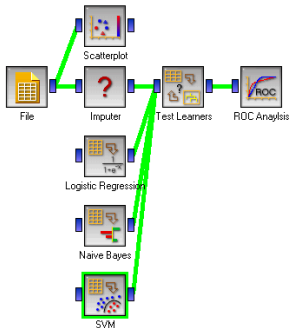


Statistical Machine Learning



- **Stéphan Cléménçon** (Telecom ParisTech - Département TSI)
 - ▶ Contact : `stephan.clemencon@telecom-paristech.fr`
Bureau : E308
 - ▶ Profil : Enseignement/Recherche/Conseil/Industrie
 - ▶ Mots-clés : processus stochastiques (markoviens, empiriques, etc.), apprentissage statistique, applications : finance, high tech, biosciences
- **Joseph Salmon** (Telecom ParisTech - Département TSI)
 - ▶ Contact : `joseph.salmon@telecom-paristech.fr`
 - ▶ Profil : Enseignement/Recherche
 - ▶ Mots-clés : régression, agrégation, traitement de l'image

Intervenants (2)

- **Florence D'Alché-Buc** (Telecom ParisTech - Département TSI)

- ▶ Contact : `florence.dalche@telecom-paristech.fr`
- ▶ Profil : Enseignement/Recherche
- ▶ Mots-clés : Noyaux, modèles graphiques, données structurées, bioinformatique

- **Slim ESSID** (Telecom ParisTech - Département TSI)

- ▶ Contact : `slim.essid@telecom-paristech.fr`
- ▶ Profil : Enseignement/Recherche
- ▶ Mots-clés : traitement de l'image, video, audio

Intervenants (3)

- **Chloé Clavel** (Telecom ParisTech - Département TSI)
 - ▶ Contact : `chloe.clavel@telecom-paristech.fr`
 - ▶ Profil : Enseignement/Recherche/Industrie
 - ▶ Mots-clés : text-mining, affective computing
- **Laurence Likforman** (Telecom ParisTech - Département TSI)
 - ▶ Contact : `laurence.likforman@telecom-paristech.fr`
 - ▶ Profil : Enseignement/Recherche
 - ▶ Mots-clés : manuscript character recognition, deep learning
- **TD** : Anna Korba, Mastane Achab, Pierre Laforgue, Moussab Djerrab, Valentin Barrière, Eugène Ndiaye, Umut Simsekli
- **Challenge** : Alexandre Gramfort, Umut Simsekli

- **Travaux Pratiques** : 2 Compte-rendus à rendre au chargé de TP une semaine après la session (8pts)
- **Challenge** (7pts)
- **Contrôle de connaissances** (5pts)

Data mining << Machine-Learning



Motivations pour le machine-learning

- Explosion des capacités de stockage
- Bases de données **massives**
 - ▶ finance, génomique, marketing, industrie ...
- Les données sont partout !
 - ▶ de grande dimension, hétérogènes, (non) structurées
 - ▶ prétraitement et modélisation a priori de la variabilité **impossibles** !
- Il existe des approches génériques et **automatisables**

Les données aujourd'hui

Les chiffres du travail (1)



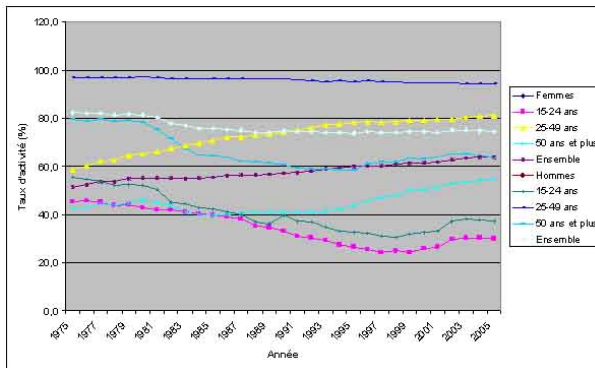
Les chiffres du travail (2)

Taux d'activité par tranche d'âge hommes vs. femmes

	A	B	C	D	E	F	G	H	I
1									
2	Taux d'activité par tranche d'âge de 1975 à 2005								
3	En %								
4		1975	1976	1977	1978	1979	1980	1981	1982
5	Femmes								
6	15-24 ans	45,5	45,7	45,2	43,9	44,2	42,9	42,1	41,87
7	25-49 ans	58,6	60,3	62,1	62,8	64,7	65,4	66,2	67,55
8	50 ans et plus	42,9	43,1	44,4	43,9	44,8	45,9	45,2	43,47
9	Ensemble	51,5	52,5	53,6	53,6	54,8	55,1	55,1	55,29
10	Hommes								
11	15-24 ans	55,6	54,7	53,7	52,2	52,5	52,0	50,4	45,02
12	25-49 ans	97,0	97,1	96,9	96,9	96,9	97,1	96,9	96,75
13	50 ans et plus	79,5	78,8	79,5	78,8	79,4	78,3	75,4	71,65
14	Ensemble	82,5	82,2	82,1	81,6	81,8	81,5	80,4	78,14

Les chiffres du travail (3)

Taux d'activité par tranche d'âge hommes vs. femmes



Le monde de la finance (1)



Wall Street à la clotûre, un lundi...

Le monde de la finance (2)

DOW JONES INDUSTRIAL AVERAGE IN (DJI: ^DJI)

Dern. Cours:	13.820,19
Heure:	21 sept.
Variation:	↑ 53,49 (0,39%)
Clôture Préc.:	13.766,70
Ouverture:	13.768,33
Var. Journalière:	13.768,25 - 13.877,17
Var. sur 1 an:	11.926,80 - 14.121,00
Volume:	419.389.397



L'imagerie médicale (1)



L'imagerie médicale (2)



Internet (1)



Internet (2)

Netscape Proxy format:

```
format=%Ses->client.ip% 146.127.62.22 %Req->vars.pauth-user% [%SYSDATE%] "%Req->reqpb.proxy-request%  
%Req->srvhdrs.clf-status% %Req->vars.p2c-cl% %Req->vars.remote-status% %Req->vars.r2p-cl%  
%Req->headers.content-length% %Req->vars.p2r-cl% %Req->vars.c2p-hl% %Req->vars.p2c-hl%  
%Req->vars.p2r-hl% %Req->vars.r2p-hl% %Req->vars.xfer-time% %Req->vars.actual-route%  
%Req->vars.cli-status% %Req->vars.svr-status% %Req->vars.cch-status%  
146.127.123.16 146.127.62.22 - [10/Dec/1997:00:30:09 -0500] "GET http://www.nba.com/bulls/ HTTP/1.0" 200 881  
200 8816 - - 321 164 359 164 1 SOCKS(146.127.11.3:1080) FIN FIN NON-CACHEABLE  
146.127.253.84 146.127.62.22 - [10/Dec/1997:00:30:12 -0500] "GET http://www.pathfinder.com/NY1/bug.html  
HTTP/1.0" 200 377 200 377 - - 392 203 418 203 1 SOCKS(146.127.11.3:1080) FIN FIN REFRESHED  
146.127.253.84 146.127.62.22 - [10/Dec/1997:00:30:12 -0500] "GET  
http://www.pathfinder.com/NY1/images/steel.gif HTTP/1.0" 304 - 304 - - - 443 142 468 142 0  
SOCKS(146.127.11.3:1080) FIN FIN UP-TO-DATE
```

Séquençage du génome humain (1)



Plate-forme de séquençage génotypage OUEST-genopole

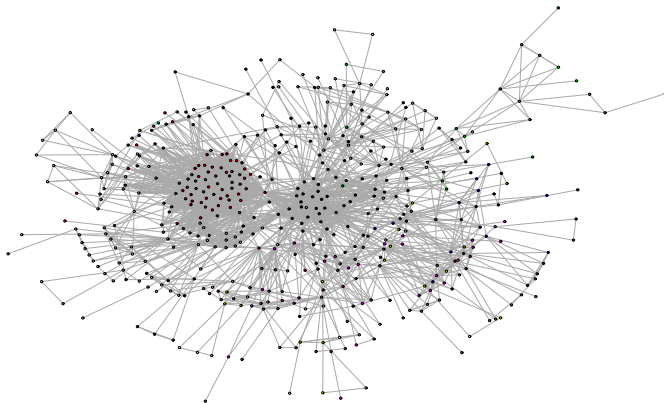
Barcoding Of Life Data Systems (1)



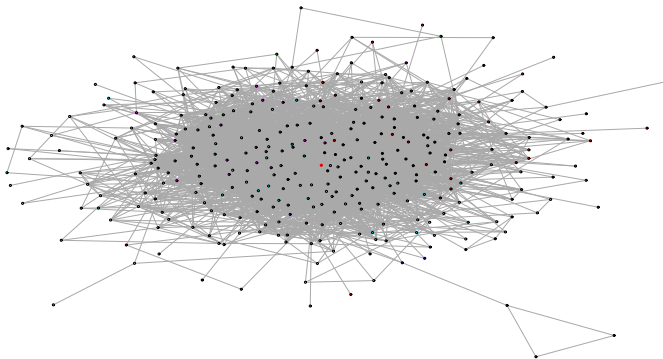
Barcoding of Life Data Systems (2)



E-marketing (1)- Livres



E-marketing (1)- Jeux vidéos



Nature des données

- Vecteurs/Matrices (e.g. image pixelisée)
- Chaînes de caractères (e.g. texte)
- Graphes/Réseaux
- Fonctions/Séries temporelles (e.g. vidéo, audio)

Les questions de machine learning

- Prédiction
- Segmentation/Clustering
- Détection d'anomalies
- Réduction de la dimension
- Sélection de variables
- Interprétation/Parcimonie
- Visualisation

Les outils



- Informatique :

- ▶ BDD
- ▶ algorithmique

- **Machine Learning :**

- ▶ méthodes effectives pour la grande dimension/volumétrie

- **Mathématiques :**

- ▶ algèbre linéaire
- ▶ **modélisation aléatoire**,
- ▶ probabilités / **statistique**
- ▶ **optimisation**
- ▶ traitement du signal

- **Introduction**

- Contexte - Repères historiques

- Éléments de statistique (Rappels)

- Nomenclature des problèmes rencontrés

- Applications (exemples)

- **Un peu de théorie : classification binaire**

- Le principe de la minimisation du risque empirique

- Théorie de Vapnik-Chervonenkis (complexité combinatoire)

- Une solution statistique.... un problème informatique !

- **Sélection de Modèle - Evaluation du Risque**

- Planification expérimentale - Validation Croisée - Minimisation

- Structurelle du Risque - Bootstrap

- **Algorithmes de Classification**

Moyennes locales, arbres de classification

Perceptron, SVM et réseaux de neurones

Méthodes ensemblistes : bagging, boosting et forêts aléatoires

Classification multi-classe

- **Algorithmes de Régression**

Adapter les algorithmes de classification à la régression (arbres de régression, *etc.*)

Régression linéaire en grande dimension : le Lasso

- **Autres Problèmes Supervisés**

Régression ordinale, ranking

- **Apprentissage non supervisé**

Clustering

Réduction de la dimension

- **Text Mining et applications à l'Opinion Mining**

Données textuelles : pré-traitements, analyse linguistique et apprentissage

Application à l'analyse des opinions pour des applications commerciales

- **Optimisation, gradient stochastique**
- **Deep Learning**
- **Analyse et prédiction dans les grands graphes**
 - Motivation, exemples de réseaux
 - Analyse de graphes : connectivité, coefficient de clustering, etc.
 - Algorithmes de prédiction : détection de communautés, prédiction de lien, propagation de label
- **Apprentissage par renforcement**
 - Q Learning
 - Bandits stochastiques

- **Modèles graphiques du diagnostic à la prédiction structurée**
- Passage à l'échelle
- Détection d'anomalie
- Apprentissage semi-supervisé

Cours de statistique "typique"

- Estimation paramétrique
- Intervalles/Domaines de confiance
- Tests d'hypothèses
- Régression
- Analyse en composantes principales

Aspects non abordés

- Classification, "distribution-free" régression
- Méthodes **non-paramétriques**
- Performances non-asymptotiques
- Sélection de modèle
- Théorie de la décision
- Optimisation
- Contraintes de calcul (temps quasi-réel, stockage et calcul distribués)

Pourquoi faire appel à l'apprentissage statistique ?

- Typologie des problèmes
- No Free Lunch !
- Choix des critères de performance
- Notion de risque
- Contrôle de la complexité
- Validation des règles de décision
- Rôle du rééchantillonnage
- Monitoring des modèles de prévision
- Intégration des contraintes computationnelles

Machine Learning... plus que des stats

- Méthodes non-paramétriques opérationnelles
- Traitement de données massives / complexes / de grande dimension
- Diversité des contextes
 - ▶ supervisé, non-supervisé, semi-supervisé, séquentiel, one-pass, multi-tâche, distribué...
- Couplage des principes inférentiels avec des algorithmes !

Machine Learning - Repères Historiques

- 1943 : Modèle neuronal artificiel - McCullough, Pitts
- 1958 : Perceptron Monocouche - Rosenblatt
- 60's : Data-mining - John Tukey
- 1971 : Loi Uniforme des Grands Nombres - Vapnik, Chervonenkis
- 1974, 1986 : Algorithme de rétropropagation et réseaux de Neurones
- 80's : CART (Breiman, Friedman, Stone, Olshen), SVM
- 90's : Algorithmes : Noyaux, Boosting, Forêts Aléatoires, *etc.*
- 1995 : Théorie de l'Apprentissage Statistique - Vapnik
- 2000's : Web data, moteurs de recherche/recommandation, publicité en ligne, *etc.*
- 10's : Renaissance des réseaux de neurones (GPU), Deep Learning

- Livres :

- ▶ The Nature of Statistical Learning Theory (2000) - Springer par V. Vapnik
- ▶ An Introduction to Statistical Learning with Applications in R (2013) - Springer par G. James, T. Hastie, R. Tibshirani, D. Witten
- ▶ The Elements of Statistical Learning (2001) - Springer par T. Hastie, R. Tibshirani, J. Friedman
- ▶ Principles and Theory for Data Mining and Machine Learning (2009) - Springer par B. Clarke, E. Fokoue et H. Zhang

- Article :

- ▶ "The curse and blessings of dimensionality" D. Donoho - IMS

- Bibliothèques au niveau de l'état de l'art :
 - ▶ **Machine Learning in Python, Scikit-Learn**
<http://scikit-learn.org/stable/>
 - ▶ **The R Project for Statistical Computing**
<http://cran.r-project.org/web/views/MachineLearning.html>
- **Large-Scale Machine Learning** : MLlib, la bibliothèque machine-learning de Spark Apache
<http://spark.apache.org/mllib/>
- Autres applications/frameworks (logiciels libres) :
 - ▶ WEKA
 - ▶ Orange
 - ▶ RapidMiner

Machine-Learning : les acteurs

- Monde académique :

- ▶ Départements : Maths (Appli), Informatique, Bioinformatique, etc.
Un savoir fondamental selon le panorama dressé par Carnegie Mellon (création du 1er Master en ML en 2000)
- ▶ Journaux : JMLR, Machine Learning, Data-Mining and Knowledge Discovery, etc.
- ▶ Conférences : NIPS, ICML, COLT, UAI, etc.

- Industrie :

- ▶ High-tech : web (google, AWS, Facebook, IBM, etc.), aéronautique
- ▶ Infrastructures (e.g. General Electric)
- ▶ Santé, médecine (personnalisée)
- ▶ Finance
- ▶ Traitement du signal, de l'image ou de la parole
- ▶ ...

Rappels de statistique

Modèle statistique

- Observation comme réalisation de X variable aléatoire de loi inconnue P^*
- On suppose X à valeurs dans (E, \mathbb{E})
- Modèle statistique = triplet $\mathcal{M} = (E, \mathbb{E}, \mathcal{P})$
où $\mathcal{P} = \{P_\theta : \theta \in \Theta\}$ famille de lois candidates pour P^*
- Θ est un paramétrage de \mathcal{P} , on note $P^* = P_{\theta^*}$
- Le modèle est paramétrique si Θ est un sev d'un espace euclidien
- Le modèle est dit non-paramétrique sinon ($\dim \infty$).
- Modèle identifiable : $\theta \mapsto P_\theta$ est injective

Vraisemblance du paramètre

- On représente \mathcal{P} par la classe des densités associées

$$\{f(x, \theta) : \theta \in \Theta\}$$

- Vraisemblance : pour x fixé,

$$L_x(\theta) = f(x, \theta) .$$

- Exemple : $X = (X_1, \dots, X_n)$ i.i.d. de loi de Bernoulli $\mathcal{B}(\theta)$

$$L(\theta) = \prod_{i=1}^n (\theta^{X_i} (1 - \theta)^{1-X_i}) = \theta^{S_n} (1 - \theta)^{n-S_n}$$

$$\text{où } S_n = \sum_{i=1}^n X_i.$$

Notion de statistique

- Soit X une observation/ un échantillon. Une **statistique** est une fonction mesurable $T : E \rightarrow \mathbb{R}^k$ de X . On dira que $T(X)$ ou $T(X_1, \dots, X_n)$ est une statistique de l'échantillon.
- Exemple : Moyenne empirique

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i$$

- Exemple : Variance empirique

$$s^2 = \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2$$

- Estimation de paramètres $g(\theta^*)$

- Exemple d'estimateur = Maximum de vraisemblance
- Dans le modèle de Bernoulli $\mathcal{B}(\theta)$ avec $\theta \in [0, 1]$:

$$\hat{\theta}_n = \bar{X}$$

- Risque quadratique et décomposition biais-variance :

$$\begin{aligned} R(\hat{\theta}_n, \theta^*) &= \mathbb{E}_{\theta^*} \left((\hat{\theta}_n - \theta^*)^2 \right) \\ &= \left(\mathbb{E}(\hat{\theta}_n) - \theta^* \right)^2 + \mathbb{V}_{\theta^*}(\hat{\theta}_n) = \frac{\theta^*(1 - \theta^*)}{n} \leq \frac{1}{4n} \end{aligned}$$

- Propriétés : consistance, normalité asymptotique (vitesse)
- Et si $\theta^* \notin \Theta$? Et si le modèle est faux ?

Intervalle de confiance - paramètre d'une Bernoulli

- Intervalle aléatoire $I(n, \alpha)$ t.q. $P(\theta^* \in I(n, \alpha)) \geq 1 - \alpha$
- Par l'inégalité de Bienaymé-Tchebychev :

$$I(n, \alpha) = \left[\bar{X} - \frac{1}{\sqrt{4n\alpha}}, \bar{X} + \frac{1}{\sqrt{4n\alpha}} \right] .$$

- Par l'inégalité de Hoeffding :

$$I(n, \alpha) = \left[\bar{X} - \sqrt{\frac{\log(2/\alpha)}{2n}}, \bar{X} + \sqrt{\frac{\log(2/\alpha)}{2n}} \right] .$$

- Par la loi limite (Φ fdr de la loi $\mathcal{N}(0, 1)$) : $I_\infty(n, \alpha) =$

$$\left[\bar{X} - \Phi^{-1}\left(1 - \frac{\alpha}{2}\right) \sqrt{\frac{\bar{X}(1 - \bar{X})}{n}}, \bar{X} + \Phi^{-1}\left(1 - \frac{\alpha}{2}\right) \sqrt{\frac{\bar{X}(1 - \bar{X})}{n}} \right]$$

- Modèle linéaire gaussien

$$\mathbf{Y} = \mathbf{X}\beta + \epsilon .$$

où $\mathbf{Y} \in \mathbb{R}^n$, $\mathbf{X} \in \mathbb{R}^{n \times p}$ sont les données
et $\beta \in \mathbb{R}^p$, $\epsilon \sim \mathcal{N}_n(0, \sigma^2 I_n)$

- On suppose : $\mathbf{X}^T \mathbf{X}$ inversible (identifiabilité)
- Estimateur des moindres carrés :

$$\hat{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}$$

$$\hat{s}^2 = \frac{1}{n} \|\mathbf{Y} - \mathbf{X}\hat{\beta}\|^2$$

Questions autour de l'estimateur des moindres carrés

Problèmes :

- Précision de la prédiction : biais faible - grande variance
- Interprétabilité si p est grand

Solutions :

- Réduction de la dimension de la matrice \mathbf{X}
- Méthodes pénalisées ("shrinkage")
- Estimation vs. Prédiction

Machine-Learning : les problèmes statistiques revisités

Cadre générique - apprentissage supervisé

- Couple de v.a. $= (X, Y) \sim P$ inconnue
- X = vecteur d'entrée à valeurs dans $\mathcal{X}(\mathbb{R}^d)$, ici $d \gg 1$
- Y = label/étiquette dans $\mathcal{Y} \subset \mathbb{R}$
- A priori, X modélise une information utile pour prédire Y
Règle prédictive : $g : \mathcal{X} \rightarrow \mathcal{Y}$ choisie dans une classe \mathcal{G}
(e.g. prédicteur linéaire $g(x) = {}^t \beta x + \alpha$)
- Fonction de perte : $\ell : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}^+$
- Risque (inconnu !) = Erreur de généralisation

$$L(g) = \mathbb{E}(\ell(Y, g(X)))$$

à minimiser sur $g \in \mathcal{G}$.

- Données $= D_n = \{(X_1, Y_1), \dots, (X_n, Y_n)\} \stackrel{i.i.d.}{\sim} P$

Exemple 1 - Régression

- Exemple : Prédiction de ventes
- \mathcal{X} = vecteur de descripteurs (information financière, indicateurs macro-économiques, ...)
- $\mathcal{Y} = \mathbb{R}$
- Fonction de perte = erreur quadratique

$$\ell(y, z) = (y - z)^2$$

- Solution optimale : $g^*(x) = \mathbb{E}(Y \mid X = x)$

Exemple 2 - Scoring

- Données de classification : $\mathcal{Y} = \{0, 1\}$ (e.g. frauduleux vs intègre)
- Probabilité a posteriori
 $\eta(x) = \mathbb{E}(Y \mid X = x) = \mathbb{P}\{Y = 1 \mid X = x\}$
- Régression Logistique

$$f(x) = \log \left(\frac{\eta(x)}{1 - \eta(x)} \right)$$

- Modélisation additive : $f(x) = \beta x + \alpha$
→ Régression logistique linéaire

Exemple 3 - Classification binaire

- Exemple : Prédiction de l'état d'un système (normal vs anormal)
- $\mathcal{Y} = \{-1, +1\}$
- Fonction de perte :

$$\ell(y, z) = \mathbb{I}\{y \neq z\}$$

- Risque d'erreur :

$$\begin{aligned} L(g) &= \mathbb{P}\{Y \neq g(X)\} \\ &= \mathbb{P}\{Y \cdot g(X) < 0\} = \mathbb{E}(\mathbb{I}_{\mathbb{R}^+}(-Y \cdot g(X))) \end{aligned}$$

Exemple 4 - Classification multi-classe

- Exemple : reconnaissance d'un caractère manuscrit

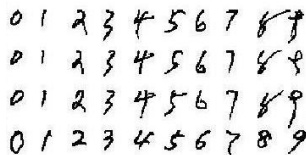
- $\mathcal{Y} = \{1, \dots, M\}$

- Fonction de perte

$$\ell(y, z) = \mathbb{I}\{y \neq z\}$$

- En pratique :

- ▶ Un contre Tous
- ▶ Un contre Un

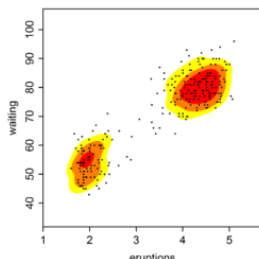


Exemple 5 - Régression ordinale

- Exemple : design d'un moteur de recherche
- label ordinal $\mathcal{Y} = \{1, \dots, M\}$
(e.g. "mauvais" vs "moyen" vs "bon")
- Fonction de perte
$$\ell(y, z) = (y - z)^2$$
- En pratique : régression + arrondi

Apprentissage non supervisé

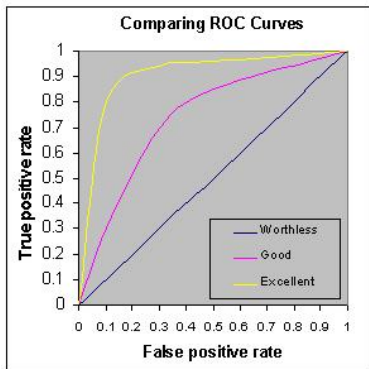
- Pas d'étiquette Y
- Modèle statistique (non-) paramétrique : $\{p(x, \theta) : \theta \in \Theta\}$
- Recouvrir la densité $f(x)$ de X à partir de $D_n = \{X_1, \dots, X_n\}$
- Fonction de perte :
$$\ell(x, \theta) = -\log p(x, \theta)$$
- Applications : clustering, modes vs. détection d'anomalie
- Sous-problème : estimation des ensembles de niveau de la densité



Exemple 7 - Ranking et scoring

- Données de classification binaire
- Set $\eta(x) = \mathbb{P}\{Y = 1 \mid X = x\}$
- Ordonnancement défini par une règle de scoring $s : \mathcal{X} \rightarrow \mathbb{R}$
- But : trouver s qui ordonne les éléments de \mathcal{X} comme η

Exemple 7 - Scoring et Courbes ROC



- Taux de vrais positifs :

$$\text{TPR}_s(t) = \mathbb{P}(s(X) \geq t \mid Y = 1)$$

- Taux de faux positifs :

$$\text{FPR}_s(t) = \mathbb{P}(s(X) \geq t \mid Y = -1)$$

Courbe Caractéristique de l'Opérateur de Réception :

$$t \mapsto (\text{FPR}_s(t), \text{TPR}_s(t))$$

De très nombreux problèmes...

- Moteurs de recommandation, filtrage collaboratif
- Graph-Mining, détection de communautés
- Analyse en Variables Latentes : (Kernel) PCA, ICA, NMF
- Règles d'association (e.g. algorithme *apriori*)
- Apprentissage par renforcement :
exploration vs exploitation
- Analyse sémantique

Des succès dans de nombreuses applications...

- Reconnaissance de la parole
- Biométrie
- Publicité en ligne
- Détection de fraude
- Risque de crédit
- Aide au diagnostic médical
- Séparation de sources
- Yield management
- *etc.*

De très nombreuses technologies...

- SVM Machines à Vecteurs Supports
- Deep Learning
- Forêts Aléatoires, Boosting
- Bandits Stochastiques
- Modèles graphiques : réseaux bayesiens/markoviens
- K-means, clustering spectral
- Indexation sémantique latente
- Apprentissage multi-tâche
- *etc.*

Et des contraintes et contextes variés...

- Données **fonctionnelles** (e.g. séries temporelles)
- Apprentissage "en-ligne" vs Apprentissage "batch"
- Nécessité d'**interpréter** les règles de décision
- Données et calculs **distribués**
- Apprentissage **semi-supervisé**
- *etc.*

Machine-Learning

Principes Statistiques et Algorithmiques

Le paradigme de l'apprentissage à travers l'exemple de la classification

- Couple de v.a. $(X, Y) \sim P$ inconnue
- X = v.a. d'entrée à valeurs dans \mathcal{X} (ex : \mathbb{R}^d with $d \gg 1$)
- Y = label binaire, à valeurs dans $\mathcal{Y} = \{-1, +1\}$
- **Objectif** : à partir d'exemples $\{(X_1, Y_1), \dots, (X_n, Y_n)\}$, construire un **classifieur** : $C : x \in \mathcal{X} \mapsto C(x) \in \{-1, 1\}$ appartenant à une **classe** \mathcal{G} de **risque** minimum

$$L(C) = \mathbb{E}[\mathbb{I}\{Y \neq C(X)\}]$$

- \mathcal{G} est en correspondance biunivoque avec la classe $\{x \in \mathcal{X} : C(x) = +1\} : C \in \mathcal{G}\}$

Apprentissage \neq Modélisation Statistique

- Idéalement, calculer $C^* = \arg \min_{C \in \mathcal{G}} L(C)$
- La fonctionnelle de risque $L(\cdot)$ est inconnue, comme P
- Soit $\eta(x) = \mathbb{P}(Y = +1 | X = x)$ **probabilité a posteriori**
- On pose $p = \mathbb{P}(Y = +1)$ (le taux espéré de positifs dans la population statistique)
- Un calcul simple (minimiser $\mathbb{E}[\mathbb{I}(\mathcal{Y} \neq g(X)) | X]$) montre que

$$C^*(x) = 2 \cdot \mathbb{I}\{\eta(x) > 1/2\} - 1, \quad x \in \mathcal{X}$$

\Rightarrow on prédit le label le plus probable au vu de l'observation $X = x$,
classifieur de Bayes

- Risque minimum théorique : $L^* = L(C^*) = 1/2 - \mathbb{E}[|\eta(X) - 1/2|]$
- La distribution de $\eta(X)$ autour de $1/2$ régit la difficulté du problème !

Minimisation du Risque d'Erreur Théorique

- Excès de risque théorique :

$$L(C) - L^* = \mathbb{E}[|\eta(X) - 1/2| \mathbb{I}\{X \in G^* \Delta G_C\}]$$

où G^* , G_C désignent les sous-ensemble de l'espace d'entrée \mathcal{X}

$$G^* = \{x \in \mathcal{X} : \eta(x) > 1/2\}$$

$$G_C = \{x \in \mathcal{X} : C(x) = +1\}$$

et $A \Delta B = (A \cap \bar{B}) \cup (\bar{A} \cap B)$ la *différence symétrique*.

- Pour bien prédire, il s'agit de recouvrir l'ensemble de niveau $G^* = \{x \in \mathcal{X} : \eta(x) > 1/2\}$, pas la probabilité a posteriori $\eta(x)$ (cf "fléau de la dimension")

Minimisation du Risque Empirique (ERM)

- Données d'apprentissage $D_n = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$
- Candidat : $C : \mathcal{X} \rightarrow \{-1, 1\}$ appartenant à une classe \mathcal{G}
- Risque empirique = Erreur d'apprentissage

$$\hat{L}_n(C) = \frac{1}{n} \sum_{i=1}^n \mathbb{I}\{Y_i \neq C(X_i)\}$$

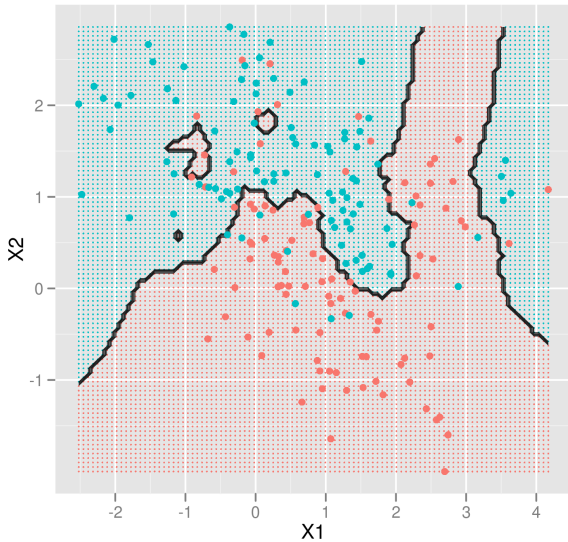
à minimiser sur la classe \mathcal{G} .

- Solution "**minimiseur du risque empirique**" :

$$\hat{C}_n = \arg \min_{C \in \mathcal{G}} \hat{L}_n(C)$$

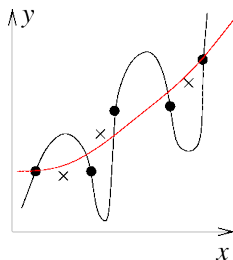
- Ajustement OK pour les données d'apprentissage, mais pour les **données (X, Y) à venir ?**

k=5



ERM - Heuristique

- L'apprentissage peut fonctionner lorsque $\hat{L}_n(C)$ est proche de $L(C)$, uniformément lorsque C décrit \mathcal{G}
- Pour un classifieur C fixé, $\hat{L}_n(C) \rightarrow L(C)$ lorsque le nombre n d'exemples croît (Loi des grands nombres, théorème de la limite centrale, etc.)
- La théorie de Vapnik-Chervonenkis fournit des garanties lorsque la classe \mathcal{G} n'est **pas trop complexe**, quelle que soit la distribution des données (X, Y)



ERM - Éléments de théorie

- Sous quelles conditions peut-on garantir que l'apprentissage fonctionne ?
C'est à dire que $L(\hat{C}_n)$ est proche de L^* ?

- Décomposition "**bias-variance**" de l'excès de risque

$$L(\hat{C}_n) - L^* \leq 2 \sup_{C \in \mathcal{G}} |L(C) - \hat{L}_n(C)| + \inf_{C \in \mathcal{G}} L(C) - L^*$$

lorsque \mathcal{G} "croît" ↗ ↘

- Le second terme dépend de la classe \mathcal{G} seulement (biais), pas des données d'apprentissage
- Le 1er terme (estimation) peut être contrôlé au moyen de résultats de **concentration** pour le processus empirique

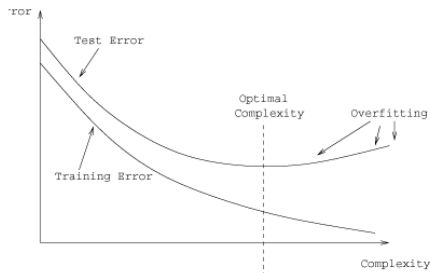
$$Z = \{L(C) - \hat{L}_n(C)\}_{C \in \mathcal{G}}$$

⇒ l'étude des **processus empiriques** est à la base de la théorie probabiliste de l'apprentissage

- Garder à l'esprit que l'**erreur test** $L(\hat{C}_n) = \mathbb{E}_{(X,Y)}[\mathbb{I}\{Y \neq \hat{C}_n(X)\}]$ est **aléatoire** (fonction complexe des données d'apprentissage)

ERM - Éléments de théorie

- La "complexité" de la classe \mathcal{G} régit le compromis "biais-variance"



- Notion de **complexité combinatoire** introduite par Vapnik et Chervonenkis dans les années 60

Théorie de l'Apprentissage Statistique



A. Chervonenkis & V. Vapnik

Théorie de Vapnik et Chervonenkis

- **Inégalité de concentration** : avec probabilité $1 - \delta$:

$$\sup_{g \in \mathcal{G}} |L_n(g) - L(g)| \leq \mathbb{E} \sup_{g \in \mathcal{G}} |L_n(g) - L(g)| + \sqrt{\frac{2 \log(1/\delta)}{n}}$$

- **Contrôle de la complexité** : $\mathbb{E} \sup_{g \in \mathcal{G}} |L_n(g) - L(g)| \leq C \sqrt{\frac{V}{n}}$ lorsque \mathcal{G} est une classe de VC dimension $V < +\infty$.
- **Complexité combinatoire** d'une collection \mathcal{G} d'ensembles de \mathbb{R}^d
 - ▶ **Trace** de \mathcal{G} sur un nuage de n points x_1, \dots, x_n

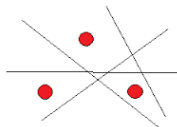
$$Tr_{\mathcal{G}}(\{x_1, \dots, x_n\}) = \{G \cap \{x_1, \dots, x_n\} : G \in \mathcal{G}\}$$

- ▶ **Coefficient d'éclatement** :

$$S_n(\mathcal{G}) = \sup_{x_1, \dots, x_n} \text{Card}(Tr_{\mathcal{G}}(\{x_1, \dots, x_n\})) \leq 2^n$$

- ▶ **VC dimension** : $V = \sup\{n \geq 1 : S_n(\mathcal{G}) = 2^n\}$

ERM - Éléments de théorie



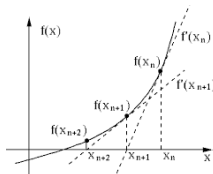
La théorie de Vapnik et Chervonenkis permet de contrôler l'erreur de généralisation

- des règles linéaires/quadratiques/paramétriques telles que celles produites par
 - ▶ l'analyse discriminante linéaire/quadratique
 - ▶ les SVM linéaires
 - ▶ les réseaux de neurones
- des règles décrites par des arbres de décision (partitionnement récursif, algorithme CART)

Elle ne permet pas d'expliquer les capacités de généralisation des SVM non linéaires, des techniques d'agrégation (e.g. Boosting, forêts aléatoires)

ERM - Un problème algorithmique !

- Le problème $\min_{g \in \mathcal{G}} \hat{L}_n(g)$ est **NP-difficile** en général
- En pratique : apprendre, c'est **optimiser** :
 - ▶ **Optimisation continue** : la dérivée $f(x) = F'(x)$ de la fonction F à optimiser est nulle en l'optimum \Rightarrow **chercher les zéros d'une fonction f**



- ▶ **Optimisation discrète** : énumérer "intelligemment"

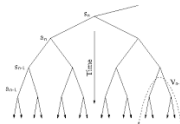
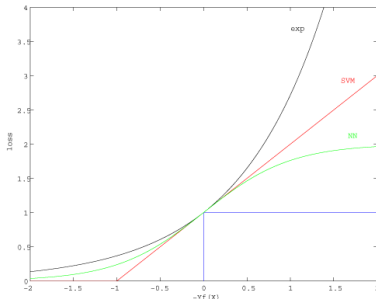


Figure 8.3: Hierarchical optimization tree schematic. In each of the N stages, state s is realized before taking action a . Each path from top to bottom corresponds to a unique realization of outcomes and decisions.

ERM en pratique

- Si $g(x) = \text{sign}(f(x))$, le risque s'écrit : $L(f) = \mathbb{E}[\mathbb{I}\{-Yf(X) > 0\}]$
- En pratique, on remplace la perte $l(u) = \mathbb{I}\{u > 0\}$ par une version "régulière" $\tilde{l}(u) \rightarrow \text{risque } \tilde{L}(f) = \mathbb{E}[\tilde{l}(-Yf(X))]$
 - ▶ SVM $\tilde{l}(u) = \max(0, 1 + u)$
 - ▶ Boosting $\tilde{l}(u) = \exp(u)$
 - ▶ Réseaux de neurones $\tilde{l}(u) = \tanh(u)$



ERM et Approximation Stochastique

- On minimise une version lissée (convexifiée) et éventuellement pénalisée du risque empirique : $\min_{f \in \mathcal{F}} \tilde{L}_n(f)$:

$$\tilde{L}_n(f) \stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^n \tilde{l}(-Y_i f(X_i)) + \text{pen}(f)$$

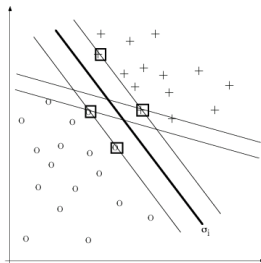
- en général, une méthode **d'approximation stochastique inductive** est mise en oeuvre

$$f_{t+1} = f_t - \rho_t \nabla_f \tilde{L}_n(f_t)$$

- De très nombreux algorithmes d'apprentissage sont basés sur ce principe
- Exemples : Logit, Neural Networks, linear SVM, etc.

Le Perceptron Monocouche

- **Le Perceptron de F. Rosenblatt ('62)** : cas binaire $Y \in \{-1, +1\}$,
règle affine $g(x) = \text{sgn}(a + \langle b, x \rangle)$, $\tilde{l}(u) = u$



Itérations

- ▶ Choisir au hasard une observation (X_i, Y_i) parmi les observations mal classées par la règle courante
- ▶ $(a, b) \leftarrow (a, b) + \rho(Y_i, Y_i X_i)$ (descente de gradient **stochastique**)

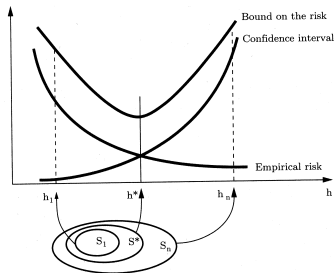
Évaluation du Risque

—

Sélection de Modèle

- Capacité de généralisation d'un modèle
- Biais, variance et complexité d'un modèle
- Le cadre "Big Data" : **Apprentissage-Validation-Test**
- **Validation croisée** : une méthode pour l'estimation de l'erreur de prédiction
- Techniques de rééchantillonnage - **Bootstrap**

Régler le bon niveau de complexité



Exemples :

- le nombre de voisins dans les " k plus proches voisins "
- le nombre de feuille terminales d'un arbre de décision
- le nombre de vecteurs support d'un SVM
- le noyau d'un SVM non linéaire
- le nombre de couches d'un réseau de neurones

Erreurs : d'apprentissage, de généralisation

- L'apprentissage est réalisé à partir d'exemples

$$\mathcal{D}_n = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$$

- Le classifieur $\hat{C}_n \in \mathcal{G}$, construit à partir d'une méthode réalisant (de façon approchée) la minimisation du risque empirique, est **aléatoire**, il dépend de \mathcal{D}_n , comme son **erreur** :

$$L(\hat{C}_n) = \mathbb{E} \left[\mathbb{I}\{Y \neq \hat{C}_n(X)\} \mid \mathcal{D}_n \right]$$

L'espérance est prise sur un couple (X, Y) indépendant de \mathcal{D}_n

Méthodes pour l'évaluation du risque, la sélection de modèle

- L'erreur d'apprentissage n'est pas un bon estimateur de l'erreur !

$$\hat{L}_n(\hat{C}_n) = \frac{1}{n} \sum_{i=1}^n \mathbb{I}\{Y_i \neq \hat{C}_n(X_i)\}$$

Elle se réduit à 0 dès que la classe \mathcal{G} est suffisamment complexe
 \Rightarrow Surajustment et faible capacité de généralisation

- L'objectif est double
 - ▶ Sélection de Modèle : choisir le meilleur modèle parmi des modèles en compétition
 - ▶ Evaluation du Risque : pour un modèle donné, estimer l'erreur de prédiction

"Big Data"

- Diviser les données en trois parties :

Apprentissage - Validation - Test

- Typiquement : 50% - 25% - 25%

- $K \geq 1$ modèles candidats : $\mathcal{G}_1, \dots, \mathcal{G}_K$

- ▶ Pour chaque $k \in \{1, \dots, K\}$, appliquer ERM aux données d'apprentissage $\Rightarrow \hat{C}^{(k)}$
- ▶ Utiliser les données de validation pour trouver le "meilleur"
 $\hat{k} \in \{1, \dots, K\}$
- ▶ Estimer son erreur au moyen des données test (indépendantes de \hat{k})

- Et si l'apprentissage était réalisé sur la quasi-totalité des données ?

→ Ré-échantillonnage

Validation Croisée

- Soit $K \geq 1$ (typiquement, 5 or 10), " K -fold cross-validation" ($K = n$ "leave-one-out" estimation)
- Diviser aléatoirement les données en K parties égales
- Pour tout $k \in \{1, \dots, K\}$,
 - ▶ apprendre $\hat{C}^{(-k)}$ à partir de toutes les données sauf celles de la k -ième partie
 - ▶ calculer l'erreur réalisée par $\hat{C}^{(-k)}$ sur les données de la k -ième partie
- Moyenner les K quantités
→ estimateur de l'erreur par validation croisée

"Pulling yourself up by your own bootstrap"

- Bootstrap (principe plug-in) : remplacer la distribution (inconnue) des données par la distribution empirique
- Exemple : choisir 80% des données aléatoirement pour apprendre le modèle et l'évaluer sur les 20% restants
- Approximation Monte-Carlo : réitérer la procédure B fois et moyenner les erreurs

Application : K -plus proches voisins

- Soit $K \geq 1$. Sur \mathbb{R}^D , on considère une **métrique** d (ex : distance euclidienne)
- Pour chaque valeur x , soit $\sigma = \sigma_x$ la permutation de $\{1, \dots, n\}$ telle que

$$d(x, x_{\sigma(1)}) \leq \dots \leq d(x, x_{\sigma(n)})$$

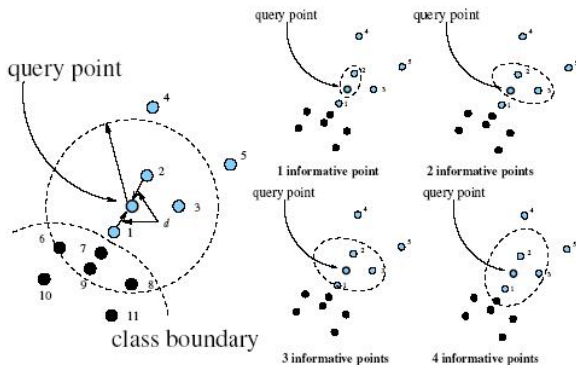
- On considère les **K -plus proches voisins**

$$\{x_{\sigma(1)}, \dots, x_{\sigma(K)}\}$$

- **Vote majoritaire** : $N_y = \text{Card}\{k \in \{1, \dots, K\}; y_{\sigma(k)} = y\}$,
 $y \in \{-1, 1\}$

$$C(x) = \arg \max_{y \in \{-1, +1\}} N_y,$$

Application : K -plus proches voisins



Application : K -plus proches voisins

- Pour $K = 1$ (sur-ajustement), l'erreur d'apprentissage est nulle.
- Pour $K = n$ (sous-ajustement), la prédiction $C(x)$ est la même dans tout l'espace (le label majoritaire dans l'échantillon d'apprentissage).
- On peut choisir K par :
 - ▶ un plan d'expérience
 - ▶ validation croisée
 - ▶ Bootstrap

A retenir

- **Optimisation** Un problème d'apprentissage statistique est défini par des données et un **critère de performance** pour la règle de décision à construire
- **Automatisation** L'algorithme apprend automatiquement le "meilleur modèle" par optimisation d'une version statistique du critère choisi
- **Généralisation** Afin d'éviter le sous/sur-apprentissage, la méthode et ses hyperparamètres sont sélectionnés via un plan d'expérience ou par validation croisée

Ces principes/concepts généraux vont s'incarner dans tous les problèmes et méthodes décrits lors des six sessions à venir !

Merci !