STA211 – Entreposage et fouille de données
Synthèse des méthodes abordées

Daniel PONT
pont.daniel@gmail.com
Tél. : 06 81 71 64 31
5 juillet 2019

Remerciements

Je tiens ici à exprimer ma vive gratitude à l'équipe pédagogique : MM. Vincent Audigier, Ndèye Niang-Keita, Nicolas Thome pour la qualité de l'enseignement et de l'organisation de l'UE ainsi que pour leur disponibilité via le site moodle et lors de la séance de regroupement.

Un grand merci également à toutes les personnes du CNAM qui ont permis de rendre STA211 accessible en enseignement à distance.

Je souhaite également témoigner ma reconnaissance à pôle emploi (particulièrement à Mmes Chantal CHALVIDAN et Marie-Noëlle BADOL) pour le financement de ma formation,

« *If you torture the data long enough, it will confess.* »

Ronald Coase

Sommaire

1. Introduction

Ce document est une synthèse du cours STA211 « Entreposage et fouille de données » organisé par le CNAM Paris (*). La fouille de données (ou *data-mining* en anglais) décrit un domaine à l'intersection de la statistique et des technologies de l'information dont le but est de découvrir des structures dans de vastes ensembles de données.

1.1 Définitions et historique

1.1.1 De la donnée à la connaissance

Au départ purement tabulaires, les **données** peuvent aujourd'hui être de tout type traitable par ordinateur (texte, image, vidéo, son,…). Il peut s'agir de *données opérationnelles* générées sans optique d'analyse a priori (ex: ticket de caisse), *de données de référence* indépendantes de données transactionnelles (ex : adresse d'un client) ou *de métas-données* portant sur les données elles-mêmes (ex: numéro de la caisse ayant généré un ticket)

Les **connaissances** sont les information émergeant des associations et relations entre les données.

Le **data-mining** est le processus d'extraction des connaissances à partir des données.

1.1.2 De la statistique classique au big data

La mondialisation, la concurrence à grande échelle, les contraintes réglementaires (par exemple dans le domaine bancaire ou pharmaceutique) ont conduit à la nécessité d'exploiter toujours plus efficacement des données toujours plus volumineuses et diverses. Ce phénomène, qui touche tous les secteurs (économie, santé, …) a évolué selon la chronologie suivante :

- **Avant 1970** : **la statistique classique** (données de l'ordre de 100 octets)

  Les données sont rares, chères et de bonne qualité car récoltées pour répondre à une question précise définie en amont.

- **1970 - 1990 : analyse de données** (données de l'ordre du Mo ou du Go)

  L'exploration exhaustive des données n'est plus possible, les méthodes d'analyse telles que l'ACP et l' ACM se développent.

- **1990 – 2010 : data-mining** (données de l'ordre du Go ou du To)

  Avec la naissance du web et l'avènement des technologies numériques, les données ne sont plus produites à des fins statistiques mais font l'objet d'*analyse secondaire.* Elles sont non représentatives , hétérogènes , de qualité moindre.

- **depuis 2010 : machine learning et big data** (données de l'ordre du To, Po, ou plus)

  Le big data est caractérisé par le 3V (*Volume, Vélocité, Variété*) voir 5V (3V plus *Véracité, Valeur*). Les données, trop massives pour êtres traitées sur une seule machine sont prises en charge par un cloud ou des clusters. Il est alors moins question d'interprétation des données que de prédiction avec des méthodes s'appuyant sur le développement de l'intelligence artificielle (ex : *machine learning*)

(*) NB : t*outes les illustrations incluses dans ce document, sauf mention explicite, proviennent du support de cours.*

1.2 Effectuer une étude de data-mining

1.2.1 Le data-mining étape par étape

Les étapes sont les suivantes :

1. Comprendre et analyser les objectifs de l'étude
2. Créer une base de données
3. Prétraiter et nettoyer les données
4. Réduire les données
5. Segmenter la population
6. Choisir et valider le modèle
7. Itérer le processus
8. Déployer les modèle

1.2.2 Les outils du data-mining

Le data-mining vise à identifier des structures au sein de données. On distingue généralement deux types de structures :

- les *modèles* communs à l'ensemble des individus visent à expliquer une variable (réponse, définie a priori) à partir d'autres variables. Les méthodes **supervisées** sont adaptées à ce type de structure et seront présentées dans le chapitre 3
- les *patterns* sont des associations entre quelques individus. Les approches **non-supervisées** sont adaptées à ce type de structure et abordées dans le chapitre 4

De nombreux outils logiciels - libres (R, Orange, Spark,...) ou propriétaires (ex : Statistica, SAS) permettent d'étudier ces structures.

2. Pré-traitement des données

Avant de commencer la fouille de données, il convient d'en avoir une bonne compréhension et de s'assurer de leur qualité. Il faut notamment identifier les difficultés techniques telles que : les valeurs manquantes, erronées ou aberrantes, les modalités rares, les distributions non Gaussiennes, les liaisons non-linéaires entre les variables et les données multi-groupes. Afin d'identifier ces difficultés, on applique des méthodes d'analyses (univariée, bivariée, multivariée). Pour décrire les données on utilise également des méthodes de classification. Le pré-traitement inclut finalement la gestion des données manquantes et au besoin la transformation des variables originales.

2.1 Analyse univariée

L'analyse univariée renseigne sur leur *distribution marginale* (par opposition à la *distribution jointe* d'un ensemble de plusieurs variables). Elle permet d'identifier les *valeurs aberrantes.*

Pour résumer la distribution des **variables qualitatives** nominales, on s'intéresse aux *fréquences absolues* et *relatives* de leurs modalités présentées dans des tableau statistiques ou des diagrammes (en barre – histogramme ou circulaire – camembert).

L'analyse univariée des **variables quantitatives** repose quant à elle sur :

- *des indicateurs de tendance centrale* (moyenne et médiane empirique, mode)
- *des indicateurs de dispersion* (variance et coefficient de variation empirique, étendue, quantiles et distances inter-quantiles)
- *des indicateurs de forme* (coefficients d'asymétrie et d'aplatissement de Fischer)

2.2 Analyse bivariée

L'analyse bivariée commence par une représentation graphique qui permet de visualiser l'existence d'un lien entre deux variables. Elle se poursuit par le calcul de coefficients qui permettent de quantifier l'importance de ce lien. Le type de graphique et les coefficients employés en fonction de la nature du lien sont résumés ci-dessous :

Nature du lien	Représentation graphique	Coefficients
lien entre variables quantitatives	nuage de points	coefficient linéaire coefficient de Kendall
lien entre variables quantitatives et qualitatives	boxplots parallèles / histogrammes	rapport de corrélation / rapport de corrélation empirique
Lien entre variables qualitatives	graphiques en mosaïque (pour visualiser les tables de contingence)	khi-deux coefficient T de Tschuprow coefficient V de Kramer

2.3 Analyse multivariée

2.3.1 Données quantitatives : ACP

L'ACP (Analyse en Composantes Principales) consiste à rechercher un sous-espace (par exemple un plan) dans le nuage des individus tel que la projection du nuage sur ce sous-espace résume « au mieux » le nuage initial. Cette approche permet à la fois de décrire et visualiser les données mais aussi de réduire leur dimension.

2.3.2 Données qualitatives : ACM

L'ACM est la méthode d'analyse factorielle adaptée à des variables uniquement qualitatives. Les données nominales sont recodées sous forme d'un *tableau disjonctif complet* : chaque variable qualitative est remplacée par autant d'indicatrices que le nombre de modalités qu'elle possède. On s'intéresse alors à l'*espace des indicatrices* et *des modalités* (plutôt qu'à l'espace des variables et à celui des individus) mais le principe est similaire à celui de l'ACP.

2.4 Méthodes de classification

Parmi les méthodes de classification utilisées dans le cades du pré-traitement, on trouve :

- les **méthodes de partitionnement** avec en tête *l'agrégation autour des centres mobiles* et à ses variantes (k-moyennes,...).
- les **méthodes hiérarchiques** (ex. *CAH - Classification Ascendante Hiérarchique*)
- les **méthodes de classification mixte** dont l'idée est de commencer par effectuer une agrégation autour des centres mobiles pour un grand nombre de classes (ex : 50) puis d'appliquer une CAH sur les groupes obtenus et définir une classification à K classes.

2.5 Données manquantes

Une première solution au problème des données manquantes est de se limiter aux individus complets. Ce n'est pas toujours possible car les individus complets ne constituent pas nécessairement un échantillon représentatif et leur nombre tend à décroître rapidement lorsque le nombre de variables augmente.

2.5.1 Taxonomie et identification des mécanismes

Parmi les mécanismes de données manquantes, on distingue :

- le mécanisme **MCAR (Missing Completely At Random)** où la probabilité d'occurrence des données manquantes est sans lien avec les données complètes.
- le mécanisme **MAR (Missing At Random)** où le dispositif des données manquantes n'est pas causé par les données non observées mais peut être dû à la partie observée.
- le mécanisme **MNAR (Missing Not At Random)** où les données manquantes sont en partie causées par des données non observées

Afin d'identifier le type de mécanisme on recourt à une analyse exploratoire (univariée, bivariée et multivariée.

2.5.2 Traitement des données manquantes

Une première catégorie de solutions pour traiter les données manquantes est l'**imputation simple**. Celle-ci peut être réalisé avec une méthode *paramétrique* (régression), *non-paramétrique* (ex : *hot-deck* : on identifie les k individus complets les plus proches de l'individu à imputer, puis on en tire un au hasard parmi les k) ou *semi-paramétrique*.

Le défaut des méthodes d'imputation simples est qu'elles ne fournissent pas de mesures d' incertitude sur les données imputées. Pour remédier à cela on peut utiliser les méthodes d'**imputation multiple**. Le principe est de proposer M (généralement 3 ou 5) tableaux imputés. On réalise ensuite la méthode d'analyse souhaitée sur chaque tableau, puis on agrège les résultats entre eux selon des *règles de Rubin.* On obtient ainsi une unique estimation ponctuelle des coefficients de régression et une unique estimation de la variabilité associée .

Au-delà de l'imputation, il existe des approches par pondération qui attribuent un poids aux individus sans données manquantes de façon à corriger le biais observé dans la mise en œuvre de la méthode du cas complet.

2.6 Transformation des variables

Ces transformations concernent aussi bien les variables quantitatives (*discrétisation*, *linéarisation*, *normalisation* – ex : transformation de *Box-Cox*) que qualitatives (*regroupement de modalités*, *analyse factorielle*).

L'intérêt est de pouvoir appliquer un modèle réservé à un type de données différent (ex : discrétisation), plus facile à interpréter (ex : linéarisation, normalisation) ou plus efficace/ stable (regroupement de modalités).

2.7 Réduction des données

Simplifier le jeu de données en diminuant son nombre de lignes ou son nombre de colonnes permet de dé-bruiter *les données*, en éliminant une partie de l'information non pertinente, *faciliter l'interprétation*, en réduisant le nombre de variables explicatives, *réduire le temps de calcul* et *faciliter le stockage* en mémoire.

La **réduction en ligne** s'effectue par échantillonnage ou classification non-supervisée.

La **réduction en colonne** est réalisée par analyse factorielle, classification ou sélection de variables.

3. Méthodes supervisées

3.1 Éléments fondamentaux

3.1.1 Introduction

Les méthodes supervisées visent à construire, à partir d'exemples (*données d'apprentissage*), des **modèles** qui *expliquent et/ou prédisent* une variable réponse Y en fonction de variables explicatives X. Voici un tableau des principales méthodes :

Régression (Y continue)	**Classification (Y quantitative)**
Paramétrique (la nature du lien entre X est Y est une fonction dont la forme est explicite)	
Régression linéaire	Régression logistique
ANOVA	Analyse linéaire discriminante
Modèles additifs généralisés	Modèles à classes latentes
	Modèles additifs généralisés
Non-paramétrique	
KNN	KNN
Arbres	Arbres
Forêts aléatoires	Forêts aléatoires
Réseau de neurones	Réseau de neurones
Support Vector Machines	Support Vector Machines
Splines	

3.1.2 Apprentissage statistique

En apprentissage statistique deux critères sont employés : la **prédiction** et **l'estimation**. La *prédiction* consiste à déterminer, pour une entrée absente des donnes d'apprentissage, la sortie la plus proche possible de la véritable valeur réponse. *L'estimation* consiste quant à elle à approcher au mieux le lien qui relie la valeur réponse aux variables dépendantes.

Ces deux critères sont équivalents dans le cadre d'une régression mais pas dans celui d'une classification (ex : une classification binaire aboutira à la même prédiction si la probabilité d'observer la valeur 1 est 0.51 ou 0.99, mais l'estimation pourra être très différente).

Un point fondamental concerne l'estimation de l'erreur de généralisation (erreur que l'on cherche à minimiser dans une optique de prédiction) : l'**erreur de généralisation doit être estimée sur un échantillon indépendant des données d'apprentissage.**

Cette erreur peut être mesurée avec différents indicateurs. Ainsi pour une classification binaire, on peut utiliser non seulement le taux de mauvais classement mais également l'AUC (Area Under Curve - qui prend en compte la précision et le rappel) ou l'indice de Gini.

3.1.3 Compromis biais-variance

Formellement l'erreur de test s'écrit : $\mathrm{E}\left[(y-\hat{f}(x))^2\right] = \mathrm{Biais}[\hat{f}(x)]^2 + \mathrm{Var}[\hat{f}(x)] + \sigma^2$ avec :

$\mathrm{Biais}[\hat{f}(x)] = \mathrm{E}[\hat{f}(x) - f(x)]$, $\mathrm{Var}[\hat{f}(x)] = \mathrm{E}\left[(\hat{f}(x) - \mathrm{E}[\hat{f}(x)])^2\right]$ et σ^2 un terme irréductible lié au bruit.

Un modèle simple (variance faible) risque le sous-apprentissage (biais élevé y compris sur les données d'apprentissage). Un modèle complexe (variance élevée) risque le sur-apprentissage (biais faible sur les données d'apprentissage mais élevé sur de nouvelles données). On souhaite trouver un modèle intermédiaire. C'est le **compromis biais-variance.**

3.1.4 Choix de modèles

3.1.4.1 Critères de pénalité

Comme indiqué dans le paragraphe précédent plus un modèle est complexe, plus sa variance a tendance à être élevée. Pour contrer ce phénomène, on évalue un **critère d'erreur qui pénalise la complexité** tels que les critères AIC (Akaïke Information Criterion) et BIC (Bayesian Information Criterion).

Il existe des algorithmes – les *méthodes pas-à-pas* – qui permettent de retenir un bon (et non nécessairement le meilleur) candidat parmi une famille de modèles. Ces méthodes optimisent un modèle initial selon un critère tel que l'AIC en ajoutant (méthode *ascendante*) ou enlevant (méthode *descendante*) successivement des variables. Il existe également une méthode mixte (la méthode *progressive*) qui suit le même principe que la méthode ascendante, sauf que l'on peut éliminer des variables déjà introduites.

Cette approche est limitée par le fait que les critères ne peuvent être utilisés que pour les modèles paramétriques, ne permettent pas de comparer des modèles issus de familles différentes et suppose que la complexité se traduit par le nombre de paramètres du modèle.

3.1.4.2 Approches empiriques

Une première approche est de constituer un **échantillon test** et un **échantillon d'apprentissage** puis pour chacun des modèle modèles candidats (potentiellement issus de famille différentes) entraîné à partir de l'échantillon d'apprentissage d'évaluer l'erreur de généralisation avec l'échantillon test. On retient alors le modèle pour lequel l'erreur est la plus faible puis on le ré-ajuste avec l'ensemble des données à disposition. Ceci présente un inconvénient : l'estimation de l'erreur de test dépend du partitionnement initial.

Les techniques de **validation croisée** sont une alternative plus sophistiquée consistant à découper l'échantillon en k blocs de même taille. L'ensemble des k-1 premiers sert d'échantillon d'apprentissage et le $k_{ème}$ sert d'échantillon test. Ainsi, on obtient k erreurs de test que l'on moyenne ce qui nous donne une estimation de l'erreur plus précise que celle obtenue par un simple découpage test-apprentissage. Dans le cas particulier de la validation *k-fold* où k est égal au nombre de données on parle de *leave-one out.*

Le découpage en échantillon en deux parties (apprentissage - test) permet de choisir un modèle mais il ne permet pas d'avoir une mesure objective de l'erreur de généralisation commise par le modèle final. Pour évaluer la performance du modèle de façon objective, il faut disposer d'un 3eme échantillon. Ainsi, on considérera généralement : un échantillon d'*apprentissage*, un échantillon de *validation* et enfin un échantillon *test*. Si le nombre d'observations le permet, ce découpage pourra être fait typiquement en proportion 1/2, 1/4, 1/4.

La procédure est la suivante :

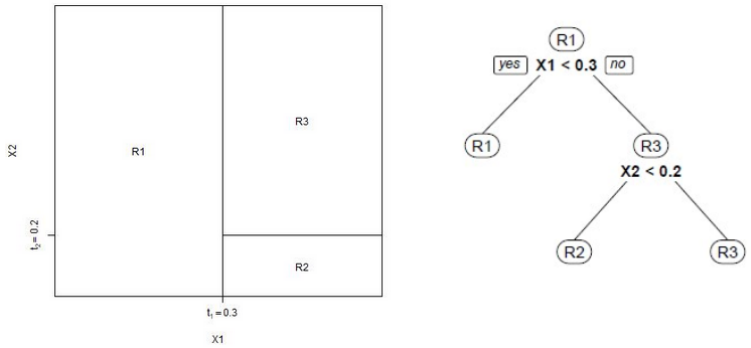
- on ajuste les différents modèles sur l'échantillon d'apprentissage
- on choisit le "meilleur" modèle à partir de l'erreur commise sur l'échantillon de validation
- on ajuste le meilleur modèle sur les données rassemblant échantillon d'apprentissage et de validation
- on évalue l'erreur de généralisation sur l'échantillon test
- on ré-ajuste ce modèle sur l'ensemble des 3 échantillons

3.2 Arbres de régression et de classification

3.2.1 Introduction

Les arbres peuvent être utilisés dans une optique de classification ou de régression. Ils font partie des méthodes dites de *segmentation* non-paramétrique. Le principe est de partitionner l'espace des variables explicatives X appelé l'espace des entrées en R régions et d'associer un modèle simple à chacune d'entre elle (généralement une constante), permettant ainsi de prédire une sortie Y.

Un point fort des arbres est de pouvoir être représenté de manière synthétique :



3.2.2 Construction d'un arbre

Afin de construire un **arbre de régression** itérativement, on doit déterminer à chaque étape selon *quelle variable* de l'espace d'entrée la division sera effectuée ainsi que la valeur de *seuil* associée. On commence par identifier pour chaque variable explicative le meilleur seuil puis on retient la variable qui permet de **minimiser l'hétérogénéité des nœuds fils,** selon le critère des moindres carrés. Ce processus de division récursive s'arrête au moment où tous les nœuds sont homogènes.

Il faut cependant veiller à ne pas construire des arbres trop profonds, car ceux-ci présentent un risque se sur-apprentissage : le nombre de paramètres est trop élevé par rapport aux données d'entraînement, résultant sur un biais faible mais une variance importante.

Il est donc primordial d'**élaguer** l'arbre. L'élagage est réalisé par composition itérative d'une séquence de *sous-arbres emboîtés* (A_1, A_2,.., A_k) en se basant sur le critère *coût-complexité.* L'objectif est de minimiser l'erreur d'ajustement en favorisant les arbres avec peu de nœuds. La sélection du meilleure sous-arbre est réalisée soit avec un échantillon de validation (en retenant le sous-arbre d'erreur test minimale ou en utilisant *la règle d'un écart type)* soit avec une procédure de validation croisée (plus subtile).

La construction d'un **arbre de classification** est similaire à celle d'un arbre de régression moyennant l'adaptation du critère d'hétérogénéité (appelée aussi *impureté*) à la nature qualitative de la variable réponse. Dans un contexte de classification, on emploie ainsi l'*erreur de mauvais classement*, l'*indice de Gini* ou la *déviance* (entropie de Shannon).

3.2.3 Propriétés

Les arbres ont pour avantages d'être faciles à expliquer à des non initiés, de produire des résultats interprétables sous forme de graphiques, sont sensibles aux effets d'interaction de par leur structure hiérarchique. De plus la sélection de variables est inhérente à la méthode, ne nécessite pas d'inversion de matrices de covariances et les donc les arbres peuvent se

construire directement sur un jeu de données où les liaisons entre variables sont fortes. Enfin les arbres fonctionnent avec des variables (explicatives et réponses) qualitatives ou quantitatives.

Les arbres ont pour défauts d'être une méthode instable (la moindre modification sur une division à des conséquences directes sur les divisions successives). De plus, un nombre d'individus assez important (plusieurs centaines) est requis pour l'apprentissage. Enfin, le partitionnement sous forme de parallélépipèdes ne permet pas de considérer des frontières autres que rectilignes.

3.2.4 *Compléments*

Ce chapitre sur les arbres de décision a présenté une méthode de partitionnement particulière *CART* (Classification And Regression Trees) qui produit des arbres binaires mais il existe bien d'autres algorithmes : *C5.0*, *CHAID* (CHi-squared Automatic Interaction Detector), *AID* (Automatic Interaction Detection), etc.…

Du point de vue de la stabilité et des performances, une famille intéressante de méthodes basée sur les arbres de décision est celle des **forêts aléatoires**. Leur principe est de générer plusieurs arbres différents en utilisant des échantillons indépendants issus des données, puis d'agréger les différentes prédictions fournies par ces différents arbres. L'assemblage des résultats fournis par de nombreux arbres permet de mieux explorer l'ensemble des règles de décision.

3.3 *Réseaux de neurones*

3.3.1 Introduction

Les premiers réseaux de neurones ont été étudiées dès la fin des années 50, ont commencé à avoir des applications pratiques durant le milieu des années 80 (système de reconnaissance des codes postaux aux États-Unis) avant de passer de mode dans les années 90. Depuis 2010 et l'émergence du Big Data, ils ont fait un retour en force notamment dans les problèmes de reconnaissance de signaux bas niveaux (ex : reconnaissance visuelle et vocale). Ils sont basés sur la notion de neurone formel inspirée des neurones biologiques situées dans le cerveau.

3.3.2 Du neurone formel au perceptron multi-couches

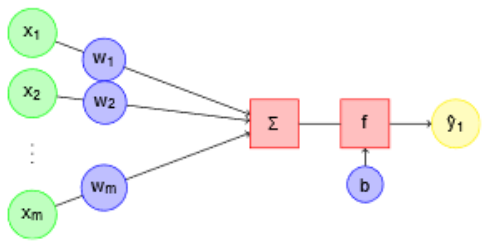
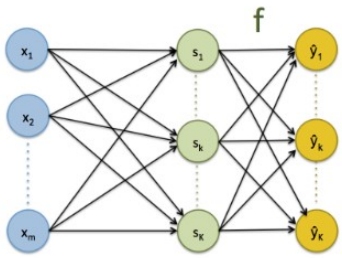


Figure: The formal neuron – Credits: R. Herault

x_i : entrée
w_i, b : poids, biais
f : fonction d'activation
y : sortie
$y = f(w^T x + b)$

Le **neurone formel** n'a qu'une sortie il est donc limité à des problèmes de classification binaire. Le **perceptron** est un réseau construit en combinant plusieurs neurones formels ce qui permet d'avoir plusieurs sorties et donc de traiter des problèmes de classification multi-classes :



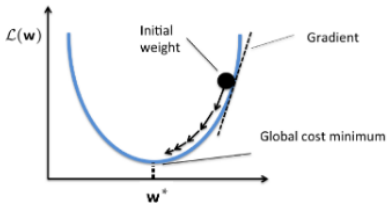
Le perceptron mono-couche est limité à des problèmes de classification avec des séparateurs linéaires. Afin de modéliser des fonctions complexes, non-linéaires, on doit utiliser un **perceptron multi-couches** dans lequel les sorties de la 1ère couche constituent les entrées de la 2ème, les sorties de la 2ème les entrées de la 3ème, etc.

3.3.3 Principes d'apprentissage des réseaux de neurones

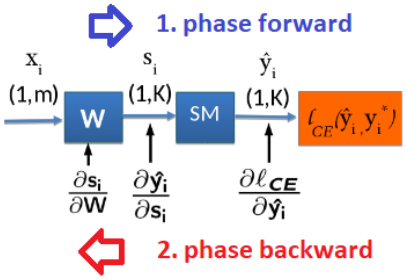
Une première technique fondamentale appliquée pendant l'apprentissage est la **méthode de descente de gradient** appliquée à la fonction d'erreur L (alias fonction de coût) qui mesure l'écart entre la sortie calculée par un neurone et celle attendue par la supervision. En effet le gradient $\nabla_w = \partial L / \partial w$ indique la direction dans lequel la fonction de coût diminue le plus en fonction des poids. Il est alors possible d'approcher le minimum de L par un algorithme itératif :

1. initialisation des poids **w**
2. jusqu'à ce que $\|\nabla_w\|^2 \approx 0$

$$w^{(t+1)} = w^{(t)} - \eta \, \partial L / \partial w$$



Afin de calculer le gradient de l'erreur pour chaque neurone du réseau, on utilise une deuxième technique : **la rétro-propagation du gradient**. Dans le cas d'une régression logistique l'algorithme est le suivant :



1. On propage les entrées de l'échantillon d'apprentissage x_i dans le réseau ce qui permet de calculer successivement s_i puis $\hat{y}_i$ et enfin $\ell_{CE}(\hat{y}_i)$: *phase forward*.
2. En utilisant la règle des dérivées chaînées, on procède ensuite à la *phase backward* :

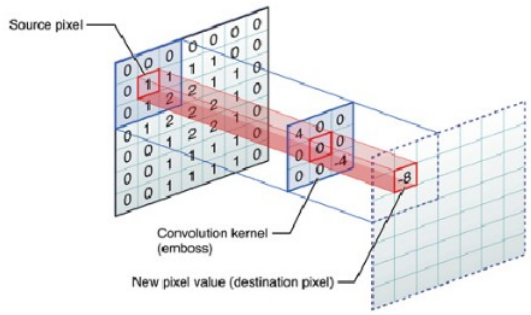
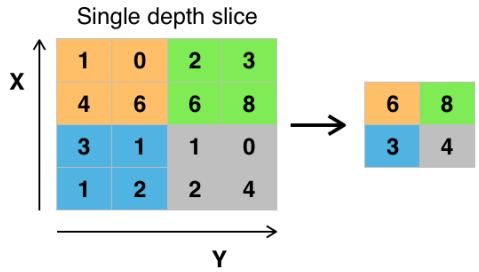
$$\frac{\partial \ell_{CE}}{\partial \mathbf{W}} = \frac{\partial \ell_{CE}}{\partial \hat{\mathbf{y}}_\mathbf{i}} \frac{\partial \hat{\mathbf{y}}_\mathbf{i}}{\partial \mathbf{s}_\mathbf{i}} \frac{\partial \mathbf{s}_\mathbf{i}}{\partial \mathbf{W}}$$

Ce mécanisme de rétro-propagation du gradient est généralisable à tout réseau possédant une ou plusieurs couches cachées.

3.3.4 Réseaux convolutifs profonds

Les réseaux de neurones complètement connectés tels que le perceptron multi-couches présentent plusieurs limitations. Ils *passent mal à l'échelle* (le nombre de paramètres augmente très rapidement avec la dimension des données et le nombre de neurones). Ils ne *prennent pas en compte les structures locales* pourtant fondamentales dans l'étude des signaux bas nivaux. De plus, une variation *même mineure* en entrée (ex : translation de 1px d'une image) peut avoir *un impact très important* en sortie.

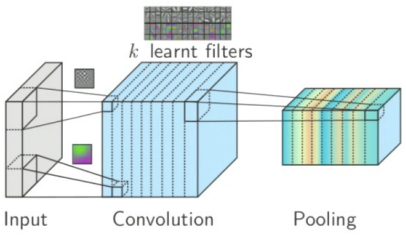
Les réseaux convolutifs profonds permettent de dépasser ces limitations, en particulier grâce à des opérations de **convolution** et de **pooling,** illustrées ci-dessous **:**

Convolution	Pooling max(source : Wikipedia)
	

Le produit de *convolution* est un opérateur bilinéaire ainsi qu'un produit commutatif. Appliqué à un problème de reconnaissance d'image, la convolution s'interprète comme l'application d'un filtre : chaque point de l'image filtrée est le résultat du produit scalaire entre une région de l'image et les coefficients du filtre. La convolution permet de réduire significativement le nombre de paramètre, de modéliser explicitement la structure locale des données et de gagner en invariance par rapport à des déformations locales.

Le *pooling* est une agrégation statistique d'un ensemble de valeurs qui produit une simple sortie scalaires (par exemple la valeur maximale, minimale ou moyenne de l'ensemble de valeurs). Le pooling réduit la taille spatiale d'une image intermédiaire, diminuant ainsi la quantité de paramètres et de calculs à effectuer dans le réseau.

Le fait de combiner une couche de convolution avec un pooling (comme représenté dans l'illustration ci-dessous) constitue le bloc élémentaire des réseaux de neurones profond :



C'est un réseau composé de ces blocs de convolution-pooling, muni d'une fonction d'activation innovante - *ReLU (Rectified Linear Unit)* et d'optimisations avancées (ex : *dropout*) qui a gagné le challenge ImageNet 2012. Ce réseau - *AlexNet de Krizhevsky et* al – a ainsi démontré la supériorité des réseaux convolutifs pour la reconnaissance d'image. Il a ouvert la voie à des réseaux de neurones toujours plus profonds (plus de cent couches) dont l'entraînement a été rendu possible par l'abondance de données d'apprentissage et le développement de GPU (processeurs graphiques) nettement plus performants que les CPU.

3.4 Méta-algorithmes

Les méta-algorithmes utilisent des méthodes d'agrégation qui consistent à combiner plusieurs prédictions fournies par différents modèles afin d'améliorer les performances.

3.4.1 Bagging

Le bagging ou *agrégation bootstrap* est une procédure permettant de réduire la variance en se basant sur le théorème central limite. Le principe est de constituer B échantillons indépendants, puis de construire un modèle par échantillon pour finalement agréger les modèles en moyennant les prédictions de chacun. Afin de garantir l'indépendance des B prédictions, on privilégie des prédicateurs variant beaucoup en fonction de l'échantillon d'apprentissage tels que les arbres. L'échantillonnage permet d'utiliser les individus non sélectionnés en apprentissage pour estimer une erreur de prédiction appelée erreur *out-of-bag.*

Dans un cadre de *classification*, le bagging présente plusieurs spécificités. D'abord la probabilité d'appartenance à une classe est souvent plus pertinente que la classe prédite elle-même. Ensuite même si l'objectif est purement prédictif, il peut être préférable d'agréger les probabilités prédites pour chaque prédicteur puis de retenir la classe la plus probable, plutôt que de considérer la classe majoritaire pour les différents prédicteurs.

3.4.2 Boosting

Le boosting est une procédure d'agrégation qui consiste à effectuer une moyenne pondérée des prédictions de différents prédicteurs (en régression) ou un vote à la majorité (en discrimination). Le principe de cette procédure d'agrégation est de construire une famille d'estimateurs construits de manière récursive : chaque estimateur est une version adaptative du précédent en donnant plus de poids aux observations mal ajustées ou mal prédites. Le poids des individus à chaque itération sert ensuite à calculer une erreur de mauvais classement qui permettra de définir le poids accordé à chaque version de l'estimateur lors de l'agrégation.

Parmi les implémentations du boosting, *Adaboost* est une des plus utilisées.

Un avantage du boosting est qu' il permet de réduire le biais ou la variance - un classificateur avec une variance faible mais un biais important peut donc s'avérer performant en termes de prédiction. Un inconvénient du boosting est sa tendance au sur-apprentissage, notamment en présence de données fortement bruitées. Le nombre d'itérations est donc un paramètre important à déterminer.

3.4.3 Comparatif bagging vs boosting

Pour terminer, listons quelques points clés de comparaison entre le bagging et le boosting :

- le boosting est purement déterministe (il utilise toutes les données d'apprentissage) tandis que le bagging est aléatoire (il s'appuie sur des échantillons),
- tous les modèles du bagging ont le même poids contrairement à ceux du boosting,
- le boosting est séquentiel tandis que le bagging est parallélisable,
- le bagging permet uniquement de réduire la variance alors que le boosting peut aussi diminuer le biais (au risque d'augmenter la variance)
- Le bagging est souvent plus efficace mais quand le boosting est efficace, il dépasse le bagging.

4. Méthodes non supervisées

Parmi les méthodes non-supervisées, on retrouve essentiellement les méthodes de classification, les méthodes de recherche de règles d'association et les méthodes d'analyse factorielle.

4.1 Classification de variables

Les techniques abordées dans ce paragraphe consistent à identifier des *groupes de variables homogènes*.

Une première méthode est la **classification ascendante hiérarchique (CAH)** : les groupements se font par agglomération progressive des variables deux à deux selon un principe analogue à la CAH des individus. Cette méthode nécessite de choisir *l'indice de dissimilarité entre variables* et le *critère d'agrégation de deux groupes* de variables. Pour le critère d'agrégation, on peut sélectionner le saut minimal, le saut maximal, le saut moyen (ou méthode de Ward dans le cas où la dissimilarité est une distance euclidienne). Quant à l'indice de dissimilarité, son choix dépend de la nature de la variable - pour les variables qualitatives χ^2, V de Cramer,… pour les variables quantitatives : concordances simples, Jaccard, etc.

Les méthodes de **classification hiérarchique descendante** (ex. : méthode VARCLUS de SAS Sarle) procèdent, quant à elles, par dichotomies successives à la construction d'un arbre hiérarchique descendant dont les segments terminaux constituent une partition des éléments à classer. La partition obtenue maximise l'homogénéité intra-classe et l'hétérogénéité inter-classe.

Un autre type de classification est le **partitionnement direct** (ex. : CLV de Vigneau et Qannari). Le principe est analogue à l'algorithme *k-moyennes* mais utilise comme critère la colinéarité entre variables au lieu de la distance.

Il est fréquent, pour le même jeu de données, de s'intéresser à la fois à la classification des variables et à celle des individus. On procède alors à du *biclustering* (ou *co-clustering* ou *classification croisée*) : chaque classe d'individus n'est construite qu'à partir des ressemblances vis-à-vis de certaines variables d'un même groupe.

4.2 Cartes de Kohonen

Les cartes de Kohonen (ou *topologiques* ou *auto-organisatrices*) peuvent être vues comme une méthode des k-moyennes contraintes, dans lesquelles les centres des classes (appelés *prototypes* ou *vecteurs référents*) sont « encouragés » à vivre dans un sous-espace de l'espace des individus.

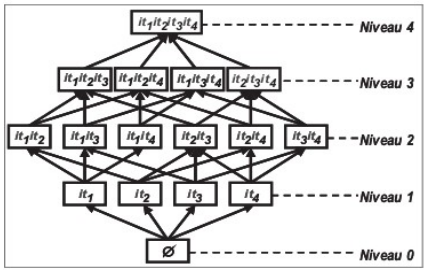
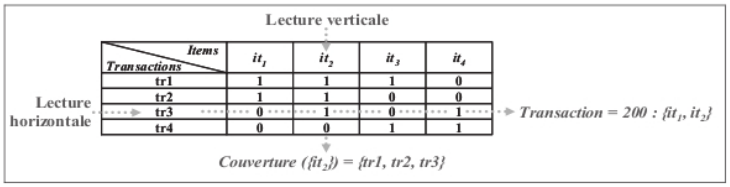
Le principe est de considérer n individus d'un espace V (de dimension p) et k vecteurs référents disposés sur un plan de V. On associe à chaque vecteur référent un couple de coordonnées dans une grille A. Puis pour chaque individu x de V on détermine le vecteur référent w_k le plus proche. On déplace ensuite en direction de x, les vecteurs référents *proches par leur coordonnées dans la grille A* de w_k. les vecteurs référents sont ainsi « attirés » par les données, mais contraints à rester rattachés aux autres via une grille en deux dimensions.

4.3 Règles d'association

Les méthodes de recherche de règles d'association ont été proposées pour découvrir quels produits étaient achetés conjointement et à quelle date dans les bases de données des ventes de supermarché (Agrawal, Imieliński, and Swami (1993)) .

On considère un ensemble d' éléments I (ou *items* – ex : les produits en vente dans un supermarché) et les sous-ensembles non vides de I (les *transactions* – ex : contenus des caddies). Une *association est une implication* de la forme : $A \rightarrow B$ où $A \subset I, B \subset I$ et $A \cap B = \emptyset$

Les règles d'associations sont représentées par des *treillis* ou des *matrices d'incidence*:

Treillis d'items	Matrice d'incidence
	

Transactions \ Items	it_1	it_2	it_3	it_4
tr1	1	1	1	0
tr2	1	1	0	0
tr3	0	1	0	1
tr4	0	0	1	1

La **pertinence** d'une règle d'association est définie par son *support* (proportion de transactions contenant les items de $A \cup B$) et sa *confiance* (proportion de transactions contenant les items de $A \cup B$ par rapport aux transactions contenant les items de A).

L**a recherche de règles d'association** consiste à trouver les itemsets dont le support est supérieur ou égal à un seuil minimum *(minsup)* puis d'extraire des règles d'association dont la confiance est supérieure ou égale à un second seuil minimum (*minconf*). NB : les deux seuils sont fixés par l'utilisateur. L'algorithme *Apriori* est l'implémentation la plus connue de cette méthode de recherche.

4.4 Analyse multi-blocs

L'analyse multi-blocs permet d'étendre les méthodes d'analyse factorielles classiques (ACP, ACM, etc.) à l'étude concomitante de plusieurs tableaux. Elle offre notamment un moyen de comparer des groupes de variables (des individus proches l'un de l'autre pour un groupe le sont- ils également pour un autre?) et des typologies d'individus (vus simultanément à travers le prisme de plusieurs groupes). Elle se déroule en quatre phases :

1. *Étude de l'inter-structure*. Cette phase analyse globalement les similitudes et différences entre tableaux via des objets (chaque objet caractérisant un tableau)
2. *Recherche d'un compromis*. Cette étape consiste à déterminer un espace commun de représentation résumant les donnée de façon transverse.
3. *Étude de l'intra-structure.* Il s'agit d'analyser le compromis afin de comparer plus finement les tableaux.
4. *Analyse des trajectoires.* A ce stade on compare les profils des individus ou les variables selon les différents groupes.

Parmi les méthodes d'analyse multi-blocs, on trouve la double analyse en composantes principales (**DACP**), la structuration de tableaux à trois indices de la statistique (**STATIS**) et l'analyse factorielle multiple (**AFM**).