Le Lexique du Data Scientist

Par Dimitri PIANETA

mars 2021

Sommaire

0-9

3V

Volume, Vélocité, Variété : ce sont les trois dimensions retenues, à l'origine par IBM, pour décrire un environnement Big Data. Le volume représente la quantité d'information stockée et/ou traitée par le système, la vélocité rend compte des besoins d'une réponse rapide (et de la nécessité d'entraîner les algorithmes en des temps raisonnables), et la variété souligne les nombreux types de données que l'on peut avoir en entrée (structurée, non structurée, libre, etc…)

4V

Pareil que 3V avec en plus Véracité : les données utilisées ne sont pas nécessairement à jour, ni même correctes (il y a beaucoup de désinformation sur Internet). La problématique est donc à considérer pour chaque système Big Data étudié.

Remarque : parfois, on entend "Valeur" à la place de Véracité mais l'idée reste la même.

A

Algorithme – Algorithm

Il s'agit d'une suite d'opérations ou d'instructions permettant de résoudre un problème ou d'obtenir un résultat. Les algorithmes sont au cœur de la data science puisqu'ils servent à décrire l'entraînement des réseaux, la manière qu'ils ont de décider d'un résultat, et de bien d'autres choses encore.

Algorithme adaptatif – Adaptive algorithm

C'est un algorithme dont le comportement change en fonction de paramètres variés (données entrantes, paramètres d'exécution, propriétés du serveur ou de la mémoire, etc…). Ils sont souvent utilisés dans le cadre du boosting car leurs performances s'améliorent avec le temps.

Algorithme génétique – Genetic algorithm

Inspiré des principes de l'évolution génétique, des générations de population sont utilisées pour confronter l'algorithme à un certain environnement dans lequel un "optimum de survie" finira par apparaître.

La théorie de l'évolution présentée par Charles Darwin en 1859 dans son livre "l'Origine des espèces", repose sur trois principes clés : l'hérédité, la variation et la sélection.

Ainsi, lorsqu'un organisme vivant se reproduit, il transmet ses caractéristiques à ses descendants, à travers ses gènes. Cependant l'hérédité des caractères n'est pas parfaite, les gènes peuvent subir des mutations et les descendants peuvent donc présenter des variations de caractères. Par conséquent, les différents individus appartenant à une population d'organismes vivants ne sont pas tous identiques, et certains d'entre eux peuvent avoir des variations qui leur permettent de mieux survivre et de se reproduire davantage dans un certain environment. Ces individus ont donc un avantage sélectif, plus de chances de transmettre leurs gènes à la génération future. À travers ce processus, les organismes s'adaptent à leur environnement, au cours des générations.

Les algorithmes génétiques, et les algorithmes évolutionnaires en général, reposent sur ces principes. Comment fonctionnent-ils ? Tout d'abord, on cherche à représenter les solutions possibles du problème d'optimisation sous la forme d'un génome, puis on génère une population de solutions potentielles au problème donné, on sélectionne ensuite les solutions les plus performantes vis à vis de la tâche à optimiser, on crée alors une nouvelle population en copiant à l'identique les solutions sélectionnées, enfin on applique des opérateurs de variation aux génomes des individus de la nouvelle population, afin de créer des solutions différentes. La population d'enfants devient à son tour la population parentale, et on itère la même procédure jusqu'à ce qu'une solution satisfaisante soit trouvée.

En pratique il existe un grand nombre de façons de représenter les solutions potentielles à un problème, les codages les plus courants sont par exemple les vecteurs de nombres (binaires, entiers ou réels) et les graphes (par exemple des structures analogues aux arbres de décision). De même, il existe différentes méthodes de sélection (tournoi, par rang, uniforme…) et de variation (mutations ponctuelles, cross-over), et le bon choix de ces différents aspects est crucial pour obtenir des résultats pertinents.

Ces algorithmes permettent d'explorer l'espace de solutions possibles de manière non exhaustive, afin d'obtenir une solution satisfaisante. Par conséquent, ils sont particulièrement utiles dans le cas d'espaces de très grande taille, présentant des optima locaux, difficiles à explorer avec des algorithmes déterministes d'optimisation. De plus, ces algorithmes s'adaptent facilement à des espaces de données qui changent dans le temps. A contrario, leur principal inconvénient est que leurs résultats dépendent fortement du choix des différents éléments qui les constituent ainsi que des paramètres associés.

Analyse de données : *Cette famille de méthodes d'apprentissage permet de dégager les aspects les plus intéressants dans la nature et la structure d'un nombre très important de données. Certaines méthodes permettent de faire ressortir des relations entre les données et de décrire de façon plus succincte les principales informations qu'elles contiennent.*

D'autres techniques permettent de regrouper les données de façon à faire apparaître clairement ce qui les rend homogènes. Toutes ces méthodes ont pour principales caractéristiques d'être multidimensionnelles et descriptives. On distingue les méthodes d'apprentissage non supervisé (segmentation, détection d'anomalie, clustering, etc.) des méthodes d'apprentissage supervisé (régression, classification…). Les premières fournissent des informations sur la structure ou la distribution des données. Quant aux méthodes supervisées, elles s'intéressent aux relations et à la dépendance des variables entre elles, mettant en évidence le lien entre la variable d'intérêt et les autres variables observées

API (Application Programming Interface)

Appelée également Interface de programmation, elle permet simplement à une application d'accéder à une autre application pour des données, des fonctionnalités. *L'API de Google aide à récupérer des images sans ouvrir son navigateur. L'API de Facebook permet de poster des commentaires sur son mur depuis un simple programme informatique.*

Apprentissage non supervisé : *A la différence de l'apprentissage supervisé, le contexte non supervisé est celui où l'algorithme doit opérer à partir d'exemples non annotés.*

Il doit faire émerger automatiquement les catégories à associer aux données qu'on lui soumet pour reconnaître qu' un chat est un chat, une voiture, une voiture comme sont capables de le faire les animaux et les humains. Le problème d'apprentissage non supervisé le plus fréquent est la segmentation (ou clustering) où l'on essaie de séparer les données en groupes (catégorie, classe, cluster…) : regrouper des images de voitures, de chats, etc. Beaucoup d'espoirs sont portés sur la détection d'anomalies pour la maintenance prédictive, la cybsersécurité, mais aussi le dépistage précoce de maladies, etc.

De manière générale, l'algorithme cherche à maximiser d'une part l'homogénéité des données au sein des groupes de données et à former des groupes aussi distincts que possible : selon le contexte, on choisit d'utiliser tel ou tel algorithme pour classer les données par exemple selon leur densité ou leur gradient de densité. Dans le cas de la détection d'anomalies, c'est plutôt le caractère extrême ou atypique des valeurs ou d'un pattern dans les données qui est recherché. La métrique sous-jacente joue un rôle clé pour déterminer ce qui est la norme et ce qui s'en éloigne.

Apprentissage par imitation : **Apprentissage par imitation** (ou Apprentissage par renforcement inverse) : *Cette forme d'apprentissage automatique utilise l'expérience des experts pour apprendre, par imitation.*

En apprentissage par renforcement (*Reinforcement Learning* ou RL), l'algorithme essaie de trouver la meilleure stratégie pour atteindre un objectif en interagissant avec l'environnement et en obtenant des « récompenses », façon de qualifier la qualité des actions prises par l'algorithme. En apprentissage par renforcement inverse (*Inverse Reinforcement Learning* ou IRL), l'algorithme observe un « expert » résoudre le problème et essaye d'apprendre à faire aussi bien que lui (voire mieux). Cet expert peut être une personne ou un algorithme qui sait, a priori, résoudre le problème et peut multiplier les exemples. Cet apprentissage aussi qualifié d'apprentissage par imitation (*Apprenticeship Learning*) a l'avantage de ne pas nécessiter de définir de récompenses, problème compliqué et crucial en apprentissage par renforcement.

Prenons un exemple bien connu de problème résolu en utilisant l'IRL : les véhicules autoguidés, ces robots qui se déplacent de façon autonome sans intervention humaine comme on en trouve beaucoup en manutention dans l'industrie ou pour la logistique. Il est facile de faire piloter ces robots par un homme (l'expert) alors qu'il est très difficile de définir des récompenses, autrement dit de qualifier la qualité de la conduite, la vitesse, le positionnement idéal, etc. qui dépendent de multiples paramètres.

En revanche, l'IRL suppose d'avoir accès et d'exploiter de nombreux exemples fournis par l'expert, ce qui peut s'avérer coûteux, en temps, en argent ou en puissance de calcul. Aujourd'hui, les chercheurs ont résolu la plupart des problèmes « d'imitation », grâce à des algorithmes efficaces. Ils travaillent désormais sur des algorithmes qui combinent RL et IRL pour tenter d'obtenir le meilleur des deux modèles d'apprentissage.

Apprentissage par renforcement : *L'apprentissage par renforcement (RL pour Reinforcement Learning) fait référence à une classe de problèmes d'apprentissage automatique, dont le but est d'apprendre, à partir d'expériences successives, ce qu'il convient de faire de façon à trouver la meilleure solution.*

Dans un tel problème, on dit qu'un « agent » (l'algorithme, au sens du code et des variables qu'il utilise) interagit avec « l'environnement » pour trouver la solution optimale. L'apprentissage par renforcement diffère fondamentalement des problèmes supervisés et non supervisés par ce côté interactif et itératif : l'agent essaie plusieurs solutions (on parle « d'exploration »), observe la réaction de l'environnement et adapte son comportement (les variables) pour trouver la meilleure stratégie (il « exploite » le résultat de ses explorations). Un des concepts clés de ce type de problèmes est l'équilibre entre ces phases d'exploration et d'exploitation. Cette méthode est particulièrement adaptée aux problèmes nécessitant un compromis entre la quête de récompenses à court terme et celle de récompenses à long terme. Parmi les exemples de problèmes traités de cette façon, on peut évoquer : apprendre à un robot à marcher en terrain difficile, à conduire (cas de la voiture autonome) ou à accomplir une tâche spécifique (comme jouer au jeu de go), piloter un agent à travers un labyrinthe, etc. Les principales familles de problèmes d''apprentissage par renforcement sont les algorithmes de bandits, les problèmes de décisions (partiellement) markovien et les arbres de jeu.

Apprentissage supervisé : *Ce cadre de machine learning part du fait que les données historiques (ou exemples) sont annotées.*

Prenons le cas de la reconnaissance d'objets : un problème supervisé correspond au cas où le label « voiture » est bien associé, en base, à des photos de voitures, le label « chat » à des photos de chat, etc. L'algorithme apprend ainsi à partir de milliers ou de millions d'exemples étiquetés : il cherche la relation qui permet de relier les images aux labels. Après avoir classifié correctement les exemples, il peut ensuite généraliser ce classement à de nouvelles données : classifier correctement des images de voiture ou de chat qu'il n'a jamais vues durant la phase d'apprentissage. C'est ce qu'on appelle la capacité de généralisation. Dans un cadre business, on parle souvent d'analyse prédictive. Parmi les exemples d'applications, citons la classification d'email en spam ou non selon le contenu du message, son expéditeur, son sujet…, le diagnostic médical selon les symptômes, etc.

Approche probabiliste fréquentiste : *Ce terme générique englobe tous les algorithmes de machine learning basés sur l'approche fréquentiste (par opposition à l'approche bayésienne). Ces méthodes (comme le Kernel Ridge Regression ou le k- Nearest Neighbour,) reposent généralement sur la « loi des grands nombres » et la théorie des « inégalités de concentration ».*

Considérons un jeu de données et un ensemble possible de distributions au sens probabiliste, à savoir les valeurs que peuvent prendre les variables aléatoires et à quelle fréquence. Deux approches probabilistes permettent de classer les variables : l'approche bayésienne et l'approche fréquentiste. La première, que l'on qualifie parfois de théorique ou déductive, combine l'information apportée par les données avec les connaissances a priori provenant soit d'études antérieures soit d'avis d'experts, dans le but d'obtenir une information a posteriori. L'approche fréquentiste, quant à elle, repose sur les observations et consiste à trouver la distribution la plus probable au vue des données, et éventuellement son intervalle de confiance correspondant (c'est à dire l'ensemble des distributions qui ont une chance significative d'être la vraie distribution).

Prenons l'exemple d'une pièce de monnaie et de la probabilité qu'elle tombe sur pile ou face. L'approche fréquentiste se basera sur l'expérience présente (par exemple, la pièce, lancée 10 fois, est tombée 6 fois sur pile) pour établir que la « vraie » probabilité d'obtenir pile. Autrement dit 6/10 soit 0,6 dans cet exemple. Conformément à la « loi des grands nombres », en lançant la pièce un nombre important de fois, cette méthode convergera, au sens mathématique du terme, vers 0,5. Concrètement, on utilise des « inégalités de concentration » pour quantifier l'incertitude du résultat autour de bornes de probabilités. Ainsi, en utilisant l'inégalité d'Azuma-Hoeffding, on sait que si on lance une pièce équilibrée 10 000 fois, la proportion de pile sera comprise entre 0,483 et 0,517 avec une probabilité supérieure à 99 %.

Généralement, du point de vue théorique, l'approche fréquentiste ne nécessite que des hypothèses assez faibles : bien que dans la plupart des cas, on travaille sous l'hypothèse de « variable aléatoire sous-gaussienne », de nombreux travaux ont montré que l'approche fréquentiste peut fonctionner avec des prérequis très faible, comme l'existence d'un « moment d'ordre 2 » [1]).

Enfin, l'une des difficultés majeures des approches fréquentistes est le problème du surapprentissage, qui est généralement contourné en utilisant une régularisation, un outil à la fois riche, flexible mais potentiellement complexe à utiliser. À noter que le débat sur les mérites relatifs entre approche fréquentiste versus bayésienne, qui date des débuts de l'apprentissage automatique, reste toujours d'actualité dans la communauté.

Arbres de décision – Decision tree

1. Une approche visuelle de compréhension des algorithmes peut être d'utiliser des arbres de décision. Les branches de l'arbre représentent les différentes règles qui vont nous guider, et

les feuilles sont le résultat final au problème d'origine. On suit un parcours dans l'arbre qui nous permet, en répondant à des questions, de parvenir à une conclusion.

2. *Cet outil d'aide à la décision ou d'exploration de données permet de représenter un ensemble de choix sous la forme graphique d'un arbre. C'est une des méthodes d'apprentissage supervisé les plus populaires pour les problèmes de classification de données.*

Concrètement, un arbre de décision modélise une hiérarchie de tests pour prédire un résultat. Il existe deux principaux types d'arbre de décision :

- Les arbres de régression (Regression Tree) permettent de prédire une quantité réelle, une valeur numérique (par exemple, le prix d'une maison ou la durée de séjour d'un patient dans un hôpital) ;
- Les arbres de classification (Classification Tree) permettent de prédire à quelle classe la variable de sortie appartient (cela permet par exemple de répartir une population d'individus, comme des clients d'une entreprise en différents types de profils).

Les décisions possibles sont situées aux extrémités des branches (les « feuilles » de l'arbre) et sont atteintes en fonction de décisions prises à chaque étape. Un arbre de décision fonctionne en appliquant de manière itérative des règles logiques très simples (typiquement des séparations de données par « hyperplan », généralisation d'un plan à plus de 2 dimensions), chaque règle étant choisie en fonction du résultat de la règle précédente. Les arbres de décision ont pour avantage d'être simple à interpréter, très rapide à entrainer, d'être non paramétrique, et de nécessiter très peu de prétraitement des données. Ils peuvent être calculés automatiquement par des algorithmes d'apprentissage supervisé capables de sélectionner automatiquement les variables discriminantes au sein de données non-structurées et potentiellement volumineuses. Ces algorithmes permettent aussi d'extraire des règles logiques qui n'apparaissaient pas dans les données brutes. Un autre usage en machine learning consiste à construire non pas un arbre mais une forêt d'arbres de décision. Une décision est alors prise en faisant « voter » l'ensemble des arbres et en choisissant la réponse majoritaire (pour un choix discret) ou la moyenne des réponses (pour une variable continue). Les résultats ainsi obtenus sont remarquables notamment lorsque les arbres de décision sont utilisés en forêts aléatoires

Architecture I/O – I/O architecture

Architecture faisant intervenir des entrées et des sorties de données.

Attrition – Churn

Le churn ou taux d'attrition correspond à la part des clients (ou d'abonnés) perdus sur une période.
Auto-Encoder : *Les auto-encodeurs sont des algorithmes d'apprentissage non supervisé à base de réseaux de neurones artificiels, qui permettent de construire une nouvelle représentation d'un jeu de données. Généralement, celle-ci est plus compacte, et présente moins de descripteurs, ce qui permet de réduire la dimensionnalité du jeu de données. L'architecture d'un auto-encodeur est constituée de deux parties : l'**encodeur** et le **décodeur.***

L'encodeur est constitué par un ensemble de couches de neurones, qui traitent les données afin de construire de nouvelles représentations dites "encodées". À leur tour, les couches de neurones du décodeur, reçoivent ces représentations et les traitent afin d'essayer de reconstruire les données de départ. Les différences entre les données reconstruites et les données initiales permettent de mesurer l'erreur commise par l'auto-encodeur. L'entraînement consiste à modifier les paramètres de

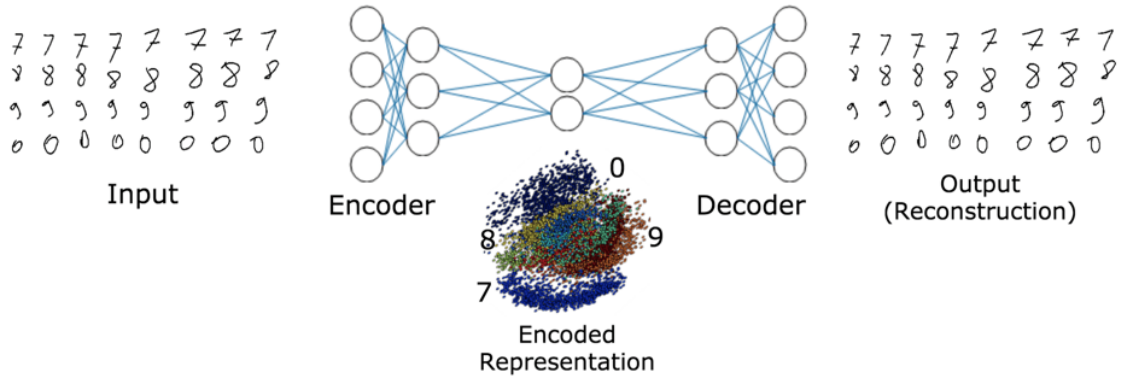
l'auto-encodeur afin de réduire l'erreur de reconstruction mesurée sur les différents exemples du jeu de données.

La plupart du temps, on ne s'intéresse pas à la dernière couche du décodeur, qui contient uniquement la reconstruction des données initiales, mais plutôt à la nouvelle représentation créée par l'encodeur.

L'architecture la plus simple d'un auto-encodeur est semblable à un perceptron multicouche. Cependant, en fonction des données traitées, on peut utiliser différentes topologies de réseaux de neurones. Par exemple, des couches convolutives afin d'analyser des images ou des couches de neurones récurrentes pour traiter des séries temporelles ou des séquences.

À noter qu'à la différence d'un grand nombre de réseaux de neurones, les auto-encodeurs peuvent être entraînés de manière non-supervisée, ce qui permet d'appliquer ces méthodes à des jeux de données non annotés.

La figure suivante schématise un auto-encodeur simple, dont l'encodeur (encoder) traite des images (inputs), afin de les représenter comme des points dans un espace à deux dimensions (encoded representation), puis décode cette représentation (decoder), afin de retrouver les données de départ (output).



AWS (Amazon Web Services)

Ce sont des services proposés par Amazon. Ils regroupent plusieurs fonctionnalités sur le Cloud : espace de stockage, puissance de calcul (pour entraîner des algorithmes par exemple), certains softwares en location.

B

Bagging : *Le mot Bagging est une contraction de Bootstrap Aggregation. Le bagging est une technique utilisée pour améliorer la classification notamment celle des arbres de décision, considérés comme des « classifieurs faibles », c'est-à-dire à peine plus efficaces qu'une classification aléatoire.*

En général, le bagging a pour but de réduire la variance de l'estimateur, en d'autres termes de corriger l'instabilité des arbres de décision (le fait que de petites modifications dans l'ensemble d'apprentissage entraînent des arbres très différents). Pour ce faire, le principe du bootstrap est de créer de « nouveaux échantillons » par tirage au hasard dans l'ancien échantillon, avec remise. L'algorithme, par exemple l'arbre de décision, est entraîné sur ces sous-ensembles de données. Les estimateurs ainsi obtenus sont moyennés (lorsque les données sont quantitatives, cas d'un arbre de régression) ou utilisés pour un « vote » à la majorité (pour des données qualitatives, cas d'un arbre de classification). C'est la combinaison de ces multiples estimateurs « indépendants » qui permet de réduire la variance. Toutefois, chaque estimateur est entrainé avec moins de données. En pratique, la méthode de bagging donne d'excellents résultats (notamment sur les arbres de décision utilisés en « forêts aléatoires »).

Biais : *Le biais est une des deux erreurs utilisée pour définir la qualité d'un algorithme d'apprentissage (l'autre étant la variance).*

Les algorithmes d'apprentissage tentent d'approcher la relation exacte entre des variables d'entrée et de sortie d'un problème, le vrai modèle en quelques sortes. Le modèle utilisé par l'algorithme est plus simple que le problème que l'on cherche à apprendre, il ne permet donc pas de rendre compte de toute sa complexité. On qualifie cette erreur faite dans les hypothèses du modèle de « biais ». Le biais sera d'autant plus faible que le modèle approchera la complexité du problème. Inversement, si le modèle est trop simple, le biais sera très élevé. Par exemple le perceptron est un modèle de classification linéaire trop simple pour des problèmes complexes de classification d'images comme CIFAR[1] : il produira un biais très élevé.

La nature de l'erreur dépend du type de problème considéré. Par exemple, dans un problème de classification d'images, l'erreur pourra être « le % de fois où le modèle se trompe en choisissant les classes » ; dans le cadre d'un problème de régression, le biais pourrait être une erreur des moindres carrés…

Quoi qu'il en soit, l'erreur totale n'est jamais nulle, ne serait qu'à cause du bruit. Cependant, elle peut être très faible. Ainsi, les derniers algorithmes de deep learning atteignent une erreur de 0,01 % sur des problèmes simples comme MNIST[2]. On définit aussi parfois le biais comme la « distance » entre le meilleur modèle pouvant être appris par l'algorithme et le vrai modèle. En machine learning, on cherche, en général, un équilibre entre biais et variance, de telle sorte que ces deux erreurs soient à peu près égales. La « régularisation » est une des nombreuses techniques utilisée pour y parvenir.

Big Data

Les 4V du big data sont **Volume**, **Vélocité**, **Variété** et **Valeurs**. On voit parfois apparaître la Véracité et la Visualisation. Il s'agit d'un terme "commercial" désignant, généralement, toute solution ou domaine s'appuyant sur de grands jeux de données ou sur des problématiques de traitement complexes et en grand nombre.

BigTable

Il s'agit d'un service Cloud de Google App Engine : c'est une base de données orientée colonnes, rapide et développée par Google. Elle n'est pas Open Source (autrement dit, on n'a pas accès à son code), mais elle héberge les services de Gmail, YouTube et même Google Earth.

Bruit : *Le bruit altère les données collectées et risque de rendre difficile l'apprentissage de la relation que l'on cherche à prédire, voire de rendre la modélisation impossible.*

Dans l'immense majorité des problèmes réels, la collecte des données est confrontée à une voire plusieurs formes de bruits. Ces bruits peuvent être d'origine « matérielle » (bruit blanc dans les signaux électriques, bruit au sens acoustique, etc.), d'imprécision ou de défaut dans les mesures issues de capteurs, d'approximation dans les données ou encore de données manquantes car non mesurées.

De fait, la plupart des modèles et des algorithmes sont conçus pour fonctionner malgré la présence de bruit. Cependant, si le bruit est trop « élevé », aucun algorithme ne peut marcher : le problème devient tout simplement mathématiquement impossible à résoudre.

Prenons l'exemple d'un problème de régression scalaire classique. Le but est de prédire une variable aléatoire ***Y*** à partir d'une variable aléatoire ***X***. Un des modèles les plus simples consiste à supposer qu'il existe une fonction $\int$, que l'on cherche à déterminer, telle que dans les données observées,

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

Où ϵ est une variable aléatoire indépendante de ***X*** qui représente le bruit. Si le bruit a ce qu'on appelle de « bonnes propriétés » (il est borné, sous gaussienne, etc.), il existe plusieurs algorithmes pour résoudre le problème et trouver la fonction $\int$ (comme Kernel Ridge Regression). À l'inverse, si ϵ n'est pas de carré intégrable, ce qui revient à dire qu'il n'a pas de bonnes propriétés, le problème est beaucoup plus compliqué à résoudre. Finalement si le bruit n'est pas absolument intégrable, il n'y a généralement pas de solution au problème.

Business analytics

Il s'agit d'une offre de produits informatiques renvoyant le plus souvent aux outils de restitution destinés à l'aide à la prise de décision. On compte notamment SAP, SAS, MicroStrategy.

C

Cassandra

Système de gestion de base de données open source de type NoSQL, un des principaux projets de la Fondation Apache. Cassandra est conçue pour gérer des quantités massives de données réparties sur plusieurs serveurs (clusters), en assurant tout particulièrement une disponibilité maximale des données et en éliminant les points individuels de défaillance.

Classification : *Cette méthode d'analyse de données regroupe des algorithmes d'apprentissage supervisé adapté aux données qualitatives. L'objectif est d'apprendre (autrement dit de trouver) la relation qui lie une variable d'intérêt, de type qualitative, aux autres variables observées, éventuellement dans un but de prédiction.* On utilise la classification lorsque la variable d'intérêt est qualitative, c'est à dire qu'elle prend ses valeurs dans un espace qui ne possède pas de métrique naturelle. Par exemple on peut essayer de prédire le genre littéraire d'un livre ; cette variable est discrète (genre « policier », genre « science-fiction », etc.) et il n'y a aucune relation entre les genres, il est difficile de définir une distance entre eux. Les algorithmes de classification les plus simples sont la régression logistique, le k-nearest neighbour (méthode des k plus proches voisins)… ; les plus complexes sont les réseaux de neurones, les support vector machine, les mixture model (modèles de mélange), le Bayesian classifier (classifieur Bayésien), etc.

Classifieur Bayésien : **Classifieur bayésien** (ou estimateur bayésien) : *Cette méthode de classification réunit une famille d'algorithmes fondés sur le Théorème de Bayes. Leur particularité est de prédire la valeur des paramètres du modèle en termes de probabilité.*

Considérons un jeu de données et un ensemble possible de distributions, au sens probabiliste, à savoir les valeurs que peuvent prendre les variables aléatoires et à quelle fréquence. Supposons que cette famille de distribution puisse être caractérisée par un paramètre (en général baptisé thêta). L'objectif de la classification paramétrique est de déterminer ce paramètre pour être capable de prédire la valeur des variables.

Il existe deux approches probabilistes de classification : l'approche fréquentiste et l'approche bayésienne. La première consiste à trouver le paramètre le plus probable au vue des données. Un classifieur *bayésien* essayera, quant à lui, d'apprendre la distribution du paramètre, en d'autres terme d'estimer la pertinence de chaque valeur possible du paramètre.

Pour ce faire, il faut commencer par lui fournir une distribution « a priori » sur les paramètres (i.e. reflétant les hypothèses que l'on a, a priori, sur le modèle ; cette distribution est appelée ***prior***). L'algorithme apprend la distribution (appelée **posterior**) via la règle de Bayes, en utilisant les données.

Les classifieurs bayésiens bénéficient de nombreux avantages. Notamment, ils ne présentent pas de risque de surapprentissage, et fournissent beaucoup plus d'informations que les classifieurs fréquentistes (puisqu'en plus de la valeur, ils en donnent la probabilité). Cependant ils nécessitent de faire des hypothèses fortes à l'avance sur les distributions des données (via le prior et la famille de distribution fournie à l'algorithme), et sont difficilement applicables aux problèmes non paramétriques.

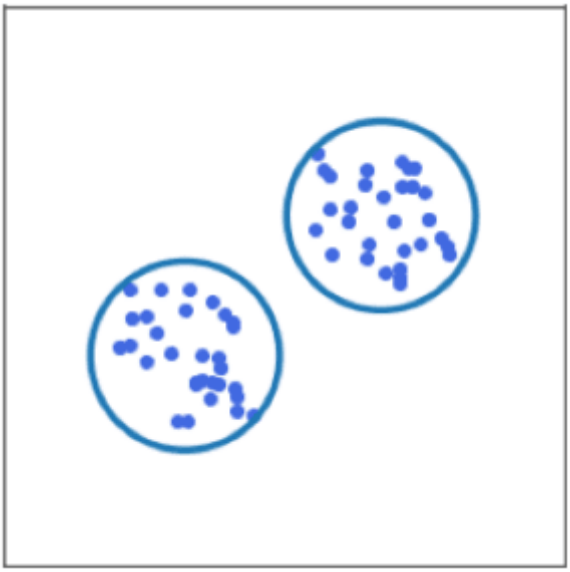
En pratique, la méthode fonctionne relativement bien alors que les hypothèses sur lesquelles elle repose sont généralement fausses dans les problèmes réels – par exemple, les distributions fournies

à l'algorithme au début ne sont généralement pas les vraies distributions des variables aléatoires sous-jacentes au problème. Ainsi, le débat sur les mérites relatifs entre classifieurs fréquentistes versus bayésiens, qui date des débuts de l'apprentissage automatique, reste toujours d'actualité dans la communauté.

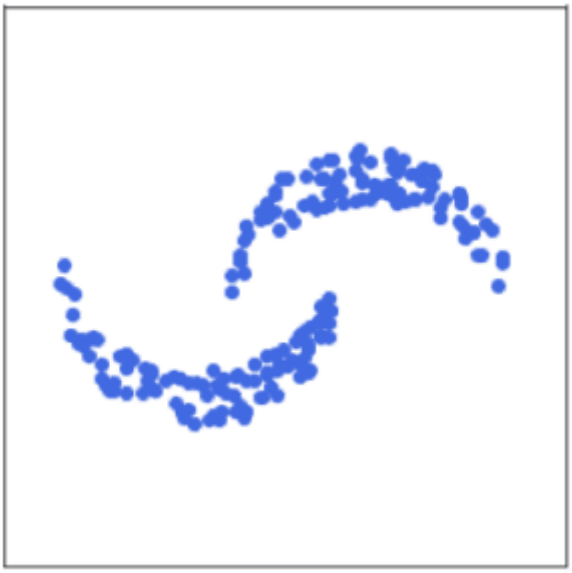
Clustering : (ou partitionnement des données) **:** *Cette méthode de classification non supervisée rassemble un ensemble d'algorithmes d'apprentissage dont le but est de regrouper entre elles des données non étiquetées présentant des propriétés similaires. Isoler ainsi des schémas ou des familles permet aussi de préparer le terrain pour l'application ultérieure d'algorithmes d'apprentissage supervisé (comme le KNN).*

Le clustering est utilisé notamment lorsqu'il est coûteux d'étiqueter le données. C'est néanmoins un problème mal défini mathématiquement : différentes métriques et/ou différentes représentations des données aboutiront à différents regroupements sans qu'aucun ne soit nécessairement meilleur qu'un autre. Ainsi la méthode de clustering doit être choisie avec soin en fonction du résultat attendu et de l'utilisation prévue des données.

Les algorithmes de clustering les plus courants sont le K-Means, les algorithmes de maximisation de l'espérance (de type EM, comme les mixtures gaussiennes) et les partitions de graphes. Voyons comment cela se traduit dans deux exemples :



Ici, les données peuvent être aisément regroupées en deux groupes (dans 2 cercles). L'algorithme K-Means (ou partitionnement en k-moyennes) qui consiste à diviser les points en k groupes appelés clusters, permet d'obtenir ce résultat rapidement et efficacement.



Dans cet autre exemple (le cas des « 2 lunes »), un clustering « naturel » serait de regrouper les données en 2 lunes. L'algorithme K-means ne permet pas de produire ce regroupement. Par contre, les algorithmes de partition de graphes peuvent y parvenir

Cluster

En réseau et système, un cluster est une grappe de serveurs (ou « ferme de calcul ») constituée de deux serveurs au minimum (appelés aussi nœuds) et partageant une baie de disques commune. Il permet d'éviter la redondance de matériel, à l'inverse de l'architecture distribuée.

D

Data Analyst

Maîtrisant les outils du Big Data et les statistiques, le Data Analyst code les algorithmes prédictifs sur la plateforme analytique.

Data mining : *Le data mining couvre l'ensemble des outils et méthodes qui permettent d'extraire des connaissances à partir de grandes bases de données. On parle aussi de fouille, forage ou prospection de données, d'extraction de connaissances à partir de données.*

C'est une analyse préliminaire où l'on explore, on cherche à confirmer des intuitions, à faire émerger des concepts (insights). C'est une façon de produire de la connaissance mais cette étape ne s'automatise pas. Certains y intègrent aussi la transformation des données en informations utiles, en établissant des relations entre les données, des corrélations (on parle aussi de patterns, de motifs, de critères) pour les catégoriser. Le data mining est une extension de l'analyse de données et des statistiques exploratoires pratiquées depuis plus de 30 ans. Il intègre (ou il est le prélude à) des techniques d'analyse issues de l'apprentissage automatique (comme le machine learning), de la reconnaissance de formes et des bases de données de diverses natures dont celles issues d'entrepôts de données (ou Data Warehouse).

Le Data Mining, également surnommé Knowledge Discovery in Data (découverte de savoir dans les données), repose sur des algorithmes complexes et sophistiqués permettant de segmenter les données et d'évaluer les probabilités futures, comme les tendances d'un marché.

Data Scientist

1. A la fois statisticien de haut vol, capable de manipuler les outils informatiques du Big Data et comprendre les enjeux business de ses analyses. Le Data Scientist (parfois traduit en scientifique des données) est l'homme clé du Big Data.

2. *Le but de cet ensemble de disciplines (la « science des données » en français) est d'extraire de la connaissance à partir des données grâce à des techniques et des théories issues des mathématiques appliquées, de la statistique et de l'informatique.*

L'objectif est de produire des méthodes automatisées de tri, d'analyse, de classification de grandes quantités de données et de sources de données, plus ou moins complexes : méthodes statistiques, traitement du signal, méthodes de référencement, apprentissage automatique, visualisation de données.

Data Steward

Orienté base de donnée, le Data Steward est responsable de la cohérence, du modèle et du contenu de l'ensemble des bases de données sous sa responsabilité. Sa mission sera, en particulier, d'exposer certaines informations et d'en agréger de nouvelles.

Datavisualisation ou data visualization : Aussi nommée “Dataviz”, il s’agit de technologies, méthodes et outils de visualisation des données. La présentation sous une forme illustrée rend les données plus lisibles et compréhensibles.

Deep learning : *Le deep learning (ou apprentissage profond) est un ensemble de méthodes d’apprentissage automatique conçues sur la base de réseaux de neurones profonds, visant à mimer la « profondeur » des couches d’un cerveau :*

Le cerveau humain est « profond », dans le sens où chaque action est le résultat d’une longue chaîne de communications synaptiques avec de nombreuses couches de traitement. Le deep learning réunit une classe d’algorithmes d’apprentissage correspondants à ces architectures profondes. Il est souvent utilisé pour un apprentissage « de bout en bout », c’est à dire l’apprentissage simultané des caractéristiques utiles des données, et de la meilleure façon de les utiliser.

Par exemple, pour distinguer une voiture d’une moto, l’algorithme peut apprendre à reconnaître les roues (caractéristique utile), puis à utiliser le nombre de roues pour les distinguer (utilisation de cette caractéristique). Cela fait appel à la fois à des connaissances en neurosciences, en mathématiques et aux progrès technologiques. Le système est constitué d’une série de modules (des couches de neurones), chacun représentant une étape de traitement. Chaque module est entraînable, comportant des paramètres ajustables (similaires aux poids des classifieurs linéaires). A chaque exemple, tous les paramètres de tous les modules sont ajustés de manière à rapprocher la sortie produite par le système de la sortie désirée. Le qualificatif « profond » vient de l’arrangement de ces modules en couches successives. Les réseaux convolutifs et les réseaux récurrents sont deux des architectures les plus en vogue actuellement pour le deep leaning. Les applications sont multiples.

Parmi les exemples récents les plus remarquables, citons le programme de reconnaissance des visages de Facebook, la victoire d’AlphaGo ou encore les outils d’aide à la conduite assistée et autonome (ADAS), la santé avec la recherche de cellules cancéreuses par la start-up DreamQuark, ou la reconnaissance de parole.

Descripteur (feature) : *Un descripteur est une quantité mesurable ou calculable qui permet de décrire en partie un objet, un signal, une donnée…*

Le choix et la conception de descripteurs est généralement une étape préliminaire cruciale de la fouille de données, nécessaire avant toute utilisation d’algorithmes. Les descripteurs sont généralement numériques (ce sont des nombres). Ils peuvent aussi prendre des formes plus complexes. Par exemple, un descripteur de couleur peut être un nombre : 1-bleu ; 2-noir, etc. Cela peut aussi être une description RGB de la couleur : trois chiffres compris entre 0 et 255, chaque chiffre représentant respectivement le dosage du rouge, du vert et du bleu. Concrètement, pour décrire une voiture, on peut imaginer différents descripteurs tels que les dimensions du véhicule, son poids, sa couleur, sa marque, l’année de fabrication, etc. Comme on peut le voir dans cet exemple, un descripteur seul ne suffit généralement pas à caractériser totalement un objet, mais une collection de descripteurs peut s’en approcher. Il n’existe pas d’algorithme pour créer les descripteurs. Par contre on peut utiliser des descripteurs « génériques » puis utiliser des algorithmes pour « sélectionner » les meilleurs. Mais, la meilleure solution reste de créer des descripteurs spécifiques.

Détection d’anomalie : *Le but de la détection d’anomalie est de repérer des données qui ne sont pas conformes à ce à quoi l’on peut s’attendre par rapport aux autres données.*

Il s'agit, par exemple, de données qui ne suivent pas le même schéma ou qui sont atypiques pour la distribution de probabilité observée. La difficulté du problème provient du fait qu'on ne connait pas au préalable la distribution sous-jacente de l'ensemble des données. C'est à l'algorithme d'apprendre une métrique appropriée pour détecter les anomalies. Parmi les exemples d'applications courantes, citons les transactions bancaires (où une anomalie sera vue comme une fraude potentielle), la surveillance des données physiologiques d'un malade (l'anomalie est un problème de santé possible), ou encore la détection de défauts dans des chaines de production. La détection d'anomalie est souvent un problème d'apprentissage de type non supervisé. Les algorithmes typiques de détection d'anomalie sont les one-class SVM, les méthodes d'apprentissage de distribution bayésienne et les random forests.

Données : *C'est la matière première de tout algorithme. Les données peuvent être de diverses natures (signaux, vidéos, séquences…) et plus ou moins structurées.*

Tout élément numérisé et stocké sur un serveur peut être appréhendé comme une donnée : les mesures physiques (acoustiques, issues de capteurs comme des accéléromètres, etc.), le texte, les séquences d'ADN, les images, la vidéo, etc. Certaines données sont binaires comme les images, les sons ou les vidéos, numérisées grâce à des approximations (en découpant les images en pixels monochromes, chaque couleur étant une suite de 24 bits), les sons en une suite de nombres… Les données structurées proviennent de bases de données relationnelles ; les données semi-structurées comportent les CSV (données de tableaux structurées pour de l'analyse quantitative), les logs (traces laissées sur les serveurs), les XML et les JSON (deux formats de stockage de données hétérogènes très utiles dans les applications). Quant aux données non structurées, on y trouve les emails, les documents et les PDF

Données structurées et non-structurées – Structured and Unstructured Data

Les données structurées correspondent aux données que l'on peut normalisé (c'est-à-dire assigner une structure) alors que les données non-structurées ne peuvent pas l'être. *Structuré : page HTML, objet JSON, base de données relationnelle… Non-structuré : fichier texte brut, photographies, vidéos, audio…*

F

First Party Data / Third Party Data

La « first-party data » correspond aux informations acquises sur les internautes visitant un site Web. Ces informations sont récoltées par l'annonceur ou les éditeurs par différents biais (formulaire d'inscriptions, cookies ou outils analytiques rattachés) et peuvent avoir trait à des données comportementales (intérêts, achats, intention d'achat, navigation…) ou déclaratives (âge, CSP…). A l'inverse, la third-party data est collectée par des acteurs spécialisés.
En résumé, la first party data est la donnée collectée par l'annonceur, la third party data est la donnée de source externe.

Flux de clics – Clickstream

Il s'agit du flux de clics généré en permanence par les internautes sur un site Internet. C'est une source précieuse d'information pour les algorithmes de Machine Learning, notamment si on veut étudier le comportement de ses internautes (on utilise également les cookies).

Fondation Apache

Il s'agit d'une organisation à but non lucratif qui développe des logiciels open source sous licence Apache. Les projets les plus connus sont le serveur web Apache HTTP Server, Apache Hadoop, OpenOffice, SpamAssassin, le moteur de recherche Solr…

Fouille de Textes – Textmining

C'est un ensemble de traitements informatiques consistant à extraire des connaissances selon un critère de nouveauté ou de similarité dans des textes produits par des humains pour des humains. Dans la pratique, cela revient à mettre en algorithme un modèle simplifié des théories linguistiques dans des systèmes informatiques d'apprentissage et de statistiques.

Framework
C'est un ensemble de bibliothèques, d'outils, de conventions, et de préconisations permettant le développement d'applications. Il peut être spécialisé ou non. C'est comme un modèle standard, qui permet la réutilisation du code par la suite.

Un Framework connu du monde Java est Spring

G

Google App Engine

Plateforme de conception et d'hébergement d'applications web basée sur les serveurs de Google. A l'inverse d'AWS, c'est gratuit pour des projets à petite échelle. Comme vu précédemment, on y retrouve BigTable pour l'hébergement de base de données.

H

Hackathon : Ce mot désigne à la fois le principe, le moment et le lieu d'un événement où un groupe de développeurs volontaires se réunit pour collaborer sur des sujets de programmation informatique pointus et innovants, sur une période généralement courte (une journée, une nuit, un week-end).

Le terme vient de hack (littéralement pirater, autrement dit trouver quelque chose de malin) et marathon (en référence au travail en général sans interruption). Organisés en équipe, les développeurs ont un objectif commun : tester des idées et produire un prototype, un algorithme dans le cas de la science des données. C'est un moyen de se tester dans un contexte d'émulation, avec parfois des récompenses à la clé. Un jury détermine les vainqueurs. Les investisseurs et les entreprises peuvent aussi dénicher des idées et des talents à ces occasions. Les hackathons peuvent également être organisés en interne comme le pratiquent Facebook, Yahoo, Google ou LinkedIn.

Malgré un délai de réalisation extrêmement court et même si la manifestation est plutôt festive et encourage l'entraide, les enjeux sont sérieux et la méthode est rigoureuse.

Hadoop

Il s'agit d'un framework Open source codé en Java et conçu pour réaliser des traitements sur des données massives. C'est l'un des frameworks les plus utilisés, et permet notamment d'implémenter le MapReduce. Il est actuellement développé par Apache, mais on retrouve chez ses concurrents des équivalents avec Pig, Hive ou Aster.

Hadoop Distributed File System (HDFS)

Composant clé de la plateforme Apache Hadoop, HDFS (Hadoop Distributed File System) est un système de fichiers distribué. Il permet de stocker de très gros volumes de données sur un grand nombre de nœuds.

HBase

Projet open source, Apache HBase est la base de données distribuée qui s'appuie sur Hadoop et son système de fichiers HDFS. La base de données est ACID et de classe NoSQL.

High-performance Analytical Application (HANA)

SAP HANA est la plateforme haute performance "In-Memory" proposée par SAP. C'est une combinaison Hardware/Software ("appliance") qui a vocation à contenir l'ensemble de l'applicatif SAP (parties ERP et BI), afin d'améliorer les performances et d'exploiter les données en temps réel.

Hive : Solution d'entrepôt de données, Apache Hive s'appuie sur Hadoop. Ce logiciel permet de structurer les données en tables, lignes, colonnes comme sur un datawarehouse traditionnel et propose aux développeurs et analystes un langage de requêtage sur les données, HiveQL (un langage proche du langage SQL).

I

Inégalité de concentration : *Les inégalités de concentration fournissent des bornes sur la probabilité qu'une statistique basée sur des tirages successifs d'une variable aléatoire dévie d'une certaine valeur. Parmi les plus connus, citons Azuma- Hoeffding (pour les variables aléatoires bornées) et McDiarmid (pour les martingales à accroissements finis).*

Contrairement à la loi des grands nombres, les inégalités de concentration sont utiles à la fois en théorie et en pratique. Elles permettent de contrôler l'incertitude liée aux quantités statistiques comme la moyenne empirique. Il existe de nombreuses méthodes de calculs d'inégalités.

Exemple : en utilisant l'inégalité d'Azuma-Hoeffding, on sait que si on lance 10 000 fois une pièce équilibrée, la proportion de tirages côté pile sera comprise entre 0,483 et 0,517 avec une probabilité supérieure à 99 %. Ces résultats permettent de calculer des intervalles de confiance autour des prédictions des modèles de machine learning. Ils sont aussi fondamentaux dans plusieurs domaines de recherches comme l'apprentissage par renforcement et les problèmes d'algorithmes de bandits.

Intelligence artificielle : *L'IA est un ensemble de techniques permettant à des machines d'accomplir des tâches et de résoudre des problèmes normalement réservés aux humains et à certains animaux (selon Y. LeCun, présentation au Collège de France).*

Ces tâches sont parfois très simples pour les humains, moins pour les machines comme reconnaître et localiser les objets dans une image, planifier les mouvements d'un robot pour attraper un objet ou conduire une voiture. Elles requièrent parfois de la planification complexe, comme par exemple pour jouer aux échecs ou au jeu de go. Les tâches les plus compliquées nécessitent beaucoup de connaissances et de sens commun, par exemple pour traduire un texte ou conduire un dialogue.

Depuis quelques années, on associe presque toujours l'intelligence aux capacités d'apprentissage. C'est grâce à l'apprentissage qu'un système intelligent capable d'exécuter une tâche peut améliorer ses performances avec l'expérience. C'est grâce à l'apprentissage qu'il pourra apprendre à exécuter de nouvelles tâches et acquérir de nouvelles compétences. Le domaine de l'IA n'a pas toujours considéré l'apprentissage comme essentiel à l'intelligence. Dans le passé, construire un système intelligent consistait à écrire un programme "à la main" pour jouer aux échecs (par recherche arborescente), reconnaître des caractères imprimés (par comparaison avec des images prototypes), ou faire un diagnostic médical à partir des symptômes (par déduction logique à partir de règles écrites par des experts). Mais cette approche "manuelle" a ses limites.

Infographie : L'infographie est un outil à forte valeur ajoutée qui vous permettra de : Communiquer de l'information de manière originale. Simplifier et comprendre l'information plus rapidement. Attirer l'attention (obtenir des partages sur les réseaux sociaux et dans les médias)

Informatique en nuage – Cloud computing

Ensemble de processus qui consiste à utiliser la puissance de calcul et/ou de stockage de serveurs informatiques distants à travers un réseau, généralement Internet.

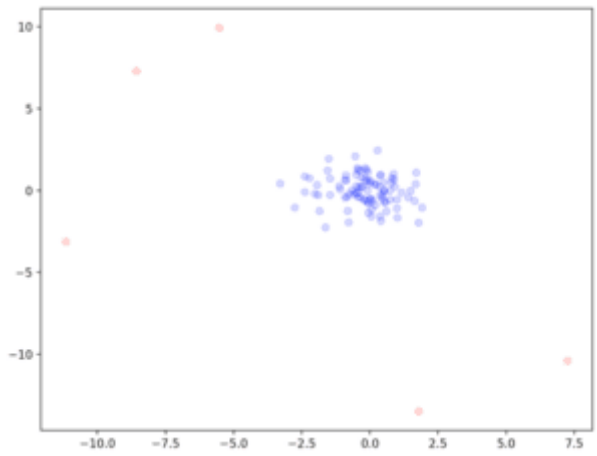
Isolation forest : *Cet algorithme non supervisé de machine learning permet de détecter des anomalies dans un jeu de données. Il isole les données atypiques, autrement dit celles qui sont trop différentes de la plupart des autres données.*

Cet algorithme calcule, pour chaque donnée du jeu, un score d'anomalie, c'est à dire une mesure qui reflète à quel point la donnée en question est atypique. Afin de calculer ce score, l'algorithme isole la donnée en question de manière récursive : il choisit un descripteur et un "seuil de coupure" au hasard, puis il évalue si cela permet d'isoler la donnée en question ; si tel est le cas, l'algorithme s'arrête, sinon il choisit un autre descripteur et un autre point de coupure au hasard, et ainsi de suite jusqu'à ce que la donnée soit isolée du reste.

Le partitionnement récursif des données peut-être représenté comme un arbre de décision et le nombre de coupures nécessaires pour isoler une donnée correspond tout simplement au chemin parcouru dans l'arbre depuis la racine jusqu'à la feuille, représentant la donnée isolée. La longueur du chemin définit le score l'anomalie : les données ayant un chemin très court, c'est à dire les données faciles à isoler, ont également de grandes chances d'être des anomalies, puisqu'elles sont très loin des autres données du jeu.

Comme pour les forêts aléatoires, il est possible d'exécuter cette démarche indépendamment en utilisant plusieurs arbres, afin de combiner leurs résultats pour gagner en performance. Dans ce cas là, le score d'anomalie correspond à la moyenne des longueurs des chemins sur les différents arbres. Cet algorithme s'avère particulièrement utile car il est très rapide et qu'il ne nécessite pas de paramétrage compliqué.

Dans l'exemple suivant, on a appliqué l'algorithme Isolation forest avec 50 arbres à un jeu de données gaussien avec deux descripteurs (2 axes) comportant quelques anomalies (les 5 points les plus extrêmes en rouge sur la figure). Une fois les scores d'anomalies calculés par l'algorithme, on constante que ce sont bien ces 5 points extrêmes qui ont le score le plus élevé.



K

K-Means (ou K-moyennes) : *C'est l'un des algorithmes de clustering les plus répandus. Il permet d'analyser un jeu de données caractérisées par un ensemble de descripteurs, afin de regrouper les données "similaires" en groupes (ou* clusters*).*

La similarité entre deux données peut être inférée grâce à la "distance" séparant leurs descripteurs ; ainsi deux données très similaires sont deux données dont les descripteurs sont très proches. Cette définition permet de formuler le problème de partitionnement des données comme la recherche de ***K*** "données prototypes", autour desquelles peuvent être regroupées les autres données.

Ces données prototypes sont appelés *centroïdes ;* en pratique l'algorithme associe chaque donnée à son centroïde le plus proche, afin de créer des *clusters*. D'autre part, les moyennes des descripteurs des données d'un *cluster*, définissent la position de leur centroïde dans l'espace des descripteurs : ceci est à l'origine du nom de cet algorithme (K-moyennes ou *K-means* en anglais).

Après avoir initialisé ses centroïdes en prenant des données au hasard dans le jeu de données, *K-means* alterne plusieurs fois ces deux étapes pour optimiser les centroïdes et leurs groupes :

1. Regrouper chaque objet autour du centroïde le plus proche.

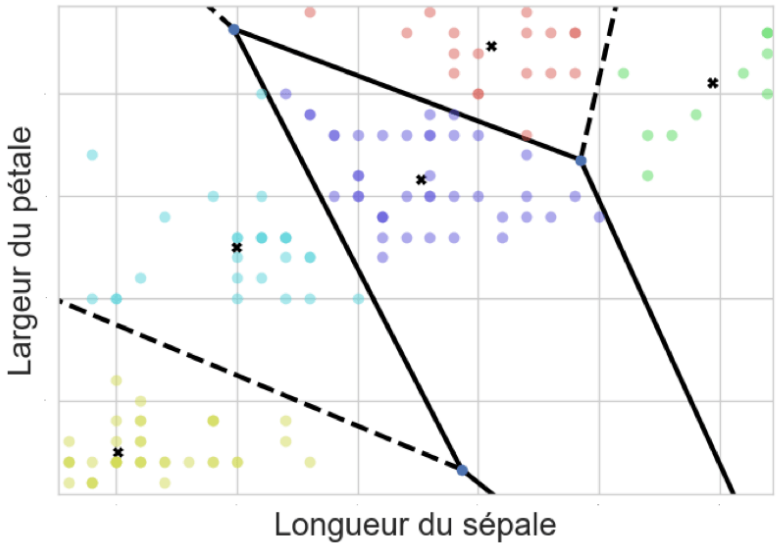
2. Replacer chaque centroïde selon la moyenne des descripteurs de son groupe.

Après quelques itérations, l'algorithme trouve un découpage stable du jeu de données : on dit que l'algorithme a convergé.

Comme tout algorithme, *K-means* présente des avantages et des inconvénients : il est simple, rapide et facile à comprendre ; cependant il ne permet pas de trouver des groupes ayant des formes complexes.

L'exemple suivant s'appuie sur le célèbre jeu de données "Iris" qui décrit des fleurs par l'intermédiaire des longueurs et largeurs de leurs pétales et sépales. Les descripteurs considérés ici sont la longueur des sépales et la largeur des pétales. Chaque point correspond à une fleur et la couleur associée au point reflète son appartenance à un groupe. Les centroïdes de chaque groupe

sont représentés par des croix, les frontières par des traits.



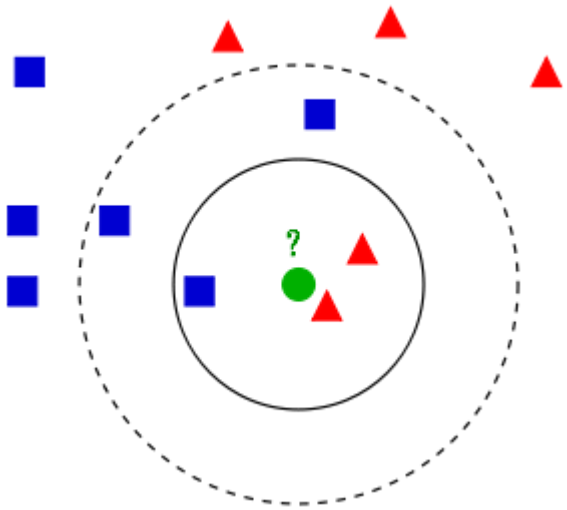
k-Nearest Neighbours : (k-NN voire KNN ou méthode des k plus proches voisins) : *k-NN est un algorithme standard de classification qui repose exclusivement sur le choix de la métrique de classification. Il est "non paramétrique" (seul k doit être fixé) et se base uniquement sur les données d'entraînement.*

L'idée est la suivante : à partir d'une base de données étiquetées, on peut estimer la classe d'une nouvelle donnée en regardant quelle est la classe majoritaire des k données voisines les plus proches (d'où le nom de l'algorithme). Le seul paramètre à fixer est k, le nombre de voisins à considérer (voir figure).

Les métriques les plus souvent choisies sont la distance usuelle dite euclidienne (comme dans la figure) et la distance de Mahalanobis (qui tient compte de la variance – du point de vue statistique – et de la corrélation entre les données). Bien que l'algorithme puisse fonctionner avec ces métriques par défaut, il est généralement bien meilleur quand il est utilisé avec une métrique adaptée aux données, métrique qui peut être calculée à partir d'heuristiques connues liées au problème (par exemple la distance euclidienne pondérée).

Les points faibles de cet algorithme sont : d'une part, son coût en puissance de calcul (pour prédire l'image d'un nouveau point, on doit calculer sa distance à tous les autres), d'autre part le fait de devoir conserver toutes les données d'entraînement en mémoire (k-NN convient donc plutôt aux problèmes d'assez petite taille).Il est également important de noter que cet algorithme est vulnérable à la « *curse of dimensionality* » : le nombre de données nécessaires pour avoir un bon estimateur croît potentiellement de manière exponentielle avec la dimension, autrement dit avec la complexité de la représentation des données. Pour ces raisons, k-NN est assez peu utilisé dans sa forme première mais plutôt avec des versions améliorées qui limitent partiellement ces défauts .

A remarquer que l'algorithme peut aussi être adapté pour la régression.



Quelle est la classe du point vert ? Celle des triangles rouges (délimitée par le cercle continu) ou celle des carrés bleus (cercle tracé en pointillés) ? Si le nombre de plus proches voisins, k, est fixé à 3, la classe du point vert est celle des triangles rouges, car ces derniers sont au nombre de 2 contre un seul carré bleu. Si k vaut 5, la classe du point vert est celle des carrés bleus, au nombre de 3 contre 2 triangles rouges. (Source de l'image : Wikipédia CC BY-SA 3.0)

L

Lac de données – Data Lake

L'approche Data Lake ou lac de données consiste à mettre en place un cluster de type Hadoop, par exemple, où vont converger toutes les données brutes que l'entreprise peut capter. Ces données seront ensuite mises à disposition de l'ensemble du SI pour entraîner des algorithmes de Machine Learning ou faire des statistiques.

Langage informatique

Notation conventionnelle destinée à formuler des algorithmes et produire des programmes informatiques qui les appliquent. D'une manière similaire à une langue naturelle, un langage de programmation est composé d'un alphabet, d'un vocabulaire, de règles de grammaire, et de significations.
Quelques exemples de language de programmation: SAS, R, SQL, Matlab, Fortran, Cobol, Python , Perl, JS, Bash, Java, C++… L'indice TIOBE permet de suivre la "popularité" des différents langages dans le temps.

Loi des grands nombres *: La loi des grand nombres est un théorème mathématique fondamental des probabilités et statistiques.*

Cette loi exprime le fait que les caractéristiques d'un échantillon aléatoire se rapprochent des caractéristiques statistiques de la population (ensemble d'individus ou d'éléments) lorsque la taille de l'échantillon augmente à l'infini. En d'autres termes, cela garantit que, lorsque le nombre de tirages effectués selon une loi de probabilité (comme les tirages successifs d'une pièce sur le côté pile ou face) tend vers l'infini, la *moyenne empirique* (moyenne calculée à partir des observations) converge vers la moyenne réelle d'une variable aléatoire suivant cette loi. Cela sous des hypothèses très faibles.

C'est un des premiers résultats qui lie les observations d'un événement – par exemple les tirages d'une pièce – avec sa variable aléatoire – ici une distribution de type Bernoulli (distribution discrète de probabilité qui prend la valeur 1 avec la probabilité p et la valeur 0 avec la probabilité q = 1 – p). C'est sur cette loi que reposent la plupart des sondages (ils interrogent un nombre suffisamment important de personnes pour connaître l'opinion ou les comportements de la population entière) ou l'assurance (en déterminant les probabilités que les sinistres garantis se réalisent ou non).

Il est à noter que la loi des grands nombres n'offre pas d'utilisation propice en pratique, car le nombre de tirages nécessaires pour approcher suffisamment la moyenne réelle est inconnu. Cette loi n'a en fait qu'une valeur asymptotique. Lorsqu'on exécute un nombre fini d'expériences, il y a des écarts par rapport au comportement moyen attendu. Ainsi, après 10 000 tirages d'une pièce équilibrée, on n'a pas la garantie d'observer 5 000 fois « pile » ni même plus de 4 000. Dans la pratique, on utilise d'autres résultats tels que les Inégalités de concentration ou le Théorème central limite.

Loi gaussienne *: La loi gaussienne (ou normale) est une des lois de probabilité les plus utilisées dans les sciences appliquées du fait de ses propriétés théoriques remarquables.*

La loi gaussienne est une loi de probabilité paramétrique. Elle est caractérisée par sa moyenne μ et sa varianceσ^2.

On la note $\mathcal{N}(\mu, \sigma^2)$

Une variable aléatoire X suivant une loi $\mathcal{N}(\mu, \sigma^2)$ a pour densité de probabilité :

$$\forall x \in \mathbb{R}, p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

Cette fonction de densité, appelée courbe de Gauss, possède une forme caractéristique rappelant celle d'une cloche. Elle se répartit de manière symétrique autour de la moyenne, point où elle atteint son maximum. Elle décroît ensuite à mesure que les valeurs sont éloignées de la moyenne. Cette concentration de la densité autour de la moyenne est une caractéristique importante de la loi gaussienne. Elle se traduit notamment par le fait qu'une majorité des valeurs observées est contenue dans un intervalle restreint autour de la moyenne. Par exemple, la probabilité d'avoir une valeur comprise dans l'intervalle **[μ – 1.96 σ, μ+ 1.96 σ]** est égale à 95 % et la probabilité d'avoir une valeur comprise dans l'intervalle **[μ – 3.29 σ, μ+ 3.29 σ]** est de 99.9 %.

M

Machine Learning

1. Discipline issue de l'intelligence artificielle, le Machine Learning ou apprentissage automatique consiste au développement d'algorithmes qui apprennent un phénomène à partir des données. L'apprentissage est automatique, à la différence du Data Mining classique, où les analyses sont réalisées par le statisticien, à posteriori.

2. *Plutôt que d'expliquer à un ordinateur avec précision comment résoudre un problème, le Machine Learning (ou apprentissage automatique) permet de lui apprendre à apprendre à résoudre un problème par lui-même. Ce champ d'étude comporte des dizaines d'algorithmes.*

   On parle aussi de systèmes entrainables car ces algorithmes sont capables de faire émerger des règles mathématiques dans les données en s'entrainant sur la base d'exemples, puis d'appliquer ces règles à de nouvelles données en s'améliorant sans cesse avec l'expérience. Parmi les algorithmes les plus courants, on trouve les SVM (Support Vector Machine), le boosting, les random forests, les réseaux de neurones, les réseaux bayésiens, etc. Ils opèrent dans des contextes variés : supervisé, semi-supervisé ou non-supervisé, en mode séquentiel ou batch (par lot), par renforcement, etc. Ce sont des systèmes « entrée-sortie » avec une donnée en entrée (image, son, texte) et une en sortie (telle que la catégorie de l'objet dans l'image, le mot prononcé, le sujet dont parle le texte).

   Toutes les tâches nécessitant d'entrer des données et de les classifier peuvent ainsi être automatisées : cela permet de doter des ordinateurs ou des machines de systèmes de perception de leur environnement comme la vision, la reconnaissance d'objets (visages, schémas, langages naturels, écriture, formes syntaxiques…), de la parole ; sur Internet, cela permet de filtrer des contenus indésirables (spam), d'ordonner des réponses à une recherche, de faire des recommandations ou de sélectionner les informations intéressantes pour chaque utilisateur (moteurs de recherche) ; de concevoir des systèmes d'aide aux diagnostics, médical notamment, des programmes de jeu, des interfaces cerveau-machine, des systèmes de détection de fraudes à la carte de crédit, d'analyse financière, de classification des séquences d'ADN, d'analyse prédictive en matière juridique et judiciaire…

Machines à vecteurs de support

Appelé aussi Support Vector Machine en anglais, les machines à vecteurs de support sont des techniques de Machine learning notamment issues de la bioinformatique et utilisées dans des problèmes de discrimination, par exemple pour classer des acheteurs dans des segments.

MapReduce

C'est une procédure de développement informatique, inventée par Google, dans laquelle sont effectués des calculs parallèles de données très volumineuses, distribués sur différentes machines dans des lieux différents (Clusters ou Cloud computing).

Trois étapes:

- Map: Diviser les données à traiter en partitions indépendantes (envoie les données et la fonction à un endroit donné),
- Exécuter les fonctions en parallèle
- Reduce: Combiner les résultats (opération inverse du Map)

Méthode des k plus proches voisins – K Nearest Neighbors (kNN)

Il s'agit d'un algorithme de classification simple. Il permet de placer un nouvel élément dans une classe en le comparant au k éléments les plus proches.

***Modèles de « bandits »** : Cette famille d'algorithmes propose des stratégies optimales pour maximiser l'espérance d'un gain lors d'une succession de choix entre plusieurs actions aux récompenses inconnues (on parle aussi de maximiser le rendement et de minimiser le regret).*

N

Nettoyage des données – Data Cleansing

C'est une phase qui consiste à supprimer les données incohérentes, corriger les erreurs comme, par exemple, des données mal saisies. Disposer d'informations d'un bon niveau de qualité est un préalable à l'élaboration d'algorithmes de Machine Learning.

Normalisation *:* **Normalisation :** *La normalisation est une méthode de prétraitement des données qui permet de réduire la complexité des modèles. C'est également un préalable à l'application de certains algorithmes.*

NoSQL

Acronyme de "Not only SQL", il désigne les bases de données de nouvelle génération (souvent volumineuses) qui se démarquent des bases de données relationnelles classiques et ne sont plus forcement interrogeables en SQL.

On dénombre 4 types de bases de données NoSQL: Orientées colonnes (cf. BigTable), Orientées graphe, Orientées clé-valeur et Orientées document.

O

OpenData

Si le mouvement données ouvertes / OpenData n'est pas directement lié au Big Data, ce dernier tire très directement profit des données publiques pour enrichir les données issues des capteurs ou les données clients avec les informations librement accessibles sur le Web.

***Overfitting** : Ce phénomène de « surapprentissage » dégrade la performance des algorithmes de machine learning.*

Dans quels cas se produit l'overfitting ? Un algorithme d'apprentissage, par exemple supervisé, cherche le modèle qui exprime le mieux la relation entre des données. L'overfitting intervient lorsque l'algorithme sur-apprend (*overfit*), autrement dit, lorsqu'il apprend à partir des données mais aussi à partir de patterns (schémas, structures) qui ne sont pas liés au problème, comme du bruit. Ainsi, l'overfitting est caractérisé par une erreur de type variance très élevée. Concrètement, on observe généralement de l'overfitting lorsqu'on utilise des modèles très complexes sur des problèmes simples mais bruités : par exemple, lors de l'utilisation de Support Vector Machine (SVM ou machine à vecteur de support) avec des noyaux polynomiaux de très haut degré dans le cadre de l'apprentissage d'un problème linéaire (c'est à dire d'un polynôme de degré 1). En d'autres termes, ce type de modèle conduit à de mauvaises performances car, trop complexe, il manque de capacité de généralisation. La technique la plus courante pour limiter le phénomène est la régularisation qui permet de réduire l'erreur de type variance.

P

Pig

Langage de scripting de la plateforme Hadoop.

Plateforme de Gestion d'Audiance (PGA) – Data Management Platform (DMP)

Outil permettant à une entreprise de regrouper l'ensemble des données issues de différents canaux (web, mobile, centre d'appels, etc.) et d'en tirer profit.

Prédictif

Les algorithmes prédictifs constituent une application directe des techniques de Machine Learning dans le Big Data. A partir d'un historique d'achats, de sessions de navigation sur un site internet, ces algorithmes vont prédirent quels seront les prochains besoins d'un consommateur. A partir de l'analyse des vibrations d'un moteur, un algorithme prédictif va diagnostiquer une panne avant qu'elle ne survienne.

Python

Langage de programmation Open Source, très utilisé dans le traitement des données en masse. Il est facile à apprendre et à utiliser, flexible et puissant.

Q

Qualité des données

C'est l'un des problèmes clés du Big Data : pour que les algorithmes fonctionnent correctement, ils doivent pouvoir s'appuyer sur des données fiables et cohérentes. Cela impose un gros travail de nettoyage en amont pour ne pas faire ce qu'on appelle du "Machine Learning on dirty data".

R

R

Langage lié à l'analyse statistique, R s'impose de plus en plus comme le langage du Big Data. Projet open source, R bénéficie d'un fort soutien du secteur universitaire ainsi que de la société Revolution Analytics, rachetée par Microsoft en 2015.

Reconnaissance automatique de la parole : La reconnaissance automatique de la parole est une technique informatique qui permet d'analyser la voix humaine captée au moyen d'un microphone pour la transcrire sous la forme d'un texte exploitable par une machine.

Reconnaissance de forme : La reconnaissance de formes est un ensemble de techniques et méthodes visant à identifier des motifs informatiques à partir de données brutes afin de prendre une décision dépendant de la catégorie attribuée à ce motif.

Régression *: Cette méthode d'analyse de données regroupe des algorithmes d'apprentissage supervisé adaptés aux données quantitatives. L'objectif est d'apprendre (autrement dit de trouver) la relation qui lie une variable d'intérêt, de type quantitative, aux autres variables observées, éventuellement dans un but de prédiction.* On utilise la régression lorsque la variable d'intérêt est quantitative, c'est à dire « à valeur » dans un espace métrique – la métrique est une notion de distance définie dans l'espace – et souvent « à valeur continue ». Par exemple on peut essayer de prédire l'âge d'un utilisateur en fonction de son comportement ; l'âge est une donnée continue avec la métrique usuelle des nombres réels (23 ans et 22 ans sont distants de 1 an). Les algorithmes de régression les plus simples sont de type régression linéaire, les plus compliqués de type régression à noyau des moindres carrés, réseau de neurones, support vector machine, etc.

Régression linéaire

Modèle de régression d'une variable expliquée sur une ou plusieurs variables explicatives dans lequel on fait l'hypothèse que la fonction qui relie les variables explicatives à la variable expliquée est linéaire dans ses paramètres. Le modèle de régression linéaire est souvent estimé par la méthode des moindres carrés.

Régression logistique

Algorithme prédictif utilisé dans le scoring des clients.

Réseaux de neurones

Algorithmes inspirés par le fonctionnement des neurones biologiques. Le fonctionnement d'un réseau de neurones éventuellement disposés en plusieurs couches est simulé. On définit le nombre de neurones, le nombre de couches et l'algorithme fonctionne en boîte noire.

Rétropropagation : *La rétropropagation du gradient de l'erreur (ou backpropagation) est un algorithme d'optimisation permettant d'ajuster les paramètres d'un réseau de neurones multicouches pour mettre en correspondance des entrées et des sorties référencées dans une base d'apprentissage.*

Pour pouvoir entraîner ces systèmes, il faut savoir comment ajuster les paramètres de chaque couche de neurones. La rétropropagation permet de calculer le gradient de l'erreur pour chaque neurone, de la dernière couche vers la première. Le calcul de ce gradient se fait par la méthode de rétropropagation, pratiquée depuis le milieu des années 80. Cela permet de corriger les erreurs selon l'importance des éléments qui ont justement participé à la réalisation de ces erreurs. Ainsi, les poids synaptiques qui contribuent à engendrer une erreur importante se verront modifiés de manière plus significative que les poids qui ont engendré une erreur marginale. Moyennant quelques précautions lors de l'apprentissage, les procédures d'optimisation finissent par aboutir à une configuration stable, généralement un extremum local, au sein du réseau de neurone.

S

Score – Scoring

Note attribuée à un prospect pour évaluer son appétence à une offre, le risque de perte de son client (attrition) ou encore un risque d'impayé. Un scoring peut notamment être calculé selon la méthode RFM (Récence, Fréquence, Montant).

Spark

Modèle de programmation Big Data publié sous licence open source sous l'égide de la fondation Apache. La solution est de type distribuée et "in-memory" et s'avère bien plus rapide qu'Hadoop.

Surapprentissage

Phénomène qui affecte certains algorithmes de Machine Learning, notamment les réseaux de neurones, et qui voit leur efficacité décroître au-delà d'un certain seuil. Engorgé par trop de données, l'algorithme perd peu à peu son pouvoir prédictif.

Système de Fichiers Distribués (SFD) – Distributed File System (DFS)

En français, système de fichiers distribués ou système de fichiers en réseau. C'est un système de fichiers qui permet le partage de fichiers à plusieurs clients au travers du réseau informatique. Contrairement à un système de fichiers local, le client n'a pas accès au système de stockage, et interagit avec le système de fichiers via un protocole adéquat. Ce sont souvent des services basés dans le Cloud.

Système de Gestion de Base de Donnée (SGBD) – Data Base Management System (DBMS)

Il s'agit d'un logiciel système destiné à stocker et à partager des informations dans une base de données, en garantissant la qualité, la pérennité et la confidentialité des informations, tout en cachant la complexité des opérations.
Les principaux types de DBMS:

- modèle hiérarchique
- modèle multidimensionnel
- modèle relationnel

T

Text mining : fouille de texte. La fouille de textes ou « l'extraction de connaissances » dans les textes est une spécialisation de la fouille de données et fait partie du domaine de l'intelligence artificielle. Cette technique est souvent désignée sous l'anglicisme text mining.

Traitement Automatique du Language Naturel (TALN) – Natural Language Processing (NLP)

Ce sont des traitements qui permettent aux machines de mieux comprendre les éléments de languages de l'homme pour mieux interagir avec lui. Les problèmes NLP sont réputés complexes du fait que les machines ne saisissent pas encore le sens des mots qu'elles manipulent. *Faire un résumé d'un texte, définir si la personne qui s'exprime est contente, implémenter un robot de discussion sont des problèmes NLP.*

Traitement de l'image : Le traitement d'images est une discipline de l'informatique et des mathématiques appliquées qui étudie les images numériques et leurs transformations, dans le but d'améliorer leur qualité ou d'en extraire de l'information.

V

Variance

La variance est une mesure servant à caractériser la dispersion d'un échantillon ou d'une distribution. Elle indique de quelle manière la série statistique ou la variable aléatoire se disperse autour de sa moyenne ou son espérance. Une variance de zéro signale que toutes les valeurs sont identiques. Une petite variance est signe que les valeurs sont proches les unes des autres alors qu'une variance élevée est signe que celles-ci sont très écartées.
La racine carrée de la variance est l'écart-type. Dans la pratique, on préfère l'écart type (lettre grecque sigma) à la variance, car l'écart type peut être comparé à l'ordre de grandeur des valeurs, ce qui n'est pas le cas de la variance.

Y

YARN

Outil de gestion des tâches d'un cluster Hadoop.