MACHINE LEARNING POUR L'ACTUARIAT

AMU, Master 1 MAS, parcours Actuariat (Jan.-Mars 2025)

Xavier Milhaud

xavier.milhaud@univ-amu.fr

www.xaviermilhaud.fr

STRUCTURE DU COURS

Volume global de 24h, prenez l'ordinateur chargé ! (TP)

Organisation pratique :

- 12h de CM (6 séances de 2h), avec comme séances
  - 1 - Notions statistiques introductives, réduction de dimension et lien avec l'assurance
  - 2 - Philosophie de l'apprentissage statistique
  - 3 - Algorithme CART
  - 4 - Algorithme des Forêts Aléatoires
  - 5 - Algorithme Gradient Boosting
  - 6 - Réseaux de neurones

- 12h de TD/TP en R (6 x 2h) :
  - 1 - Réduction de dimension en paramétrique
  - 2 - Méthodes CART
  - 3 - Méthode ensembliste, exemple Random Forest
  - 4 - Introduction aux GBM - les Gradient Boosting Trees
  - 5 - Implémentation approfondie des GBM
  - 6 - Réseaux de neurones

- Sanctionné par un projet en R : résolution d'une problématique opérationnelle avec de vraies données.

1. Introduction au problème statistique : fléau de la dimension et convergence des estimateurs, réduction de dimension
2. Actuariat - données et assurance
3. Philosophie de l'apprentissage statistique
4. Première brique en Machine Learning : arbres de décision
5. Bagging $+$ randomization de CART : forêts aléatoires
6. Agrégation de modèles par boosting
7. Réseau de neurones et Deep Learning

BIBLIOGRAPHIE - EXEMPLES

Livres :

- An introduction to Statistical Learning, (with Applications in R), ;James, Written, Hastie, Tibshirani
- The Elements of Statistical Learning : Data Mining, Inference and Prediction ; Hastie, Tibshirani, Friedman
- Classification & Regression Trees ; Breiman, Friedman, Olshen, Stone
- Artificial Intelligence : A Modern Approach ; Russell and Norvig
- Speech and Language Processing ; Jurafsky and Martin
- Pattern Recognition and Machine Learning ; Bishop C.

1. Introduction au problème statistique : fléau de la dimension et convergence des estimateurs, réduction de dimension
   - Motivation statistique des modèles d'apprentissage
   - Estimation et grande dimension
   - Extrapolation de la statistique classique (échantillons raisonnables) aux grands échantillons
   - Notions de biais et variance d'un estimateur

CONTEXTE CLASSIQUE D'ETUDE DES RISQUES

L'analyse d'engagements d'un assureur nécessite de comprendre l'impact de caractéristiques **X** sur le risque Y.

Les bases de données des assureurs comportent généralement

- les **caractéristiques** de l'assuré,
- les **options** du contrat,
- les conditions de **marché**.

Informations **X** *jouent un rôle crucial* dans les prév. de sinistralité Y
$\Rightarrow$ méthodes doivent tenir compte de ces caractéristiques
(historiquement modélisation paramétrique par régression).

GENERALITES

Pourquoi modéliser ?
$\Rightarrow$ A partir d'une série d'observations, phénomène trop complexe pour une description analytique par un modèle déterministe...

Objectif en statistique : modélisation, parfois décomposable, pour

1. **explorer** : décrire variables, leurs liaisons, positionner obs. ;
2. **expliquer** : tester l'influence d'une variable ds un modèle supposé connu ;
3. **prévoir** et sélectionner : un meilleur ensemble de prédicteurs.

Historiquement, modèles paramétriques avec var. expl. $+$ bruit $\Rightarrow$ inférer les paramètres depuis les observ. en contrôlant au mieux les propriétés (comportement) de la partie aléatoire.

MOTIVATION DU COURS

Observons n réalisations de $(\mathbf{X}, Y) \in \mathbb{R}^{p+1}$.
D'habitude, on considère que

- le rapport des dimensions (n, p) est raisonnable,
- les hyp. du modèle sont vérifiées (échantillon/résidus supposés suivre des lois sous la forme d'une famille connue),

Alors les techniques statistiques tirées du modèle linéaire général sont optimales (max. de vraisemblance)... Avec des échantillons de taille restreinte $\Rightarrow$ difficile de faire beaucoup mieux.

Mais dès que hyp. distributionnelles ne sont pas vérifiées / relations entre les variables ou la variable à modéliser ne sont pas linéaires, ou encore dès que le volume des données est important, d'autre méthodes viennent concurrencer la stat. classique...

1. Introduction au problème statistique : fléau de la dimension et convergence des estimateurs, réduction de dimension
   - Motivation statistique des modèles d'apprentissage
   - Estimation et grande dimension
   - Extrapolation de la statistique classique (échantillons raisonnables) aux grands échantillons
   - Notions de biais et variance d'un estimateur

PARAMETRIQUE VS NON-PARAMETRIQUE

Cadre : on veut estimer 1 fct. m, par ex. $m(x) = E[Y \mid X = x]$, ou $m(x) = P(Y = 1 \mid X = x)$.

- Estimation paramétrique : on cherche m parmi une famille indexée par un param. de dim. finie → **ex** : rég. lin., $m(x) = a + bx$. Un candidat s'identifie à 2 paramètres (a, b).
- Estimation non paramétrique : pas d'hypothèse (ou bc –), cherche $m(x)$ parmi ttes les fonct. possibles (dim. infinie) ⇒ décompositions dans des bases fonctionnelles (ex GAM) :

$$y = m(x) = \sum_{k=0}^{\infty} w_k g_k(x) \quad \text{et donc} \quad \hat{m}(x) = \sum_{k=0}^{h\star} \hat{w}_k g_k(x)$$

LA DIMENSION, FACTEUR LIMITANT

Paramètres importants du problème : ses dimensions... **Notons** :

- n nombre d'observations ou taille de l'échantillon,
- p nombre de variables observées sur cet échantillon.

$\rightarrow$ n grand : pas de pb a priori, bien au contraire (théo asymptot.) !
$\rightarrow$ p grand pose problème (fléau de la dimension) !

L'estimateur du max. de vrais. conserve sa prop. de normalité asymptotique si $p^2/n \rightarrow 0$ lorsque $p, n \rightarrow \infty$ (Portnoy, 1988).
$\Rightarrow$ Données "massives" : $p > \sqrt{n}$.

Concept de sparsité $\simeq$ dimension effective $\Rightarrow$ compter le nb de var. expl. réel du pb, à défaut de compter le nb total de var. expl. !

1. Introduction au problème statistique : fléau de la dimension et convergence des estimateurs, réduction de dimension
   - Motivation statistique des modèles d'apprentissage
   - Estimation et grande dimension
   - Extrapolation de la statistique classique (échantillons raisonnables) aux grands échantillons
   - Notions de biais et variance d'un estimateur

THE P-VALUE PROBLEM

"*A key issue with applying small-sample statistical inference to large samples is that even minuscule effects can become statistically significant. The increased power leads to a dangerous pitfall as well as to a huge opportunity. The issue is one that statisticians have long been aware of :* the p-value problem.*Chatfield (1995, p. 70) comments,* question is not whether differences are significant (they nearly always are in large samples), but whether they are interesting. Forget statistical significance, what is the practical significance of the results ?"
Mingfeng Lin, Henry Lucas, Jr. et Galit Shmueli , 2010 galitshmueli.com

Source : blog d'Arthur Charpentier.
Idée : bonne puissance de test implique qu'1 gd échantillon (n grand) fait systématiquement conclure à un effet significatif d'un facteur de risque, quand bien même cet effet serait négligeable...

RAPPEL SUR LA PUISSANCE D'UN TEST

On peut résumer le rôle des probabilités de bonne et mauvaise décision dans le tableau suivant (β est la **puissance** du test) :

Vérité / Décision	H_0	H_1
H_0	$1-\alpha$	$1-\beta$
H_1	α	β

Risque / Erreur 1ère espèce : décider H_1 vraie alors que H_0 vraie (proba. α).

Erreur seconde espèce : décider H_0 vraie alors que H_1 vraie (proba. erreur de seconde espèce : $1-\beta$).

La puissance β dépend

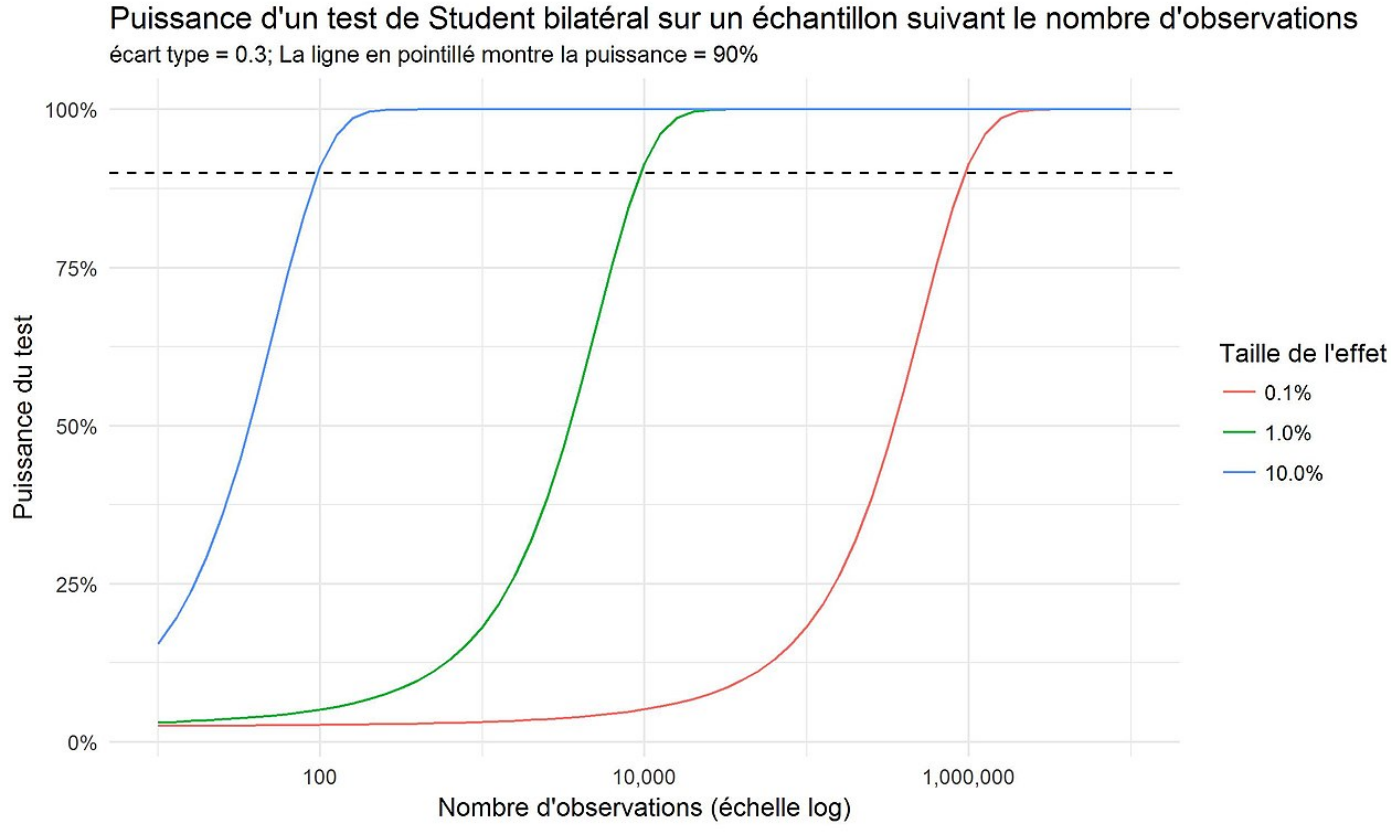
1. du nombre d'observations (d'individus),
2. du risque α : en general quand $\alpha \nearrow$, la puissance $\beta \nearrow$ aussi : on ne gagne pas partout !
3. et de l'ampleur de l'effet (différence entre les 2 groupes pour un essai clinique par ex.) relativement aux autres grandeurs.

Remarque 1 : puissance statistique β permet de calculer le nb d'observations nécessaire dans une étude (on fixe β désirée, le risque de 1$^{\text{ère}}$ espèce et les paramètres associés aux groupes).

Remarque 2 : calcul de la puissance peut s'appliquer à grand nombre de tests statistiques (comparaison de moyennes, comparaison de proportions, modèle logistique, modèle de régression, ...), lorsque l'hyp. alternative est assez restrictive.

ILLUSTRATION AVEC UN TEST DE STUDENT

Peut servir comme test sur les coefficients d'une rég. linéaire.
Même avec un effet faible (1%), on dispose souvent en assurance de + de 10 000 observ., donc d'une bonne puissance...



AUTRE FORMULATION DU MEME PB : FALSE DISCOVERY RATIO (FDR)

Le test de significativité,

$$H_0 : \beta_k = 0 \qquad VS \qquad H_1 : \beta_k \neq 0$$

est basé sur le test de Student, issu de la statistique $t_k = \frac{\hat{\beta}_k}{se_{\hat{\beta}_k}}$.

Cette statistique suit une loi de Student, T, à ν degrés de liberté (où $\nu = d + 1$, avec d le nombre de paramètres) : $T \sim t_\nu$.

La p-valeur du test correspond à $P(|T| > |t_k|)$.

En grande dimension, l'intérêt est limité car le FDR est grand...

Exemple : avec un niveau de significativité de 5%, 5% des variables sont faussement significatives !

Application : supposons que nous disposons de 100 variables explicatives, avec seulement 5 d'entre elles réellement significatives...

→ Normalement, ces 5 variables passeront le test de Student.
→ Mais 5 autres le passeront aussi (test faussement positif) ⇒ 10 variables sont donc détectées significatives !

⇒ Le FDR est de 50% !

Pour corriger cet effet, on peut consulter [BH95]...

AUTRE EXEMPLE ET CONCLUSION

Un coefficient de corrélation égal à 0,002 est significativement différent de 0 si $n = 10^6$, mais il est totalement inutile...

"A researcher might choose to retain a causal covariate which has a strong theoretical justification even if is statistically insignificant"

"Statistical significance plays a minor or no role in assessing predictive performance. In fact, it is sometimes the case that removing inputs with small coefficients, even if they are statistically significant, results in improved prediction accuracy" (Shmueli, 2010)

1. Introduction au problème statistique : fléau de la dimension et convergence des estimateurs, réduction de dimension
   - Motivation statistique des modèles d'apprentissage
   - Estimation et grande dimension
   - Extrapolation de la statistique classique (échantillons raisonnables) aux grands échantillons
   - Notions de biais et variance d'un estimateur

ERREUR D'UNE MODELISATION

On peut décomposer l'erreur dans la modélisation de $m(x)$:

Erreur de spécification $+$ Erreur d'estimation du modèle.

$\rightarrow$ **Erreur spécification** : vient d'hyp. sur la classe d'estimateurs de la fct m. Inmesurable par déf. puisque m inconnue.

$\rightarrow$ **Erreur d'estimation** du modèle (si le modèle est "vrai", cad bien spécifié). Erreur d'autant $+$ importante que la technique est compliquée et/ou nécessite beaucoup de données.

Rq : un modèle non paramétrique a une erreur de spécification $\simeq$ 0, au prix d'une éventuelle inflation de l'erreur d'estimation.

DECOMPOSITION DE L'ERREUR D'ESTIMATION

Soit un estimateur $\hat{\theta}$ (var. aléatoire) de θ.

On a coutume de considérer comme mesure d'erreur d'estimation le risque quadratique d'un estimateur (MSE : erreur quadratique moyenne ; ou MSEP : MSE sur de nvelles données n'ayant pas servi à construire l'estimateur), par

$$MSE(\hat{\theta}) = E[(\hat{\theta} - \theta)^2].$$

Cette erreur se décompose en 2 termes, biais et variance :

$$MSE(\hat{\theta}) = E[(\hat{\theta} - \theta)]^2 + Var(\hat{\theta}),$$

soit approximativement son biais au carré plus sa variance.

Globalement, $+$ un modèle est complexe, $+$ son biais diminuera et $+$ sa variance grandira.

$\Rightarrow$ **Il faut optimiser le dosage entre biais et variance** !

$\Rightarrow$ Cela revient à contrôler la complexité du modèle !

Ex : contrôler le nb de variables (explicatives) dans le cadre paramétrique $\Rightarrow$ a conduit à la déf. de critères de sélection tels que le Cp de Mallows, Akaïke (AIC), Schwartz (BIC), ...

Rq : hormis la classe, choix du bon modèle dans une classe est primordial. Pb d'optimisation doivent donc prendre en compte la complexité de la classe dans laquelle la solution est recherchée.

LIEN ENTRE CES NOTIONS

Quelque soit la méthode, tous les auteurs soulignent l'importance de construire des modèles parcimonieux (dimension raisonnable).

En effet $+$ un modèle est complexe, $+$ il est flexible $\Rightarrow$ faible erreur d'ajustement (bon "fit") $\Rightarrow$ synonyme d'un biais faible...

Par contre ce modèle peut s'avérer défaillant pour généraliser, s'appliquer à des données nouvelles (synonyme de gde variance).

$\Rightarrow$ **Combinaison de modèles** (bagging, boosting) contourne ce pb au prix d'une $\nearrow$ du volume de calculs et de l'interprétabilité.

2 Actuariat - données et assurance

- La révolution numérique : nouvelles données en assurance
- Impact du Big Data sur le secteur assurantiel

DEFIS STATISTIQUES

L'arrivée des Big Data a permis la découverte pour le "grand public" de méthodes statistiques fondées sur l'apprentissage statistique (Machine Learning quand appliqué en pratique).

Mais il faut garder en tête que

1. il ne faut pas créer une usine à gaz...
2. un modèle statistique est d'autant $+$ robuste qu'il est simple,
3. ces méthodes ne sont pas encore parfaitement adaptées à la gestion de tt type de données.

$\Rightarrow$ Beaucoup de travail préalable à faire avant un emploi judicieux !

BIG DATA, QU'ES A QUO ?

Big Data, définition simplifiée : données non traitable en une passe et dans un temps raisonnable sur une station de travail.

Deux époques :

$<$ 2005, ordinateurs 32-bit. Taille $n > 10^7$, $p > 100 =$ 8Go.
$>$ 2005, ordinateurs 64-bit : bc $+$ de mémoire physique, mais unités de calcul limitées.

Deux motivations principales d'utlisation : description, prévision.
Deux aspects : spatial (volume) et temporel (flux).

CARACTERISATION DES BIG DATA

On a coutume de parler de Big Data lorsqu'on dispose de données...

- en grand **volume** (énorme base de données),
- en grande **variété** (numérique, texte, images, vidéos, ...),
- en grande **vitesse** (fréquence d'arrivée de l'information, évolution des données).

Règle des 3V...qui doit déboucher sur la création de "V"aleur de par l'exploitation de ces données.

DEFIS PRATIQUES

- Défi opérationnel, essentiellement informatique :
  - système d'information, architecture, capacité de stockage...
  - calculs distribués (MapReduce) $\Rightarrow$ Hadoop, Spark, ... ;
- Une réflexion sur la donnée :
  - qualité de la donnée et gestion de son aspect non-structuré : comment homogénéiser des formats différents à l'origine ?
  - sélection en fonction de sa pertinence, gestion,
  - visualisation : SQL (Structured Query Language), noSQL..
- Un enjeu éthique : anonymisation principalement (tests génétiques en assurance maladie,...) $\Rightarrow$ réglementation RGPD.

D'OU VIENNENT LES NOUVELLES DONNEES ?

Essentiellement de données externes... Les assureurs possèdent déjà des données internes (peu exploitées, $\simeq 20\%$), et accèdent maintenant à d'autres sources riches en information :

1. **Objets connectés** : télématique, Apple Watch, ...
2. Réseaux sociaux et **navigation internet** : pouvoir de nuisance des consommateurs ;
3. Assurance de **biens partagés** : AirBnB, AutoLib', ...
4. L'**Open Data** : crawling, scrapping... (Datagouv, ...).

C'est l'intégration au sein d'un même SI qui est très compliqué.

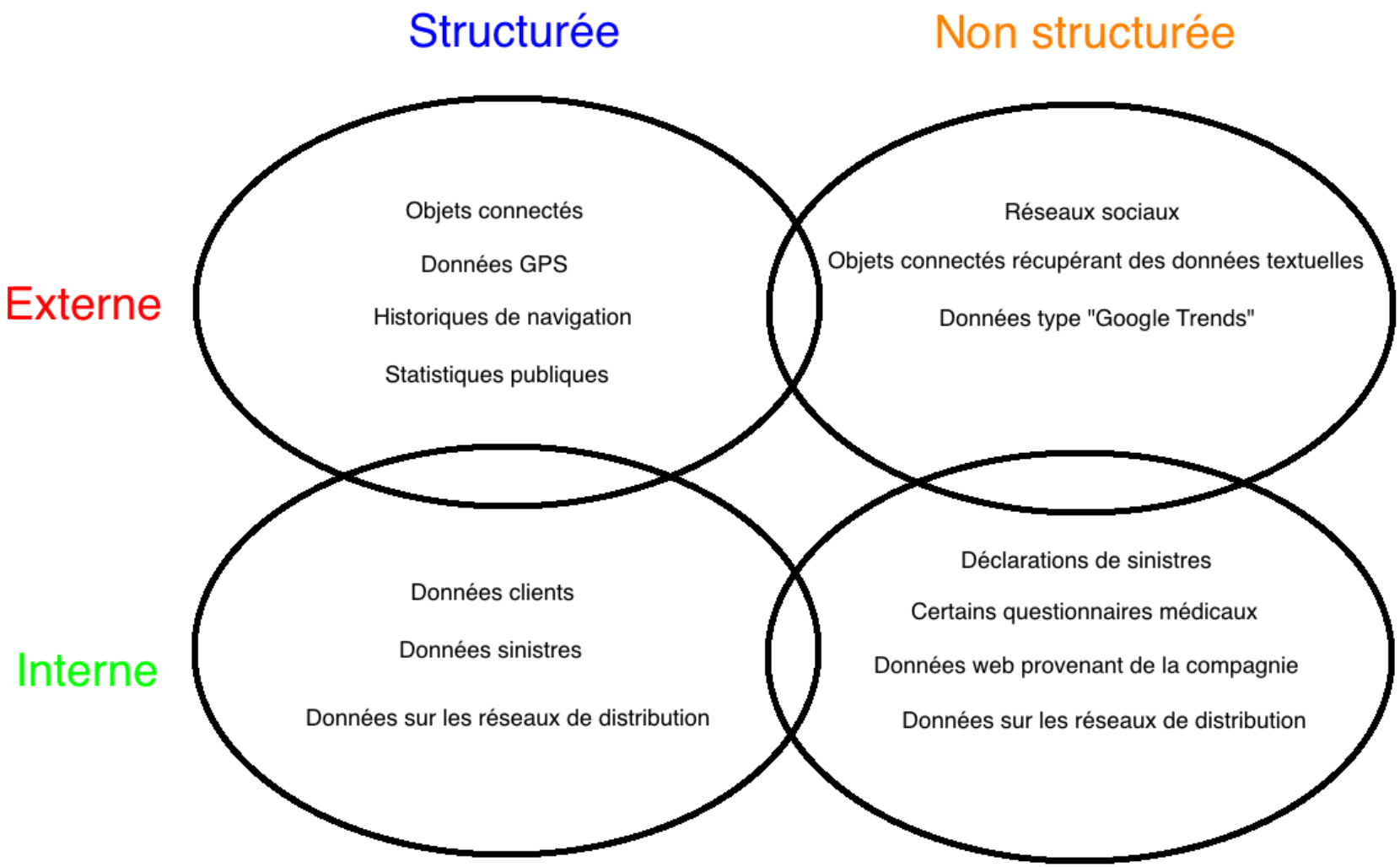
QUESTIONS ESSENTIELLES

Ces nouvelles données posent des questions fondamentales quant à leur utilisation, notamment

- Fiabilité des données
  → s'assurer auprès des services ayant fourni les données de leur fiabilité, de leur authenticité ;
- Cohérence
  → s'assurer du contenu de ces données ;
- Sécurité
  → cyber-risque, ...

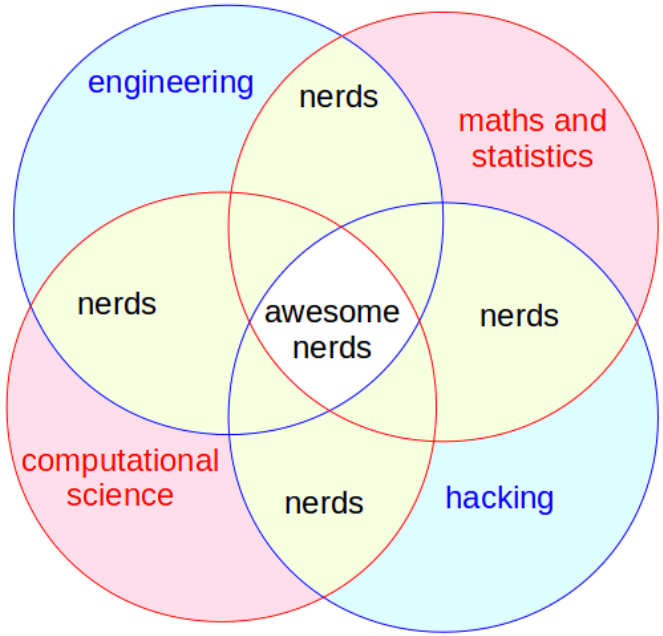
⇒ Risque opérationnel également accru !

CLASSIFICATION DES DONNEES



DATA SCIENTIST ET DATAVIZ

"*statistics is the* grammar of data science.*It is crucial to making data speak coherently. But it takes statistics to know whether this difference is significant, or just a random fluctuation.* (. . .) *What differentiates data science from statistics is that data science is a holistic approach. We're increasingly finding data in the wild, and data scientists are involved with gathering data, massaging it into a tractable form, making it tell its story, and presenting that story to others.*" Mike Loukides, 2010
radar.oreilly.com



Source : blog d'Arthur Charpentier.
Idée : le data scientist ne se limite pas à la statistique, il cherche à faire parler ses données en général... (data visualisation)

2 Actuariat - données et assurance

- La révolution numérique : nouvelles données en assurance
- Impact du Big Data sur le secteur assurantiel

APPORT PRINCIPAL DE CES NOUVELLES DONNEES EN ASSURANCE

Un des gros problèmes de l'assureur (par rapport au banquier) est la faible fréquence de ses intéractions avec l'assuré...

En effet, ils ne se voient en général que 2 fois en tout pour tout :
$\rightarrow$ Une fois à la souscription ;
$\rightarrow$ Une fois lors du sinistre s'il a lieu.
$\Rightarrow$ Très difficile pour l'assureur de bien connaitre l'assuré !

Technologies liées au Big Data vont augmenter significativement la fréquence de ces intéractions...et **atténuer les particularités** de l'assurance : antisélection et aléa moral.
(en plus de l'inversion du cycle de production !)

IMPACT SUR LA CHAINE DE VALEUR

Le Big Data a un impact à plusieurs niveaux pour un assureur, parmi ses tâches “historiques” impactées :

- segmentation, tarification (Pay-As-You-Drive, HomeBox),
- provisionnement : micro-level reserving,
- détection de fraude (par géolocalisation par exemple),
- ciblage marketing (compréhension des comportements),
- scoring d’assurés : la construction d’un bon score reste issue d’une approche stat. **couplée à une connaissance métier**.

Remarque : échelle de temps de l’assurance parfois bc plus longue que dans d’autres secteurs (attention aux dérives du risque).

LA DATA SCIENCE, JUSQU'OÙ ?

La base de l'assurance est la mutualisation...
...Or l'enjeu principal du Big Data est de mieux comprendre les mécanismes à l'échelle de l'individu !

"We are moving from an era of private data and public analyses to one of public data and private analyses" (Andrew Gelman)

Il y a donc un risque énorme (surtout en tarification), qui est...

...la PERTE de MUTUALISATION.

Où s'arrêtera la segmentation... ?

3 Philosophie de l'apprentissage statistique

- Principes généraux
- Théorie de Vapnik et surapprentissage
- Agrégation d'estimateurs
- Comment analyser les résultats ?
- Rappels sur le bootstrap
- Agrégation : cas des variables catégorielles

MACHINE LEARNING, NOUVELLE APPROCHE

- Abandon d'une approche de "modélisation" pour 1 approche qui cherche à laisser parler les données ("**data-driven**"), typique du monde non-paramétrique.
- Big Data : pour rendre compte d'une réalité complexe, on s'autorise des modèles – simples, voire peu intelligibles.

$\Rightarrow$ Une logique de prévision domine, plus qu'une logique d'analyse et d'explication des phénomènes.

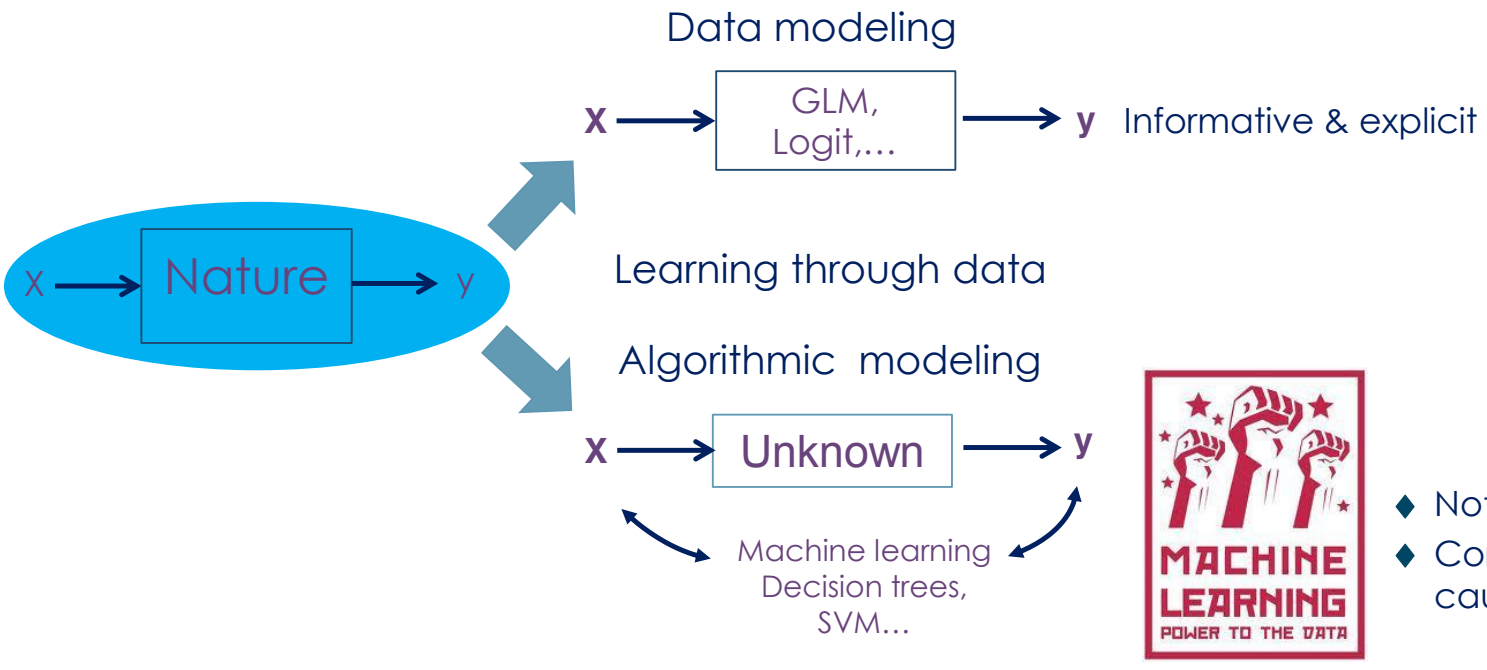
RAPPEL : NON PARAMÉTRIQUE, PARAMÉTRIQUE

- Estimation paramétrique : on cherche m parmi une famille indexée par un paramètre de dimension finie.
  $\rightarrow$ *Exemple* : régression linéaire, $m(x) = a + bx$.
  Une fonction candidate s'identifie à 2 paramètres (a, b).

- Estimation non paramétrique : on ne fait plus d'hypothèse (ou bc –), on cherche $m(x)$ parmi ttes les fonctions possibles (dim. infinie).

Exemple connu d'estimateurs **non paramétriques** : estimateurs à noyaux, ...

ILLUSTRATION



PROBLEME SUPERVISÉ VS NON SUPERVISÉ

Deux types de pb : présence ou non d'une variable à expliquer Y qui a été, conjointement avec X, observée sur les mêmes objets.

Paradigme du cas supervisé : apprendre à généraliser à partir d'exemples du phénomène observé.
S'applique

- **à la régression** : cas où la réponse est continue ;
- **à la classification** : cas où la réponse est catégorielle.

Cas non supervisé : n'observe pas la valeur de la variable d'intérêt (ex. modèles mélange : classer les indiv. dans les composantes $\Rightarrow$ on ne connait pas leur composante d'appartenance)

EN PRATIQUE...

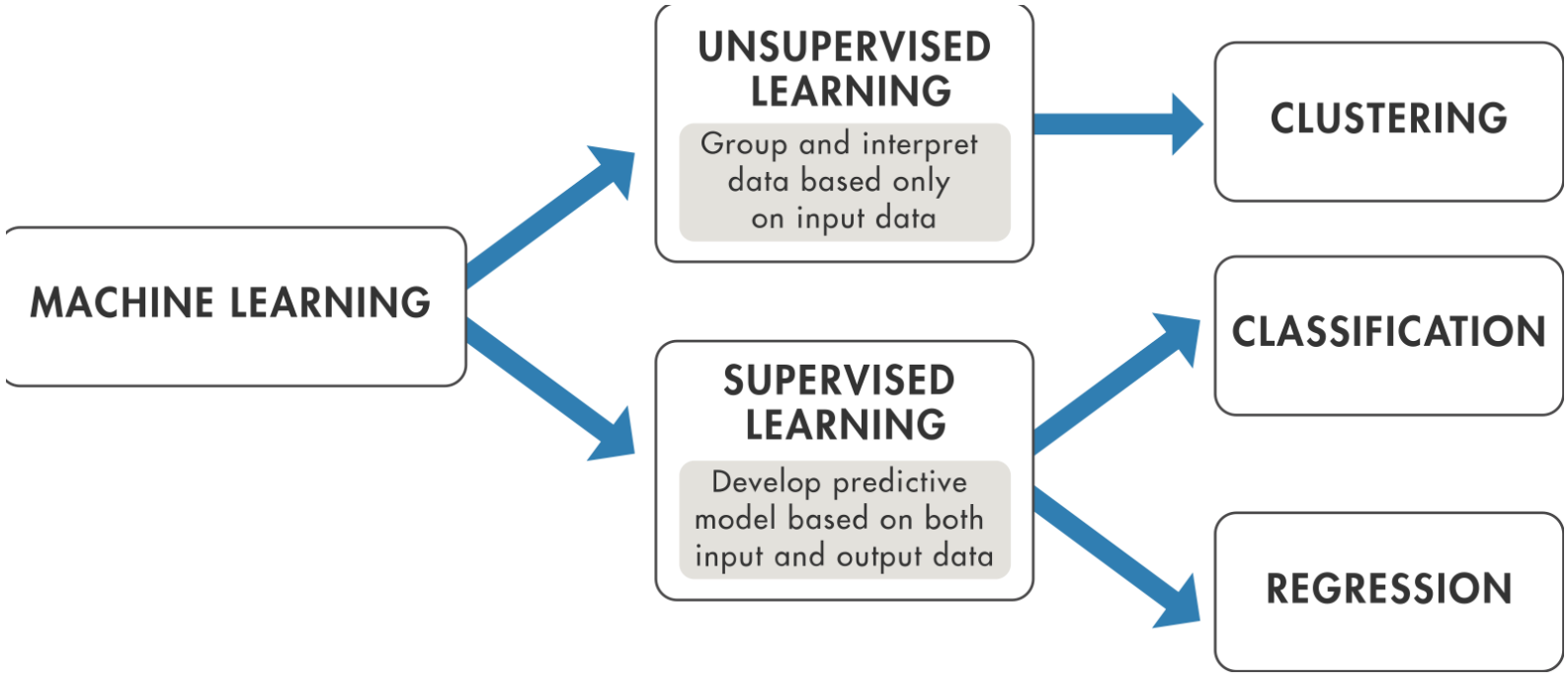
Dans le 1er cas (supervisé) $\Rightarrow$ trouver une fonction f susceptible, au mieux selon un critère à définir, de reproduire Y ayant observé X :

$$Y = f(X) + \epsilon$$

où ϵ symbolise le bruit ou erreur de mesure.

Dans le cas contraire (absence d'Y) $\Rightarrow$ non-supervisé.
Objectif : recherche d'une typologie/taxinomie des observations...
Comment regrouper celles-ci en classes homogènes mais les $+$ dissemblables entre elles $\rightarrow$ pb de clustering.

SCHEMA RECAPITULATIF ET METHODES ASSOCIEES



STATISTIQUE CLASSIQUE VS APPRENTISSAGE

- **Statistique classique** : recherche le modèle génératif des données. Construit l'estimateur sur 1 jeu de données unique. Une théorie asymptotique permet de juger sa qualité (IC,...).

- **Apprentissage stat.** : recherche de bonnes prévisions...
  - on ne cherche pas le modèle qui génére les données !
  - les exemples du phénomène observé sont représentés par l'échantillon d'appren. : on souhaite faire apprendre à l'algo. la relation entre X et Y, puis la généraliser (prévision de Y) à des occurrences de X pr lesquelles Y inconnue.
  - la qualité n'est plus jugée via des critères asymptotiques, mais à l'aune d'une mesure d'adéquation à l'échantillon test.

AUTREMENT DIT...

Statistique classique : approches privilégiant la compréhension !

- Permet une compréhension du mécanisme générateur des données, avec une représentation si possible parcimonieuse ;
- Le modèle doit être “simple” et interprétable (odd-ratio, ...)

Machine learning : approches privilégiant la prévision !

- pour de nouveaux individus : pouvoir de généralisation,
- les modèles sont en fait des algorithmes.

“Modern statistical thinking makes a clear distinction between the statistical model and the world. The actual mechanisms underlying the data are considered unknown. The statistical models do not need to reproduce these mechanisms to emulate the observable data”, (Breiman, 2001)

“Better models are sometimes obtained by deliberately avoiding to reproduce the true mechanisms”, (Vapnik, 2006)

3 Philosophie de l'apprentissage statistique

- Principes généraux
- Théorie de Vapnik et surapprentissage
- Agrégation d'estimateurs
- Comment analyser les résultats ?
- Rappels sur le bootstrap
- Agrégation : cas des variables catégorielles

QUALITE D'ESTIMATION ET GRANDE DIMENSION
MONDE NON PARAMETRIQUE

Théorème : soit $X \in \mathbb{R}^d$, et m une fonction k fois dérivable à dérivées bornées. La **vitesse optimale de convergence d'un estimateur non paramétrique** $\hat{m}$ est

$$\hat{m}(x) - m(x) = O(n^{-k/(2k+d)}) \quad p.s.$$

- Si la fonction m est régulière (par ex. infiniment dérivable) à d fixé, la vitesse de convergence est en $\sqrt{n}$.
- Si d est "grand" par rapport à n, la performance d'estimation est considérablement dégradée.

ÉCHANTILLONS ET POUVOIR DE GENERALISATION

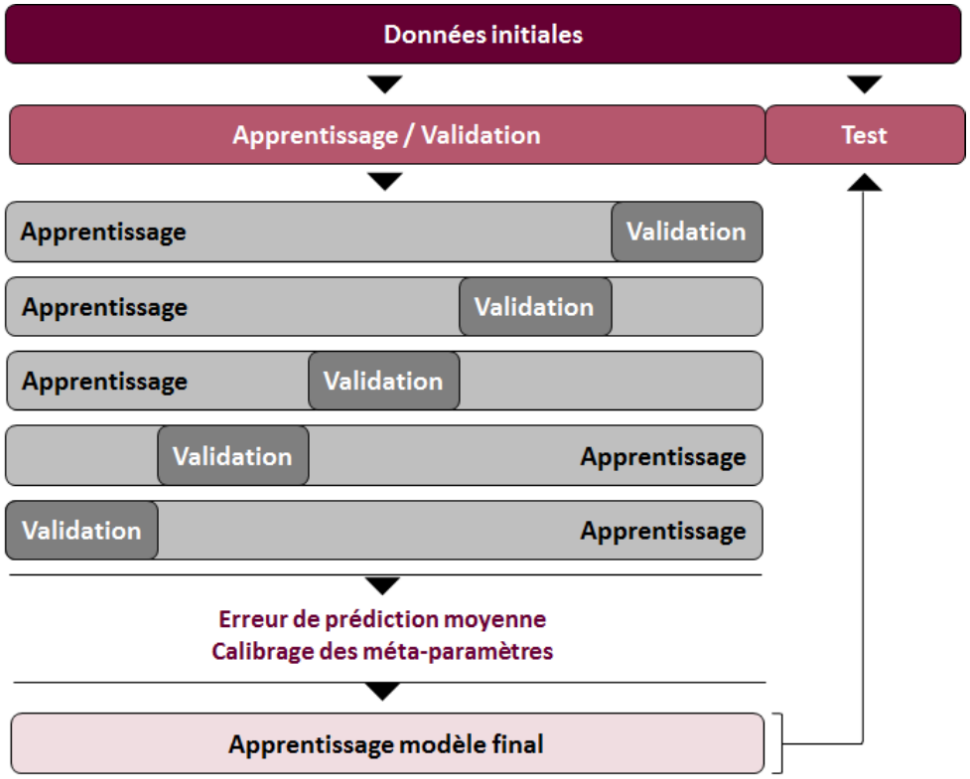
Les méthodes d'apprentissage statistique induisent le choix de paramètres de tuning (param. "utilisateur")... Ils jouent un rôle important dans le pouvoir de généralisation du modèle.

Pour choisir leur valeur, on peut soit

- recourir à la validation croisée, ou
- on créé plusieurs échantillons :
  - un échantillon d'apprentissage pour construire le modèle ;
  - un échantillon de validation pr optimiser les paramètres de tuning ("tuning" du modèle) ;
  - un échantillon test $\perp\!\!\!\perp$ pour évaluer la performance du modèle avec les paramètres de tuning choisis.

PRINCIPE DE LA VALIDATION CROISÉE (5-fold)

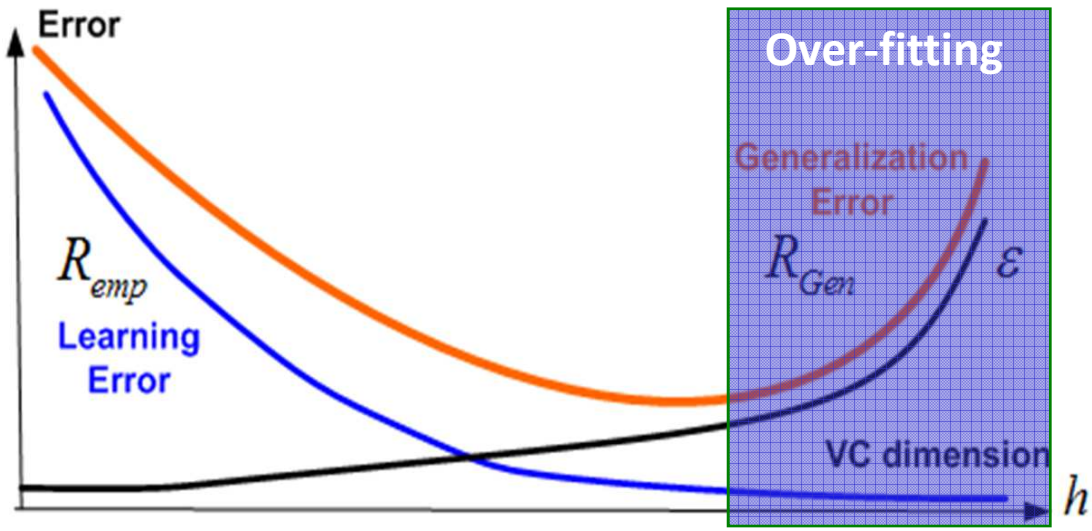
Utilisée pr la sélection de modèle ! Permet de choisir le param. et/ou modèle optimal.



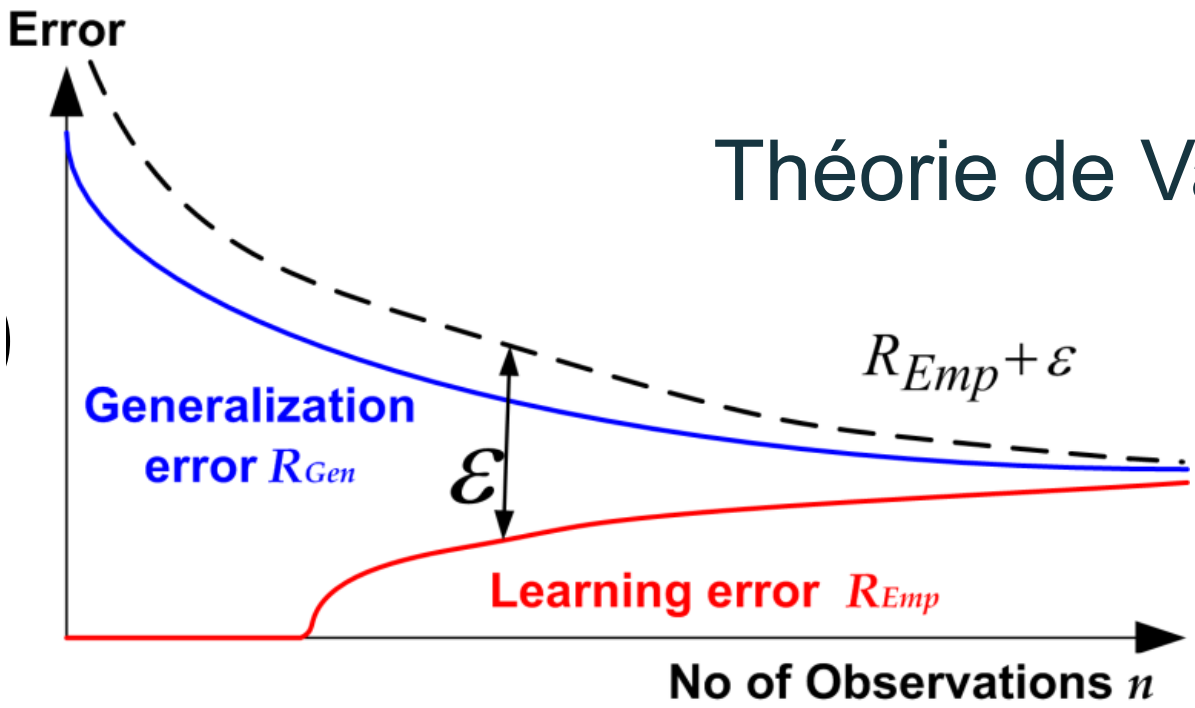
THÉORIE DE VAPNIK
ERREURS EN FONCTION DE LA VC DIMENSION (*h*)

$$R_{Gen}(\theta) \leq R_{emp}(\theta) + \varepsilon(n,h)$$

$$\varepsilon(n,h) = \sqrt{\frac{1 + ln\left(2n/h\right)}{n/h} - \frac{ln\,\eta}{n}}$$



ET EN FONCTION DE n ?



Théorie de Vapnik

QUELQUES PREMIERES REMARQUES

On voit très bien à travers l'inégalité de Vapnik que :

- l'erreur de généralisation croit quand la dimension augmente : $\Rightarrow$ les modèles de grande dim. ont un faible biais au prix d'une grande variance (et inversement).
- l'erreur est dépendante du rapport n/h (rapport du nombre de données sur complexité du modèle),
- on $\nearrow$ la capacité prédictive si $h \nearrow$ mais moins vite que n,
- on peut $\nearrow$ la complexité du modèle si on $\nearrow$ aussi n.

3 Philosophie de l'apprentissage statistique

- Principes généraux
- Théorie de Vapnik et surapprentissage
- Agrégation d'estimateurs
- Comment analyser les résultats ?
- Rappels sur le bootstrap
- Agrégation : cas des variables catégorielles

L'AGREGATION

- Approche "modèle" VS agrégation :
  - **modèle** : déterminer une distribution de probabilité "simple" et unique qui rende compte des données ;
  - **agrégation** : faire la synthèse de plusieurs approches, ne plus se reposer sur un modèle unique.
- Les 2 approches ne sont **pas totalement antagonistes**.

- Les approches d'estimation basées sur l'agrégation sont $+$ précises mais $+$ difficilement interprétables (ex : agréger 3 modèles de régression paramétriques, comment ?).

Rq : on dit que les modèles simples (ex : logit) sont interprétables. Loin d'être vrai car les covariables sont svt corrélées, donc la valeur des param. ne reflète pas exactement leur impact !

META-MODELES ou METHODES D'ENSEMBLE

Soit $\hat{m}_j(x)$ l'estimateur obtenu en utilisant le modèle j. Pour agréger B modèles et obtenir l'estimateur ensembliste

$$\hat{m}_a(x) = \sum w_j \hat{m}_j(x),$$

on peut mener :

- construction parallèle, $\perp\!\!\!\perp$ de $+$sieurs estimateurs individuels, puis combinaison $\Rightarrow$ bagging
- construct. séquentielle, puis combinaison $\Rightarrow$ boosting !
- construct. parall., puis imbrication (meta-modèle) $\Rightarrow$ stacking

Rq : $\sum_{j=1}^{B} w_j = 1$ avec w_j poids affecté à l'estimateur j (version fréquentiste du Bayesian Model Averaging).

	Avantages	Inconvénients
"Modèle" unique	Interprétation des paramètres Analyse de l'impact des variables Communication plus aisée	Biais important (erreur de modèle) Choix entre deux modèles ?
Agrégation	Moins de biais car hypothèses plus faibles Plus de difficulté liée au choix de modèle (à nuancer)	Interprétation complexe Analyse de l'impact des variables plus compliquée (mais possible) Communication sur le modèle hardue

SYNTHÈSE DES PRINCIPALES DIFFÉRENCES

A travers ce que nous venons de voir, les différences essentielles de l'apprentissage statistique par rapport à une approche classique de modélisation résident dans les points suivants :

- les hypothèses : beaucoup moins d'hypothèses ($\perp\!\!\!\perp$ entre observations, entre facteurs de risque, hypothèses de distribution paramétrique, ...)
- l'agrégation potentielle de modèles : on construit plusieurs modèles et on synthétise,
- l'interprétabilité des résultats : on perd en interprétabilité à cause de l'agrégation.

3 Philosophie de l'apprentissage statistique

- Principes généraux
- Théorie de Vapnik et surapprentissage
- Agrégation d'estimateurs
- Comment analyser les résultats ?
- Rappels sur le bootstrap
- Agrégation : cas des variables catégorielles

MESURE DE L'INCERTITUDE - CAS D'UN MODÈLE

On dispose de résultats asymptotiques...

- En paramétrique, théorie du max. de vraisemblance. On a en général des IC sur le paramètre estimé...
  Exemple : modèle linéaire,

$$\sqrt{n}(\hat{\beta}_1 - \beta_1) \quad \approx \quad \mathcal{N}(0, \sigma^2)$$

  Donc $\mathbb{P}(|\hat{\beta}_1 - \beta_1| \geq \epsilon) \approx \mathbb{P}(|Z| \geq \epsilon)$ où $Z \sim \mathcal{N}(0, \sigma^2/n)$.

  $\rightarrow \sigma^2$ indique la précision de l'estimation : à estimer ! $\rightarrow$ D'où la possibilité d'évaluer $\mathbb{P}(|\hat{m}_1(x) - m_1(x)| \geq \epsilon)$.

- En non paramétrique, théorie de Vapnik-Chervonenkis.

CAS DE L'AGREGATION D'ESTIMATEURS

C'est différent...(notons $\hat{m}_a(x)$ l'estimateur agrégé)

- En général, pas de résultat du type $\hat{m}_a(x) - m(x) \sim \mathcal{N}(0, \sigma^2)$.
- La qualité se mesure en premier lieu par rapport à un échantillon de validation.

Vocabulaire : soit un échantillon de $n + m$ observations, avec

- un échantillon d'apprentissage : sous-échantillon de n observations à partir desquelles on construit $\hat{m}_a$.
- un échantillon de validation : le reste (m observations) sur lequel on juge de la qualité de l'estimateur.

EXEMPLE EN RÉGRESSION

On dispose d'un échantillon de $(n+m)$ observations i.i.d., de même loi qu'un vecteur aléatoire (Y, X).

But : estimer $m(x) = E[Y|X = x]$.

(1) On tire au sort n observations, d'où l'échantillon $(Y_i, X_i)_{1 \leq i \leq n}$.

(1bis) Les m autres observations $(Y_i, X_i)_{n+1 \leq i \leq n+m}$ constituent l'échantillon de validation.

(2) Construction de $\hat{m}_a(x)$ à partir de $(Y_i, X_i)_{1 \leq i \leq n}$.

(3) Calcul de l'erreur de prédiction sur l'échantillon de validation :

$$e(\hat{m}_a) = \sum_{i=n+1}^{n+m} (Y_i - \hat{m}_a(x_i))^2.$$

Plus cette quantité est petite, plus l'estimateur est jugé bon.

Questions :

1. Pourquoi ce critère ?
2. Pourquoi ne pas directement regarder l'erreur sur l'échantillon d'apprentissage ?
3. Choix de n et m ?

Q1 : POURQUOI CE CRITÈRE ?

- $m(x) = E[Y|X = x]$: “meilleure façon d’approcher Y par une fonction de X, au sens de l’écart quadratique” ;
- Si $\hat{m}_a$ est bon estimateur, alors $\hat{m}_a(X_i)$ est proche de $m(X_i)$ $\forall i$. Or $m(x)$ étant la fonction la $+$ proche de Y sachant $X = x$, $+$ $e(\hat{m}_a)$ est petit, $+$ $\hat{m}_a$ devrait ê proche de m (inconnue !).
- Si on calcule d’autres quantités, le coût quadratique ne sera pas forcément utilisé pour l’erreur.
  $\rightarrow$ Ex. : pour estimer la médiane de $Y|X = x$, on minimisera

$$\sum_{i=n+1}^{n+m} |Y_i - \hat{m}_a(x_i)|.$$

Q2 : POURQUOI NE PAS REGARDER L'ERREUR SUR L'ÉCHANTILLON D'APPRENTISSAGE ?

En d'autres termes, pourquoi ne pas prendre $m = 0$?

- Risque = **surapprentissage** (on capte le bruit au lieu du signal), le signal etant l'information principale...

- Exemple d'estimateurs faisant de l'overfitting : arbre maximal dans les estimateurs CART possibles...

Q3 : CHOIX DE n ET m

Pas de règle gravée dans le marbre (choix classique, et très arbitraire : $m = n/2$), mais

- en général, $n > m$, et
  $\rightarrow$ la proportion de l'échantillon d'apprentissage tend vers 50% quand la taille globale des données est grande ;
  $\rightarrow$ elle tend vers 80 voire 90% le cas contraire.

- pourquoi a-t-on besoin d'un n grand ?
  $\rightarrow$ besoin de $+$ de données pour calculer un estimateur $\hat{m}_a$ précis (sa CV est, en général, en $n^{-\alpha}$ pour un certain $\alpha > 0$).
- pourquoi m ne doit pas être trop petit ?
  $\rightarrow$ Pour que la validation ait un sens...

AGREGATION : ESTIMATEURS SUR ÉCHANTILLONS ⫫

$\rightarrow$ Limite : on rappelle que $\hat{m}_a(x) = \frac{1}{B}\sum_{j=1}^{B} \hat{m}_j(x)$, où les $\hat{m}_j(x)$ sont **corrélés si calculés sur le même échantillon**...

Si les $\hat{m}_j$ sont ⫫ car calculés sur $\neq$ échantillons ⫫ (en notant $\sigma_j^2(x)$ la variance de $\hat{m}_j(x)$) :

$$Var(\hat{m}_a(x)) = \frac{1}{B^2}\sum_{j=1}^{B} \sigma_j^2(x).$$

En somme, si $\sigma^2(x) = \sup_{j=1,\ldots,B} \sigma_j^2(x)$, $Var(\hat{m}_a(x)) \leq \frac{\sigma^2(x)}{B}$:

$\Rightarrow$ Variance estimateur agrégé << variance estimateur unique.

LIMITE PRATIQUE

Néanmoins, il est difficile de calculer des estimateurs sur des échantillons $\neq$, car la taille des données n'est évidemment pas infinie en pratique... D'où :

- prendre B sous-échantillons pour calculer B estimateurs $\neq$ est une solution de riche (n doit être très grand pour l'$\perp\!\!\!\perp$) ;
- la solution de couper l'échantillon en sous-échantillons atteint vite ses limites.

$\Rightarrow$ Une solution : le rééchantillonnage (par exemple bootstrap).

Rq : l'indépendance entre les estimateurs unitaires n'est pas garantie car certains échantillons bootstrap peuvent fortement se ressembler...

APPLICATION : REECHANTILLONNAGE BOOTSTRAP

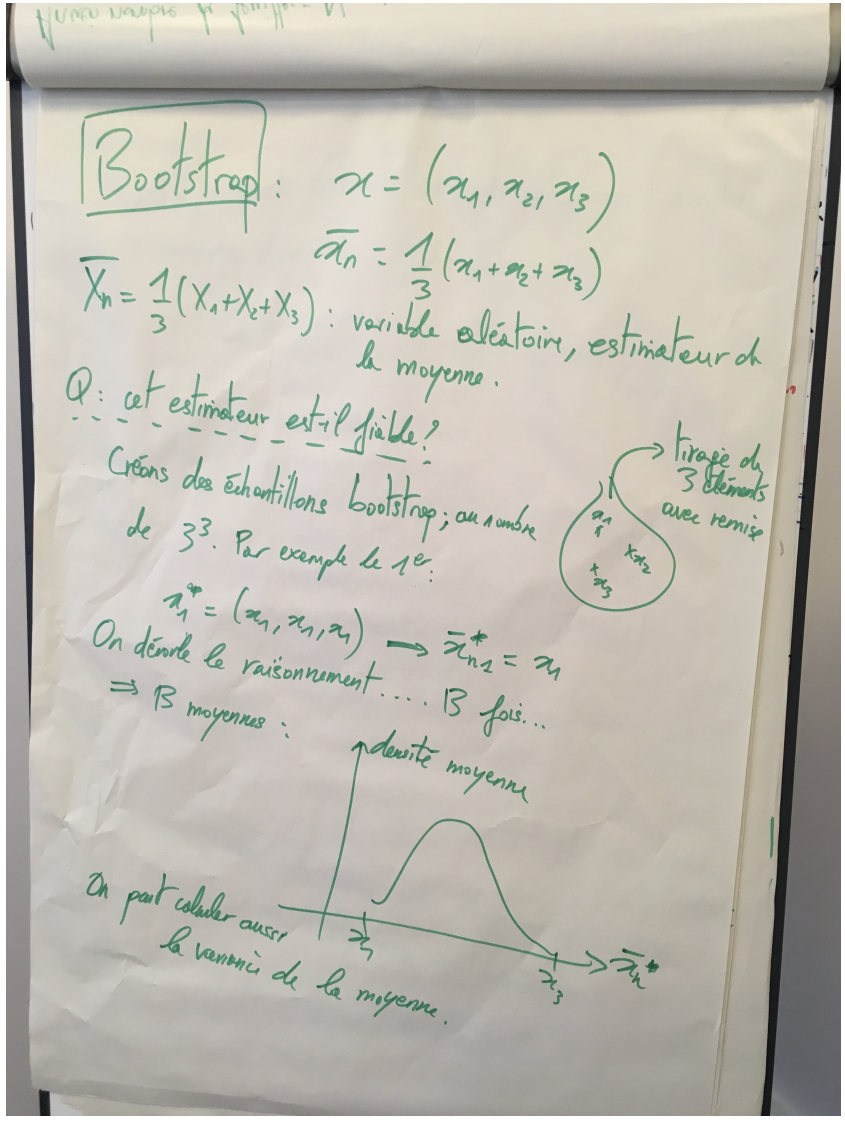
- Si on souhaite agréger B estimateurs, on génère B échantillons bootstrap suivant la méthode ci-dessous.
- On utilise l'échantillon j pour calculer l'estimateur $\hat{m}_j$.

Bootstrap : pour $j = 1, ..., B$ et $i = 1, ..., n$, on tire $Z_i^{(j)}$ i.i.d. de loi unif. sur $\{1, ..., n\}$. Le j$^{\text{ème}}$ échantillon bootstrap est $(Y_i^{(j)}, X_i^{(j)})_{1 \leq i \leq n}$ où

$$Y_i^{(j)} = Y_{Z_i^{(j)}} \qquad X_i^{(j)} = X_{Z_i^{(j)}}.$$

En moyenne, $e^{-1} = 36,7\%$ des observations initiales ne sont pas tirées dans un échantillon bootstrap donné.

ILLUSTRATION



POURQUOI LE BOOTSTRAP ?

Idées derrière le bootstrap :

- On va créer artificiellement des échantillons semblables à celui d'origine en simulant des données, ce qui permettra de construire des modèles cohérents entre eux.
- Problème : les échantillons étant corrélés, les estimateurs seront corrélés (même si différents !)...

On aura besoin d'introduire des éléments supplémentaires pour décorréler au mieux les estimateurs !

PRÉVISIONS AGRÉGÉES SUR VARIABLE BINAIRE

Soit le contexte suivant :

- une v.a. Y qui vaut 0 ou 1, avec X = caractéristiques.
  - ex. 1 : $Y = 1$ si accident dans l'année, 0 sinon.
  - ex. 2 : $Y = 1$ si défaut de paiement dans l'année, 0 sinon.
  - ex. 3 : $Y = 1$ si le client souscrit un contrat, 0 sinon.

- $1^{\text{ère}}$ solution : déterminer $E[Y|X] = P(Y = 1|X)$ pour chaque modèle. Puis approche similaire à précédemment : moyenner.
  $\rightarrow$ Cette solution fournit un estimateur $\hat{m}_a(x)$ qui prend des valeurs entre 0 et 1.

  $\Rightarrow$ Si $X = x$ et $\hat{m}_a(x) > 0.5$, on prédit $Y = 1$. Sinon $Y = 0$.

DEUXIÈME SOLUTION : LE VOTE MAJORITAIRE

- Au lieu d'agréger les estimations des espérances conditionnelles $\hat{m}_j$, on agrège les **prédictions** associées.
- i.e. on définit, pour $j = 1, ..., B$, $\quad \hat{p}_j(x) = 1_{\hat{m}_j(x)>0,5}$.
- Pour $X = x$, on prédit Y par

$$\hat{p}_a(x) = \begin{cases} 1 & \text{si majorité de } \hat{p}_j(x) \text{ égaux à 1} \\ 0 & \text{sinon.} \end{cases}$$

- Rq : c'est ce que fait randomForest(.) de rpart.

GÉNÉRALISATION AUX VARIABLES CATÉGORIELLES

On s'intéresse à une variable Y prenant un nombre fini de modalités, $\{1, ..., k\}$, avec X = caractéristiques.

- Exemple : Y = gravité sinistre, classé sur échelle de 1 à k.
- Stratégie : transformation en un problème binaire.

$$Z_l = 1_{Y=l}$$

pour $l = 1, ..., k$, on estime $E[Z_l \,|\, X] = P(Y = l \,|\, X)$ pour tout l.

- Si on note $\hat{m}_{j,l}(x)$ l'estimateur de $P(Y = l \,|\, X = x)$ basé sur la méthode j, la prédiction $\hat{p}_j(x)$ associée est

$$\hat{p}_j(x) = \underset{l=1,...,k}{\arg\max}\ \hat{m}_{j,l}(x).$$

4 Première brique en Machine Learning : arbres de décision

- Algorithme CART
- Exemples
- Formalisation : construction de l'arbre
- Lien avec le problème de régression classique
- Gestion du surapprentissage : réduction de dimension
- Réponse catégorielle
- Outils et mesures de performance des modèles
- Extensions et conclusion

ALGO ISSU DE L'INTELLIGENCE ARTICIFIELLE

7 principes éthiques de la Commission Européenne pour l'IA :

1. contrôle/supervision humaine : l'IA n'a pas de conscience !
2. résistance et sécurité des algorithmes : fiabilité pour gérer les erreurs et incohérences ;
3. gestion des données, protection de la vie privée : utilisateurs en mesure de contrôler leurs propres données ;
4. transparence algo : expliquer ce que fait l'IA, traçabilité
5. diversité, non-discrimination et équité ;
6. bien-être social et environnemental : l'IA doit être mise au service de la société dans son ensemble ;
7. l'"accountability" : principe de responsabilité, mise en place de procédures internes à l'entreprise pour démontrer le respect des règles relatives à la protection des données.

OBJECTIF ARBRE : CLASSIF. DES INDIVIDUS

Regrouper des indiv. hétérogènes en classes homogènes de risque pour résumer l'info d'une BdD gigantesque.

$\exists$ de nombreuses techniques de classification, parmi lesquelles :

- pour la classification non-supervisée :
  - $\rightarrow$ les algorithmes dits des k-plus proches voisins (non param.) ;
  - $\rightarrow$ les techniques ascendantes d'arbre de classification (CAH) ;
  - $\rightarrow$ model-based clustering (paramétrique) ;

- pour la classification supervisée :
  - $\rightarrow$ modèles paramétrique de choix (LOGIT) ;
  - $\rightarrow$ réseaux de neurones ; SVM (non paramétrique) ;
  - $\rightarrow$ arbres descendants (CART, CHAID, ...). Non param.

ARBRE ET CLUSTERING : PREMIERS ÉLÉMENTS

Pour estimer notre quantité d'intérêt, on choisit d'utiliser un arbre...
Mais qu'est-ce qu'un arbre ?

1. Une racine : contient l'ensemble de la population à segmenter (le portefeuille global) $\Rightarrow$ c'est le point de départ ;
2. Un tronc et des branches : contiennent les règles de division qui permettent de segmenter la population ;
3. Des feuilles : contiennent les sous-populations homogènes (sur leurs caractéristiques et la réponse) créées, fournissent l'estimation de la quantité d'intérêt.

RÉGLES ET LECTURE D'UN ARBRE CART

Un arbre de classification / régression se lit de la racine vers les feuilles (l'inverse d'une CAH...).

A chaque ramification, une règle de division apparait : dans CART,

- cette règle ($\simeq$ question) admet une réponse binaire (oui/non),
- elle n'est basée que sur un facteur de risque (une covariable).

Un noeud est l'intersection d'un ensemble de règles. L'estimation de la quantité d'intérêt se lit dans les noeuds terminaux (feuilles).

N'importe quel individu de la population initiale appartient à une unique feuille : les sous-populations créées sont disjointes.

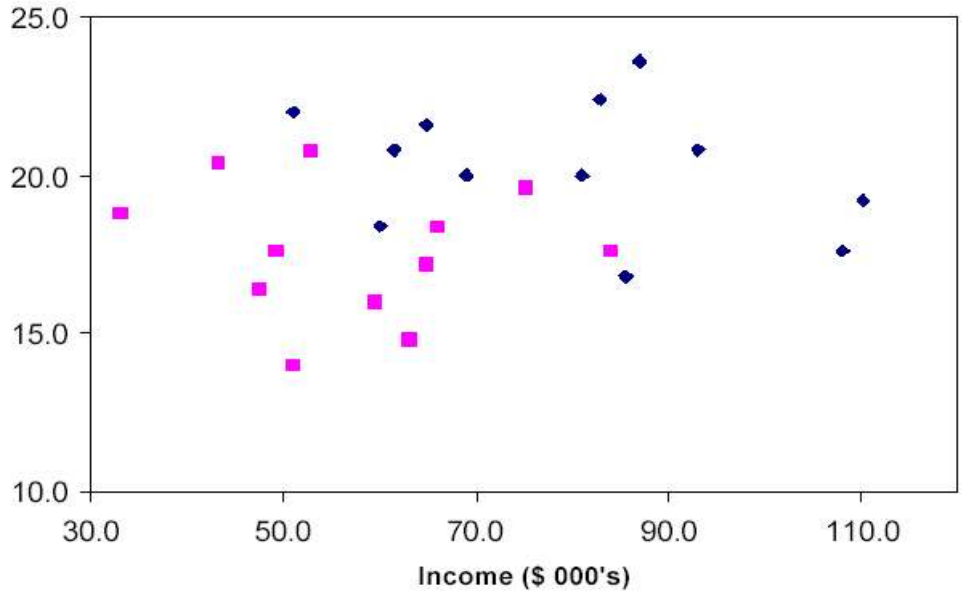
4 Première brique en Machine Learning : arbres de décision

- Algorithme CART
- Exemples
- Formalisation : construction de l'arbre
- Lien avec le problème de régression classique
- Gestion du surapprentissage : réduction de dimension
- Réponse catégorielle
- Outils et mesures de performance des modèles
- Extensions et conclusion

EXEMPLE 1 : ARBRE DE CLASSIFICATION

A travers cet exemple, on veut intuiter comment un arbre se construit... Cherchons à prévoir “propriétaire” | salaire + surface.

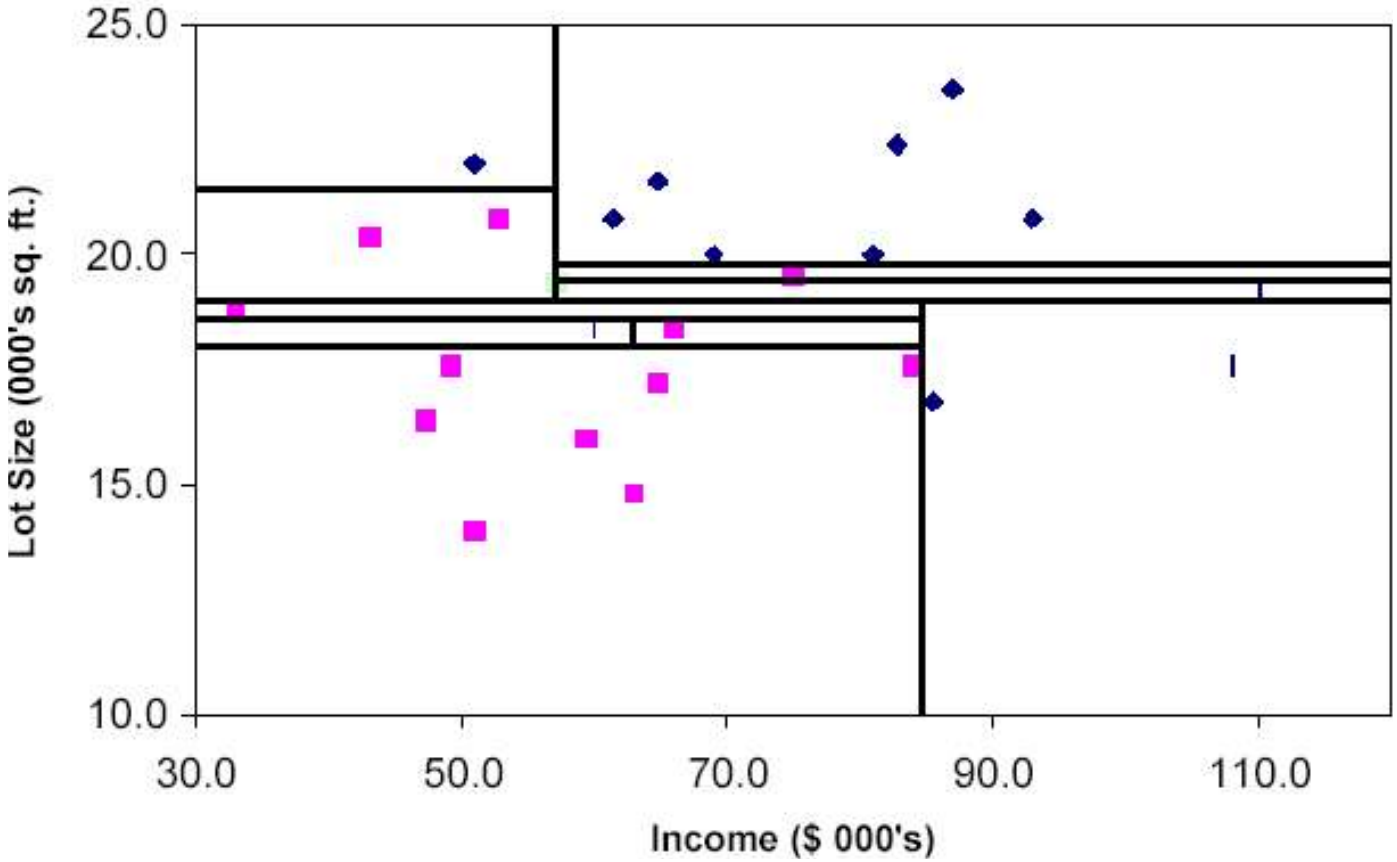
Income ($ 000's)	Lot Size (000's sq. ft.)	Owners=1, Non-owners=2
60	18.4	1
85.5	16.8	1
64.8	21.6	1
61.5	20.8	1
87	23.6	1
110.1	19.2	1
108	17.6	1
82.8	22.4	1
69	20	1
93	20.8	1
51	22	1
81	20	1
75	19.6	2
52.8	20.8	2
64.8	17.2	2
43.2	20.4	2
84	17.6	2
49.2	17.6	2
59.4	16	2
66	18.4	2
47.4	16.4	2
33	18.8	2
51	14	2
63	14.8	2



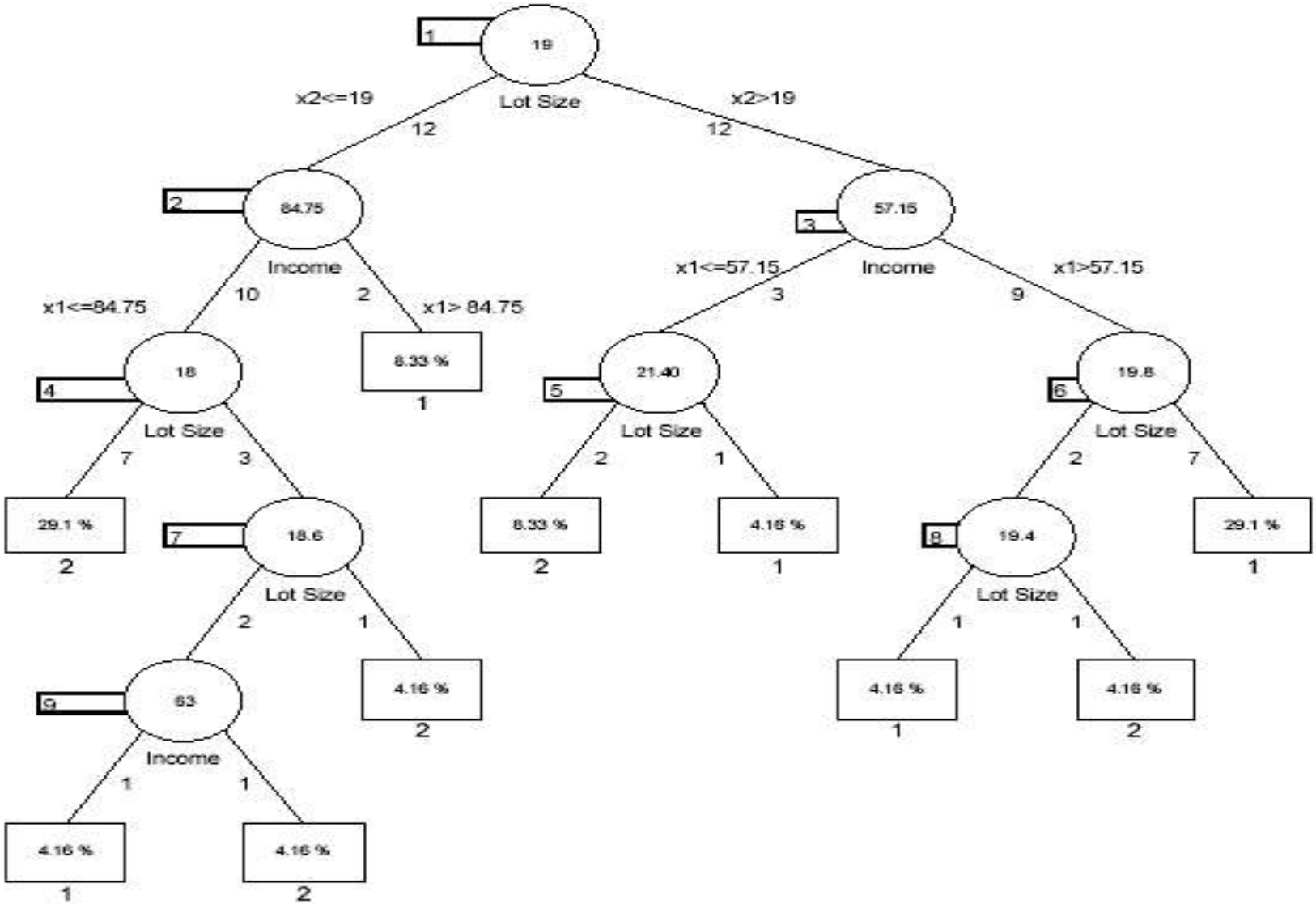
CHOISIR LA SEGMENTATION DE L'ESPACE

1. Choisir une var. explicative j donnée à m valeurs : soit elle est
   - numérique ou catégorielle ordonnée : partitionnements de l'espace associé à cette covariable se situent entre 2 de ses valeurs successives observées $\Rightarrow m - 1$ possibilités ;
   - catégorielle non ordonnée : partitionnements de χ_j sont toutes les combinaisons de modalités, au nb de $2^m - 1$;
2. Je teste tous ces partitionnements : j'y associe un critère d'homogénéité par rapport à ma quantité d'intérêt (réponse) ;
3. Je choisis le partitionnement qui conduit à la plus grande homogénéité dans les sous-espaces créés ;
4. Je répète les étapes (1)-(3) pour chacune des covariables dont je dispose : j'obtiens une liste de k homogénéités max. ;
5. Je choisis à la fin la covariable et son partitionnement qui maximise l'homogénéité globalement.

PARTITIONNEMENT ET ARBRE MAXIMAL



Partitionnement qui maximise l'homogénéité dans chq rectangle.


1
19
Lot Size
x2<=19
12
x2>19
12
2
84.75
Income
10
2
x1<=84.75
x1> 84.75
4
18
Lot Size
7
3
29.1 %
2
7
18.6
Lot Size
2
1
9
63
Income
1
1
4.16 %
1
4.16 %
2
4.16 %
2
8.33 %
1
3
57.15
Income
x1<=57.15
3
x1>57.15
9
5
21.40
Lot Size
2
1
8.33 %
2
4.16 %
1
6
19.8
Lot Size
2
7
8
19.4
Lot Size
1
1
4.16 %
1
4.16 %
2
29.1 %
1

4 Première brique en Machine Learning : arbres de décision

- Algorithme CART
- Exemples
- Formalisation : construction de l'arbre
- Lien avec le problème de régression classique
- Gestion du surapprentissage : réduction de dimension
- Réponse catégorielle
- Outils et mesures de performance des modèles
- Extensions et conclusion

NOTATIONS

- $i \in [\![1, n]\!]$: identifiant de l'individu / l'assuré ;
- $j \in [\![1, k]\!]$: identifiant du facteur de risque (continu ou discret) ;
- Y_i : réponse OBSERVEE du i$^{\text{ème}}$ individu (continue/discrète) ;
- $\mathbf{X}_i = (X_{i1}, ..., X_{ik})$: vecteur des facteurs de risque de l'indiv. i ;
- $\mathcal{X}$: espace des covariables (facteurs de risque) ;
- $l \in [\![1, L]\!]$: identifiant des feuilles de l'arbre ;
- $\mathcal{X}_l$: ensemble de la partition correspondant à la feuille l.

ARBRE DE RÉGRESSION AVEC Y CONTINUE

En régression, la quantité d'intérêt est

$$\pi_0(\mathbf{x}) = E_0[Y \mid \mathbf{X} = \mathbf{x}] \tag{1}$$

En supposant une relation lin. (se restreignant à une classe d'estimateurs), on a

$$\hat{\pi}(\mathbf{x}) = \hat{\beta}_0 + \mathbf{x}^T\hat{\beta},$$

et on estime les paramètres de régression par MCO.
En toute généralité, on ne peut pas considérer ts les estimateurs potentiels de $\pi_0(\mathbf{x}) \Rightarrow$ arbres sont 1 autre classe d'estimateurs : ce sont des **fonct. constantes par morceaux**.

Construire un arbre maximal génére une suite d'estimateurs selon une procédure spécifique : divisions successives de l'espace $\mathcal{X}$.

CONSTRUCTION DE L'ARBRE : CRITÈRE DE DIVISION

La ramification de l'arbre est **basée sur la définition d'un critère d'homogénéité**, cohérent avec l'estimation de la quantité d'intérêt.

Dans l'estimation de (1), *MCO* tjs utilisé car solution donnée par

$$\boxed{\pi_0(\mathbf{x}) = \arg\min_{\pi(\mathbf{x})} E_0[\Phi(Y, \pi(\mathbf{x})) \,|\, \mathbf{X} = \mathbf{x}],} \qquad (2)$$

où $\Phi(Y, \pi(\mathbf{x})) = (Y - \pi(\mathbf{x}))^2$.

La fonction de perte Φ correspond donc à l'erreur quadratique (fn. convexe), et le critère est la minimisation de l'EQM.

La $\neq$ est ici que l'on va estimer $\pi_0(\mathbf{x})$ en plusieurs étapes !

ETAPES DE CONSTRUCTION DE L'ARBRE

On résume donc l'enchainement des étapes de construction de l'arbre :

1. on part de la racine ;
2. on cherche la meilleure première segmentation (donnant le meilleur gain d'homogénéité) ;
3. on segmente ;
4. on itère sur chacun des 2 noeuds fils ;
5. on itère sur les fils des noeuds fils, et ainsi de suite...

Par construction l'hétérogénéité diminue à chaque segmentation, pour atteindre sa valeur minimale sur l'arbre maximal.

LIEN ENTRE RÉGRESSION ET ARBRE

Arbre $=$ ensemble de règles. Pour chaque noeud m, une règle R_m est associée à un sous-ensemble $\mathcal{X}_m \subseteq \mathcal{X}$.

Notation : dans la suite, $E_n[Y]$ désigne la moyenne empirique de Y, et $\mathcal{X}_{pa(m)}$ est le sous-ensemble associé au noeud parent de m.

L'arbre est associé à la fonction de régression

$$\hat{\pi}(\mathbf{x}) = \sum_{m=1}^{M} \hat{\beta}_m^{tree} \, R_m(\mathbf{x}) \tag{3}$$

où $\hat{\beta}_m^{tree} = E_n[Y \,|\, \mathbf{x} \in \mathcal{X}_m] - E_n[Y \,|\, \mathbf{x} \in \mathcal{X}_{pa(m)}]$ si $m \neq$ racine,
$\hat{\beta}_m^{tree} = E_n[Y]$ sinon.

Cela équivaut en régression classique à chercher

$$\hat{\beta}^{tree} = \arg\min_{\beta^{tree}} E_n\left[\left(Y - \sum \beta_m^{tree} R_m(\mathbf{x})\right)^2\right].$$

Depuis (3), en $\sum$ sur ts les noeuds, il reste les feuilles... :

$$\hat{\pi}(\mathbf{x}) := \hat{\pi}^L(\mathbf{x}) = \sum_{l=1}^{L} \hat{\gamma}_l \, R_l(\mathbf{x}) \tag{4}$$

$\Rightarrow$ Décomposition en bases fonctionnelles de $\mathbf{x}$ $\Rightarrow$ non-param !

- L est le nombre de feuilles de l'arbre, l leur indice,
- $R_l(\mathbf{x}) = \mathbb{1}(\mathbf{x} \in \mathcal{X}_l)$: règle d'appartenance au ss-ensemble $\mathcal{X}_l$,
- $\hat{\gamma}_l = E_n[Y \,|\, \mathbf{x} \in \mathcal{X}_l]$: moyenne empirique de Y dans la feuille l,
- Ss-ensembles $\mathcal{X}_l \subseteq \mathcal{X}$ disjoints ($\mathcal{X}_l \cap \mathcal{X}_{l'} = \emptyset$, $l \neq l'$) et exhaustifs ($\mathcal{X} = \cup_l \mathcal{X}_l$).

(4) généralisable qlq soit la quantité d'intérêt. Ainsi, **tout arbre peut être vu comme un estimateur par morceaux**.

→ Interprétation :

- chaque morceau est une feuille, dont la valeur est la moyenne empirique des valeurs de Y de cette feuille (cas quantitatif),

- chq div. d'1 noeud t minimise la $\sum$ variances intra-noeuds résultantes $\Rightarrow$ max. $\searrow$ hétérogénéité $H_t = 1/|t| \sum_{i \in t}(y_i - \bar{y}_t)^2$:

$$\max_{div.} \Big(H_t - (H_{t_g} + H_{t_d}) \Big) \Leftrightarrow \min \left(\frac{|t_g|}{n} \sum_{i \in t_g} (y_i - \bar{y}_{t_g})^2 + \frac{|t_d|}{n} \sum_{i \in t_d} (y_i - \bar{y}_{t_d})^2 \right)$$

où t_g et t_d désignent respectivement les fils gauche et droite du noeud parent t.

La construction étant **récursive**, on génère une suite d'estimateurs depuis le nd racine : soit une suite $\{\Pi^K\}$ de ss-espaces t.q. $\Pi^K \subseteq \Pi$,

$$\Pi^K = \left\{ \pi^L(.) = \sum_{l=1}^{L} \gamma_l \, R_l(.) \ : \ L \in \mathbb{N}^*, \, L \leq K \right\}. \tag{5}$$

A K fixé, on cherche $\pi_0^K(\mathbf{x}) = \underset{\pi(\mathbf{x}) \in \Pi^K}{\arg\min} \, E_0[\Phi(Y, \pi(\mathbf{x})) \,|\, \mathbf{X} = \mathbf{x}]$.

$\Rightarrow$ Version empirique $\hat{\pi}^K$: $\hat{\pi}^K(\mathbf{x}) = \underset{\pi(\mathbf{x}) \in \Pi^K}{\arg\min} \, E_n[\, \Phi(Y, \pi(\mathbf{x}))\,]$. Ou :

$$\boxed{\hat{\pi}^K(\mathbf{x}) = \underset{\gamma = (\gamma_1, \ldots, \gamma_L)}{\arg\min} \; E_n[\, \Phi(Y, \pi^L(\mathbf{x}))\,].} \tag{6}$$

CART ne cherche pas ts les estimateurs possibles avec $L \leq K$: approche ce minimum petit à petit.

ARRÊT DE LA PROCÉDURE DE SEGMENTATION

Comme déjà évoqué, l'algorithme CART ne fixe pas de règle d'arrêt arbitraire pour la procédure de division de l'espace.

L'algorithme arrête ainsi de diviser les feuilles quand :

- il n'y a qu'une observation dans la feuille, ou
- les individus de la feuille ont les mêmes valeurs de facteurs de risque (covariables **X**).

On construit ainsi l'*arbre "maximal"*, qui sera ensuite élagué.

Arbre maximal : estimateur par morceaux le $+$ complexe de la suite d'estimateurs construits $\rightarrow$ CV garantie (Breiman et al. 1984).

ILLUSTRATION ESTIMATEUR PAR MORCEAUX : EXEMPLE 2

Exemple en assurance : prévision de décès et modélisation des taux de mortalité. Résultats de l'article EAJ Olbricht (2012).

Portefeuille de SwissRe avec les caractéristiques suivantes :

- comprenant 1 463 964 enregistrements,
- couvrant une période de 4 ans,
- les variables explicatives en jeu sont le sexe et l'âge.

Les résultats obtenus par CART sont comparés à la table de mortalité actuelle "German standard life table DAV 2008 T".

ARBRE ÉLAGUÉ (PAS MAXIMAL !)

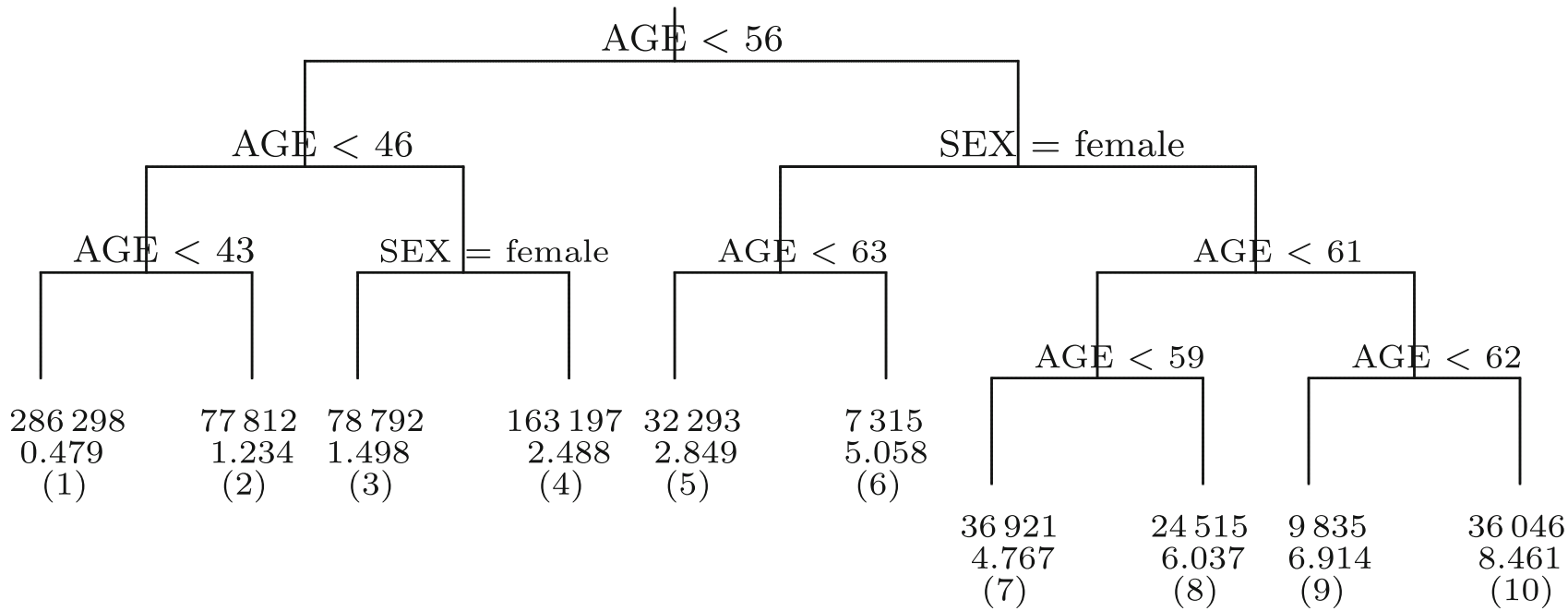
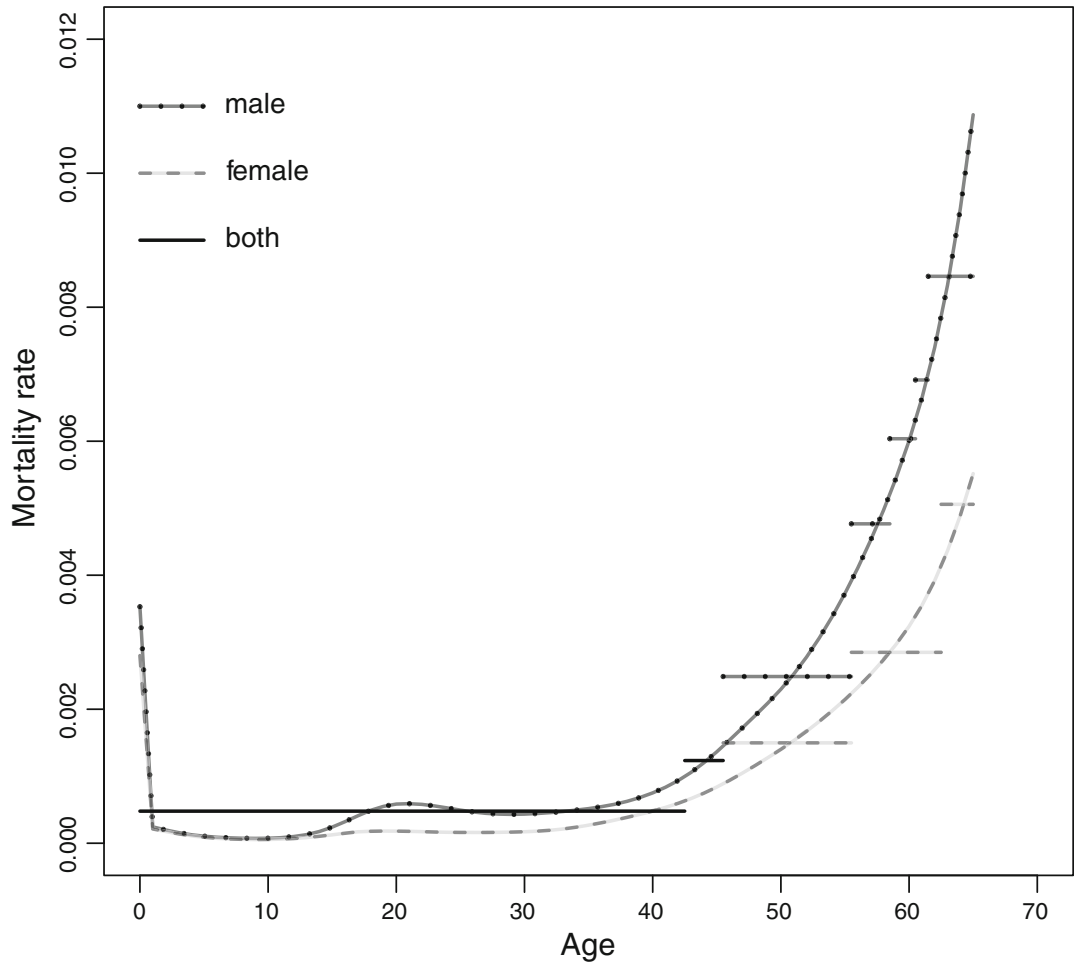


Fig. 8 Final tree for the standard life table example. For each terminal node the number of cases and the mortality rate (per mille) are given (the numbers in *brackets* are the labels for the nodes used in Table 6)

COURBES DE MORTALITÉ CORRESPONDANTES



Courbe continue : table réglementaire ; par morceaux : CART.

REMARQUE IMPORTANTE

Notez la différence majeure qu'il existe entre ce type de modélisation et une modélisation dite paramétrique.

En effet, on s'autorise toute forme de dépendance ici, alors qu'un modèle paramétrique (ex : GLM) impose une forme de dépendance entre Y et $\mathbf{X}$...
$\Rightarrow$ Peut s'avérer inadapté dans de nombreux cas pratiques ! (ex : tarification d'un contrat auto en incluant l'âge ds le modèle de fréquence, ss forme de classes d'âge).

En revanche, dans l'exemple de mortalité ici, il serait préférable d'avoir un modèle paramétrique...

PERFORMANCE DE LA PRÉVISION CART

La performance s'évalue sur le "test set" à droite du tableau :

Table 6 Performance of the tree from Fig. 8

Node	Learning set			Independent test set			
	No. of elements in node	No. of deaths in node	Estimated mortality rate (per mille)	No. of elements in node	No. of deaths in node	Tree prediction (Fig. 8)	Classical prediction (DAV 2008 T)
1	286,298	137	0.479	254,995	143	122	127
2	77,812	96	1.234	75,882	60	94	79
3	78,792	118	1.498	79,202	146	119	116
4	163,197	406	2.488	155,912	361	388	389
5	32,293	92	2.849	33,163	119	94	96
6	7,315	37	5.058	7,440	26	38	36
7	36,921	176	4.767	41,759	163	199	188
8	24,515	148	6.037	20,708	118	125	118
9	9,835	68	6.914	8,354	59	58	55
10	36,046	305	8.461	33,525	219	284	299
Total	753,024	1,583		710,940	1,414	1,521	1,503

4 Première brique en Machine Learning : arbres de décision

- Algorithme CART
- Exemples
- Formalisation : construction de l'arbre
- Lien avec le problème de régression classique
- Gestion du surapprentissage : réduction de dimension
- Réponse catégorielle
- Outils et mesures de performance des modèles
- Extensions et conclusion

SÉLECTION DE MODÈLE (α FIXÉ)

L'arbre maximal construit (de taille $K(n)$) génère une suite d'estimateurs $(\hat{\pi}^K(\mathbf{x}))_{K=1,\dots,K(n)} \Leftrightarrow$ chaque sous-arbre.

But : éviter estimateur trop complexe (surapprentissage) $\Rightarrow$ trouver meilleur sous-arbre selon un **arbitrage adéquation / prévision** :

$$R_\alpha(\hat{\pi}^K(\mathbf{x})) = E_n[\,\Phi(Y, \hat{\pi}^K(\mathbf{x}))\,] + \alpha\,(K/n),$$

où α param. de complexité, K dim. de l'estimateur (nb de feuilles).

Pour α fixé, l'estimateur final optimise un critère coût-complexité :

$$\boxed{\hat{\pi}^K_\alpha(\mathbf{x}) = \underset{(\hat{\pi}^K)_{K=1,\dots,K(n)}}{\arg\min} \; R_\alpha(\hat{\pi}^K(\mathbf{x})).} \qquad (7)$$

RESULTATS REMARQUABLES

$\rightarrow$ Pour α fixé, l'arbre $\hat{\pi}_{\alpha}^{K}(\mathbf{x})$ est unique et le calcul est rapide !

Exemples :

- $\alpha = \infty$: le modèle sélectionné sera la racine ;
- $\alpha = 0$: le modèle sélectionné sera l'arbre maximal.

$\rightarrow$ Puisque n'importe quelle *suite de sous-arbres emboîtés* de l'arbre maximal a au max. K membres, toutes les valeurs possibles de α peuvent être groupées en m intervalles ($m \leq K$) :

$$I_1 = [0, \alpha_1] \quad I_2 = (\alpha_1, \alpha_2] \quad ... \quad I_m = (\alpha_{m-1}, +\infty]$$

$\Rightarrow$ Chaque $\alpha \in I_j$ partage le même sous-arbre optimal.

PROCEDURE D'ELAGAGE

Raisonnement : impossible de parcourir ts les sous-modèles de l'arbre max. (nb sous-arbres exponent. $\nearrow$ avec nb feuilles) $\Rightarrow$

1. on part de l'arbre maximal construit ;
2. on considère **une 1$^{\text{ère}}$ valeur de** α : conduit à sélectionner un sous-arbre optimal de l'arbre maximal (cf équation (7)).
3. **à partir de ce sous-arbre optimal**, on prend une autre valeur de α ($+$ grande) qui conduit à sélectionner un sous-arbre optimal de ce sous-arbre.
4. Et ainsi de suite... Cela crée une suite croissante de α_z !

$\Rightarrow$ Par construction, on obtient une suite $\searrow$ de sous-arbres optimaux emboîtés (de l'arbre maximal vers la racine).

Dans cette liste d'estimateurs, on choisit finalement $\hat{\alpha}$ (et l'arbre optimal qui va avec) tel que

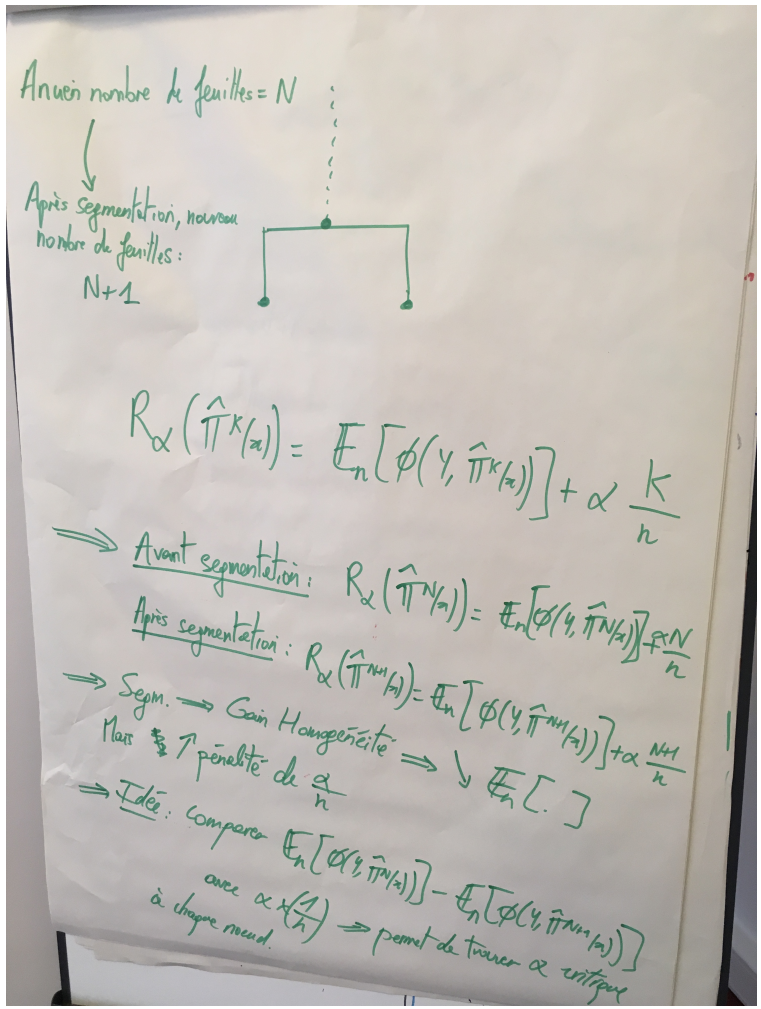
$$\boxed{\hat{\pi}^K_{\hat{\alpha}}(\mathbf{x}) = \underset{(\hat{\pi}^K_{\alpha_z})_{\alpha=\alpha_1,\dots,\alpha_Z}}{\arg\min} \; R_{\alpha_z}(\hat{\pi}^K_{\alpha_z}(\mathbf{x})).} \tag{8}$$

Remarque : en pratique,
$\rightarrow$ il faut déterminer les valeurs possibles de α !
$\rightarrow$ et $\hat{\alpha}$ est choisi en regardant cette erreur, mais moyennée via une validation croisée (pr minimiser une erreur de généralisation).

Consistance : Gey et Nedelec (2005) ; Molinaro, Dudoit et VanDerLaan (2004).

PROPOSITION DES VALEURS DE α

La suite des valeurs de α est obtenue lors de la construction de l'arbre maximal, avec le raisonnement suivant :



TUNING : CHOIX DE L'HYPERPARAMETRE α

$\rightarrow$ Tuning du modèle : sélection du paramètre de complexité α.
$\rightarrow$ Elagage : sélection de modèle pour un α fixé.
Comment choisir le meilleur paramètre de tuning α ?

Application à CART : une particularité... En effet, la validation croisée induit des séquences d'arbres emboîtés différentes.
$\Rightarrow$ L'erreur moyenne n'est pas calculée pour chaque sous-arbre avec un nb de feuilles donné, mais pour chaque valeur α_z fixée *issue de la séquence produite initialement par tout l'échantillon*.

Le choix de α répond à l'équation (8) (où l'erreur est moyennée) $\Rightarrow$ fournit le bon α et donc l'arbre optimal !
$\rightarrow$ En pratique, choisis 1er point en-dessous de min+1SE (Therneau : An Introduction to Recursive Partitioning Using the RPART Routines).

FORMULATION ALGORITHMIQUE (V-fold)

1. Construction de l'arbre maximal T_{max} ;
2. Construction de la séquence $T_K, ..., T_1$ d'arbres emboîtés associée à une séquence de valeurs (α_z) ;
3. Pour $v = 1, ..., V$ (où v désigne le segment de l'échantillon initial servant à la validation),
   - pr chq nouvel éch. d'apprentissage, construire T_{max} et estimer la séquence d'arbres associée à la séq. des pénalisations α_z,
   - estimation de l'erreur sur la partie validation de l'échantillon ;
4. Calcul de la séquence des moyennes de ces erreurs ;
5. L'erreur minimale désigne la pénalisation α_{opt} optimale ;
6. Retenir l'arbre associé à α_{opt} ds la suite initiale $T_K, ..., T_1$.

VALIDATIONS CROISÉES DANS `rpart`

Pour amener plus de robustesse au choix du paramètre de complexité α, on procède par validations croisées.

Principe de la validation croisée : meilleur compromis biais / variance. On diminue la variance de l'estimateur en recherchant une valeur réaliste de l'erreur basée sur plusieurs calibrations.

Dans le cadre de l'algorithme CART, cela consiste en les étapes :

1. Construire l'arbre maximal (modèle complet) sur l'échantillon ;
2. Déduire les intervalles I_1, I_2, ..., I_m à partir des α_z.
3. Construire la suite (β_z) (pour se placer dans les intervalles $]\alpha_k, \alpha_{k+1}]$) telle que

$$
\begin{aligned}
\beta_1 &= 0 \\
\beta_2 &= \sqrt{\alpha_1 \alpha_2} \\
\dots &= \dots \\
\beta_{m-1} &= \sqrt{\alpha_{m-2} \alpha_{m-1}} \\
\beta_m &= \infty
\end{aligned}
$$

4. Diviser l'échantillon d'origine en s sous-groupes $G_1, G_2, ..., G_s$ de taille s/n (n est la taille de l'échantillon de base).
5. Sur chaque sous-groupe i :
   - construire l'arbre maximal sur l'ensemble des sous-groupes sauf le groupe i, et déterminer les sous-arbres T_{β_1}, T_{β_2}, ..., T_{β_m},
   - prédire la quantité d'intérêt pour chaque observation du groupe i dans chaque modèle T_{β_z}, $1 \leq z \leq m$;
   - calculer l'erreur pour chaque sous-arbre.
6. Pour chaque β_z, sommer les erreurs des G_i. Prendre le paramètre de complexité β d'erreur minimale, et choisir T_β comme meilleur sous-arbre sur l'échantillon de base.

ARBRE DE CLASSIFICATION : Y DISCRÈTE

Supposons que $Y \in \{A, B\}$.

Dans le cas discret, la quantité d'intérêt est

$$\pi_0(\mathbf{x}) = E_0[1_{Y=A} \,|\, \mathbf{X} = \mathbf{x}] = \mathbb{P}(Y = A \,|\, \mathbf{X} = \mathbf{x})$$

Ici il faut adapter le critère d'homogénéité, donc la perte Φ.

On considère classiquement de

- l'indice de Gini,
- l'entropie.

ENTROPIE

La fonction d'entropie est classiquement définie pour $p \in [0, 1]$ par

$$f(p) = -p \log(p).$$

Appliqué aux CART, dans un pb à 2 classes $\{A, B\}$ pour Y, on définit l'hétérogénéité du noeud t (convention $0 \log(0) = 0$) comme

$$H_t = -2 \sum_{l=\{A,B\}} |t| \, p_t^l \log(p_t^l),$$

où p_t^l est la proportion de la classe l dans le noeud t.

On maximise $\searrow$ hétérogénéité, soit $\max_{div.} \; H_t - (H_{t_g} + H_{t_d})$.

CONCENTRATION DE GINI

La concentration de Gini est définie pour $p \in [0, 1]$ par

$$f(p) = p\,(1 - p).$$

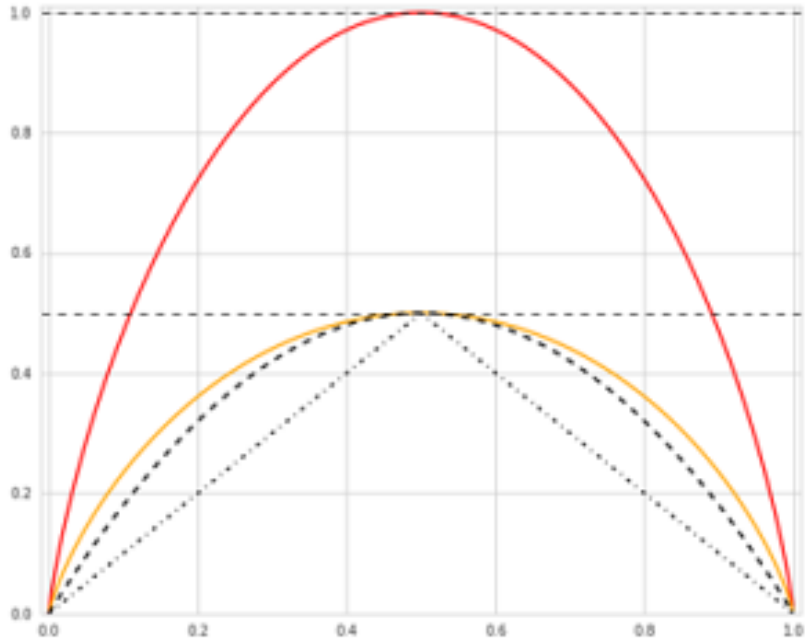
Appliqué aux CART, on définit l'hétérogénéité comme

$$H_t = \sum_{l=\{A,B\}} p_t^l\,(1 - p_t^l).$$

Rq :
- La concentration de Gini est la variance d'une Bernoulli...
- Proportions remplaçable par des proba. conditionnelles si proba. a priori des classes connues ($\neq$ proba. observées). Sinon, proba. de chq classe estimées sur l'éch. (revient à prendre proportion).

GRAPHIQUE DE L'ERREUR

Ds tous les cas, la quantité à optimiser sera convexe/concave.



⇒ Zones intéressantes : extrémités de $[0, 1]$.

AFFECTATION POUR PREVISION

Concernant l'affectation de l'observation à prédire à l'une des classes, il y a donc 3 distinctions possibles en fonction de l'information à disposition :

- soit on affecte la classe la plus représentée dans la feuille,
- soit on affecte la classe a posteriori la plus probable (au sens bayésien) si l'on dispose de probabilités a priori (pas les proba. de représentation dans l'échantillon) des classes,
- soit on affecte la classe la moins coûteuse si des coûts de mauvais classement sont donnés.

4 Première brique en Machine Learning : arbres de décision

- Algorithme CART
- Exemples
- Formalisation : construction de l'arbre
- Lien avec le problème de régression classique
- Gestion du surapprentissage : réduction de dimension
- Réponse catégorielle
- Outils et mesures de performance des modèles
- Extensions et conclusion

REPONSE QUANTITATIVE

Les mesures classiques de performance d'un modèle si Y est quantitative sont :

- l'**Erreur Quadratique Moyenne** (EQM, ou MSE) :

$$MSE(\hat{\pi}^K(\mathbf{x})) = \sum_i (Y_i - \hat{\pi}^K(\mathbf{x}_i))^2$$

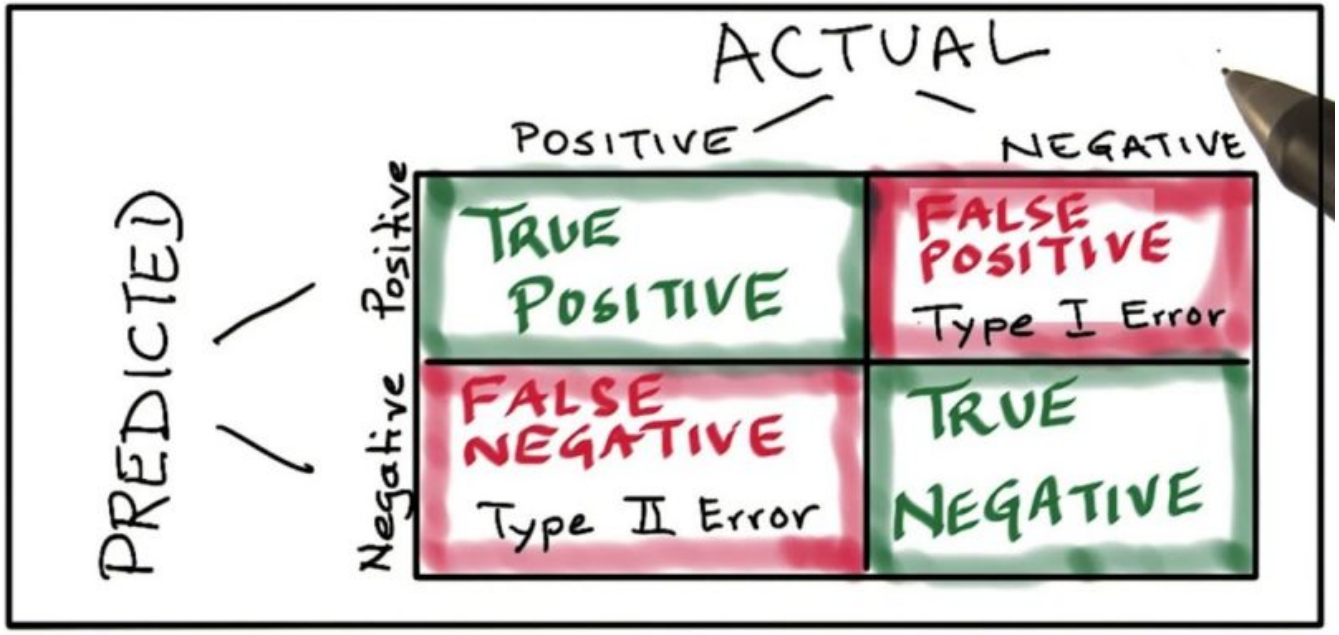
- l'**Erreur Absolue Moyenne** (EAM, ou MAE) :

$$MAE(\hat{\pi}^K(\mathbf{x})) = \sum_i |Y_i - \hat{\pi}^K(\mathbf{x}_i)|$$

Rq : évidemment, ces erreurs se mesurent sur un échant. test, $\perp\!\!\!\perp$ des échant. ayant servi à construire et tuner/optimiser le modèle...

REPONSE CATEGORIELLE : MATRICE DE CONFUSION

Dans un pb de classif., on utilise svt la matrice de confusion comme mesure de performance $\Rightarrow$ résume les indiv. mal classés et ceux bien classés par le modèle :



REMARQUES

En utilisant cet outil, on peut calculer facilement :

- le taux de mauvaise classification :

$$(FP + FN)/(FP + FN + TP + TN)$$

- l'indice de sensibilité : $TP/(TP + FN)$
- l'indice de spécificité : $TN/(TN + FP)$

Ds la pratique, on optimise svt le modèle par rapport à 1 des 2 indices, qui mène à la prudence du modèle (svt la spécificité, qui mesure la prédiction d'un événement rare...).

LIMITES DE CETTE MESURE

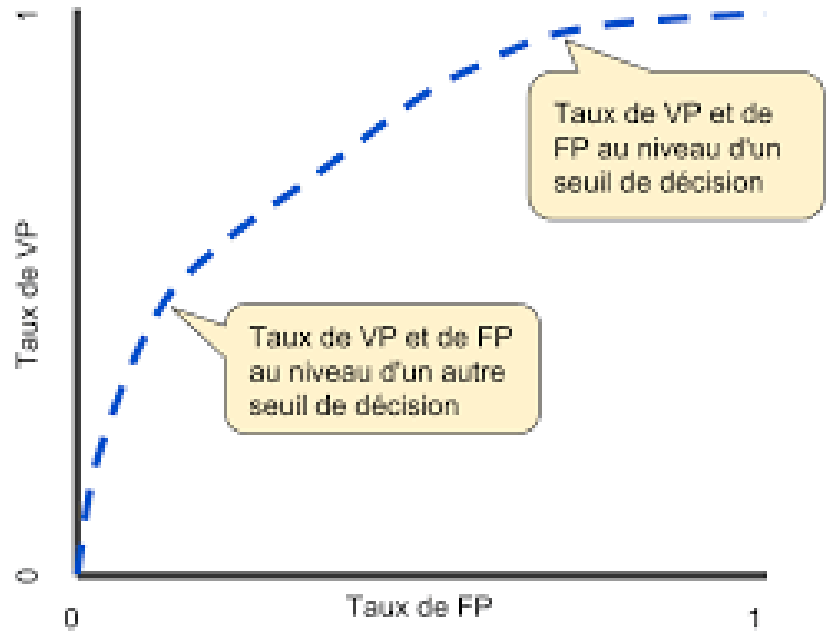
Principalement 2 limites à l'utilisation de cette matrice :

1. dépendante d'un seuil d'affectation : pour classer les prév. du modèle, on définit ce seuil. Dans un pb à 2 classes, svt 0,5 $\Rightarrow$ bien connu que ce n'est svt pas seuil optimal ($\Rightarrow$ ROC).

2. ds un pb où classes de Y sont largement disproportionnées, le modèle prédira tjs la même classe et donnera 1 erreur de classif. globalement très faible... Peu réaliste, car souvent c'est l'événement rare qu'il nous intéresse de prédire... Donc en fait l'erreur sur cette prévision est maximale, puisque l'événement en question n'est jamais prédit !

COURBE ROC ET AUC

ROC (Receiving Operator Curve) résume taux de VP (sensibilité) et FP (1-spécificité) pour ts les seuils d'affectation :



AUC (Area Under Curve) : $\in$ [0,5 (modèle aléatoire) ; 1 (parfait)].

AUTRES OUTILS : C-INDEX, F_1-SCORE

Au lieu d'utiliser la matrice de confusion pour optimiser un modèle, on peut aussi utiliser une mesure différente qui répond à une autre logique...

- le C-index (descendant de l'AUC...) : cf thèse Anani
- ex : article Pierrick ;
- F_1 score...permet de tuner les hyperparamètres en optimisant ce score ! (cf article Yohan Le Faou)

4 Première brique en Machine Learning : arbres de décision

- Algorithme CART
- Exemples
- Formalisation : construction de l'arbre
- Lien avec le problème de régression classique
- Gestion du surapprentissage : réduction de dimension
- Réponse catégorielle
- Outils et mesures de performance des modèles
- Extensions et conclusion

EXTENSIONS : AUTRES FONCTIONS DE PERTE Φ

$$\boxed{\pi_0(\mathbf{x}) = \arg\min_{\pi(\mathbf{x})} E_0[\Phi(Y, \pi(\mathbf{x})) \,|\, \mathbf{X} = \mathbf{x}]}$$

$\rightarrow$ Estimation de moyenne : $\quad \pi_0(\mathbf{x}) = E_0[Y \,|\, \mathbf{X} = \mathbf{x}]$
Critère de division (MCO) : $\quad \Phi(Y, \pi(\mathbf{x})) = (Y - \pi(\mathbf{x}))^2$.

$\rightarrow$ Quantile : $\quad \pi_0(\mathbf{x}) = Q_Y(\alpha \,|\, \mathbf{X} = \mathbf{x}) = \inf\{y : F(y \,|\, \mathbf{X} = \mathbf{x}) \geq \alpha\}$
$\Phi_\alpha(y, \pi(\mathbf{x})) = \alpha |y - \pi(\mathbf{x})| \mathbb{1}(y > \pi(\mathbf{x})) + (1 - \alpha) |y - \pi(\mathbf{x})| \mathbb{1}(y \leq \pi(\mathbf{x}))$

$\rightarrow$ Estimation de densité de la loi de Y :
$\Phi(Y, \pi(\mathbf{x})) = -\log \pi(Y, \mathbf{x})$, avec π la densité jointe de $(Y, \mathbf{X})$.

$\Rightarrow$ En pratique, version empirique de ces mesures pr l'estimateur !

DONNEES MANQUANTES : LES SURROGATE SPLITS

Dans la pratique, on n'observe pas certaines variables explicatives pour certains individus $\Rightarrow$ on ne peut pas les faire descendre dans l'arbre pour en déduire une prévision...

Dans ce cas, on impute la donnée manquante ou on utilise une surrogate split (obligatoirement basée sur une autre covariable !).

Correspond à la division **la** $+$ **voisine** de celle initialement choisie, en termes de concordance des individus envoyés dans chacun des noeuds fils $\Rightarrow$ imite au mieux la meilleure d'origine, mesurée par une mesure d'association entre 0 et 1 (1 est un clône).

PROBLEMATIQUES CLASSIQUES A GERER

Problème de biais de l'estimateur CART lorsqu'une variable explicative catégorielle contient trop de modalités... Tendance à attirer la règle de division à cette variable notamment.

Problème lorsque unbalanced response : on se retrouve qu'avec la racine et on ne segmente pas ! Que faire si on a juste la racine ?... cf https ://stats.stackexchange.com/questions/28029/training-a-decision-tree-against-unbalanced-data

Problème de censure, troncature...

CONCLUSION SUR CART

+ Algorithme simple, résultat facile à interpréter (règles, fournit pouvoir discriminant facteurs de risque).
+ Procédure statistique consistante théoriquement.
+ Méthode non-paramétrique, et invariante par transformation monotone des covariables (rangs utilisés) $\Rightarrow$ robustesse.
+ Adapté à la gestion de bc de var. explic. : sélection variables "intégrée" à l'algo. et intéractions implicitement considérées.
+ Extensions possibles avec adaptation de la perte.

- Algo récursif : peut passer à côté de l'optimum global...
- Instabilité aux données d'apprent. (variance estimateur) du fait de structure hiérarchique $\Rightarrow$ gagner en robustesse.

UN MOT SUR LA ROBUSTESSE PREDICTIVE

Certaines techniques ont été développées afin de stabiliser la prévision donnée par un estimateur arbre.

En effet, la construction d'un arbre optimal peut varier fortement quand bien même le jeu de données initial varie peu...
$\Rightarrow$ Proposer des estimateurs agrégés $\Rightarrow \searrow$ variance estimateur !

Pour éviter de corréler les estimateurs simples qui composeront l'estimateur agrégé, on peut intégrer par exemple

1. choix aléatoire des covariables considérées lors d'1 division ;
2. tirage aléatoire de sous-jeux de données.

UN MOT SUR LES STRATEGIES D'AGREGATION

Deux stratégies s'opposent dans le raisonnement :

→ **Stratégie d'agrégation aléatoire** (bagging : boostrap aggregating) : créer des échantillons, construire le modèle sur chq échantillon, combiner les modèles (ex : type forêts aléatoires).

→ **Stratégie alternative, apprentissage incrémental** (boosting) : apprentissage sur 1 paquet, prévision sur paquet 2, puis apprendre des exemples mal prédits du paquet 2, actualiser modèle, puis recommencer sur les paquets suivants ⇒ apprendre, mémoriser (ex : GBM).

LE BAGGING PLUS EN DETAIL
[FH00], [Bre94]

Le bagging conduit structurellement à diminuer la variance d'un estimateur.

En effet, n'importe quelle estimateur peut s'écrire à l'aide d'un développement de Taylor...Le premier terme étant la partie linéaire, les termes suivants étant des termes d'ordre supérieur. Le bagging ne touche pas au premier terme, mais considère l'espérance des termes suivants... Faisant ainsi diminuer la variance !

Conclusion : plus la quantité à estimer est linéaire (problème simple et dimension raisonnable), moins le bagging est efficace !

LE BOOSTING PLUS EN DETAIL
[FS97], [Fri01], [Sha03]

Le bagging conduit structurellement à diminuer la variance et le biais d'un estimateur.

...

5 Bagging + randomization de CART : forêts aléatoires

- Principe
- Construction de la fôret aléatoire
- Force, corrélation et erreur de la forêt
- Interprétabilité de modèles ensemblistes

PRINCIPES DES RANDOM FORESTS

Agrégation d'estimateurs CART.
L'objectif des forêts aléatoires est de proposer un estimateur "moyenné" afin d'améliorer la robustesse de l'estimation de la quantité d'intérêt ($\searrow$ variance estimateur agrégé).

Il s'agit d'intégrer une multitude de prévisions obtenues dans une estimation finale. Approche intéressante pour 2 raisons principales :

- on peut dégager un classement robuste du pouvoir explicatif de chacun des facteurs de risque,
- sa consistance a été démontrée dans plusieurs articles.

MAIS N'OUBLIEZ PAS...

"RF is an example of a tool that is useful in doing analyses of scientific data."

"But the cleverest algorithms are no substitute for human intelligence and knowledge of the data in the problem."

"Take the output of random forests not as absolute truth, but as smart computer generated guesses that may be helpful in leading to a deeper understanding of the problem."

Leo BREIMAN.

PRÉVISIONS : Y CONTINUE VS Y DISCRÈTE

Soit $\hat{Y}_i$ l'estimateur obtenu pour l'indiv. i par un CART **maximal** (pour diminuer le biais).
On construit N arbres CART en modifiant l'échantillon à chaque fois. Pour chaque obs. i, l'estimateur forêts aléatoires vaut :

- une moyenne dans le cas où Y est continue :

$$\hat{Y}_i^{RF} = \frac{1}{N} \sum_{n=1}^{N} \hat{Y}_{i,n}^{CART}$$

- un vote majoritaire si Y est discrète :

$$\hat{Y}_i^{RF} = \underset{k=A,B}{\arg\max} \left(\# \hat{Y}_{i,n}^{CART} = k \right)$$

5 Bagging + randomization de CART : forêts aléatoires

- Principe
- Construction de la fôret aléatoire
- Force, corrélation et erreur de la forêt
- Interprétabilité de modèles ensemblistes

RF = BAGGING + RANDOMIZATION

Les forêts aléatoires étaient basées sur plusieurs arbres CART.
Chacun de ces arbres est construit comme suit.

1. Construire un échantillon bootstrap de même taille que l'apprentissage (répliquer l'éch. selon mesure empirique) ;
2. Construire l'arbre CART sur cet échantillon bootstrap : considérons qu'il y a k facteurs de risque, avec $m << k$:
   - à chaque noeud, on tire aléatoirement m facteurs de risque parmi les k disponibles ;
   - on cherche la division optimale basée sur ces m covariables ;
   - où s'arrête-t-on dans la construction (cf slide suivante) ?
3. agréger ces arbres pour construire l'estimateur forêt.

Remarque : m ne change pas entre les $\neq$ arbres de la forêt.

TROIS STRATÉGIES D'ÉLAGAGE

Chaque arbre est-il élagué ? On distingue 3 stratégies $\neq$

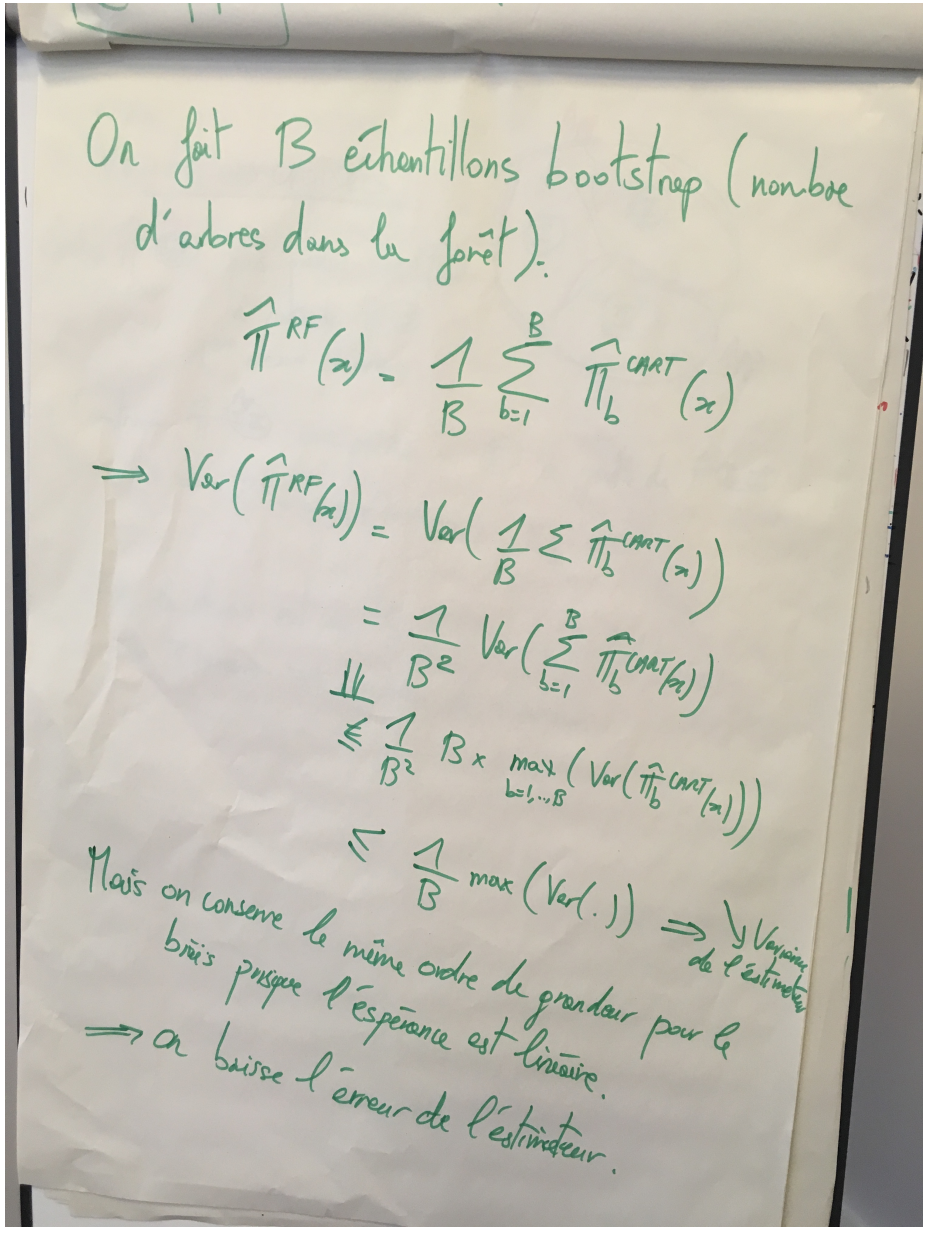
1. Laisser construire l'arbre maximal pour chacun des échant..
   $\rightarrow$ Bon compromis volume des calculs / qualité des prév. : faible biais et grande variance de chaque estimateur.

2. Construire un arbre d'au plus q feuilles $\rightarrow$ Cf plus loin...

3. Construire l'arbre maximal à chaque fois, puis l'élaguer par validations croisées $\rightarrow$ pénalise lourdement la quantité de calculs sans gain substantiel de qualité de prévision...

Rq : stratégie (1) implémentée par défaut dans `randomForest(.)`.

BOOTSTRAP, AGGRÉGATION ET RANDOM FORESTS



5 Bagging + randomization de CART : forêts aléatoires

- Principe
- Construction de la fôret aléatoire
- Force, corrélation et erreur de la forêt
- Interprétabilité de modèles ensemblistes

ERREUR DE LA FORÊT

L'erreur associée à la forêt dépend de 2 paramètres :

- la corrélation entre les arbres de la forêt : $+$ cette corrélation $\nearrow$, $+$ l'erreur est grande ;
- la capacité de chq arbre ds la forêt à donner une estimation proche de réalité (force) : $+$ l'arbre est précis, – erreur gde.

Par rapport au paramètre de tuning "m", on observe que

- abaisser *m* réduit la corrélation et la force,
- agrandir *m* augmente la corrélation et la force.

$\Rightarrow$ Arbitrage à trouver sur $m \rightarrow$ minimiser erreur O(ut)-O(f)-B(ag)

Rq : l'autre paramètre de tuning est le nombre d'arbres de la forêt.

L'ERREUR OOB

Au sein de la construction de chaque arbre CART de la forêt, on ne considère qu'une portion de l'échantillon bootstrap correspondant $\Rightarrow$ le reste constitue les **données "out-of-bag"**.

C'est sur ces données "out-of-bag" que sont calculées :

- une estimation non-biaisée de l'erreur de l'arbre,
- une estimation de l'importance des facteurs de risque.

Ici, pas de validation croisée pour avoir une estimation non-biaisée de l'erreur : on prend les obs. et prévisions chaque fois qu'elles sont dans l'éch. OOB $\rightarrow$ calcul erreur indiv. $\rightarrow$ moy. erreurs indiv.

TIRAGE ALÉATOIRE COVARIABLES CHAQUE ÉTAPE

$\rightarrow$ Randomization permet de diminuer la corrélation entre les arbres (rappel : les arbres sont ensuite agrégés), et de traiter le pb de covariables corrélées qui induisent un biais.

La variance de la moyenne de B estimateurs $\perp\!\!\!\perp$ (v.a.) vaut

$$Var(\frac{1}{B}\sum_{b=1}^{B} X_b) = \frac{1}{B^2} Var(\sum_{b=1}^{B} X_b) \simeq \frac{\sigma_b^2}{B}.$$

En revanche, si ces arbres sont corrélés 2 à 2, de coefficient de corrélation ρ, on obtient :

$$Var(\frac{1}{B}\sum_{b=1}^{B} X_b) \simeq \rho\sigma_b^2 + \frac{1-\rho}{B}\sigma_b^2.$$

Ainsi,

- si $\rho \to 0$, alors on retrouve le cas initial,
- si $\rho \to 1$, alors on a beau $\nearrow$ B, il restera toujours $\rho\sigma^2$.

Cela limite donc fortement l'avantage du bagging... !

La procédure de bagging est encore plus fructueuse si p (nb de facteurs de risque) est grand !

Conclusion : lors de l'agrégation, on $\searrow$ ainsi la variance de l'estimateur tout en conservant le meme ordre de grandeur pour le biais...l'erreur globale de l'estimateur diminue donc !

Grâce à cette randomization, la stratégie d'élagage peut être + élémentaire qu'en pur bagging (avec d'autres modèles), on pourrait adopter la stratégie (2) d'élagage...

Preuve :

→ Pourquoi "randomiser" ? prenons la variance de la moyenne de B variables i.d, chacune de variance σ^2 : $\mathrm{Var}\left(\frac{1}{B}\sum_{b=1}^{B} X_b\right) = \frac{1}{B^2}\mathrm{Var}\left(\sum_{b=1}^{B} X_b\right) = \frac{B}{B^2}\mathrm{Var}(X_b) = \frac{\sigma^2}{B}$

Si les variables sont i.d., mais corrélées 2 à 2 avec corrélation ρ :

$$\mathrm{Var}\left(\frac{1}{B}\sum X_b\right) = \frac{1}{B^2}\mathrm{Var}\left(\sum X_b\right) = \frac{1}{B^2}\Big[\underbrace{\mathrm{Var}(X_1)}_{\sigma^2} + \dots + \underbrace{\mathrm{Var}(X_B)}_{\sigma^2}\Big) + 2\underbrace{\mathrm{Cov}(X_1, X_2)}_{\rho} + \dots + 2\mathrm{Cov}(X_i, X_j)\Big]$$

$$= \frac{1}{B^2}\left[B\sigma^2 + 2\sum_{1 \le i < j \le B} \mathrm{Cov}(X_i, X_j)\right]$$

avec $\mathrm{Cov}(X_i, X_j) = E\big[(X_i - E(X_i))(X_j - E(X_j))\big]$

$\mathrm{Corr}(X_i, X_j) = \dfrac{\mathrm{Cov}(X_i, X_j)}{\sigma(X_i)\sigma(X_j)}$, $\sigma(X_i) = \sqrt{\sigma^2}$, $\sigma(X_j) = \sqrt{\sigma^2}$, σ^2

$$= \frac{\sigma^2}{B} + \frac{1}{B^2} \times 2 \times \sum_{1 \le i < j \le B} \sigma^2 \underbrace{\mathrm{Corr}(X_i, X_j)}_{\rho}$$

$$= \frac{\sigma^2}{B} + \frac{1}{B^2} \times 2 \times \rho\sigma^2 \times \underbrace{[1 + 2 + \dots + (B-2) + (B-1)]}_{\text{nb de termes de la somme} = \sum_{j=1}^{B-1} j = (B-1)\left(\frac{1+(B-1)}{2}\right)}$$

$$= \frac{\sigma^2}{B} + \frac{1}{B^2} \times 2 \times \rho\sigma^2 \times (B-1)\,\frac{1+(B-1)}{2}$$

$$= \frac{\sigma^2}{B} + \frac{\rho\sigma^2}{B^2}\left[(B-1) + (B-1)^2\right] = \frac{\sigma^2}{B} + \rho\sigma^2\left[\frac{1}{B} - \frac{1}{B^2} + \frac{B^2}{B^2} - \frac{2B}{B^2} + \frac{1}{B^2}\right]$$

$$= \frac{\sigma^2}{B} + \rho\sigma^2\left[\frac{1}{B} + 1 - \frac{2}{B}\right] = \frac{\sigma^2}{B} + \rho\sigma^2\left[1 - \frac{1}{B}\right] = \frac{\sigma^2}{B} + \rho\sigma^2 - \frac{\rho\sigma^2}{B}$$

$$= \rho\sigma^2 + \frac{(1-\rho)}{B}\sigma^2, \text{ à comparer avec le cas } \perp\!\!\!\perp = \frac{\sigma^2}{B} !$$

- si $\rho \to 1$, on a beau augmenter B, il restera toujours $\rho\sigma^2$! ⇒ limite l'avantage du bagging... ⇒ motive la randomisation pour ↓ ρ.

REMARQUES ADDITIONNELLES

→ La randomization permet de gérer également les covariables corrélées.
→ L'importance des facteurs de risque peut être calculée de 2 façons différentes.

- Mesurer l'importance par permutation des covariables (**shuffling**).

  → Permutation aléatoire des valeurs de la covariable entre individus, puis on prédit : + la qualité de prévision est dégradée, + le facteur de risque est important.

  En pratique, on calcule l'erreur OOB du b^e arbre sans et avec permutation de la covariable, puis on regarde l'écart. Puis on moyenne sur tous les arbres.

- Utiliser pour chaque variable dans chaque arbre la valeur de décroissance de l'indice de Gini.

En pratique : il est + simple de moyenner la $\searrow$ de Gini car elle est déjà calculée lors de la construction de l'arbre.

$\rightarrow$ Gestion des données manquantes : imputées comme suit,

- échantillon d'apprentissage : moyenne ou proximités ;
- échantillon de valid. : $\neq$ suivant que l'on observe Y ou non.

Rq : le fichier d'aide de `randomForest` détaille tout cela...

RÉSUMÉ SUR LE BAGGING

Finalement, le principe du bagging présente des avantages et des inconvénients...

(+) Simple à mettre en oeuvre et à comprendre ;

(+) Se programme facilement, qlq soit la méthode ;

(+) Diminue la variance de l'estimateur ;

(-) Temps de calcul parfois important : nécessité d'agréger un grand nombre de modèles avant de stabiliser l'erreur OOB ;

(-) stockage de tous les modèles (mémoire...) ;

(-) Perte de l'interprétabilité, sorte de boîte noire.

5 Bagging + randomization de CART : forêts aléatoires

- Principe
- Construction de la fôret aléatoire
- Force, corrélation et erreur de la forêt
- Interprétabilité de modèles ensemblistes

INTERPRETABILITE

Contrairement à des modèles simples de type régression linéaire ou arbres de décision (donc paramétrique ou non), les modèles ensemblistes sont difficilement interprétables...

En particulier, bien qu'il soit possible d'extraire une mesure d'importance des variables explicatives pour expliquer la réponse, il est complexe de déterminer l'impact quantifié d'une variation de valeur d'une covariable sur la réponse...

METHODES

Il existe plusieurs techniques permettant d'améliorer l'interprétabilité d'un modèle ensembliste.

On peut citer par exemple :

Essai 1 : extraire plusieurs arbres représentatifs de la forêt, suite à un clustering sur les arbres (Weinberg AI, Last M. Selecting a representative decision tree from an ensemble of decision-tree models for fast big data classification. J Big Data. 2019 ;6(1) :23

Essai 2 : Méthode de LIME

Essai 3 : Méthode de SHAP (Shapley)

Il y a aussi les PDP, ICE, ...

6 Agrégation de modèles par boosting

LES GRADIENT BOOSTING MACHINE (GBM) [SF12]

Nous avons vu un exemple de combinaison de modèles basé sur une stratégie aléatoire (bagging : par ex. avec forêts aléatoires).
$\rightarrow$ L'enjeu de l'agrégation par boosting est tout à fait $\neq$: il s'agit d'une stratégie adaptative (boosting).

$\Rightarrow$ Améliore l'ajustement par 1 construction adaptative séquentielle d'estimateurs, puis une combinaison de ces estimateurs pour éviter le surapprentissage.

Rq : les principes de bagging / boosting concernent tte modélisation...mais ont principalement un intérêt dans le cas de modèles instables (ex : CART...) !

EVOLUTION PAR RAPPORT AU BAGGING

On traite le problème du biais de l'estimateur en plus de traiter la réduction de variance.

En effet, l'agrégation par bagging ne corrige pas le biais... puisque l'espérance est un opérateur linéaire, et le biais est défini par une espérance.

Or, dans le cas d'arbres individuels simples ("weak learner"), le biais peut être important.

$\Rightarrow$ Le boosting construit une famille de modèles récurrente : chaque modèle est une version adaptative du précédent.

PROCEDURE DU BOOSTING

On peut écrire le boosting comme suit.

A la 1ère étape, on estime le modèle m_1 pour $\mathbf{y}$, à partir de $\mathbf{x}$.
$\Rightarrow$ On en déduit le vecteur d'erreurs $\epsilon_\mathbf{1}$.

A la 2ème étape, on estime le modèle m_2 pour $\epsilon_\mathbf{1}$, à partir de $\mathbf{x}$.
$\Rightarrow$ On en déduit le vecteur d'erreurs $\epsilon_\mathbf{2}$.

On réitère ce procédé...et on obtient à l'étape k :

$$m^{(k)}(\mathbf{x}) = \underbrace{m_1(\mathbf{x})}_{\sim y} + \underbrace{m_2(\mathbf{x})}_{\epsilon_\mathbf{1}} + ... + \underbrace{m_k(\mathbf{x})}_{\epsilon_{\mathbf{k-1}}} = m^{(k-1)}(\mathbf{x}) + m_k(\mathbf{x}).$$

STRATÉGIE ADAPTATIVE

Pour s'adapter de proche en proche, on donne + de poids dans l'estimation suivante aux observ. mal prédites précédemment.

Intuitivement, l'algorithme concentre ses efforts sur les observ. les + difficiles à ajuster, tout en limitant l'overfitting par l'agrégation...

Les ≠ algo. de boosting diffèrent par leurs caractéristiques :

- la façon de pondérer l'importance des indiv. mal estimés ;
- la façon de pondérer les modèles lors de l'agrégation ;
- leur objectif (prédire Y réelle, binaire, ...)) ;
- la fonction de perte qui mesure l'erreur d'ajustement (+ ou - sensible aux valeurs atypiques par ex.)

ALGORITHME D'ORIGINE : ADABOOST

Au départ, cet algorithme est proposé pour un problème de discrimination à 2 classes.

Notons δ la fonction de discrimination, à valeurs dans $\{-1, 1\}$.

Algorithme :

1. Soit y_0 à prévoir (connaissant x_0), et $z = \{(x_1, y_1), ..., (x_n, y_n)\}$ un échantillon.

2. On initialise les poids (équipondération au départ) :

$$\omega = \{\omega_i = \frac{1}{n};\ i = 1, ..., n\}$$

3. De $m = 1$ à M (m est le m$^{\text{ième}}$ modèle) :
   1. on estime δ_m sur l'échantillon pondéré par ω.
   2. on calcule le taux d'erreur apparent : $\hat{\epsilon}_p = \sum_{i=1}^{n} \omega_i \mathbb{1}_{\delta_m(x_i) \neq y_i}$.
   3. on calcule les logit relatifs au modèle m : $c_m = \ln\left(\frac{1-\hat{\epsilon}_p}{\hat{\epsilon}_p}\right)$.
      Ainsi, $\hat{\epsilon}_p \nearrow \Rightarrow\ c_m \searrow \Rightarrow$ on pondérera + les bons modèles.
   4. on met à jour les pondérations :

$$\omega_i = \omega_i \ \exp\left(c_m \mathbb{1}_{\delta_m(x_i) \neq y_i}\right)$$

      Ainsi, on pondère + les observations mal classées...
   5. $m \leftarrow m + 1$, retour à l'étape 1 de la boucle.

4. Résultat du vote :

$$\hat{\Phi}_M(x_0) = \text{signe}\left[\sum_{m=1}^{M} c_m \, \delta_m(x_0)\right].$$

Remarque : il faut vérifier à chaque étape que le modèle courant fait mieux qu'une prévision aléatoire, i.e.

$$\hat{\epsilon}_p < 0,5.$$

Effectivement, le poids c_m du modèle correspondant devient négatif sinon !

De nombreuses adaptations de cet algo. ont été proposés, avec des fonctions de perte adaptées aux cas où :

- Y quantitative,
- Y qualitative à plusieurs modalités,
- ...

AUTRES PONDÉRATIONS

Parfois, on utilise des classifieurs pour lesquels il est difficile (voire impossible) d'intégrer une pondération des observations...

La stratégie revient à créer aléatoirement des échantillons (un peu comme en bootstrap), en procédant comme suit :

- chaque modèle sera construit sur un nouvel échantillon ;
- la proba. de tirer (avec remise) chaque observ. est inversement proportionnelle à sa qualité d'ajustement dans l'itération précédente.

C'est ce qu'on appelle des arcing classifiers (adaptively resample and combine) (voir les travaux de Breiman).

ADABOOST AVEC Y CONTINUE

On est donc dans un cadre de régression, où Y est quantitative.

Algorithme :

1. Soit y_0 à prévoir (connaissant x_0), et $z = \{(x_1, y_1), ..., (x_n, y_n)\}$ un échantillon.

2. On initialise un vecteur de proba. p par une loi uniforme (équipondération) : $p = \{p_i = 1/n\}$
3. Pour $m = 1$ à M (m est le m$^{\text{ième}}$ modèle) :
   1. on tire avec remise dans z un échantillon $z_m^\star$ suivant p .
   2. on estime $\hat{\Phi}_m$ sur $z_m^\star$.
   3. on calcule sur l'échantillon initial z les quantités :

- $l_m(i) = Q(y_i, \hat{\Phi}_m(x_i))$, pour $i = 1, ..., n$ et Q la perte ;
- $\hat{\epsilon}_m = \sum_i p_i l_m(i)$
- $\omega_i = g(l_m(i))\, p_i$, avec g continue décroissante ;
- on met à jour les proba. de tirage : $p_i = \frac{\omega_i}{\sum_i \omega_i}$.

4. Prévision du modèle agrégé : $\hat{\Phi}_M(x_0)$ est la moyenne (ou médiane) des prévisions $\hat{\Phi}_m(x_0)$, pondérée par des poids $\ln(1/\beta_m)$ (cf ci-dessous pour β_m).

Remarques :

- Q : souvent perte quadratique, mais p-ê une autre fonction !
- $\beta_m = \frac{\hat{\epsilon}_m}{L_m - \hat{\epsilon}_m}$, avec $L_m = \sup_i l_m(i)$, et $g(l_m(i)) = \beta_m^{\frac{1-l_m(i)}{L_m}}$
- Condition supp. ajoutée : arrêt et réinitialisation à des poids uniformes si $\hat{\epsilon}_m < 0.5 L_m$ (erreur trop dégradée) ;
- β_m : indicateur de la performance du prédicteur m sur z.

VISION PAS À PAS D'ADABOOST

Comme nous l'avons vu, cet algorithme fonctionne pas-à-pas : c'est la raison pour laquelle on l'appelle le Gradient Boosting (déplacement : opposé de la pente de fonction à minimiser, cf expansion de Taylor et algo. de Newton).

Une manière d'écrire l'optimisation avec cette vision à l'étape m (rappel : c est lié à la performance du modèle) :

$$(c_m, \gamma_m) \quad = \quad \arg\min_{(c,\gamma)} \sum_{i=1}^{n} Q(y_i, \hat{\Phi}_{m-1}(x_i) + c\delta(x_i, \gamma)).$$

GENERALISATION : BOOSTING, GRADIENT ADAPTATIF

En d'autres termes, on construit une suite de modèles

$$m^{(k)}(\mathbf{x}) = m^{(k-1)}(\mathbf{x}) + \alpha \; f^{\star}(\mathbf{x})$$

où

$$f^{\star}(\mathbf{x}) = \arg\min_{f \in \mathcal{W}} \left(\sum_{i=1}^{n} l\Big(y_i - m^{(k-1)}(\mathbf{x}_i), f(\mathbf{x}_i)\Big) \right),$$

avec l fonction de perte, et $\mathcal{W}$ un ensemble de weak-learners...

Rq : ces weak-learners sont svt des CART ("stumps")...Gradient Tree Boosting !

PARAMETRES DE TUNING

Dans cet algorithme, voici les paramètres de tuning :

- nombre d'itérations : combien d'étapes M considérer ?
- profondeur des arbres, ...
- "shrinkage" $\alpha \in [0, 1]$: assurer une convergence lente !
  Plutôt que $\epsilon_1 = y - m_1(\mathbf{x})$, on considère

$$\epsilon_1 = y - \alpha\, m_1(\mathbf{x}).$$

Rq :
- Algo. très performant car peut corriger biais et variance.
Néanmoins, les multiples param. de tuning rendent la gestion du surapprentissage difficile...
- En reg. lin., $\epsilon \perp\!\!\!\perp \mathbf{X} \Rightarrow$ impossible d'apprendre de nos erreurs !

XGBOOST

eXtreme Gradient BOOSTing

En...

En...

7 Réseau de neurones et Deep Learning

- Introduction
- Neurone formel et fonctionnement d'un perceptron multicouches
- Estimation des paramètres
- Paramétrage du réseau
- Deep learning et autres types de réseaux

BIBLIOGRAPHIE ACTUARIELLE

	Pricing	Reserving	Telematics	Mortality Forecasting	Quantitative Risk Management
Feed-forward Nets	• Ferrario, Noll and Wüthrich (2018) • Noll, Salzmann and Wüthrich (2018) • Wüthrich and Buser (2018)	• Castellani, Fiore, Marino et al. (2018) • Doyle and Groendyke (2018) • Gabrielli and Wüthrich (2018) • Hejazi and Jackson (2016, 2017) • Wüthrich (2018) • Zarkadoulas (2017)	• Gao and Wüthrich (2017) • Gao, Meng and Wüthrich (2018) • Gao, Wüthrich and Yang (2018)		• Castellani, Fiore, Marino et al. (2018) • Hejazi and Jackson (2016, 2017)
Convolutional Neural Nets			• Gao and Wüthrich (2019)		
Recurrent Neural Nets		• Kuo (2018a, 2018b)		• Nigri, Levantesi, Marino et al. (2019)	
Embedding Layers	• Richman (2018) • Schelldorfer and Wüthrich (2019) • Wüthrich and Merz (2019)	• Gabrielli, Richman and Wüthrich (2018) • Gabrielli (2019)		• Richman and Wüthrich (2018)	
Autoencoders			• Richman (2018)	• Hainaut (2018) • Richman (2018)	

Source : Ronald Richman.

UN PEU D'HISTOIRE

Réseaux de neurones sont une branche de l'IA (Intellig. Artific.) qui a pour but de simuler le comportement du cerveau humain.

→ Approche connexionniste (connaissance répartie), avec des couches... :

1. entrée,
2. coeur,
3. sortie.

Ds les années 1970, mise en oeuvre difficile car puissance des ordinateurs limitée ⇒ développement de l'approche séquentielle ou symbolique → systèmes experts à connaissance localisée.

EXPERTISE HUMAINE

But : automatiser le principe de l'expertise humaine via 3 concepts :

1. une base de connaissances : propositions logiques élémentaires,
2. une base de faits : données, observations,
3. un moteur d'inférence : applique les règles expertes sur la base des faits.

$\Rightarrow$ En déduit de nv faits (expérience) jusqu'à réaliser l'objectif !

Pb : complexité...(algorithmiquement, et en termes de modélisation)

PRINCIPAUX RÉSULTATS UTILISÉS

Finalement, les réseaux de neurones se sont développés grâce à l'essor de l'informatique...
Et l'approche connexionniste a été relancée, grâce notamment aux deux résultats théoriques principaux suivants :

- l'estimation du gradient par rétropropagation de l'erreur (Hopkins, 1982) ;
- l'analogie avec les modèles Markoviens en mécanique statistique (Hopfield, 1982).

Remarque : large variété d'applications, technique complémentaire de méthodes stats usuelles (MLE, ...).

RÉSEAU NEURONAL

Réseau neuronal : association de neurones formels $\Rightarrow$ créé un graphe + ou - complexe d'objets élémentaires.

Les $\neq$ réseaux se distinguent par 4 composantes :

1. organisation du graphe (couches, ...) ;
2. niveau de complexité (nb neurones, ...) ;
3. type des neurones (transition, activation) ;
4. objectif (apprentissage supervisé ou non, ...).

LE NEURONE FORMEL

Défini sur la base du fonctionnement d'un neurone biologique !
C'est un modèle caractérisé par :

- un état interne, noté $s \in \mathcal{S}$;
- des signaux d'entrée, notés $x_1, ..., x_p$;
- une fonction d'activation :

$$s = h(x_1, ..., x_p) = f(\alpha_0 + \sum_{j=1}^{p} \alpha_j x_j) = f(\alpha_0 + \alpha^T x).$$

Voc : on appelle α le vecteur des poids, α_0 le biais du neurone.
Rq : les poids α sont estimés durant l'apprentissage : mémoire ou "connaissance répartie" du réseau.

TYPES DE NEURONE

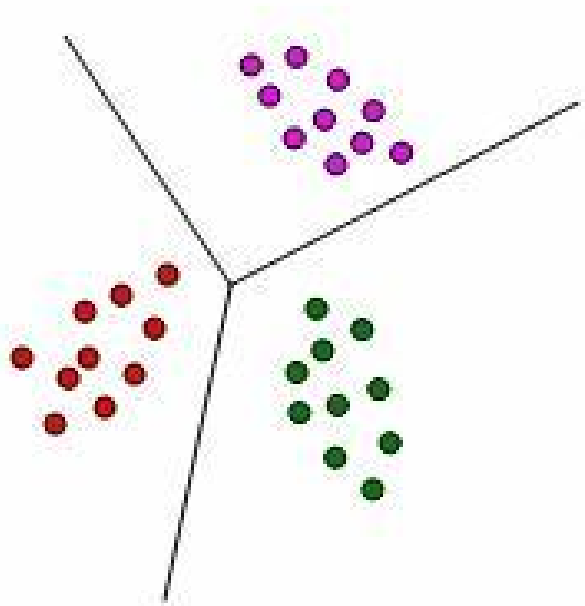
$\neq$ types de neurone se distinguent par leur fonction d'activation f :

- type linéaire : $f(x) = x$
- type sigmoïde : $f(x) = (1 + e^{-x})^{-1}$
- type seuil : $f(x) = \mathbb{1}_{[0,+\infty[}(x)$
- type radiale : $f(x) = \sqrt{\frac{1}{2\pi}} \exp\left(-\frac{1}{2}x^2\right)$
- type stochastique : $f(x) = 1$ avec proba $(1 + e^{-x/H})^{-1}$, 0 sinon ; ...

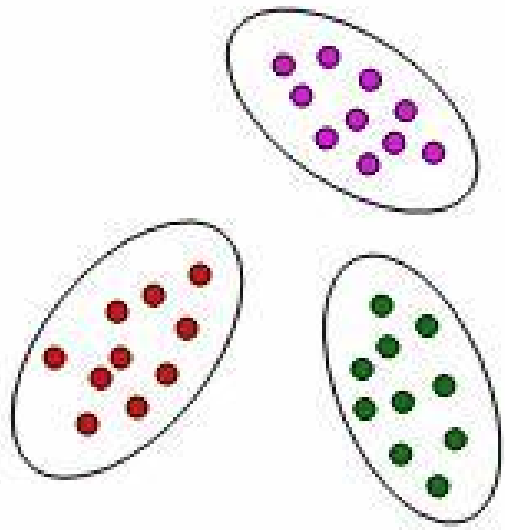
Rq : en data mining, les 2^{iers} types de réseaux sont les + utilisés car fonction d'activation est différentiable $\rightarrow$ adapté à un algo. d'apprentissage impliquant la rétropropagation du gradient.

DIFFERENCES DE SEPARATION

En fonction de l'activation choisie, les données sont séparées différemment. Exemple ici : fonction linéaire VS fonction radiale...



PMC

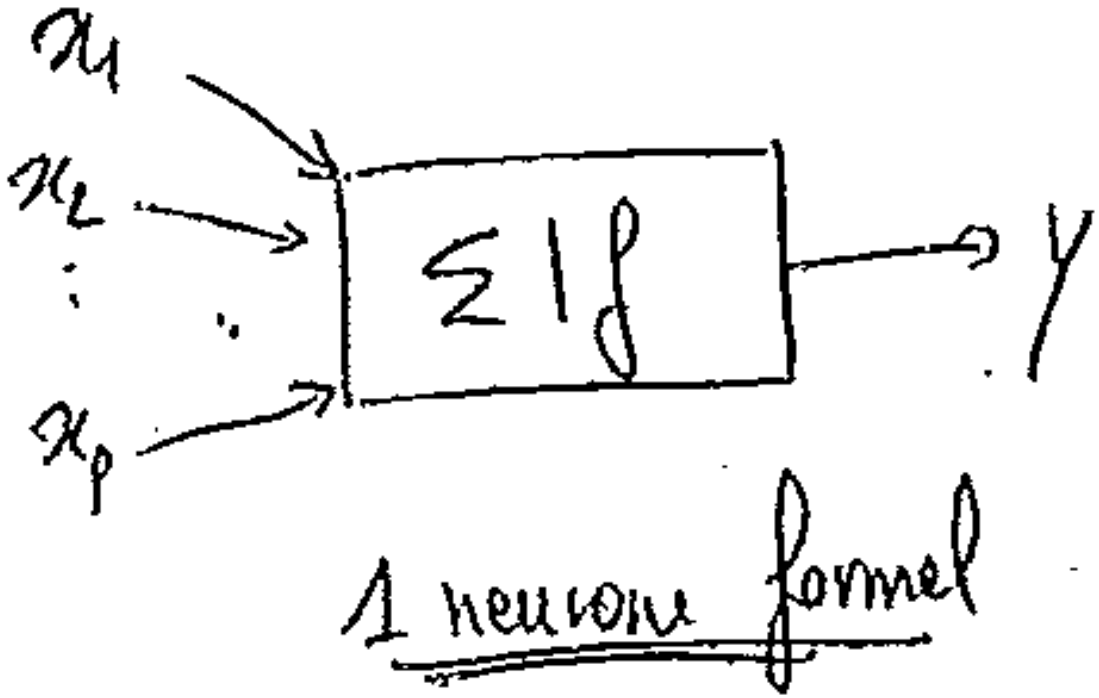


RBF

AUTRES FONCTIONS D'ACTIVATION

Fonction	Définition	Description	Intervalle de définition
Identité	a	L'activation du neurone est transmise directement en sortie	$(-\infty,+\infty)$
Sigmoïdale logistique	$\frac{1}{1+e^{-a}}$	Une courbe en "S"	$(0,1)$
Tangente hyperbolique	$\frac{e^{a}-e^{-a}}{e^{a}+e^{-a}}$	Une courbe sigmoïdale similaire à la dfonction logistique. Produit généralement de meilleurs résultats que la fonction logistique en raison de sa symétrie. Idéale pour les perceptrons multicouches, en particulier, pour les couches cachées	$(-1,+1)$
Exponentielle	e^{-a}	La fonction exponentielle négative	$(0,+\infty)$
Sinus	$\sin(a)$	S'utilise éventuellement si les données sont distribuées radialement. N'est pas utilisé par défaut	$[0,1]$
Softmax	$\frac{\exp(a_i)}{\sum \exp(a_i)}$	Essentiellement utilisé (mais pas uniquement) pour des tâches de classification. Permet de construire des réseaux de neurones avec plusieurs sorties normalisées ce qui le rend particulièrement adapté à la création de classifications par les réseaux de neurones avec des sorties probabilistes.	$[0,1]$
Gaussienne	$\frac{1}{\sqrt{2\pi}\sigma}\exp\left[\frac{(x-\mu)^2}{2\sigma^2}\right.$	Ce type de fonction d'activation isotropique Gaussienne n'est utilisé que par les unités cachées d'un réseau de	

SCHÉMA DE FONCTIONNEMENT



RÉSULTAT FONDAMENTAL

Le résultat suivant est la base de l'approche de modélisation par un réseau de neurones.

Théorème d'approximation "universelle" :

Toute fonction régulière peut être approchée uniformément avec une précision arbitraire et dans un domaine fini de l'espace de ses variables par un réseau de neurones comportant une couche de neurones cachés (en nombre fini et possédant tous la même fonction d'activation), et un neurone de sortie de type linéaire.

LE PERCEPTRON MULTICOUCHES (PMC)

Intéressons nous ici à un réseau “statique” (ou *feedforward*, i.e. pas de boucle rétroactive), dans un but d’apprentissage supervisé.

En voici quelques caractéristiques :

1. architecture : PMC composé de couches successives, où 1 couche : ens. de neurones sans connexion entre eux ;

2. fonction de transfert : un PMC réalise une transformation des variables d’entrée :

$$Y = \Phi(X_1, X_2, ..., X_p;\ \alpha),$$

avec $\alpha = (\alpha_{jkl})$ pr la j$^{\text{è}}$ entrée (x_j) du k$^{\text{è}}$ neurone de l$^{\text{è}}$ couche.

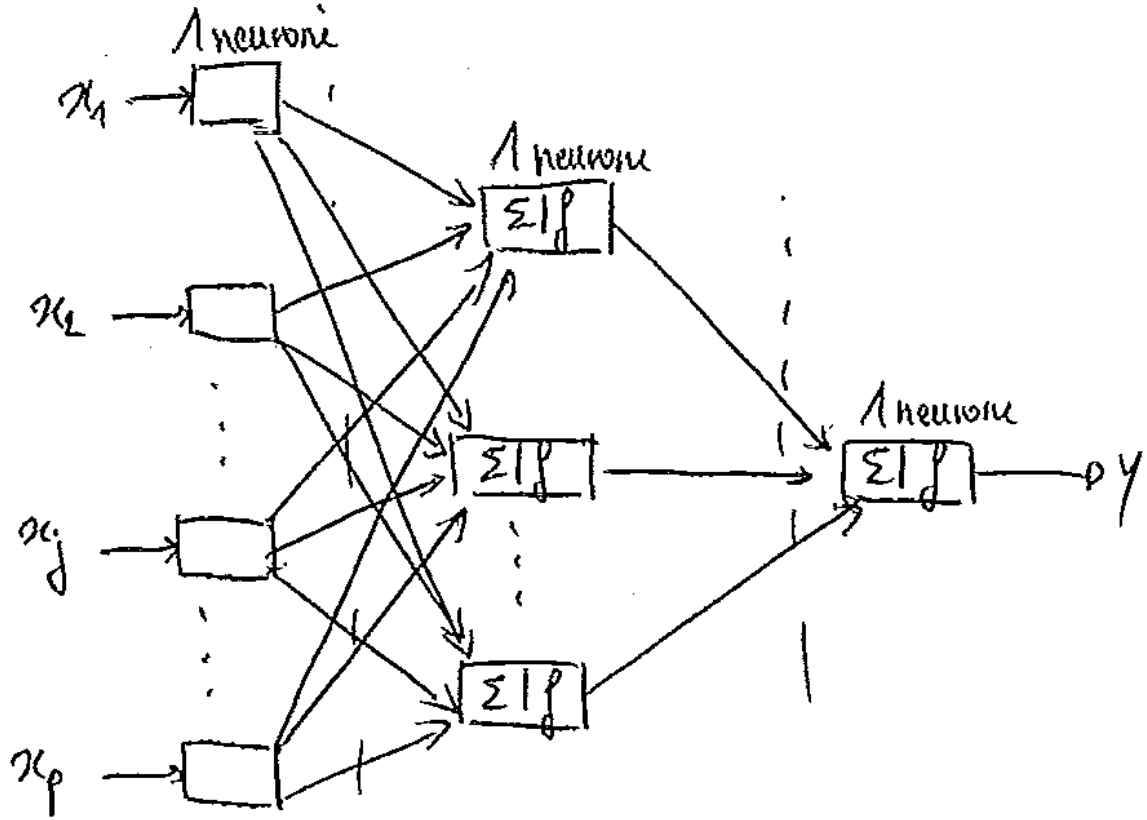
3. généralisation : cas de la régression, avec un perceptron à une couche cachée de q neurones, un neurone de sortie.
   $\Rightarrow$ La fonction de transfert s'écrit

$$Y = \Phi(x; \alpha, \beta) = \beta_0 + \beta^T z, \quad \text{avec } z_k = f(\alpha_{k0} + \alpha_k^T x),$$

pour $k = 1, ..., q$ (identifiant neurone ds couche cachée).

Usuellement, on a

- en rég. : dernière couche avec 1 seul neurone, avec $f = Id$; tandis que neurones couche cachée ont une fonct. sigmoïde ;
- en classif. binaire : neurone de sortie muni de la fonction d'activation sigmoïde ;
- en discrimination à m classes : m neurones de sortie, munis de sigmoïde.


1 neurone
1 neurone
1 neurone
Σ | f
Σ | f
Σ | f
Σ | f
x_1
x_2
x_j
x_p
y
couche d'entrée
couche cachée
couche de sortie
non-comptabilisée parmi les couches car ne modifie pas les entrées.
TRANSFERT

INTERPRETATION

D'un point de vue statistique, on peut donc voir les réseaux de neurones comme 2 étapes distinctes :

1. du feature engineering automatisé,
2. une régression linéaire des nouvelles variables.

La première étape est processée par les couches d'entrée et les couches internes du réseau, alors que la dernière étape est gérée par la couche de sortie...

On pourrait donc récupérer les variables transformées juste avant la couche de sortie, et remplacer le neurone de sortie par un autre modèle prédictif !

APPRENTISSAGE DU RÉSEAU

Supposons qu'on dispose d'une base d'apprentissage de n observations, $(x_i^1, x_i^2, ..., x_i^p; y_i)_{i=1,...,n}$.
Prenons le cas de la régression (généralisable à tte fonction de perte dérivable, dc aussi à la discrimination cf Gini) et le réseau à

- une couche cachée à q neurones,
- une sortie linéaire.

$\Rightarrow$ Les paramètres (poids) sont optimisés par moindres carrés : l'"apprentissage" minimise donc la perte quadratique

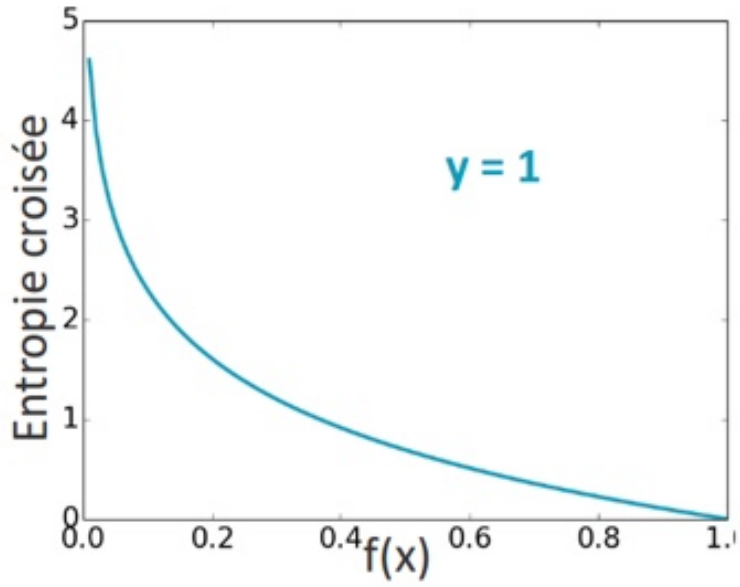
$$Q(\alpha,\beta) = \sum_{i=1}^{n} Q_i(\alpha,\beta) = \sum_{i=1}^{n} (y_i - \Phi(x_i;\alpha,\beta))^2,$$

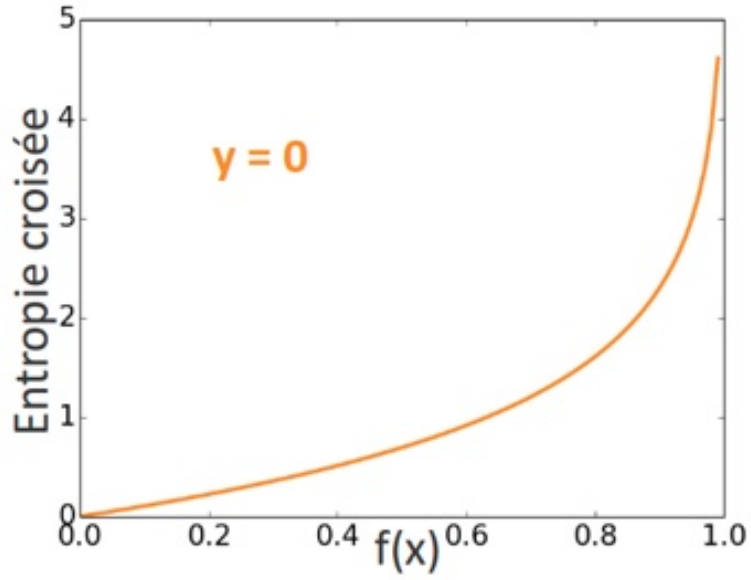
avec $\alpha = (\alpha_{jk})_{j=0,...,p;k=1,...,q}$, et $\beta = (\beta_k)_{k=0,...,q}$.

FONCTION DE PERTE CONVEXE - CLASSIFICATION

Dans le cas de la **classification**, nous allons choisir l'**entropie croisée.** Dans le cas binaire l'entropie croisée est définie par

$$\text{Erreur}(f(x^{(i)}), y^{(i)}) = -y^{(i)} \log f(x^{(i)}) - (1 - y^{(i)}) \log(1 - f(x^{(i)})).$$





Quand y=0, l'entropie croisée est d'autant plus élevée que f(x) est proche de 1. Réciproquement, quand y=1, l'entropie croisée est d'autant plus grande que la prédiction est proche de 0.

ESTIMATION : EVALUATION DES GRADIENTS

Les algorithmes utilisés pour l'optimisation sont généralement basés sur une évaluation du gradient par rétropropagation.

On détaille l'algorithme le plus utilisé : la rétropropagation de l'erreur !

Consiste en évaluer la dérivée de la fonction de coût en **une seule observation à la fois** par rapport à l'ensemble des paramètres, puis ajuster les paramètres, puis réévaluer avec les nouveaux paramètres sur une nvelle observation, et ainsi de suite.
Notons $z_{ki} = f(\alpha_{k0} + \alpha_k^T x_i)$, et $z_i = (z_{i1}, ..., z_{iq})$.

Ainsi, z_i sont les valeurs pour l'individu i dans chaque neurone de la couche cachée, et z_{ki} la valeur de l'individu i dans le neurone k.

Etudions les dérivées partielles de l'erreur :

$$\begin{aligned}\frac{\partial Q_i}{\partial \beta_k} &= \frac{\partial (y_i - \Phi(x_i;\alpha,\beta))^2}{\partial \beta_k} = \frac{\partial (y_i - (\beta_0 + \beta^T z_i))^2}{\partial \beta_k} \\ &= -2(y_i - \Phi(x_i))\,(\beta^T z_i)\,z_{ki} \\ &= \delta_i\, z_{ki}\end{aligned}$$

$$\begin{aligned}\frac{\partial Q_i}{\partial \alpha_{kj}} &= \frac{\partial (y_i - (\beta_0 + \beta^T z_i))^2}{\partial \alpha_{kj}} = \frac{\partial (y_i - (\beta_0 + \beta^T\,(f(\alpha_{k0} + \alpha_k^T x_i)))^2}{\partial \alpha_{kj}} \\ &= -2(y_i - \Phi(x_i))\,(\beta^T z_i)\,\beta_k\, f'(\alpha_k^T x_i)\, x_{ij} = \delta_i\,\beta_k\, f'(\alpha_k^T x_i)\, x_{ij} \\ &= s_{ki}\, x_{ij}\end{aligned}$$

$\rightarrow s_{ki}$: terme d'erreur sur chaque neurone caché pr l'indiv. i.
$\rightarrow \delta_i$: terme d'erreur du modèle courant à la sortie pr l'indiv. i.

Ces deux termes vérifient les équations dites de rétropropagation de l'erreur. On pose

$$s_{ki} = f'(\alpha_k^T x_i)\,\beta_k\,\delta_i$$

$\Rightarrow$ Pour estimer les valeurs des gradients, on a donc besoin d'évaluer δ_i et s_{ki}.

Cela se fait en 2 étapes :

1. une passe avant : valeurs courantes des poids permet de déterminer la sortie du réseau $\hat{\Phi}(x_i)$;
2. puis une passe retour : avec $\hat{\Phi}(x_i)$ et les valeurs courantes des poids, on évalue δ_i, puis s_{ki} par rétropropagation des δ_i... On obtient ainsi l'évaluation des gradients. Reste à optimiser.

ALGORITHMES D'OPTIMISATION

On sait évaluer les gradients $\Rightarrow$ reste à utiliser un algo adapté !

+ simple : utilisation itérative du gradient (e.g.Newton-Raphson) : en tout point de l'espace des paramètres, le vecteur gradient de Q pointe dans la direction de l'erreur croissante $\Rightarrow$ suffit de se déplacer dans le sens opposé pour $\searrow Q$! Ainsi,

$$\beta_k^{(r+1)} = \beta_k^{(r)} - \tau \sum_i \frac{\partial Q_i}{\partial \beta_k^{(r)}}$$

$$\alpha_{kj}^{(r+1)} = \alpha_{kj}^{(r)} - \tau \sum_i \frac{\partial Q_i}{\partial \alpha_{kj}^{(r)}}$$

τ : taux d'apprentissage (schéma minimisation f convexe).

APPLICATION : ALGORITHME DE RÉTROPROPAGATION ÉLÉMENTAIRE DU GRADIENT

Initialisation :

Tirage aléatoire uniforme sur $[0, 1]$ pour les poids α_{jkl} (normaliser dans $[0, 1]$ les données d'apprentissage).

Boucle :

- Tant que ($Q > erreurMax$) ou ($niter < niterMax$), faire
  - ranger la base d'apprentissage dans un nouvel ordre aléatoire,
  - pour chaque indiv. $i = 1, ..., n$, faire
    - calculer $\epsilon(i) = y_i - \Phi(x_i^1, ..., x_i^p; (\alpha)(i-1))$ en propageant les entrées vers l'avant ;
    - l'erreur est rétropropagée dans les $\neq$ couches pour affecter à chaque entrée une "responsabilité" dans l'erreur globale ;
    - mise à jour de chaque poids $\alpha_{jkl}(i) = \alpha_{jkl}(i-1) + \Delta\alpha_{jkl}(i)$.

REMARQUE SUR LE TAUX D'APPRENTISSAGE

Le taux d'apprentissage (learning rate) est un paramètre de tuning.

Il peut

- soit être fixé par l'utilisateur au début de l'algorithme ;
- soit varier en cours d'exécution.

Si τ est grand, alors on converge + vite vers une solution, mais elle est moins précise.
Et inversement.

PARAMÈTRES D'UN RÉSEAU DE NEURONES

Si on récapitule, on doit spécifier/ déterminer...

1. ...variables d'entrée et de sortie (leur faire subir d'éventuelles transformation de normalisation) ;
2. ...architecture du réseau :
   - nb de couches cachées : aptitude à traiter des non-linéarités ;
   - nb de neurones par couche cachée ;

   $\Rightarrow$ Impacte le nb de param. à estimer !
3. ...3 autres paramètres : erreur max. tolérée, nb d'itérations max. de l'algo, un terme éventuel de régularisation ("decay", à intégrer dans la fonction de coût $\Rightarrow$ Ridge, norme 2 des poids) ;
4. taux d'apprentissage τ.

COMPLEXITÉ D'UN RÉSEAU DE NEURONES

Les 2 choix sur le nombre de couches cachées, et le nombre de neurones par couche cachée, jouent sur la complexité du réseau.

$\Rightarrow$ Donc ces choix jouent sur la recherche du meilleur compromis biais-variance de l'estimateur par réseau neuronal...

$\Rightarrow$ Jouent donc également sur l'arbitrage qualité d'adéquation / qualité prédictive.

En pratique, on ne règle pas simultanément ces paramètres : on cherche à contrôler le phénomène de surapprentissage $\Rightarrow$ on fera des échantillons bootstrap, ou des validations croisées, ou échantillon test, pour estimer l'erreur.

RÉGLAGES

Pour ce qui concerne...

→ ...la durée d'apprentissage (`maxit` dans R) : arrêter par ex l'apprentissage lorsque l'erreur de validation réaugmente.
→ ...le nb de couches : d'après le théo. d'approx. univ., on peut se restreindre à un petit nb de couches cachées (1 ou 2 max.).
→ ...le nb de neurones par couche cachée : minimiser l'estimation de l'erreur de prévision par validations croisées par exemple.

Conclusion : à chaque architecture spécifiée correspond un réseau de neurones optimal. On fait varier ensuite les param. : on choisit au final l'optimal des optimaux (comme CART avec l'élagage).

LE DEEP LEARNING ?

Le Deep Learning n'est rien d'autre qu'un réseau de neurones ultra complexe.

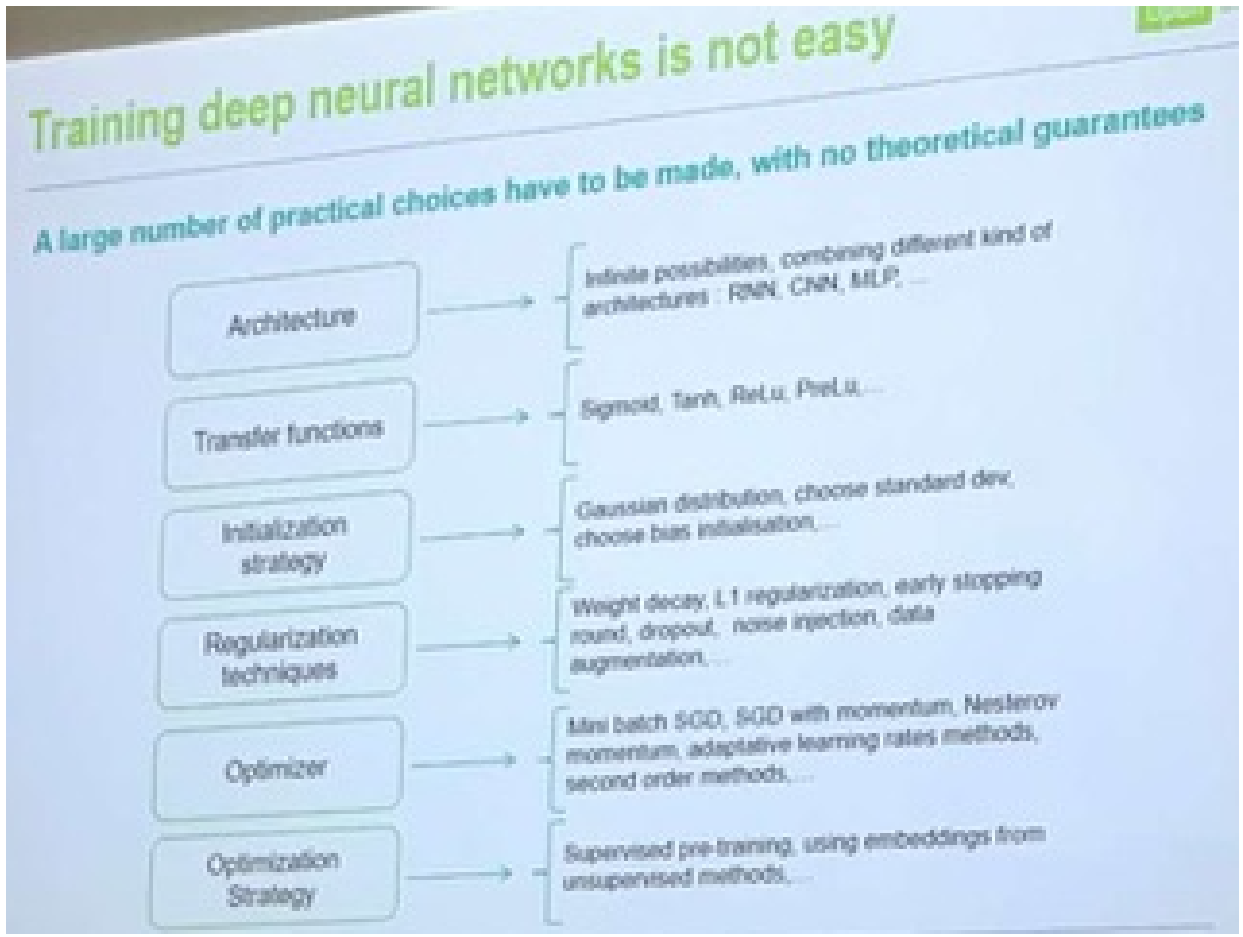
Il y a

- énormément de couches,
- et énormément de neurones.

Cela implique des millions de paramètres potentiels, et ne peut se calibrer qu'en cas de données gigantesque...

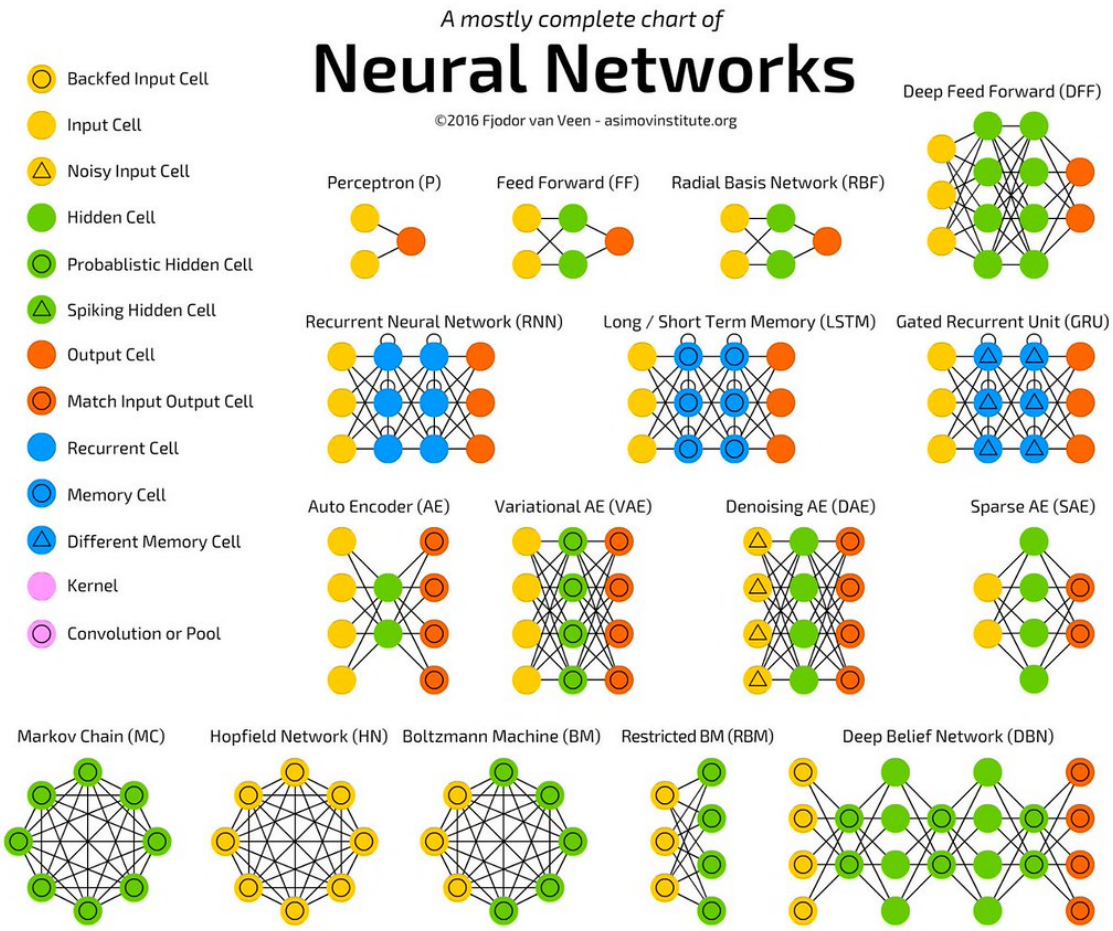
CONCLUSION SUR LE DEEP LEARNING

Comme on peut s'en douter, il n'est en fait pas du tout facile de bien se servir d'un réseau Deep Learning...



TYPES DE RESEAUX NEURONAUX

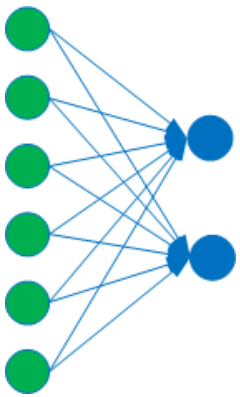
∃ multitude de types de réseaux, avec des caractéristiques ≠ : boucles de rétro-apprentissage, ...



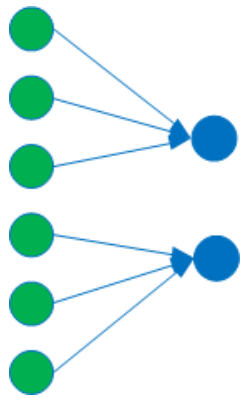
TYPES DE CONNEXION

∃ aussi une multitude de types de connexions :

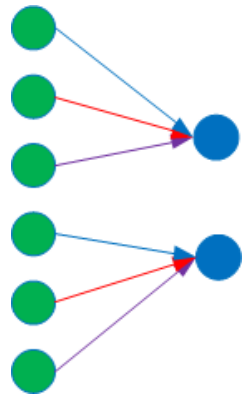
Connexions complètes
- Poids : 2*6 = 12



Connexions locales
- Poids : 2*3 = 6



Poids partagés (réseaux de convolution)
- Poids : 3



CONCLUSION GENERALE

Some characteristics of different learning methods.
Key: ●= good, ●=fair, and ●=poor.

Characteristic	Neural Nets	SVM	CART	GAM	KNN, Kernel	Gradient Boost
Natural handling of data of "mixed" type	●	●	●	●	●	●
Handling of missing values	●	●	●	●	●	●
Robustness to outliers in input space	●	●	●	●	●	●
Insensitive to monotone transformations of inputs	●	●	●	●	●	●
Computational scalability (large N)	●	●	●	●	●	●
Ability to deal with irrelevant inputs	●	●	●	●	●	●
Ability to extract linear combinations of features	●	●	●	●	●	●
Interpretability	●	●	●	●	●	●
Predictive power	●	●	●	●	●	●

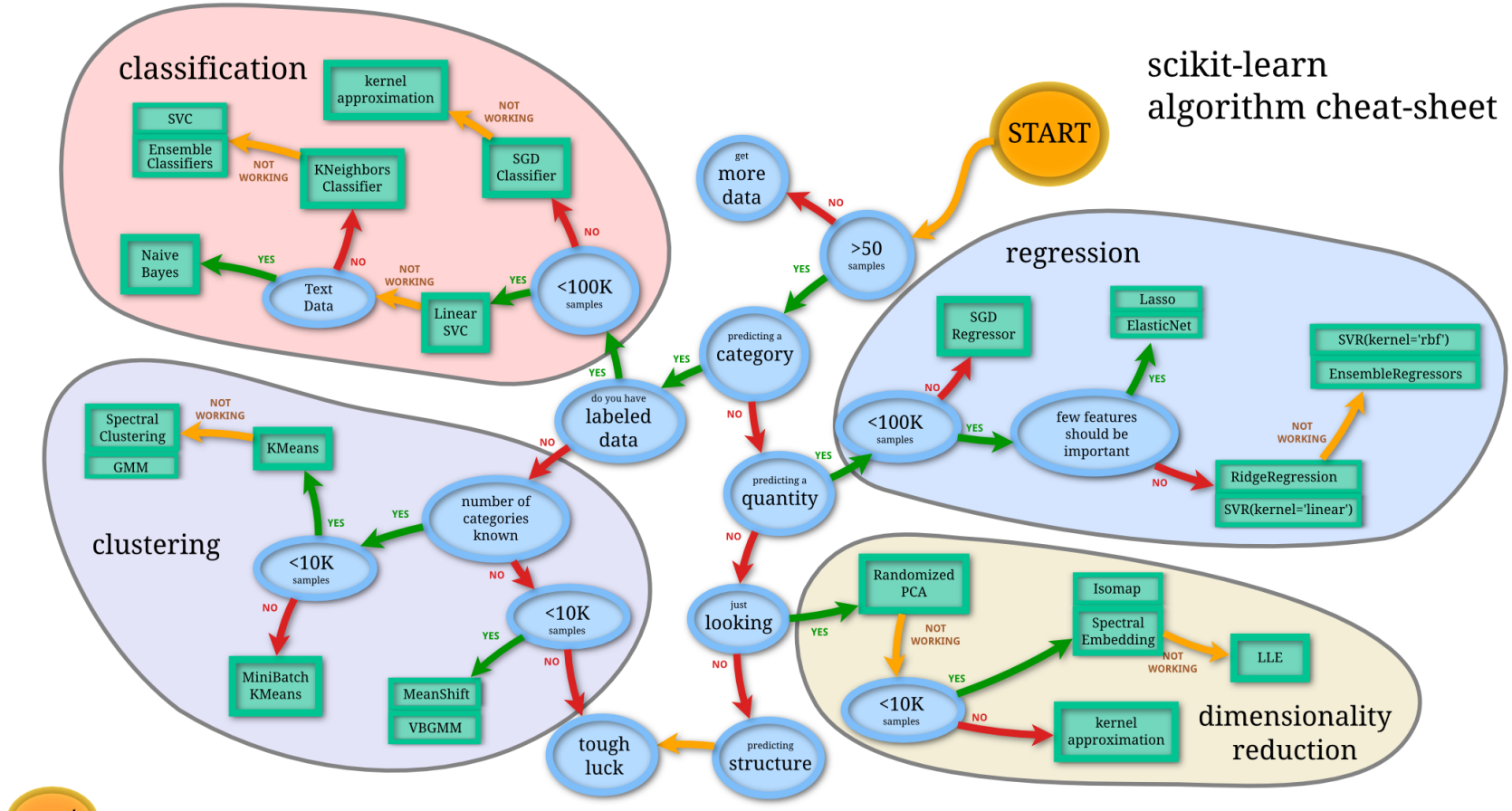
On peut aussi mentionner les points suivants, caractéristiques des méthodes d'apprentissage statistiques :

+ non-paramétrique,
+ peu d'hypothèses,
+ "data-driven",
+ faible biais normalement,

- instabilité (potentielle large variance),
- gestion du surapprentissage,
- ressources informatiques,
- interprétabilité.

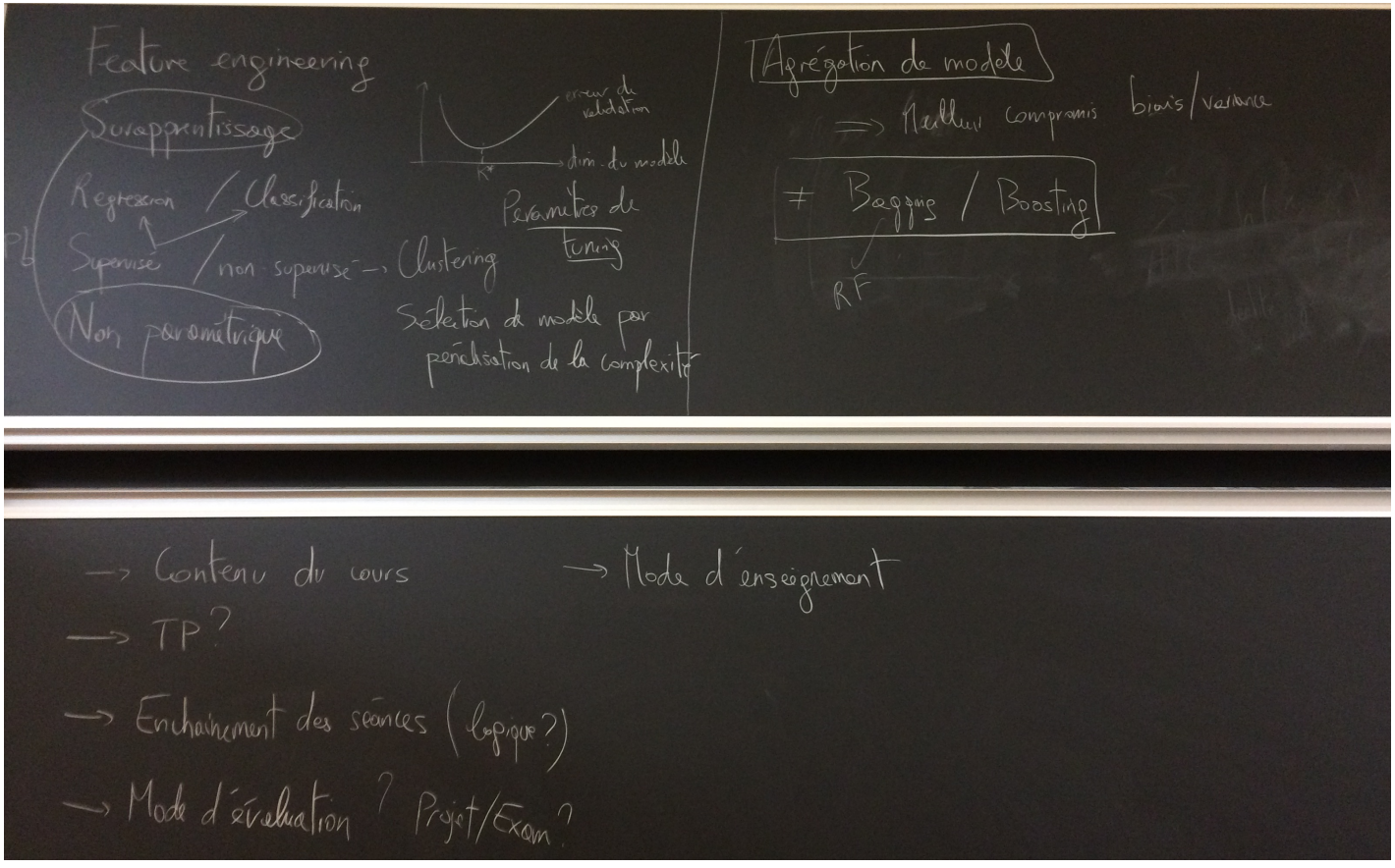
Notions-clefs retenus du cours ?
Retours sur le cours ? (contenu, TP, ...)

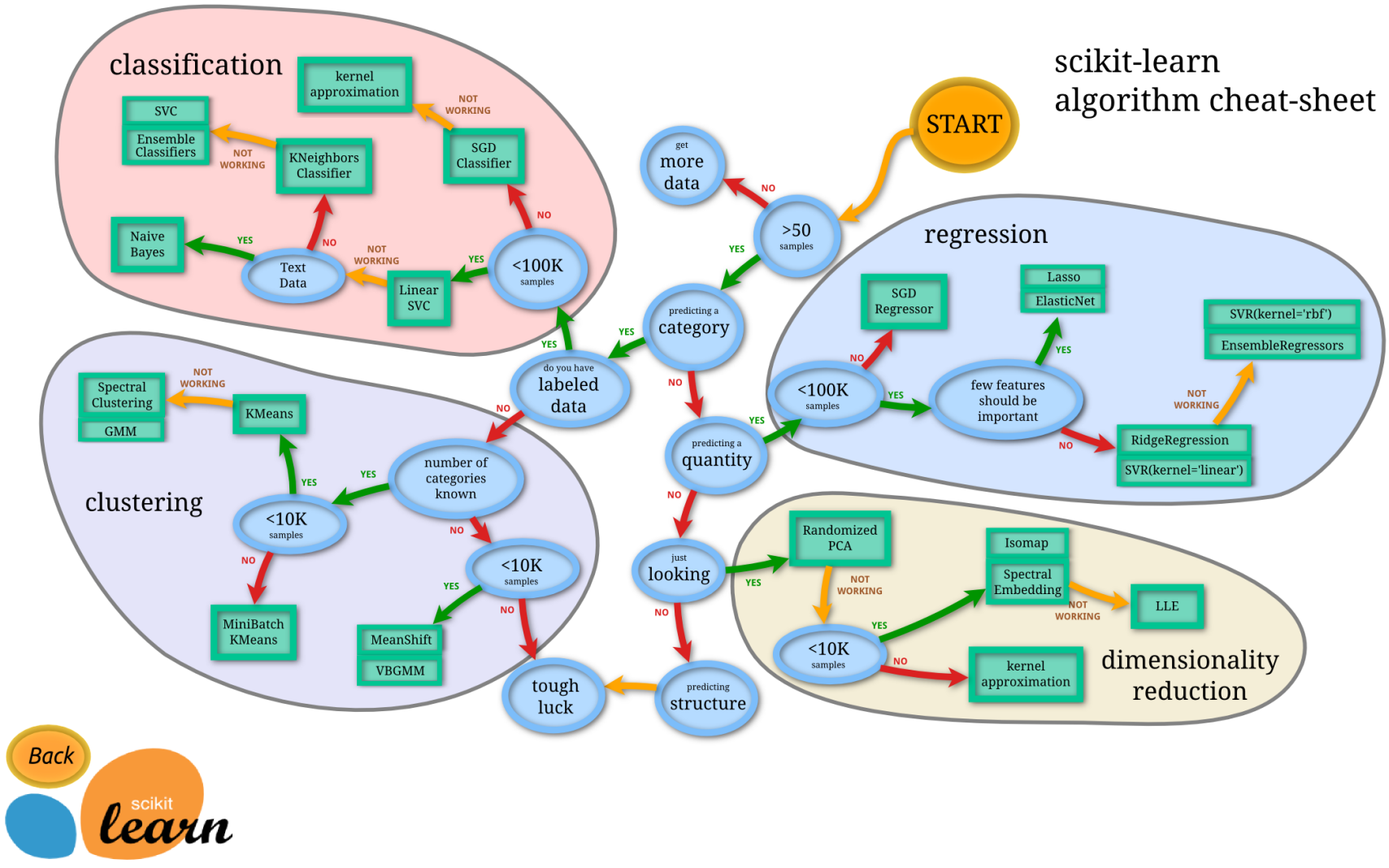
QUELS MODELES POUR QUELLES APPLICATIONS ?



NOTIONS PHARES DU COURS

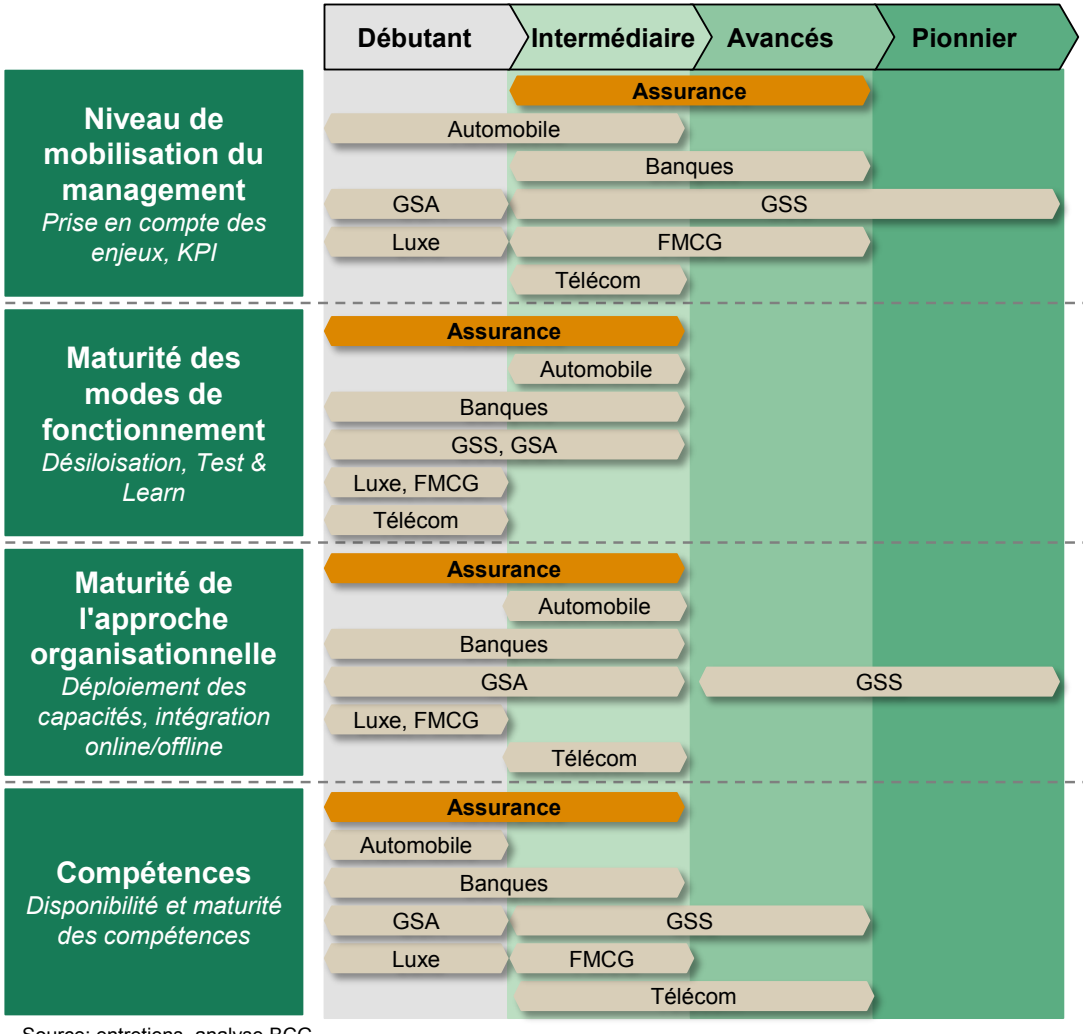
Videos Deep Learning (cf dossier mac mes videos)




scikit-learn
algorithm cheat-sheet
START
get more data
NO
>50 samples
YES
predicting a category
YES
NO
do you have labeled data
YES
NO
<100K samples
YES
NOT WORKING
Linear SVC
NOT WORKING
Text Data
YES
NO
Naive Bayes
NO
SGD Classifier
NOT WORKING
kernel approximation
NOT WORKING
KNeighbors Classifier
NOT WORKING
Ensemble Classifiers
SVC
classification
predicting a quantity
YES
NO
<100K samples
NO
YES
SGD Regressor
Lasso
ElasticNet
YES
few features should be important
NO
NOT WORKING
RidgeRegression
SVR(kernel='linear')
SVR(kernel='rbf')
EnsembleRegressors
regression
just looking
YES
NO
Randomized PCA
NOT WORKING
Isomap
Spectral Embedding
NOT WORKING
LLE
YES
<10K samples
NO
kernel approximation
dimensionality reduction
predicting structure
tough luck
NO
YES
number of categories known
YES
NO
<10K samples
NO
YES
MeanShift
VBGMM
<10K samples
NO
YES
MiniBatch KMeans
KMeans
YES
NOT WORKING
Spectral Clustering
GMM
clustering
Back
scikit learn

ANNEXES

POSITION DES ASSUREURS



+ Prise de conscience des enjeux au niveau Comex et mise en place d'un plan de transformation depuis 12-18 mois
- A date seule une partie du périmètre de business véritablement adapté (typiquement Auto et IARD dans certains cas)

+ Progrès collectif sur les approches fonctionnelles par la mise en place de plateaux cross-fonction
- Pas encore d'alignement structurel entre fonctions (parcours de carrière, KPIs croisés etc.)
- Pas de logique systématique Test & Learn/POC – les acteurs plus avancés sont par exemple les banques

+ Réorganisation dans les 12 derniers mois permettant l'intégration du numérique dans la fonction marketing
- A terme intégration du canal online dans l'écosystème agenc – les GSS y sont en avance

= Bien couverts sur les compétences E-business et Digital Marketing mais besoin de les transformer pour mieux répondre aux enjeux numériques
- Priorité de recrutement dans le Big data

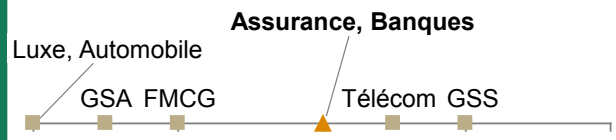
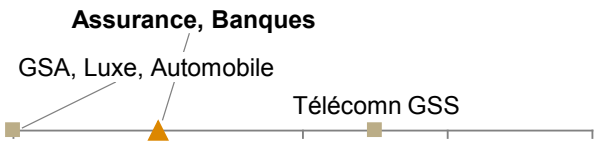
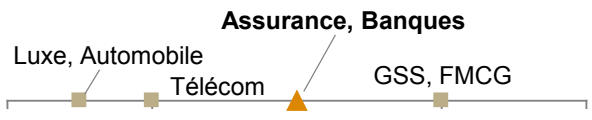
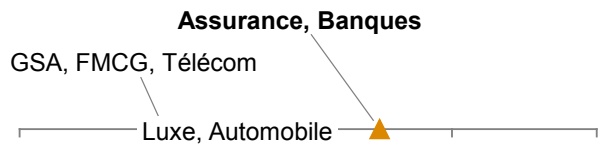
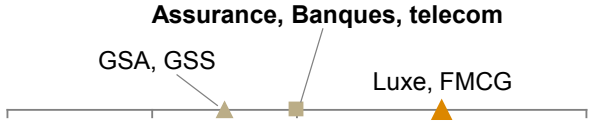
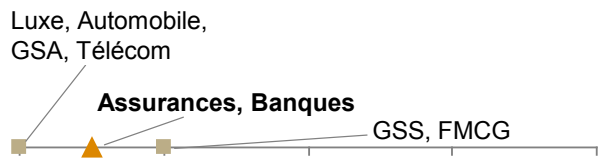
Source: entretiens, analyse BCG

COMPÉTENCES BIG DATA

Les compétences	Activités spécifiques	Les compétences	Activités spécifiques
1 **E-business** — Activités de vente sur internet (web, site propre, sites tiers)	• E-commerce • E-merchandising and site optimization • Omnichannel/Multi-channel strategy	5 **Big Data & Analytics** — Collecte, analyse et exploitation des données	• Data scientist • Web Analytics • Data quality • Business Intelligence
2 **Digital Customer Experience** — Design des interfaces et parcours digitaux	• UX designer / ergonome • Web developers	6 **Mobile interfaces** — Ensemble des interfaces propres aux canaux Smartphones et tablettes	• Mobile app / msite developer • Mobile UX • Mobile data / geolocalisation
3 **Digital Branding, Marketing** — Activités marketing liées aux canaux digitaux (web, réseaux sociaux)	• Social media marketing (community mgr / E-reputation / Advocacy Marketing) • Traffic acquisition (SEO, SEM, emailing, comparators, partnerships, affiliates) • Digital branding (display, video) • Programmatic / Real Time Bidding • E-CRM	7 **Digital tools** — Développement et maintenance des outils et logiciels digitaux permettant la transformation numérique en interne comme en externe	• Data technology (Hadoop,) • E-CRM (Neolane, Unica,) • Digital Front-Ends (Salesforce, ...) • Cloud • Digital security
4 **Digital content** — Création de contenu, de nouveaux produits/ services digitaux, digitalisation de produits/services existants	• Digital product or service manager • Web / App Editor • Digital Innovation / new digital product conception	8 **Digital support** — Ressources en support des activités numériques	• Digital Recruting • Digital legal • Digital purchasing

Source: entretiens, analyse BCG

ASSUREURS SUR CES COMPÉTENCES ?

Les compétences	Maturité de la compétence par rapport aux enjeux	Les compétences	Maturité de la compétence par rapport aux enjeux
1 E-business		5 Big Data & Analytics	
2 Digital Customer Experience		6 Mobile interfaces	
3 Digital Branding & Marketing		7 Digital tools	
4 Digital content		8 Digital support	

Source: entretiens, analyse BCG

Quelques articles de référence I

Y. Benjamini and Y. Hochberg.

Controlling the false discovery rate : a practical and powerful approach to multiple testing.

Journal of the Royal Statistical Society, Series B, 57 :289–300, 1995.

Leo Breiman.

Bagging predictors.

Technical report, UC Berkeley, 1994.

J.H. Friedman and P. Hall.

On bagging and nonlinear estimation.

pages –, 2000.

J. Friedman.

Greedy function approximation : A gradient boosting machine.

The Annals of Statistics, 29 :119–139, 2001.

Quelques articles de référence II

Y. Freund and R.E. Shapire.

A decision-theoretic generalization of on-line learning and an application to boosting.

J. Comput. Syst. Sci., 55(1) :119–139, 1997.

R.E. Shapire and Y. Freund.

Boosting.

MIT Press, Cambridge, 1st edition, 2012.

R.E. Shapire.

The boosting approach to machine learning : An overview.

Technical report, 2003.