Modèles formels pour le Big Data

Contenu

Contenu

- Introduction aux modèles formels pour le Big Data
- Rappels de statistiques élémentaires (statistique descriptive unidimensionnelle et bidimensionnelle)
- Modélisation du problème d'apprentissage
- Apprentissage supervisé : concept formel et application
  - Régression linéaire
  - Régression logistique
  - Arbres de décision
  - Analyse en composantes principales
  - Réseaux de neurones

Partie 1 : Introduction aux modèles formels pour le Big Data

Plan

1. Introduction générale
2. Modèles formels : définition et exemple
3. Sources de données
4. Big Data
5. Science des données
6. Initiation au langage R

Introduction générale

- **Méthodes formelles** :
  Méthodes/techniques permettant de raisonner rigoureusement, à l'aide de logique mathématique (descriptions mathématiques formelles), pour démontrer leur validité par rapport à une certaine spécification.

- **Données** (*en anglais Data*) :
  Ensemble de données numériques (personnelles ou professionnelles) crées par l'utilisation des nouvelles technologies (e.g. web, messageries, e-commerce, réseaux sociaux, objets connectés, géolocalisation, ...).

Modèles formels : définition

- **Modélisation** (en anglais Modelling) :
  "A scientific theory is formalised as a mathematical model of **reality**, *from which can be deduced or calculated the observable properties and behaviour of a* **well-defined class of processes** *in the physical world" [C. A.R Hoare]*
- **Deux principales notions de modèles en informatique** :
  1. **Un modèle** : Un modèle est une approximation de la réalité par une structure mathématique.
  2. **Un objet** : Un modèle O de réalité R, si O permet de répondre aux questions que l'on se pose sur R.
- **Exemple** :

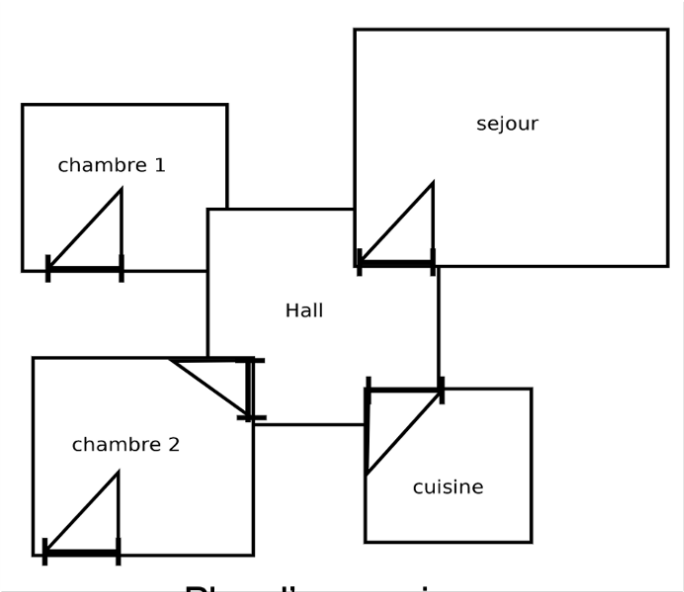
En **mathématique** : systèmes d'équations portant sur des grandeurs (énergies, masses, ...) ou des lois hypothétiques.

$$f(x) = \begin{cases} y = x - 1, & x < 0 \\ y = x + 1, & x >= 0 \end{cases}$$

$$f : entier \rightarrow entier$$

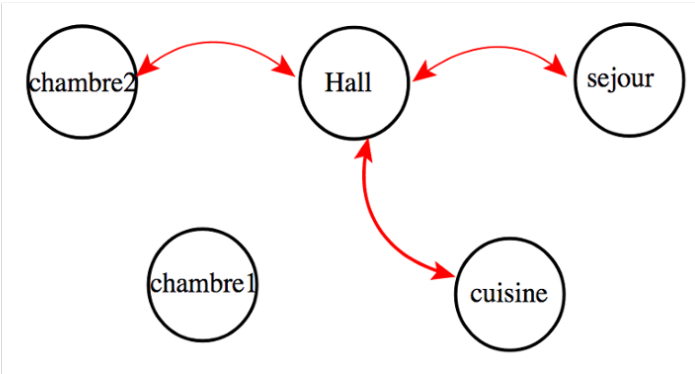
$$x \mapsto y$$

Modèles formels : exemple



Plan d'une maison

Modélisation



Modèle à base de graphe

Raisonnement:
Peut-on aller du Hall vers toutes les pièces ?

Modèles formels : exemple

Comment vérifier que l'identité $(a+b)^2 = a^2 + b^2 + 2(a*b)$ est correcte ?

Une vérification naïve pourrait consister à examiner toutes les valeurs possibles de a, à les croiser avec toutes les valeurs possibles de b et, pour chaque couple, à calculer $(a+b)^2$, puis $a^2 + b^2 + 2(a*b)$ et à s'assurer que l'on obtient le même résultat. Si les domaines de a et de b sont grands, cette vérification peut être très longue. Et si les domaines sont infinis (par exemple les réels), cette vérification ne peut pas être exhaustive.

En vérification formelle, on utilise des valeurs symboliques et on applique les règles qui régissent le $+$ et l'$*$. Ici, les règles pourraient être:

- $\forall x,\ x^2 = x * x \quad (R1)$
- $\forall x, y, z,\ x * (y+z) = x * y + x * z \quad (R2)$
- $\forall x, y,\ x * y = y * x \quad (R3)$
- $\forall x,\ x + x = 2x \quad (R4)$
- $\forall x, y,\ x + y = y + x \quad (R5)$

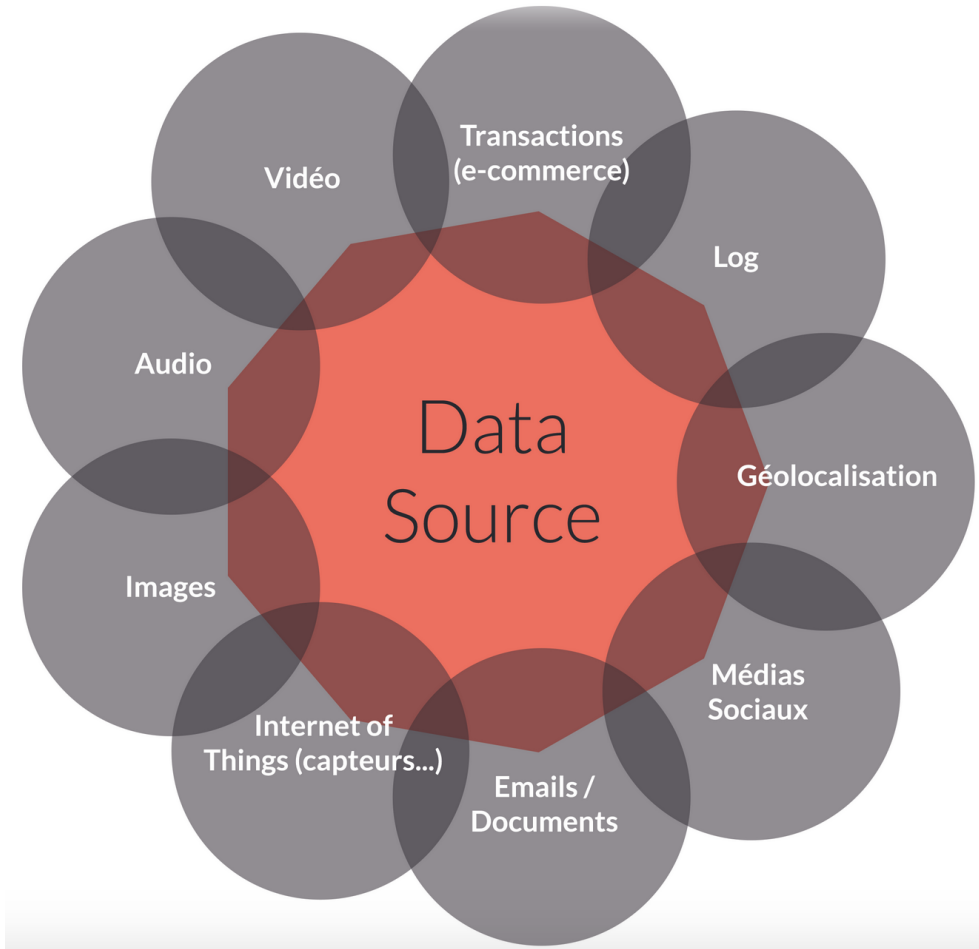
En se servant de ces règles, on arrive à montrer que $(a+b)^2 = a^2 + b^2 + 2(a*b)$.

Démonstration

$$
\begin{aligned}
(a+b)^2 &= (a+b) * (a+b) && (R1)\\
&= (a+b) * a + (a+b) * b && (R2)\\
&= a * (a+b) + b * (a+b) && (R3)\\
&= a * a + a * b + b * a + b * b && (R2)\\
&= a^2 + a * b + b * a + b^2 && (R1)\\
&= a^2 + a * b + a * b + b^2 && (R3)\\
&= a^2 + 2(a * b) + b^2 && (R4)\\
&= a^2 + b^2 + 2(a * b) && (R5)
\end{aligned}
$$

Sources de données

- Type de données les plus répandues :
  - Données **structurées** (fXML, JSON, bases de données, ...)
  - Données **non structurées** (données multimédia : image, vidéo, son ; pages web, mails, ...)



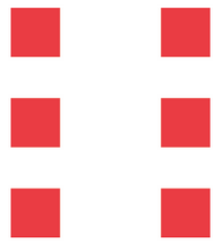
Big Data : Les 5 V du Big Data

Volumétrie

Une volumétrie importante qui ne peut être traitée par les solutions classiques.
Tera, Peta Octets de données.
Une volumétrie croissante.

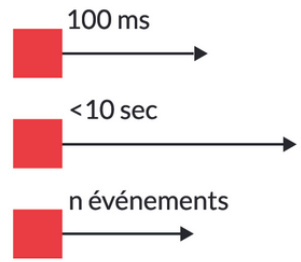
Variabilité

Des données hétérogènes, JSON, CSV, texte...
Des formats non encore connus.



Vélocité

Capacité à traiter des données en quasi temps réel.
Analyse prédictive/ approximative sur un échantillon de données.

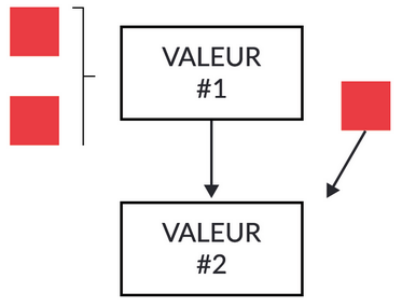


Véracité

Toutes les données n'ont pas la même valeur, le même poids dans l'algorithme de traitement.

Valeur

Les données sont valorisées pour assurer pertinence, création de valeur pour les clients et pour les entreprises.



- **Volume** : augmentation exponentielle des données (jusqu'à plusieurs milliers de téra octets)
  logs, réseaux sociaux, e-commerce, catalogue produit, analyse des données, monitoring, ...
  Les technologies traditionnelles (Business Intelligence, bases de données) n'ont pas été pensées pour de telles volumétries.
- **Variabilité/Variété** : Les données à traiter sont de natures multiples (structurées et non structurées)
  Les données non structurées peuvent faire l'objet d'une analyse sémantique permettant de mieux les structurer et les classer, entraînant une augmentation du volume de données à stocker.
- **Vélocité** : La vitesse de traitement élevée permet d'offrir des capacités temps réel d'analyse et de traitements des données.
  Dans certains cas l'accès et le partage des données doivent se faire en temps réel
- **Valeur** : permettre de monétiser les données (e.g. les données d'une entreprise).
  Ce n'est pas une notion technique mais économique.
  On va mesurer le retour sur investissements de la mise en ¡uvre du Big Data et sa capacité à s'autofinancer par les gains attendus pour l'entreprise.
  Plus on souhaite apporter de la valeur aux données, plus le coût et la complexité de la chaîne augmente (Chaîne de valorisation des données : données brutes –> données préparées –> modèle prédictif)
- **Véracité** : la capacité à disposer de données fiables pour le traitement
  S'intéresser à la provenance des données afin de déterminer s'il s'agit de données de confiance (e.g. en fonction du critère de confiance, on accordera plus ou moins d'importance à la donnée dans les chaînes de traitement)
  Exemple : cas des données incomplètes (dont l'anonymisation a enlevé une partie de la valeur statistique) et le cas de données trop anciennes.

Science des données

Origine de la Data Science

Le terme de *data scientist* à été "inventé" par D. Patil (LinkedIn) [1] et J. Hammerbacher (Facebook) en cherchant comment caractériser les métiers des données pour les offres d'emploi : *Analyste, ça fait trop Wall Street ; statisticien, ça agace les économistes ; chercheur scientifique, ça fait trop académique. Pourquoi pas "data scientist" ?*

Une "définition" attribuée à J. Wills (Cloudera) est souvent reprise :*Data scientist (n) : Person who is better at statistics than any software engineer and better at software than any statistician*

Article de l'OBS [1]

Comment avez-vous inventé le mot "data scientist", c'est à dire scientifique des données ?

En 2008, je travaillais pour la société LinkedIn, et j'ai co-inventé le terme avec Jeff Hammerbacher, qui dirigeait le département data chez Facebook. Nos équipes respectives grossissaient, et les ressources humaines nous disaient : "Il faut que vous mettiez un nom sur le métier de vos gars !" Alors, on s'est dit : analyste, ça fait trop Wall Street ; statisticien, ça agace les économistes ; chercheur scientifique, ça fait trop académique. Pourquoi pas "data scientist" ?

1. http://tempsreel.nouvelobs.com/tech/20170411.OBS7885/dj-patil-ex-sorcier-des-chiffres-d-obama-j-ai-invente-le-concept-de-data-scientist.html

La science des données (en anglais data science)

Une nouvelle discipline qui s'appuie sur des outils mathématiques, de statistiques, d'informatique et de visualisation des données. Elle est en plein développement, dans le monde universitaire ainsi que dans le secteur privé et le secteur public (due aux moyens de calcul et aux volumes de données).

- 1930-1970 (Octets) : Statistiques (modèle linéaire Gaussien)
  un test, une décision, donc une réponse
- 1970s (Ko) : Les premiers outils informatiques pour échapper à l'impérialisme du modèle linéaire (ex. Analyse des données en France, Exploratory Data Analysis (EDA) aux États-Unis)
  L'objectif est de décrire ou explorer, prétendument sans modèle, des données déjà plus volumineuses.
- 1980s (Mo) : Intelligence Artificielle (apprentissage des réseaux de neurones)
  La Statistique développe des modèles non-paramétriques ou fonctionnels.
- 1990s (Go) : Data Mining et Premier changement de paradigme
  aide à la décision, logiciels de fouille de données
- 2000s (To) : Deuxième changement de paradigme
  Apprentissage Statistique
- 2010s (Po) : Troisième changement de paradigme
  apprentissage non supervisées ou supervisées, optimisation (applications industrielles, e-commerce, géo-localisation)

Environnement logiciel

- Logiciels de fouille de données (data maining) :
  - Logiciels incluant en plus des modèles linéaires classiques, les différents algorithmes d'apprentissage au fur et à mesure de leur apparition.
  - WEKA, Clementine de SPSS, Enterprise Miner de SAS, Insightfull Miner de SPLUS, SPAD, Statistica Data Miner, Statsoft, ...
- Langages R et Python :
  - **R** : toute méthode d'apprentissage est implémentée en R sous la forme d'une librairie (package) librement accessible.
  - **Python** : a été développé pour le traitement et l'analyse des signaux, des images et des séries temporelles. Il permet de paralléliser facilement la préparation de grosses données sans les charger en mémoire avant de passer à la phase d'exploration puis de modélisation qui est elle toujours traitée en chargeant les données en mémoire.

  - R est préféré pour modéliser et interpréter des modèles statistiques tandis que Python est préféré pour des modélisations efficaces à seule fin prédictive au détriment de l'interprétation.

Langage R : installation

- **RStudio**
  https ://rstudio.com/products/rstudio/download/
- **RStudio via Anacanda**
  https ://www.anaconda.com/products/individual
  1. Anaconda Installer : choisissez la version graphique correspondante à votre système d'exploitation à installer
  2. Lancer Anacanda-Navigator
  3. Installer en suite RStudio
  4. Ensuite lancer RStudio ou bien Jupyter notebook
- **R en ligne**
  https ://cocalc.com/doc/jupyter-notebook.html
  1. Cliquer sur "Run Jupiter Now"
  2. Clique en suite sur : Welcome to CoCalc !
  3. Créer un nouveau document en cliquant sur New (create new file with extension Jupiter Notebook .ipynb)
  4. Sélectionner Kernel (Select a Kernel) : choisissez R

Á faire

- Tutoriels d'initiation à R.
  https ://github.com/wikistat/Intro-R