

Technologies de l'IA

Introduction

Philippe Besse & Brendan Guillouet

Université de Toulouse
INSA – Dpt GMM
Institut de Mathématiques – ESP
UMR CNRS 5219

IA & grosses données

Succès de l'IA : rencontre de

- Données massives
- Puissance de calcul
- Algorithme d'apprentissage statistique

Définition des *Grosses Données*

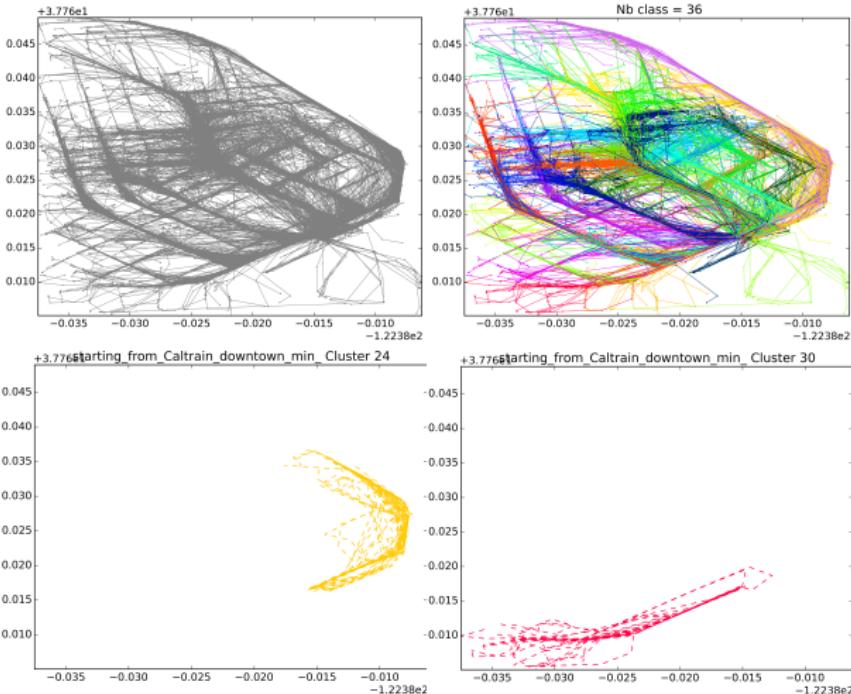
- **Volume** : croissance exponentielle
- **Variété** : signaux, images, graphes
- **Vélocité** : décision séquentielle, optimisation stochastique
- **Validité, Valorisation** —> Prévision —> Apprentissage

Introduction

Technologies des données massives
Fonctionnement des ateliers

Définitions, objectifs

Science des Données massives



Variété : Classification de trajectoires GPS

Confusion de domaines très variés

- E-commerce : recommandations et Réseaux sociaux
- Publique : administrations, santé et (*open data*)
- Recherche Météo, Biologie, Astronomie...
- Industrie : défaillance, fraudes, maintenance...

Objectifs de ces ateliers

- Aborder les technologies récentes de l'IA
- Traiter des cas d'usage réalistes

Cinq ateliers

- ➊ Compléments Python, introduction à **Spark** pour données distribuées
- ➋ Analyse d'images (MNIST, cats vs. dogs) avec Keras & Tensor FLow :
- ➌ Calcul intensif avec la *Google cloud platform*
- ➍ Système de recommandation (base movieLens) avec Spark/MLlib
- ➎ **NLP** (base Cdiscount) traitement du langage naturel avec Spark, Python, Gensim

Évaluation

- Classement au **Défi IA 2019**
- Présentation orale



Réellement massives ?

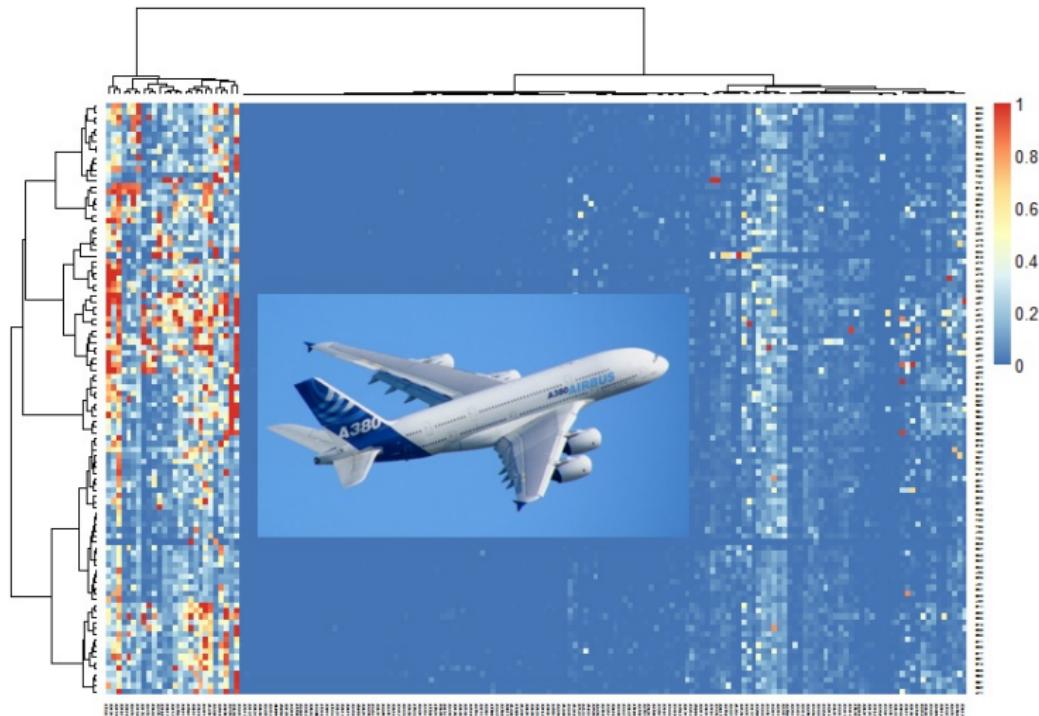
- **Seuils** technologiques (RAM, Disque)
- **Préparation** (*munging*) des données (Python)
- Données **distribuées** :
- **Hadoop, MapReduce, scalability**



Ferme de données

Objectif

- Apprentissage sur données massives
- Question : Quelles (meilleures) technologies utiliser ?
- Matériel : Poste personnel, GPU, serveur(s), *Cloud*
- Données distribuées (*Hadoop*) ?
- Point de vue du "statisticien"
 - Prototype puis passage à l'échelle du même code
 - Cas d'usage : MovieLens, Cdiscount, MNIST, Cats/Dogs
 - Algorithmes : *munging*, NMF, Logit, RF, SVM, *Gradient boosting*, *Deep Learning*
 - environnements : R, Python, Spark, XGBoost, TensorFlow, Keras
- Comparaisons de difficiles à impossibles
- Scripts dans <http://github.com/wikistat>



Albus : Analyse des messages d'incidents en vol (700 000 en 6 mois)

Nouvelle (?) Science des Données

- 1995 *Data Mining* : GRC & suites logicielles
- 2010 *Data Science* : Publicité en ligne, *cloud computing*
 - Données **préalables** (fouille), dimensions (omiques) $p \gg n$,
 - Pas de **nouvelles** méthodes, seules celles **échelonnables**
 - **Data** pas toujours *Grosses* mais *datification* du quotidien
 - **Éthique** et virtualisation / transparence de la décision
- "Science" des données : **nouveau** terme d'erreur
 - Erreur **d'optimisation** + Compromis biais / variance
 - **Optimisation** stochastique (Robbins Monro) ;
non différentiable (parcimonie)
 - **Parallélisation** et/ou distribution des calculs

Nouveau modèle économique

- Marges réduites sur matériels (IBM)
- Logiciels sous licence GNU, MIT, Apache... (Microsoft, SAS)
- Vendre du *service* :
 - Enthought ([Canopy](#)), Continuum analytics ([Anaconda](#)),
 - Horton Works, Cloudera ([Hadoop](#), [Spark...](#))
 - Databricks ([Spark](#)), Oxdata ([H2O](#))
 - Revolution Analytics ([RHadoop](#)) – Microsoft
- Nouveau marché du *cloud computing*
 - Platform [aaS](#), Software [aaS](#), Service [aaS...](#)
 - Amazon Web Service
 - Microsoft Azure, Google Cloud Computing
 - IBM Analytics, SAS Advanced Analytics...
- Développement *industriel* vs. Recherche *académique*

Introduction

Technologies des données massives

Fonctionnement des ateliers

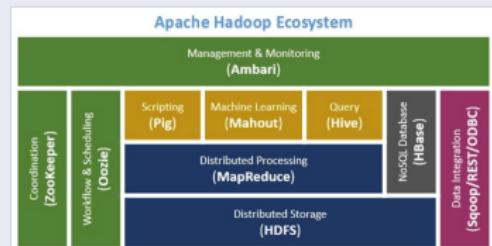
Écosystème Hadoop, MapReduce
MapReduce pour les nuls
Spark

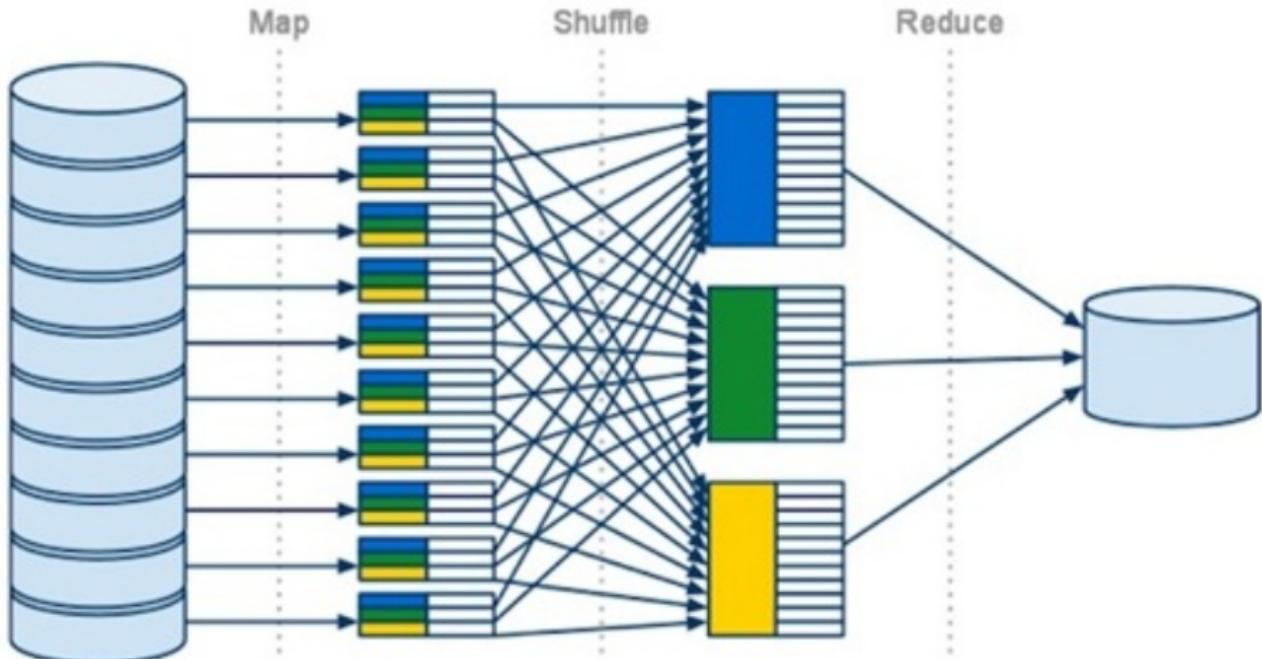


Écosystème des données massives et de l'IA (Turck, 2018)

Hadoop, MapReduce

- Environnement : *Google* puis **Apache** (2009)
- ***Hadoop Distributed File System*** (HDFS)
- Données hétérogènes **distribuées**
- Tolérance aux pannes matérielles
- Scalabilité (multiplier les nœuds)
- **Parallélisation** : *Map Reduce*
- Communication par (**clef, valeur**)
- Déplacer les algorithmes, pas les données
- Données *immuables*
- Lecture unique ou *streaming*

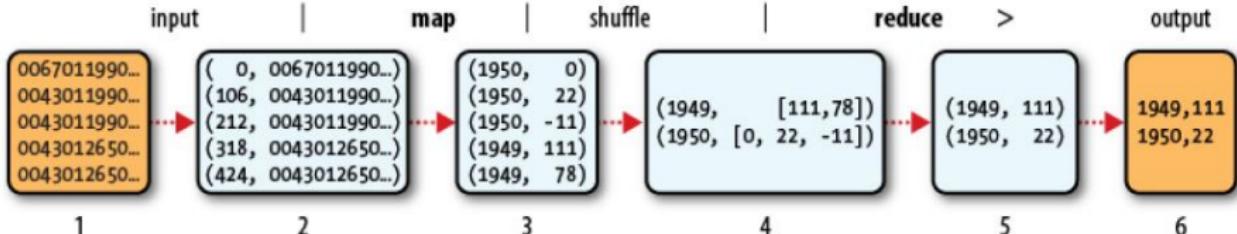




Hadoop Distributed File System (HDFS) & MapReduce

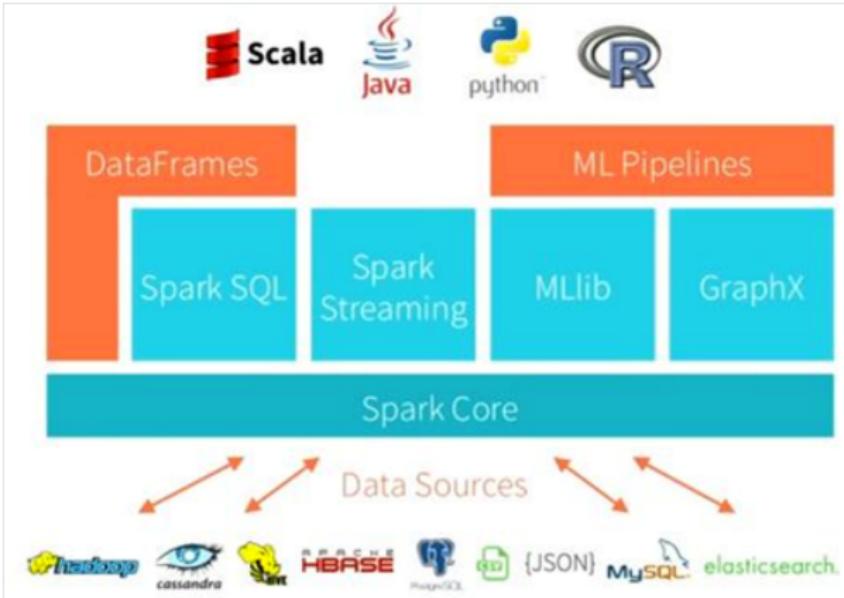
MapReduce

- ① Input \Rightarrow list (k1, v1)
- ② Map () \Rightarrow list (k2, v2)
- ③ *Shuffle, Combine* \Rightarrow (k2, list(v2))
 - Implicite
 - Clefs identiques vers même nœud réducteur
- ④ Reduce () \Rightarrow list (k3, v3)



Classification par centres mobiles ($\approx k\text{-means}$)

- Définition d'une **distance** euclidienne
- Algorithme de **Forgy** (1965)
 - Initialisation des k centres
 - Itération des étapes *MapReduce*
 - **Map** : Affectation de chaque individu (**valeur**) au centre (**clef**) le plus proche
 - **Reduce** : Calcul des **centres** des individus de même **clef**
 - Mise à jour des centres
- **Problème** : accès disques à chaque itération
- **Solution actuelle de Spark** : *Resilient Distributed Dataset*,
(Zaharia *et al.*, 2012)



La technologie **Spark** et son écosystème

Librairie MLlib

- Spark 2.2
- MLlib (Resilient Distributed Dataset) : *k-means*, SVD, NMF (ALS), Régression linéaire et logistique (l_1 et l_2), SVM linéaires, Classifieur Bayésien Naïf, Arbre, Forêt Aléatoire, Boosting
- Évolution de SparkML : *DataFrame*, *pipeline*
- Peu de méthodes mais passage à l'échelle "Volume"

Progression "pédagogique"

- Statistique de **petites** données avec **R**
- Traitement et stat de données plus **grosses** : **Python**
- Traitement et stat de données **distribuées** : **PySpark**
- Autres **algorithmes** avec Python : *XGBoost, deep learning*

Matériels

- **Ordinateurs personnels** avec R, Python 3.6 (Anaconda)
- **GMM** salles TP avec 16 cartes GPU ; R, Python 3.6, Spark 2 (Hadoop virtuel)
- **Cloud** : *Google platform*

Rappel de l'objectif principal

Apprendre à s'auto-former à des technologies
en perpétuelle (r)évolution

