Sistemas Inteligentes para la Gestión de la Empresa

2016 - 2017

- Tema 1. Introducción a la Inteligencia de Negocio
- Tema 2. Depuración y Calidad de Datos
- Tema 3. Análisis Predictivo para la Empresa
- Tema 4. Modelos avanzados de Analítica de Empresa
- Tema 5. Análisis de Transacciones y Mercados
- Tema 6. Big Data

Tecnologías para el procesamiento y analítica de datos en big data

Francisco Herrera

Research Group on Soft Computing and Information Intelligent Systems (SCI²S)

http://sci2s.ugr.es

Dept. of Computer Science and A.I.

University of Granada, Spain

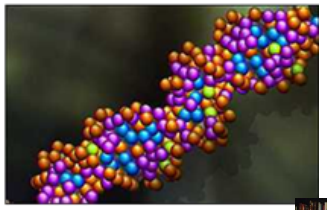
Email: herrera@decsai.ugr.es

DECSAI
Universidad de Granada

Big Data

Nuestro mundo gira en torno a los datos

- Ciencia
  - Bases de datos de astronomía, genómica, datos medio-ambientales, datos de transporte, …

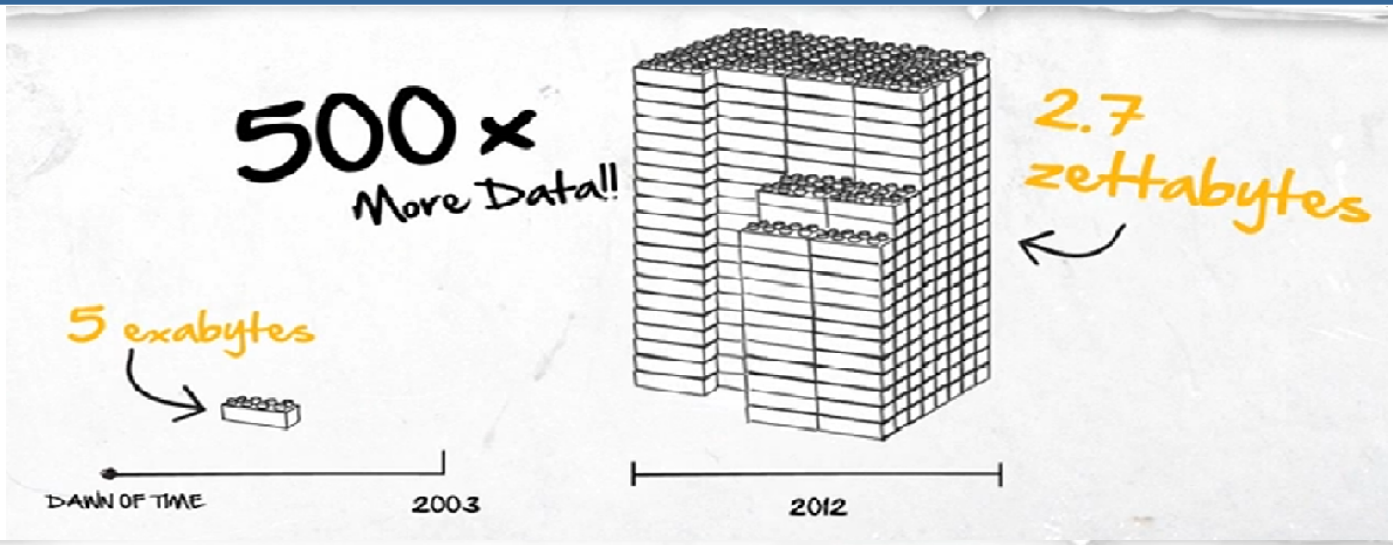


- Ciencias Sociales y Humanidades
  - Libros escaneados, documentos históricos, datos sociales, …
- Negocio y Comercio
  - Ventas de corporaciones, transacciones de mercados, censos, tráfico de aerolíneas, …

- Entretenimiento y Ocio
  - Imágenes en internet, películas, ficheros MP3, …

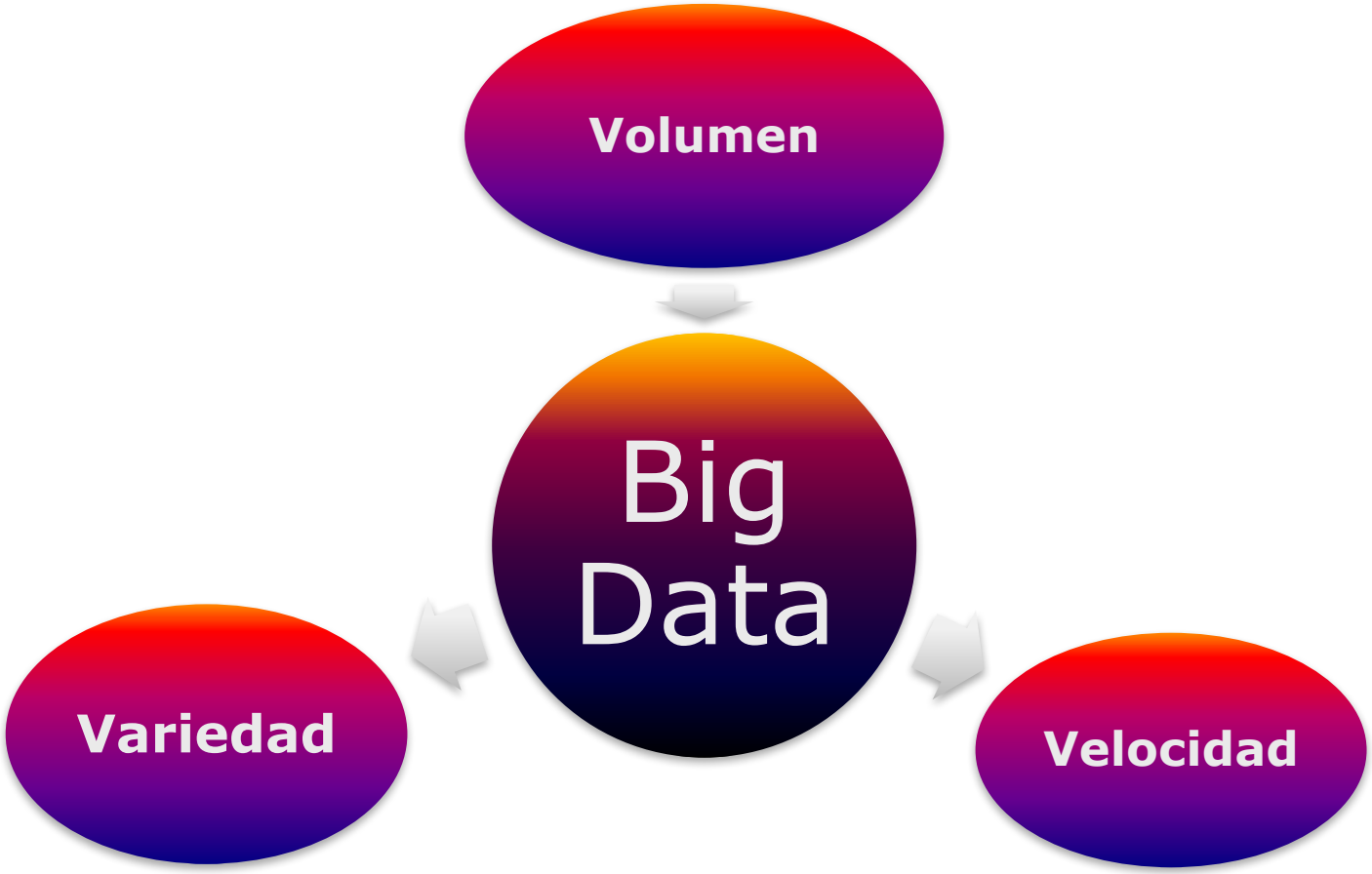
- Medicina
  - Datos de pacientes, datos de escaner, radiografías …
- Industria, Energía, …
  - Sensores, …

Big Data: La explosión de los datos



Big Data

Big Data en 3 V's



Doug Laney, Gartner 2001

Índice

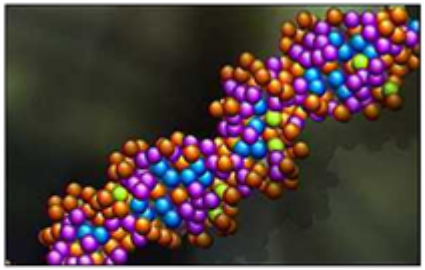
- Big Data. Big Data Science
- ¿Por qué Big Data? Google crea el Modelo de Programación MapReduce
- Tecnologías para Big Data: Ecosistema Hadoop (Hadoop, Spark, …)
- Big Data Analytics: Librerías para Analítica de Datos en Big Data.
- Casos de estudio: Random Forest, Clustering
- Algunas aplicaciones: Salud, Social Media, Identificación
- Imbalanced Big Data: Caso de estudio
- Comentarios Finales

Índice

- **Big Data. Big Data Science**
- ¿Por qué Big Data? Google crea el Modelo de Programación MapReduce
- Tecnologías para Big Data: Ecosistema Hadoop (Hadoop, Spark, …)
- Big Data Analytics: Librerías para Analítica de Datos en Big Data.
- Casos de estudio: Random Forest, Clustering
- Algunas aplicaciones: Salud, Social Media, Identificación
- Imbalanced Big Data: Caso de estudio
- Comentarios Finales

¿Qué es Big Data? 3 V's de Big Data

Ej. Genómica



- 25,000 genes in human genome
- 3 billion bases
- 3 Gigabytes of genetic data

Ej. Astronomía

- Astronomical sky surveys
- 120 Gigabytes/week
- 6.5 Terabytes/year

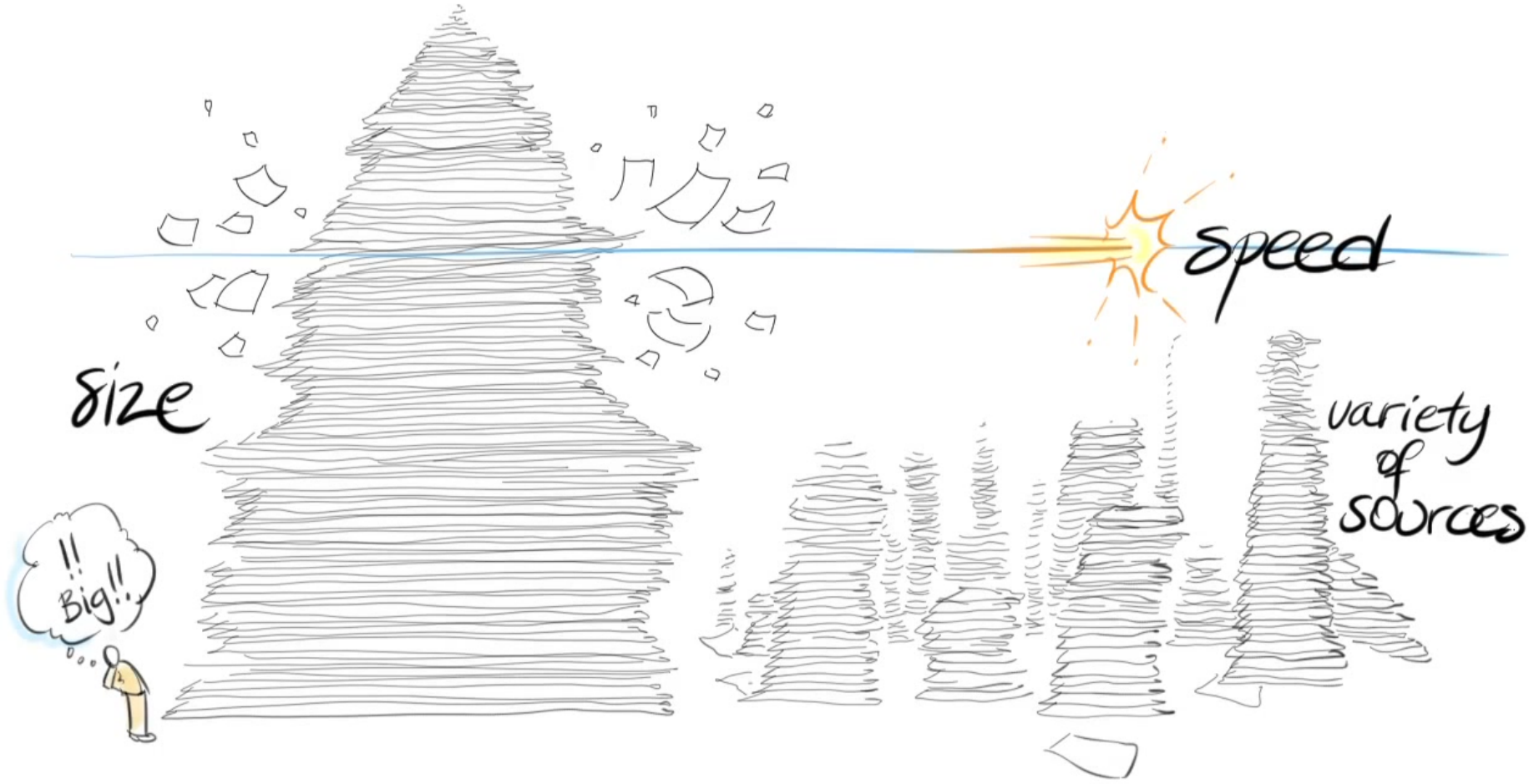
Ej. Transacciones de tarjetas de crédito

- 47.5 billion transactions in 2005 worldwide
- 115 Terabytes of data transmitted to VisaNet data processing center in 2004

¿Qué es Big Data? 3 V's de Big Data

Ej. E-Promociones: **Basadas en la posición actual e historial de compra ➔ envío de promociones en el momento de comercios cercanos a la posición**

¿Qué es Big Data? 3 V's de Big Data



¿Qué es Big Data? 3 V's de Big Data

Ej. Huella digital de pasajeros

Interior encarga un megacerebro capaz de localizar terroristas entre los pasajeros

¿Un proyecto de ciencia ficción? El Ministerio del Interior cree que no. Que es posible tener un megacerebro electrónico capaz de localizar —a través de cálculos estadísticos y de cruzar ingentes cantidades de datos en milisengundos—...

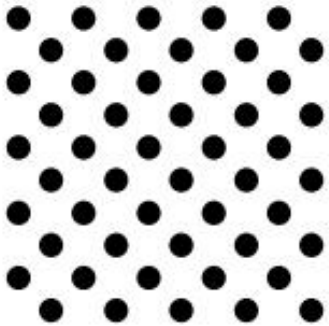
> *"identificación automática del perfil demográfico y sociológico del pasajero"*

http://article.wn.com/view/2015/04/04/Interior_encarga_un_megacerebro_capaz_de_localizar_terrorist_3/

¿Qué es Big Data? 3 V's de Big Data

Some Make it 4V's: Veracity

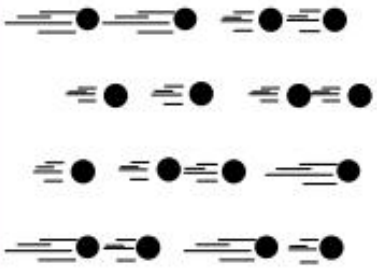
Volume



Data at Rest

Terabytes to exabytes of existing data to process

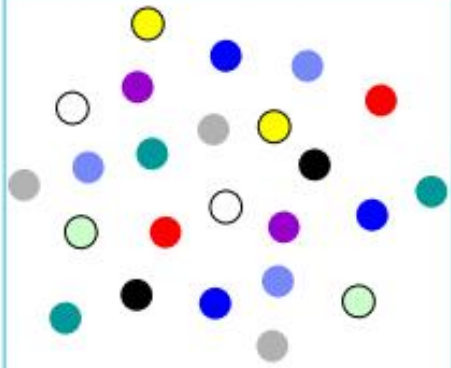
Velocity



Data in Motion

Streaming data, milliseconds to seconds to respond

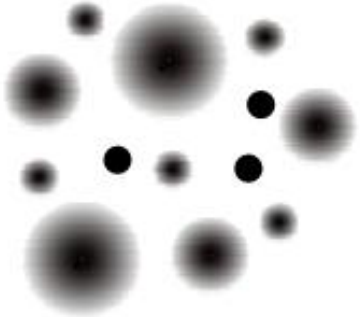
Variety



Data in Many Forms

Structured, unstructured, text, multimedia

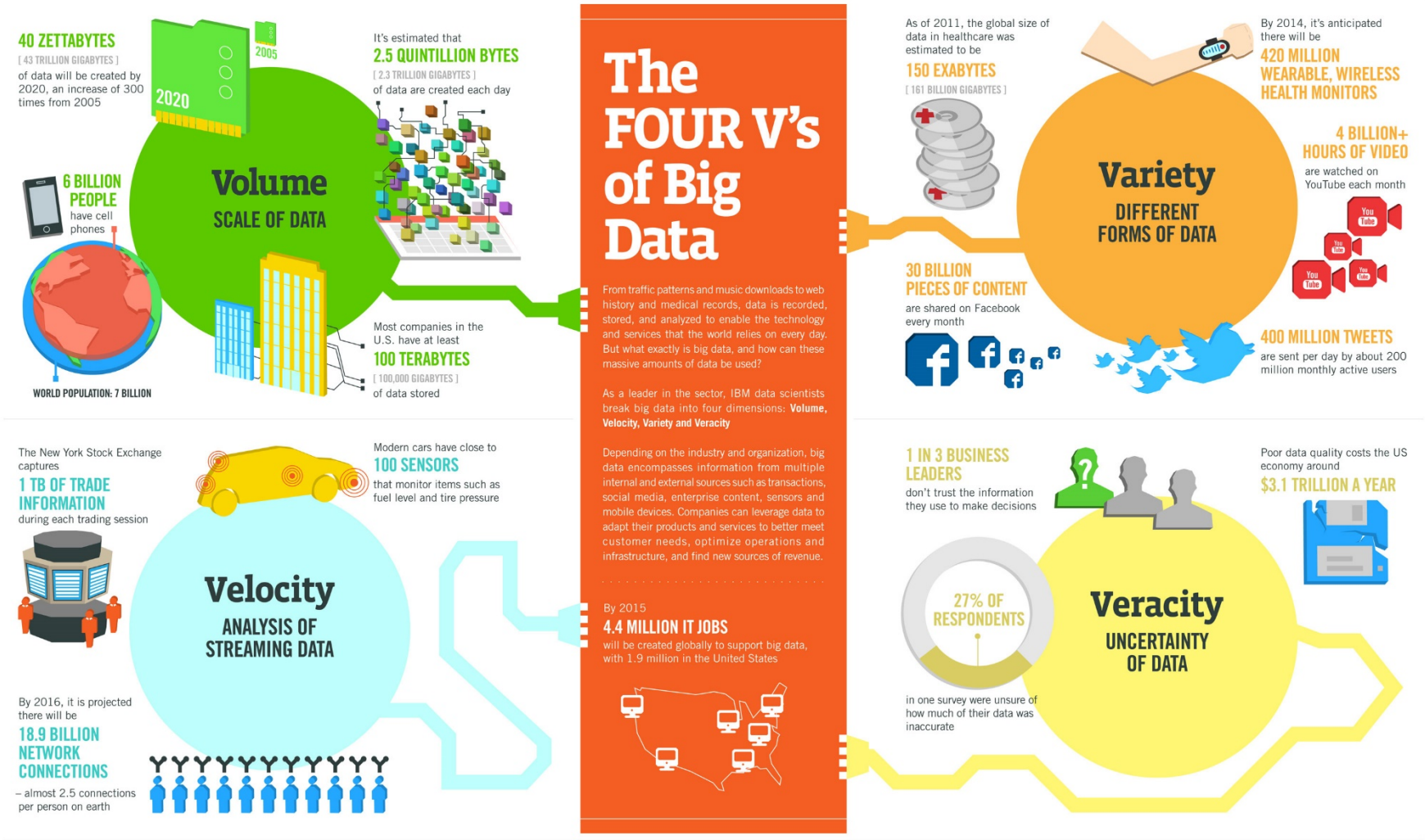
Veracity*



Data in Doubt

Uncertainty due to data inconsistency & incompleteness, ambiguities, latency, deception, model approximations

¿Qué es Big Data?



¿Qué es Big Data?

V --> Valor

Aproximaciones y tecnologías innovativas

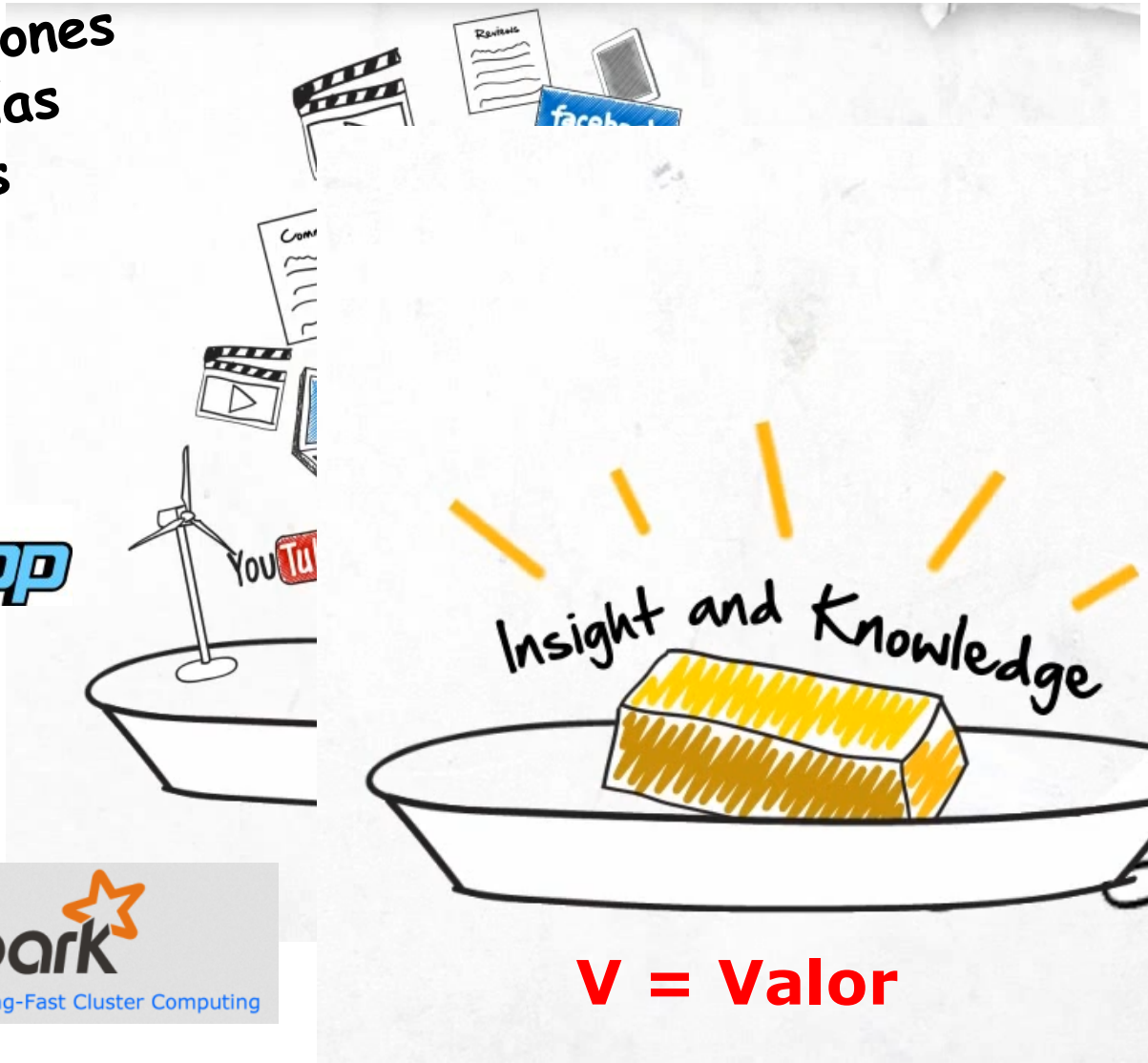
MapReduce

hadoop

mahout

APACHE GIRAPH

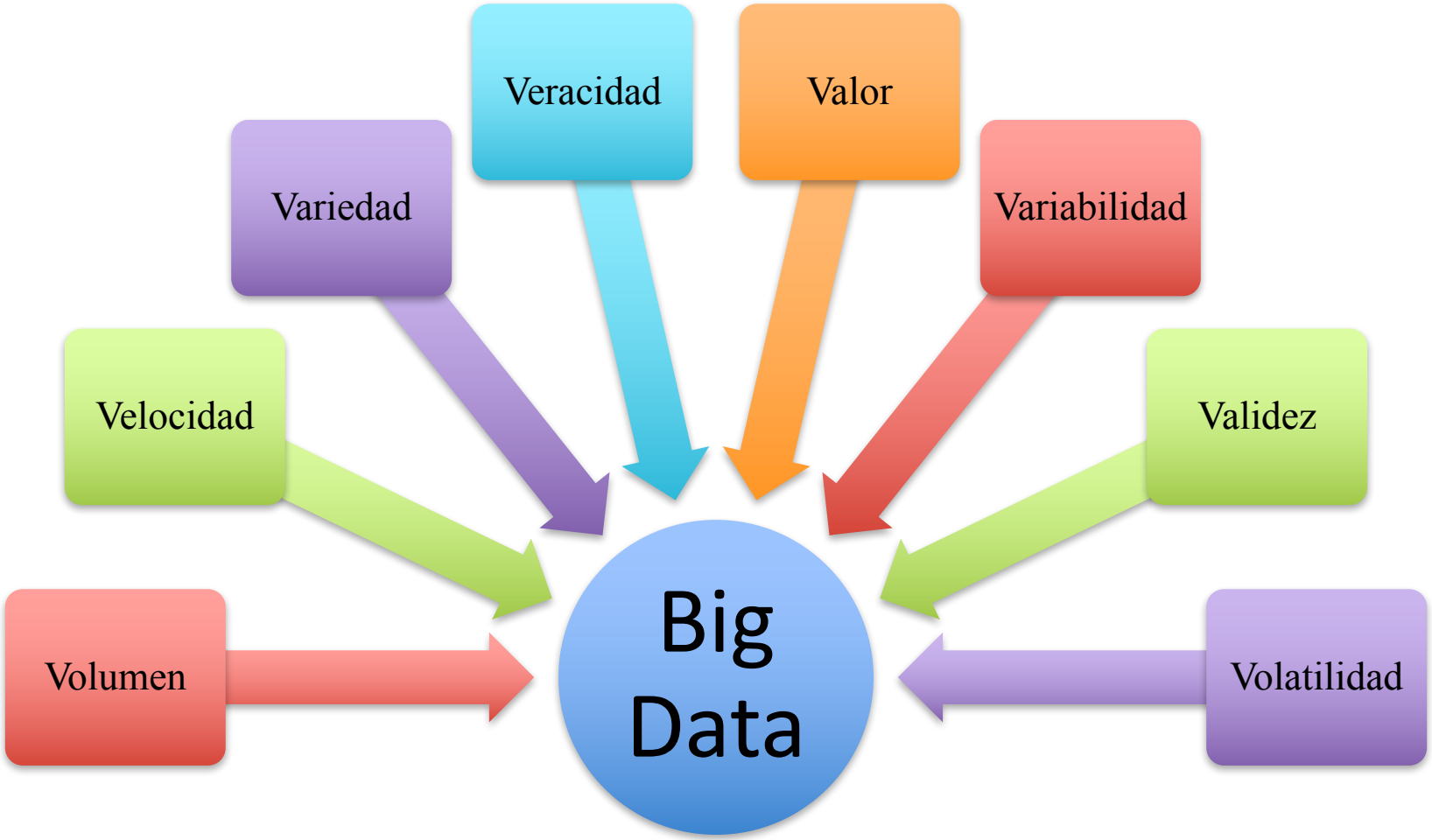
Spark
Lightning-Fast Cluster Computing



V = Valor

¿Qué es Big Data?

Las 8 V's de Big Data

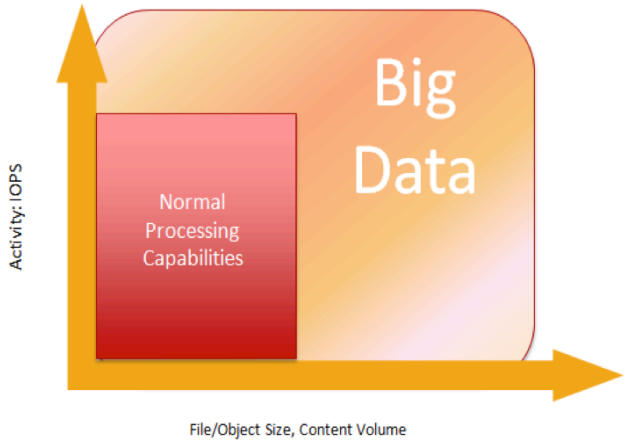


¿Qué es Big Data?

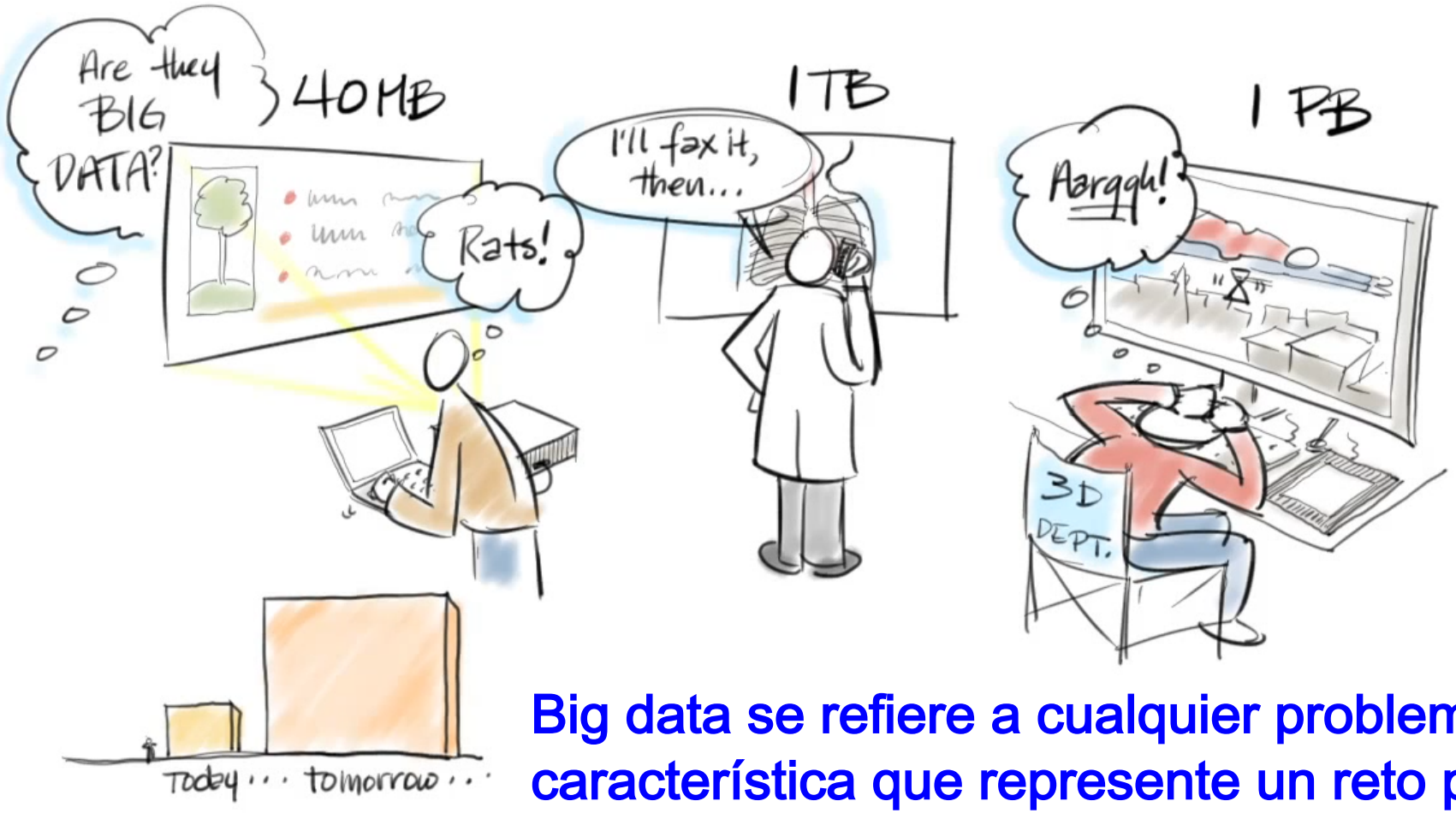
No hay una definición estándar

Big data es una colección de datos grande, complejos, **muy difícil de procesar a través de herramientas de gestión y procesamiento de datos tradicionales**

"***Big Data***" son datos cuyo volumen, diversidad y complejidad **requieren nueva arquitectura, técnicas, algoritmos y análisis** para gestionar y extraer valor y conocimiento oculto en ellos ...



¿Qué es Big Data?



Big data se refiere a cualquier problema o característica que represente un reto para ser procesado con aplicaciones tradicionales

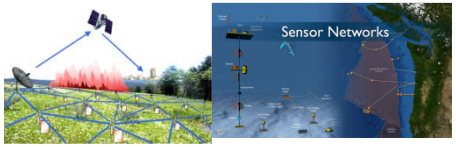
¿Qué es Big Data?

¿Quién genera Big Data?

Redes sociales y multimedia
(todos generamos datos)

Instrumentos científicos
(colección de toda clase de datos)

Dispositivos móviles
(seguimiento de objetos)



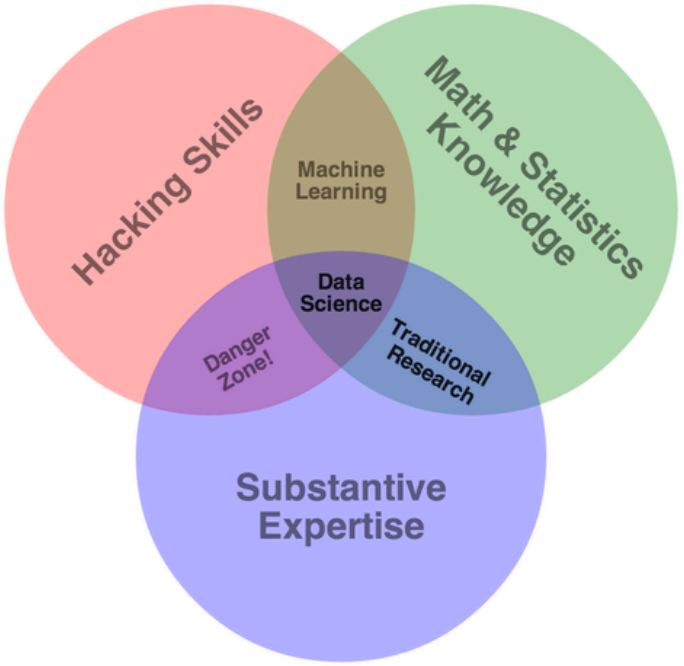
Redes de sensores
(se miden toda clase de datos)

El progreso y la innovación ya no se ven obstaculizados por la capacidad de recopilar datos, sino por la capacidad de gestionar, analizar, sintetizar, visualizar, y descubrir el conocimiento de los datos recopilados de manera oportuna y en una forma escalable

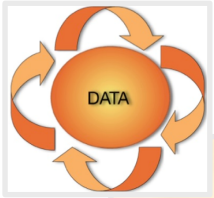
(Big) Data Science

Data Science combines the traditional scientific method with the ability to explore, learn and gain deep insight for (Big) Data

It is not just about finding patterns in data … it is mainly about explaining those patterns



Data Science Process



Data Preprocessing

- Clean
- Sample
- Aggregate
- Imperfect data: missing, noise, …
- Reduce dim.
- …

> 70% time!

Data Processing

- Explore data
- Represent data
- Link data
- Learn from data
- Deliver insight
- …

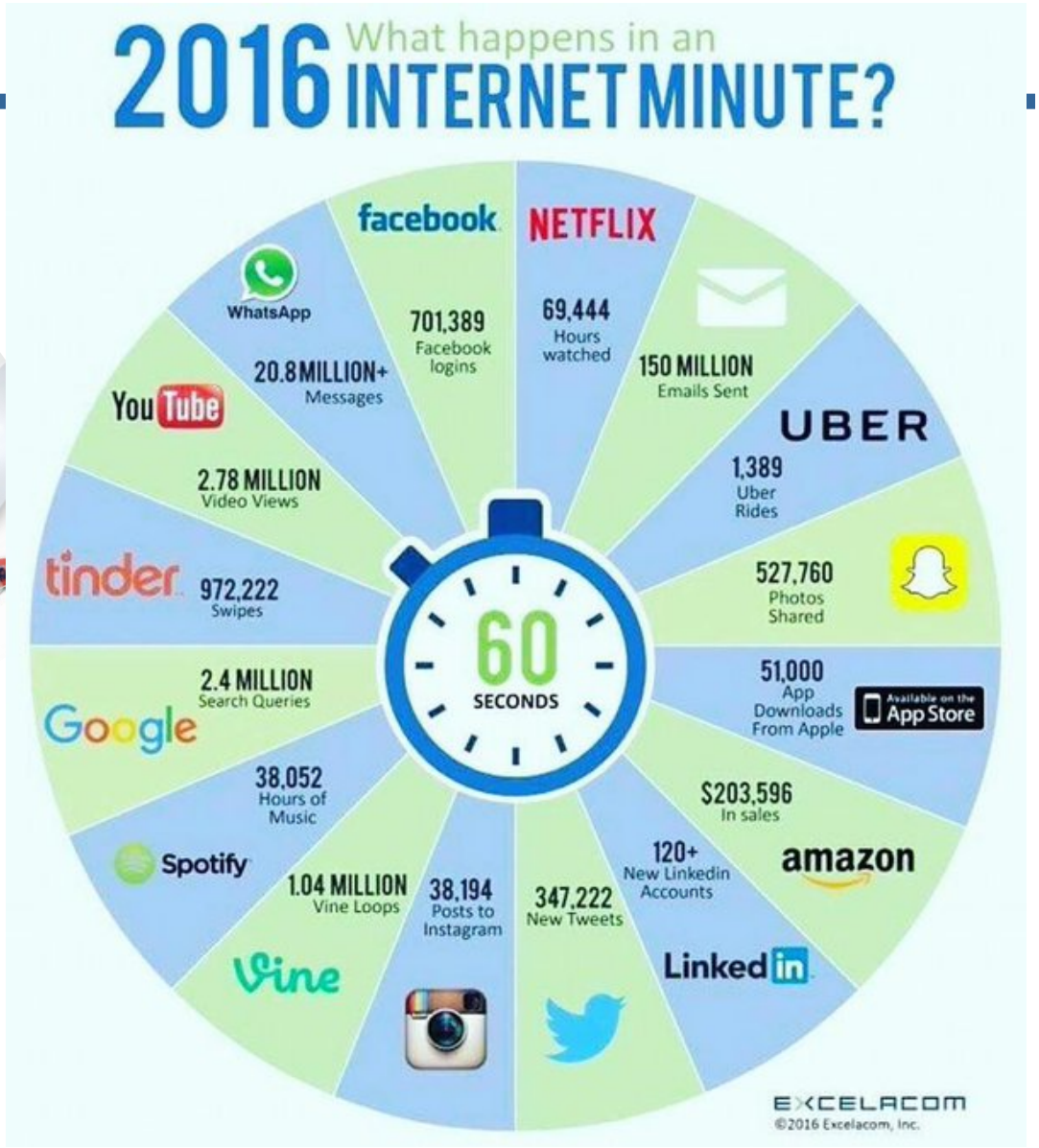
Data Analytics

- Clustering
- Classification
- Regression
- Network analysis
- Visual analytics
- Association
- …

Índice

- **Big Data. Big Data Science**
- **¿Por qué Big Data? Google crea el Modelo de Programación MapReduce**
- Tecnologías para Big Data: Ecosistema Hadoop (Hadoop, Spark, …)
- Big Data Analytics: Librerías para Analítica de Datos en Big Data.
- Casos de estudio: Random Forest, Clustering
- Algunas aplicaciones: Salud, Social Media, Identificación
- Big Data en el grupo de investigación SCI^{2}S
- Comentarios Finales

¿Por qué Big Data?



http://www.dailyinfographic.com/what-happens-in-an-internet-minute-infographic

¿Por qué Big Data?

- **Problema:** Escalabilidad de grandes cantidades de datos
- **Ejemplo:**
  - Exploración 100 TB en 1 nodo @ 50 MB/sec = 23 días
  - Exploración en un clúster de 1000 nodos = 33 minutos
- **Solución → Divide-Y-Vencerás**

Una sola máquina no puede gestionar grandes volúmenes de datos de manera eficiente

¿Por qué Big Data?

- **Problema:** Escalabilidad de grandes cantidades de datos
- **Ejemplo:**
  - Exploración 100 TB en 1 nodo @ 50 MB/sec = 23 días
  - Exploración en un clúster de 1000 nodos = 33 minutos
- **Solución → Divide-Y-Vencerás**

¿Cómo podemos procesar

1000 TB or 10000 TB?

¿Por qué Big Data?

- Escalabilidad de grandes cantidades de datos
  - Exploración 100 TB en 1 nodo @ 50 MB/sec = 23 días
  - Exploración en un clúster de 1000 nodos = 33 minutos

Solución → Divide-Y-Vencerás

¿Qué ocurre cuando el tamaño de los datos aumenta y los requerimientos de tiempo se mantiene?

Hace unos años: Había que aumentar los recursos de hardware (número de nodos). Esto tiene limitaciones de espacio, costes, …

Google 2004: Paradigma **MapReduce**

MapReduce

- Escalabilidad de grandes cantidades de datos
  - Exploración 100 TB en 1 nodo @ 50 MB/sec = 23 días
  - Exploración en un clúster de 1000 nodos = 33 minutos

Solución → Divide-Y-Vencerás

MapReduce

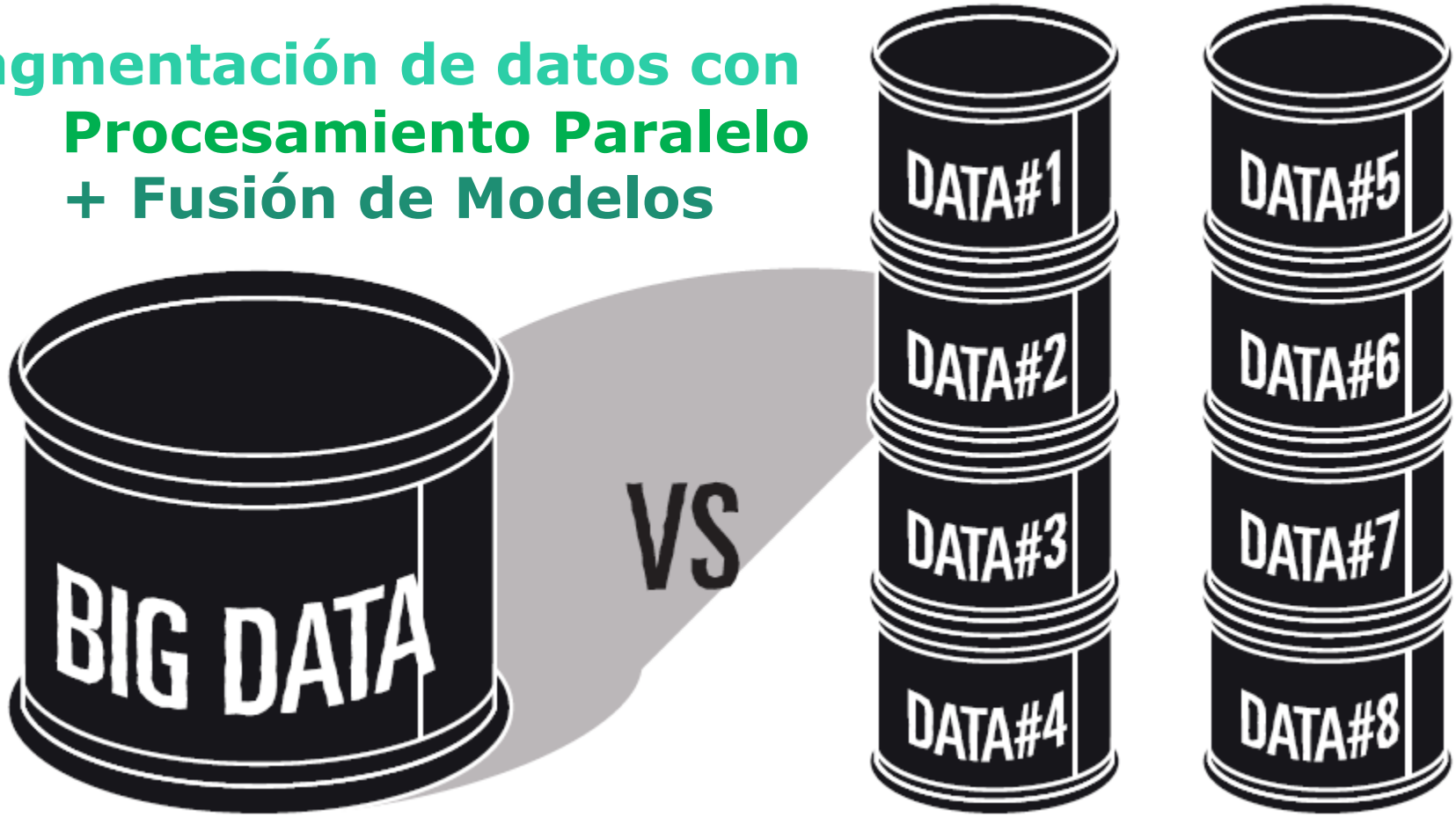
- Modelo de programación de datos paralela
- Concepto simple, elegante, extensible para múltiples aplicaciones

- **Creado por Google (2004)**
  - Procesa 20 PB de datos por día (2004)
- **Popularizado por el proyecto de codigo abierto Hadoop**
  - Usado por Yahoo!, Facebook, Amazon, …

MapReduce

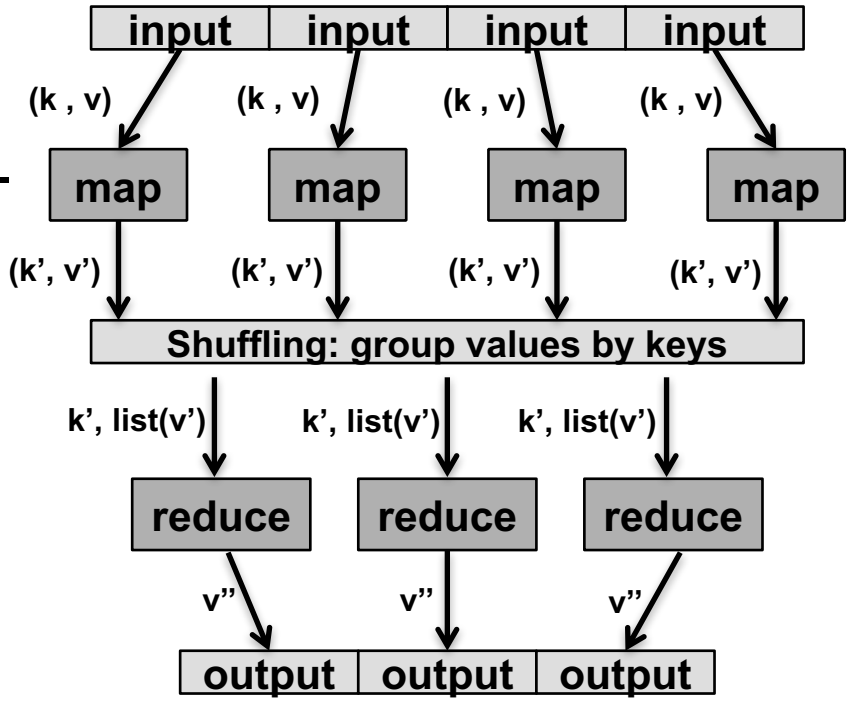
MapReduce es la aproximación más popular para Big Data

Fragmentación de datos con
Procesamiento Paralelo
+ Fusión de Modelos



MapReduce

- MapReduce es el entorno más popular para Big Data
- Basado en la estructura Valor-llave.
- Dos operaciones:
  1. **Función Map : Procesa bloques de información**
  2. **Función Reduce function: Fusiona los resultados previous de acuerdo a su llave.**
- + Una etapa intermedia de agrupamiento por llave **(Shuffling)**



map (k, v) → list (k', v')
reduce (k', list(v')) → v"

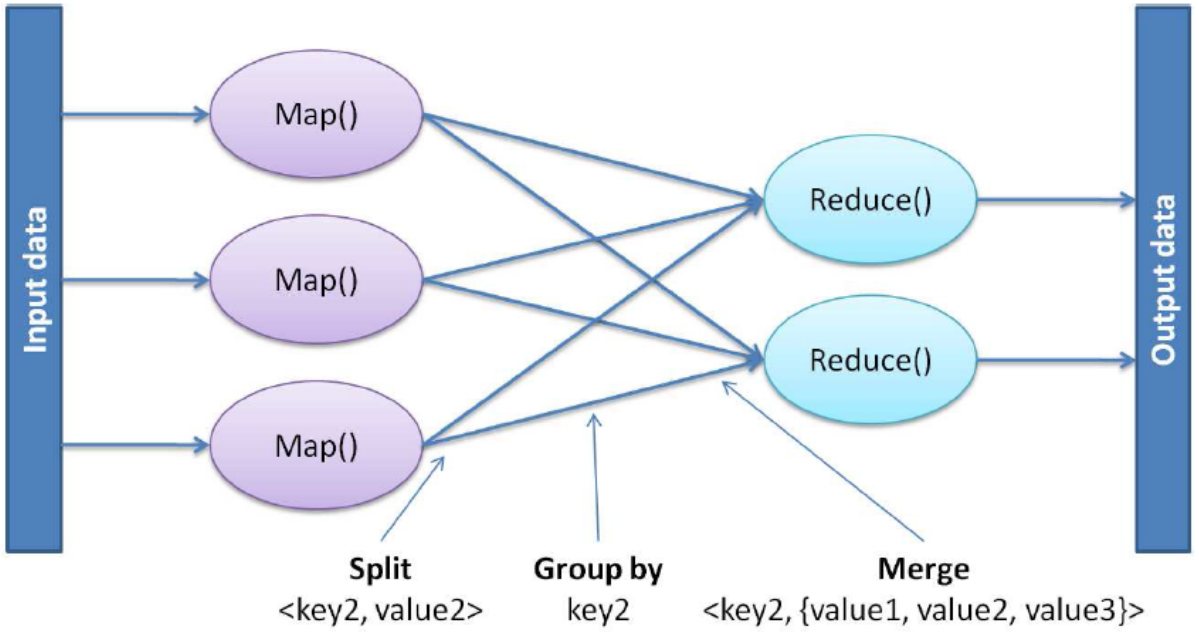
J. Dean, S. Ghemawat, MapReduce: Simplified data processing on large clusters, Communications of the ACM 51 (1) (2008) 107-113.

MapReduce

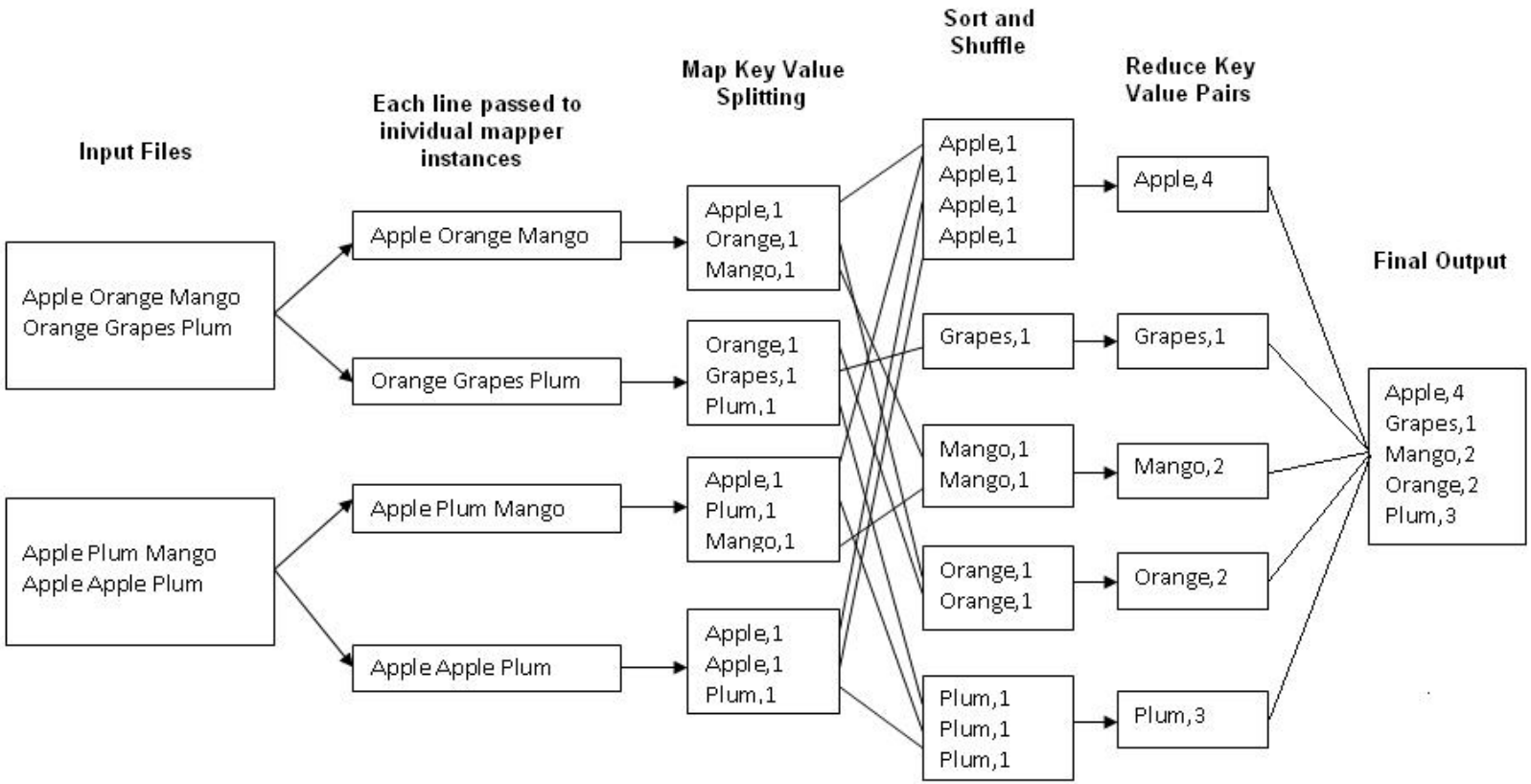
Flujo de datos MapReduce

map (k,v) → list(k',v')

reduce (k',list(v')) → v''



MapReduce

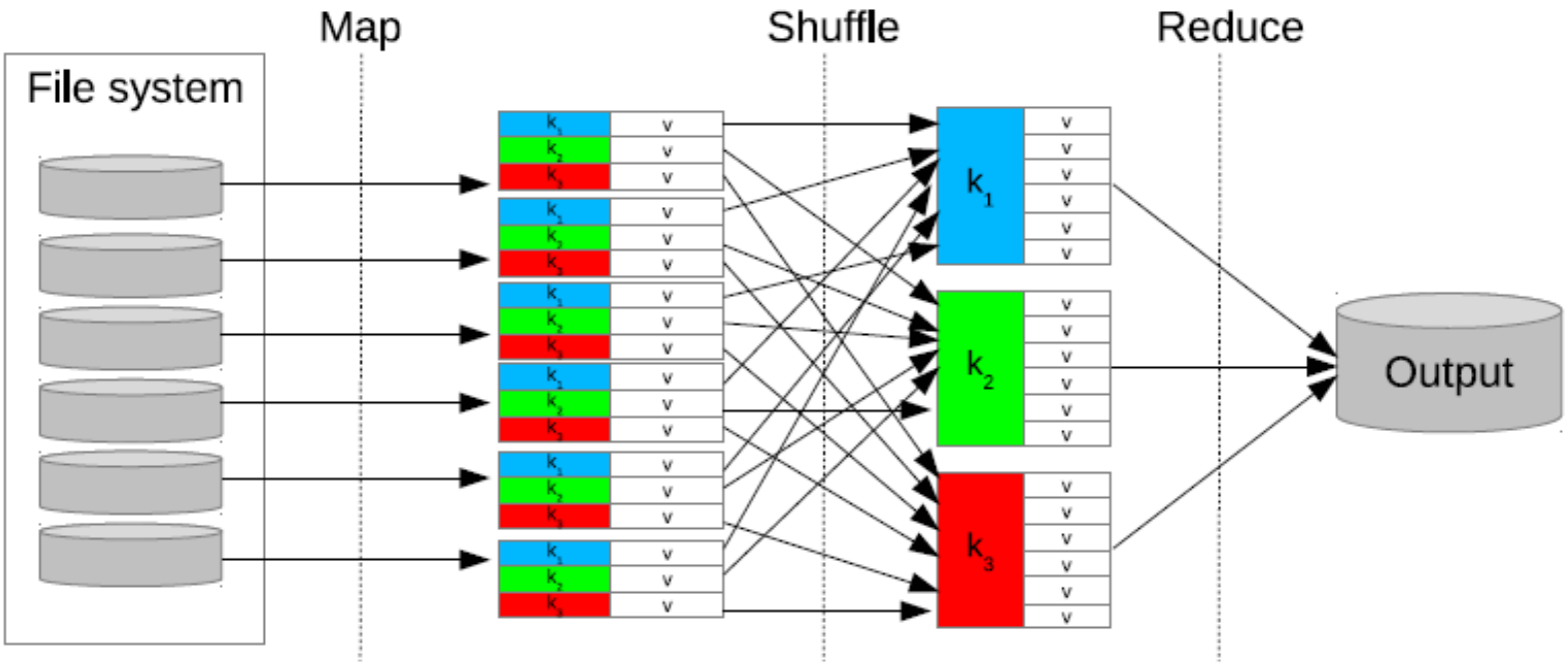

Input Files
Each line passed to individual mapper instances
Map Key Value Splitting
Sort and Shuffle
Reduce Key Value Pairs
Final Output
Apple Orange Mango Orange Grapes Plum
Apple Plum Mango Apple Apple Plum
Apple Orange Mango
Orange Grapes Plum
Apple Plum Mango
Apple Apple Plum
Apple,1 Orange,1 Mango,1
Orange,1 Grapes,1 Plum,1
Apple,1 Plum,1 Mango,1
Apple,1 Apple,1 Plum,1
Apple,1 Apple,1 Apple,1 Apple,1
Grapes,1
Mango,1 Mango,1
Orange,1 Orange,1
Plum,1 Plum,1 Plum,1
Apple,4
Grapes,1
Mango,2
Orange,2
Plum,3
Apple,4 Grapes,1 Mango,2 Orange,2 Plum,3

Características

- **Paralelización automática:**
  - Dependiendo del tamaño de ENTRADA DE DATOS➔ se crean mutiples tareas MAP
  - Dependiendo del número de intermedio <clave, valor> particiones ➔ se pueden crear varias tareas REDUCE
- **Escalabilidad:**
  - Funciona sobre cualquier cluster de nodos/procesadores
  - Puede trabajar desde 2 a 10,000 máquinas
- **Transparencia programación**
  - Manejo de los fallos de la máquina
  - Gestión de comunicación entre máquina

MapReduce

Flujo de datos en MapReduce, transparente para el programador



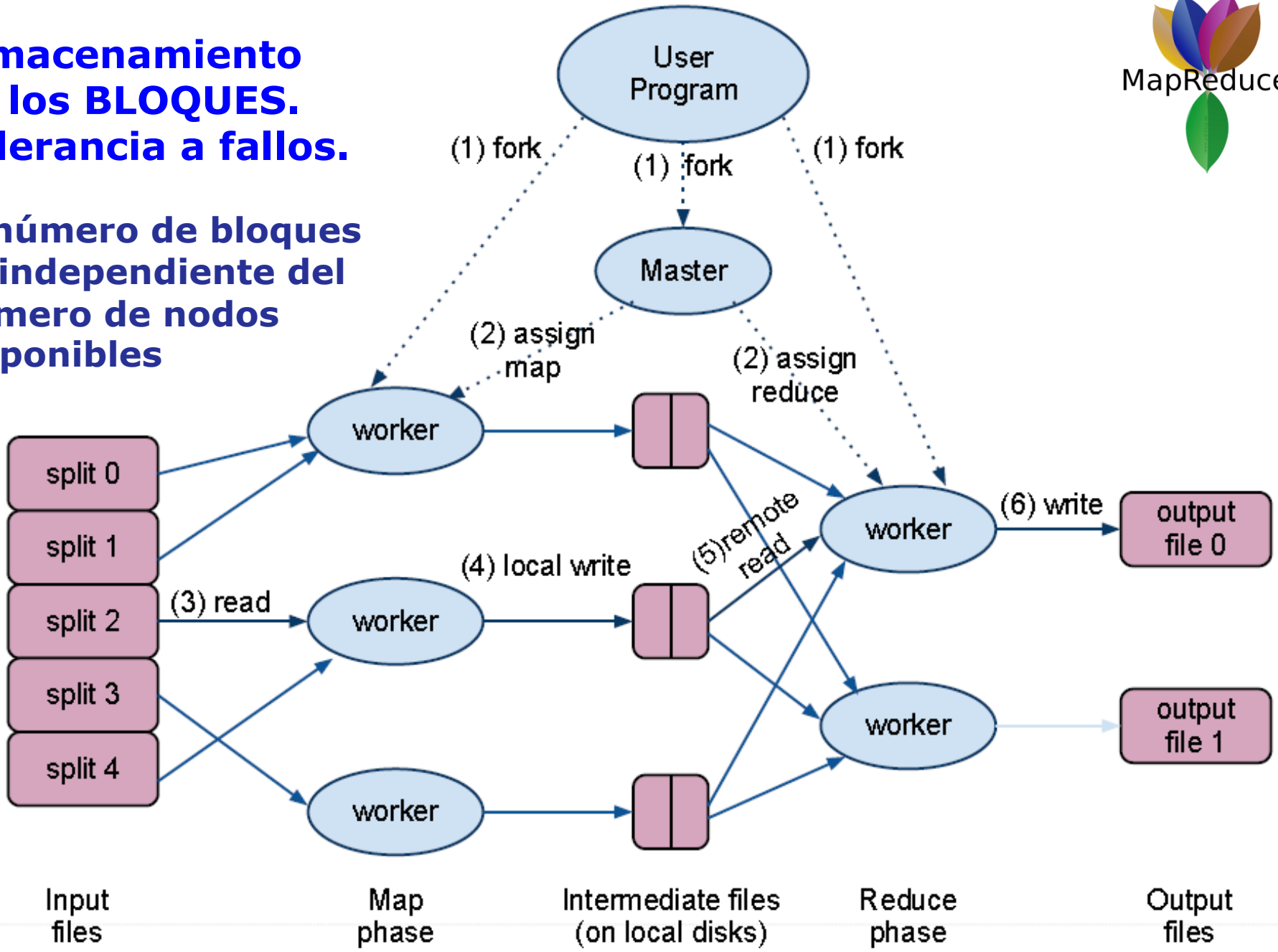
Ficheros de entrada Particionamiento en bloques

Ficheros Intermedios

Ficheros salida

Almacenamiento de los BLOQUES. Tolerancia a fallos.

El número de bloques es independiente del número de nodos disponibles

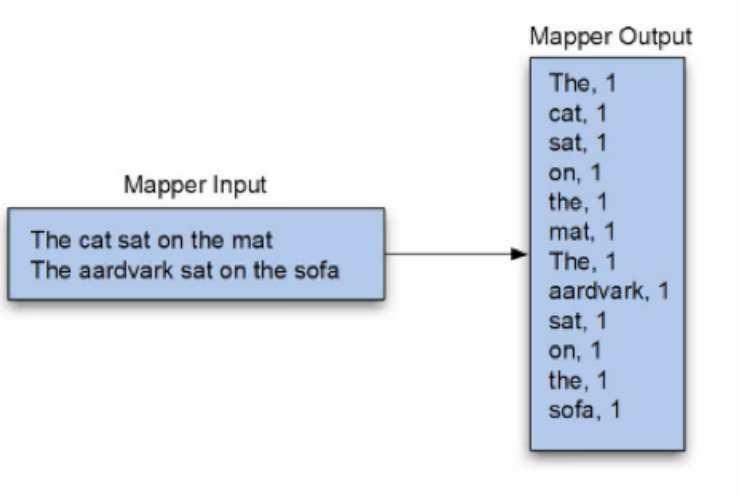


Almacenamiento con copias, normalmente r=3

MapReduce

Aspectos a analizar:

Proceso map: Puede crear conjuntos de datos muy pequeños, lo cual puede ser un inconveniente para algunos problemas.

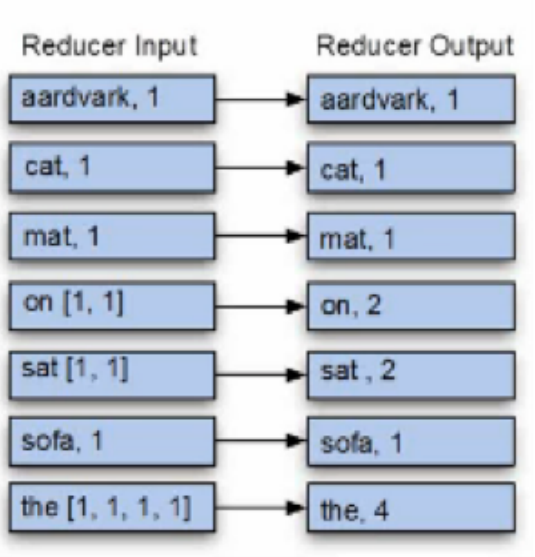


Ejemplo: Problemas de Clasificación. Nos podríamos encontrar con el problema de falta de densidad de datos y de clases con muy pocos datos (clasificación no balanceada).

MapReduce

Aspectos a analizar:

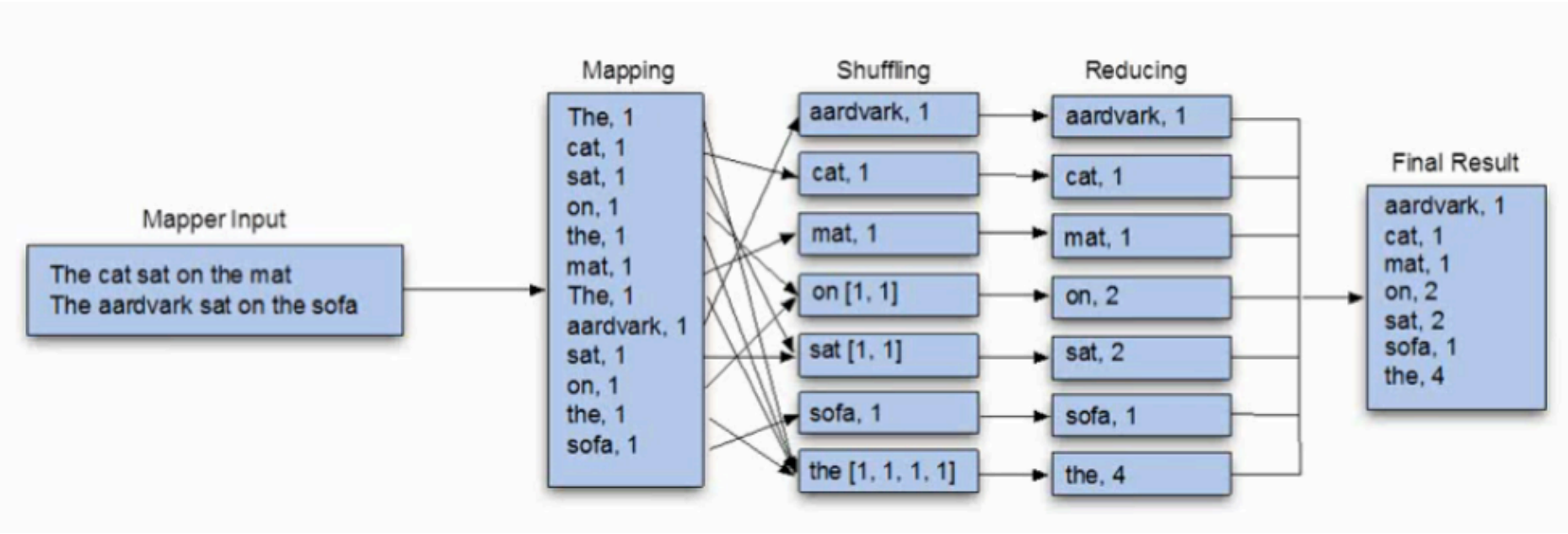
Proceso reduce: Debe combinar las soluciones de todos los modelos/procesos intermedios



Esta es la fase más creativa porque hay que conseguir crear un modelo global de calidad asociado al problema que se desea resolver a partir de los modelos intermedios.

MapReduce

Una imagen completa del proceso MapReduce



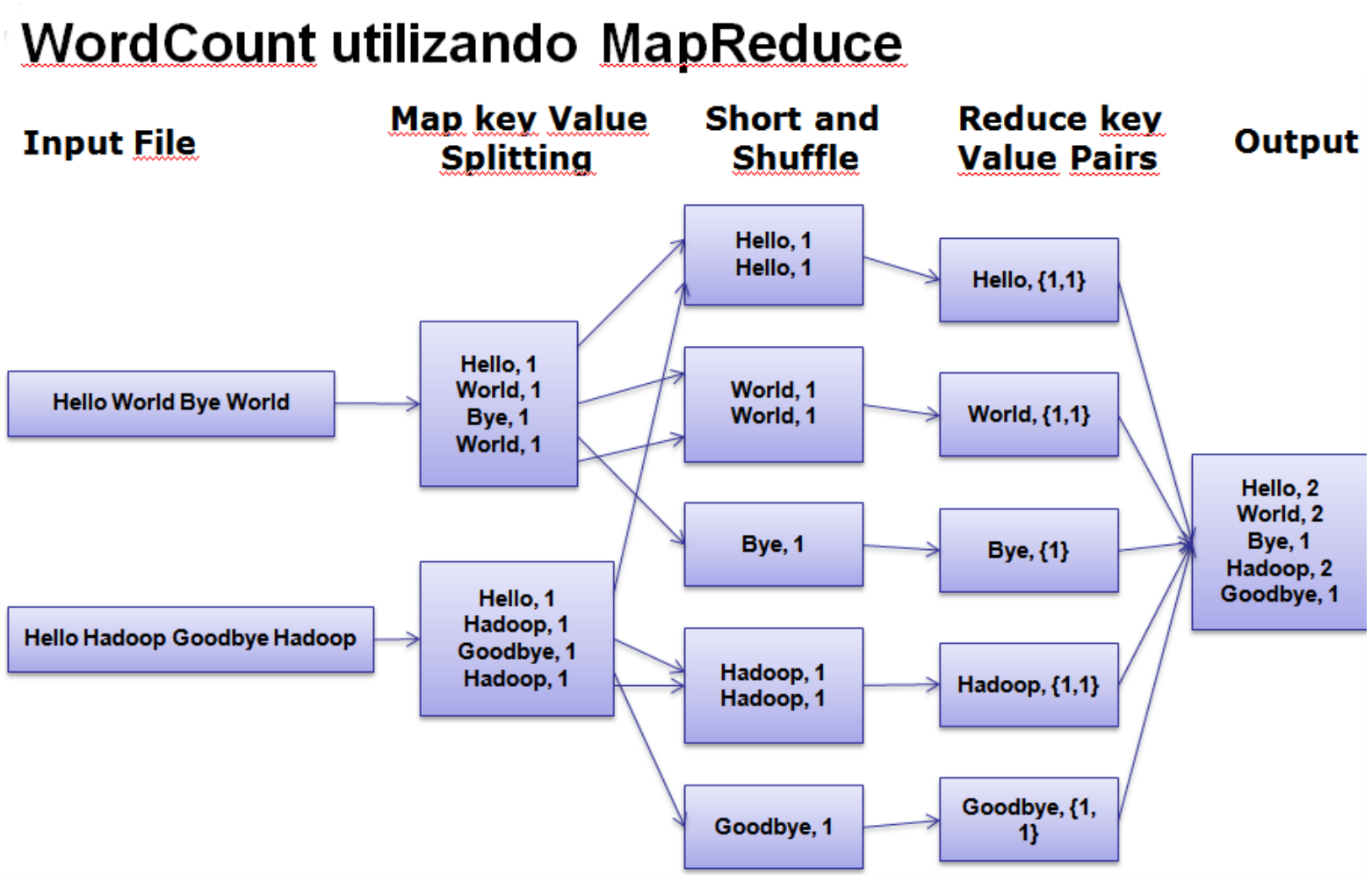
MapReduce

MapReduce: WordCound pseudo-code!

Pseudo-code:

```
map(key, value):
// key: document ID; value: text of document
    FOR (each word w in value)
        emit(w, 1);

reduce(key, value-list):
// key: a word; value-list: a list of integers
        result = 0;
        FOR (each count v on value-list)
            result += v;
        emit(key, result);
```

MapReduce


WordCount utilizando MapReduce
Input File
Map key Value Splitting
Short and Shuffle
Reduce key Value Pairs
Output
Hello World Bye World
Hello Hadoop Goodbye Hadoop
Hello, 1
World, 1
Bye, 1
World, 1
Hello, 1
Hadoop, 1
Goodbye, 1
Hadoop, 1
Hello, 1
Hello, 1
World, 1
World, 1
Bye, 1
Hadoop, 1
Hadoop, 1
Goodbye, 1
Hello, {1,1}
World, {1,1}
Bye, {1}
Hadoop, {1,1}
Goodbye, {1, 1}
Hello, 2
World, 2
Bye, 1
Hadoop, 2
Goodbye, 1

MapReduce

Resumiendo:

- **Ventaja frente a los modelos distribuidos clásicos:** El modelo de programación paralela de datos de MapReduce oculta la complejidad de la distribución y tolerancia a fallos.
- **Claves de su filosofía**: Es
  - ***escalable:*** *se olvidan los problemas de hardware*
  - ***más barato:*** *se ahorran costes en hardware, programación y administración (Commodity computing).*
- **MapReduce no es adecuado para todos los problemas, pero cuando funciona, puede ahorrar mucho tiempo**

Bibliografía: A. Fernandez, S. Río, V. López, A. Bawakid, M.J. del Jesus, J.M. Benítez, F. Herrera, **Big Data with Cloud Computing: An Insight on the Computing Environment, MapReduce and Programming Frameworks.** *WIREs Data Mining and Knowledge Discovery 4:5 (2014) 380-409*

MapReduce

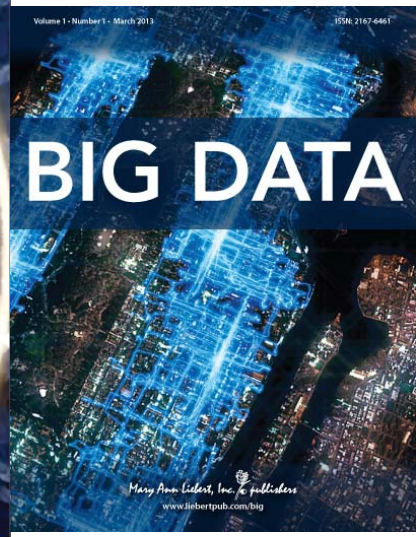
Limitaciones

"If all you have is a hammer, then everything looks like a nail."

MAPREDUCE IS GOOD ENOUGH?

If All You Have is a Hammer, Throw Away Everything That's Not a Nail!

Jimmy Lin
The iSchool, University of Maryland
College Park, Maryland



Los siguientes tipos de algoritmos son ejemplos en los que MapReduce no funciona bien:

Iterative Graph Algorithms
Gradient Descent
Expectation Maximization

Limitaciones de MapReduce

Algoritmos de grafos iterativos. Existen muchas limitaciones para estos algoritmos.
Ejemplo: Cada iteración de PageRank se corresponde a un trabajo de MapReduce.

Se han propuesto una serie de extensiones de MapReduce o modelos de programación alternativa para acelerar el cálculo iterativo:

Pregel (Google)

Implementación: http://www.michaelnielsen.org/ddi/pregel/

Malewicz, G., Austern, M., Bik, A., Dehnert, J., Horn, I., Leiser, N., and Czajkowski, G. Pregel: A system for large escale graph processing. ACM SIGMOD 2010.

Limitaciones de MapReduce

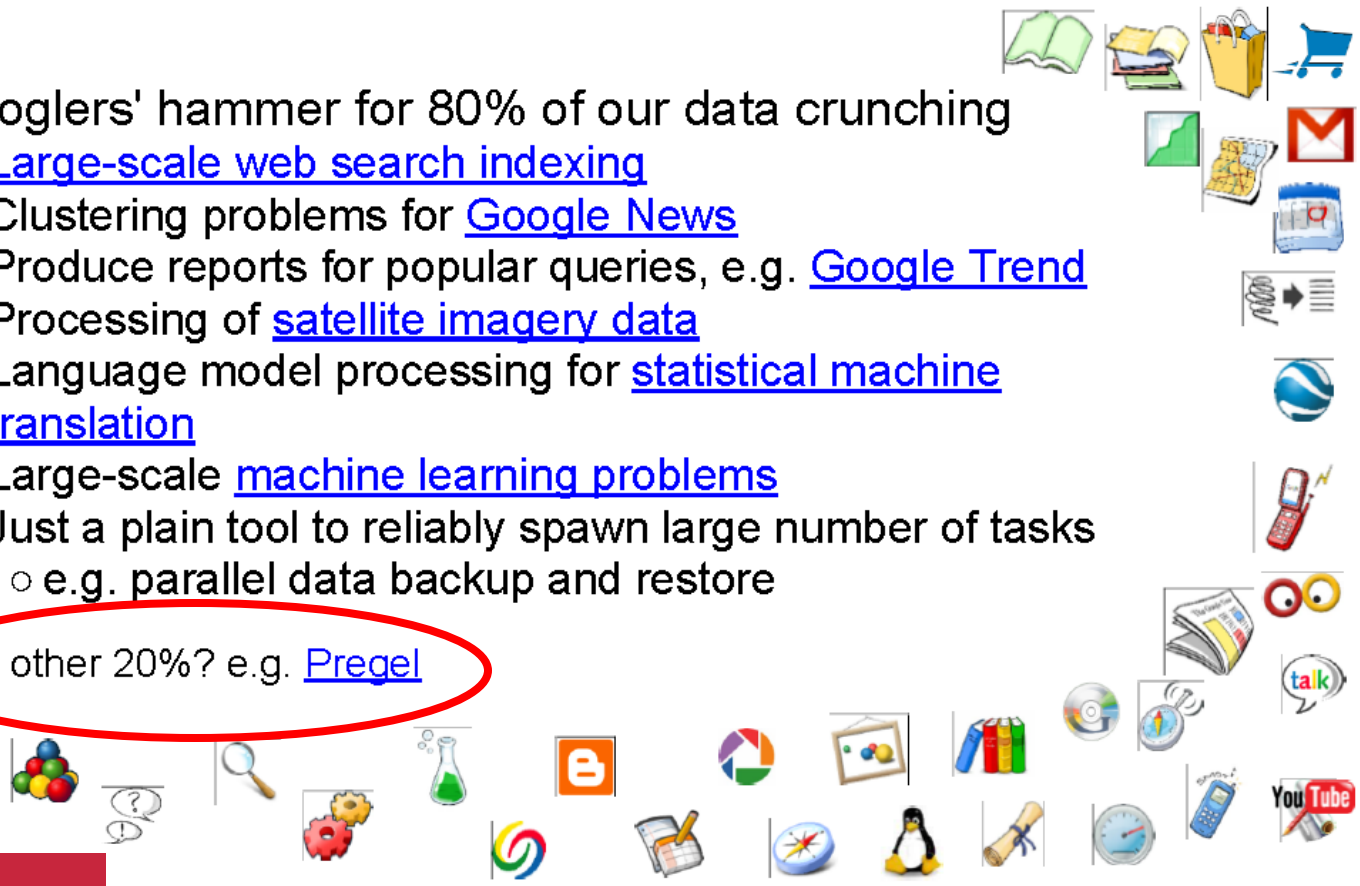
MapReduce inside Google

Google

Googlers' hammer for 80% of our data crunching

- Large-scale web search indexing
- Clustering problems for Google News
- Produce reports for popular queries, e.g. Google Trend
- Processing of satellite imagery data
- Language model processing for statistical machine translation
- Large-scale machine learning problems
- Just a plain tool to reliably spawn large number of tasks
  - e.g. parallel data backup and restore

The other 20%? e.g. Pregel



Enrique Alfonseca
Google Research Zurich

Limitaciones de MapReduce

En Resumen

Principales características

- Arquitectura escalable
- Planificación optimizada
- Elasticidad y disponibilidad
- Flexibilidad
- Seguridad y Autenticación

Limitaciones

- Aprendizaje automático: Computación iterativa
- Procesamiento de grafos
- Procesamiento en tiempo real (streams)
- Funcionalidad de comunicación de procesos
- Dificultad en la implementación tipo MR

Bibliografía: A. Fernandez, S. Río, V. López, A. Bawakid, M.J. del Jesus, J.M. Benítez, F. Herrera, **Big Data with Cloud Computing: An Insight on the Computing Environment, MapReduce and Programming Frameworks.** *WIREs Data Mining and Knowledge Discovery 4:5 (2014) 380-409*

Índice

- Big Data. Big Data Science
- ¿Por qué Big Data? Google crea el Modelo de Programación MapReduce
- **Tecnologías para Big Data: Ecosistema Hadoop (Hadoop, Spark, …)**
- Big Data Analytics: Librerías para Analítica de Datos en Big Data.
- Casos de estudio: Random Forest, Clustering
- Algunas aplicaciones: Salud, Social Media, Identificación
- Imbalanced Big Data: Caso de estudio
- Comentarios Finales

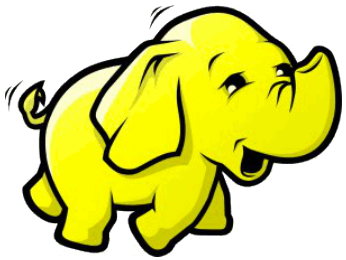
Hadoop

Hadoop es una implementación de código abierto del paradigma computacional MapReduce

http://hadoop.apache.org/

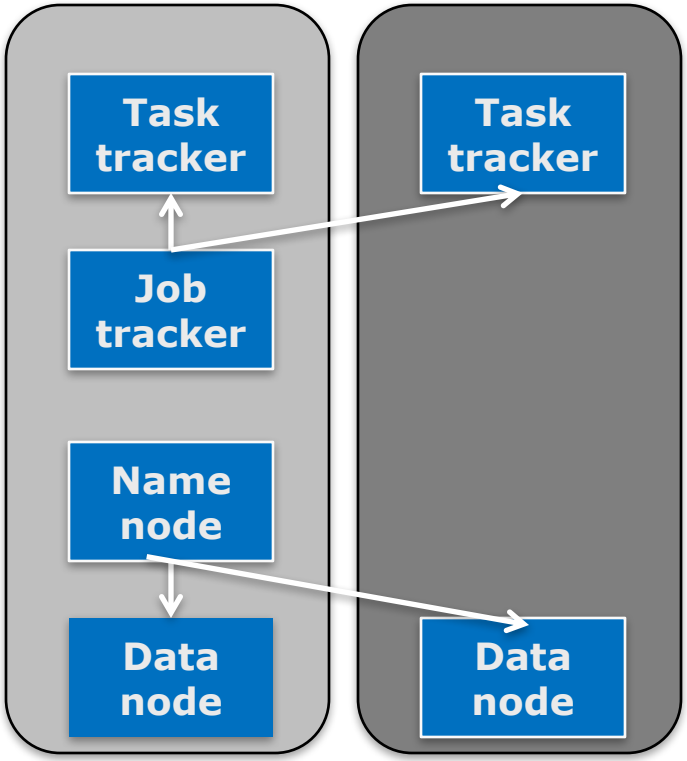
Hadoop

Hadoop Distributed File System (HDFS) es un sistema de archivos distribuido, escalable y portátil escrito en Java para el framework Hadoop



Map Reduce Layer

HDFS Layer



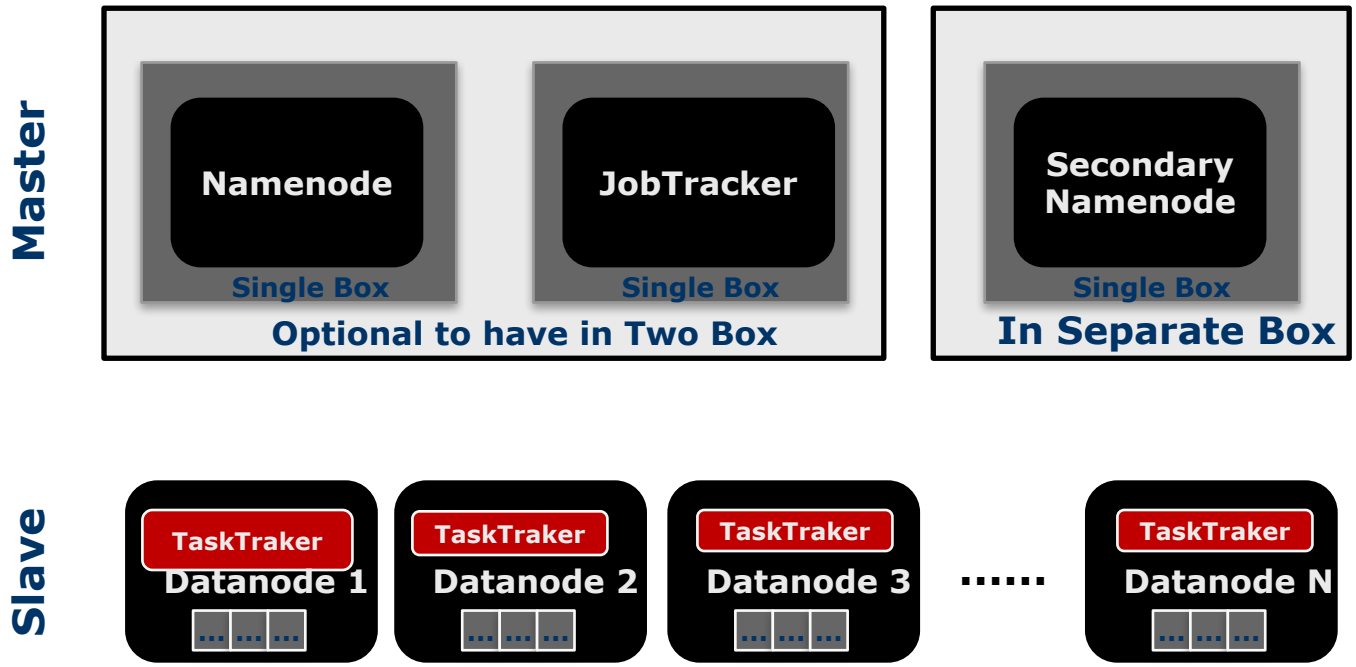
Creado por Doug Cutting (chairman of board of directors of the Apache Software Foundation, 2010)

http://hadoop.apache.org/

Hadoop

Hadoop: A master/slave architecture

- **Master**: NameNode, JobTracker
- **Slave**: {DataNode, TaskTraker}, ..., {DataNode, TaskTraker}



Hadoop

http://sortbenchmark.org/

Primer hito de Hadoop: July 2008 - Hadoop Wins Terabyte Sort Benchmark

Uno de los grupos de Yahoo Hadoop ordenó 1 terabyte de datos en 209 segundos, superando el récord anterior de 297 segundos en la competición anual de ordenación de un terabyte (Daytona). Esta es la primera vez que un programa en Java de código abierto ganó la competición.

2008, 3.48 minutes

Hadoop

910 nodes x (4 dual-core processors, 4 disks, 8 GB memory)

Owen OMalley, Yahoo

2007, 4.95 min

TokuSampleSort

tx2500 disk cluster

400 nodes x (2 processors, 6-disk RAID, 8 GB memory)

Bradley C. Kuszmaul , MIT

	Daytona
Gray	2013, 1.42 TB/min **Hadoop** 102.5 TB in 4,328 seconds 2100 nodes x (2 2.3Ghz hexcore Xeon E5-2630, 64 GB memory, 12x3TB disks) Thomas Graves Yahoo! Inc.

http://developer.yahoo.com/blogs/hadoop/hadoop-sorts-petabyte-16-25-hours-terabyte-62-422.html

Ecosistema Hadoop

El proyecto Apache Hadoop incluye los módulos:

Hadoop Common: Las utilidades comunes que apoyan los otros módulos de Hadoop.
Hadoop Distributes File System (HDFS): El sistema de ficheros que proporciona el acceso
Hadoop YARN: Marco para el manejo de recursos de programación y grupo de trabajo.
Hadoop MapReduce: Un sistema de basado en YARN o para el procesamiento en paralelo de grandes conjuntos de datos.

http://hadoop.apache.org/

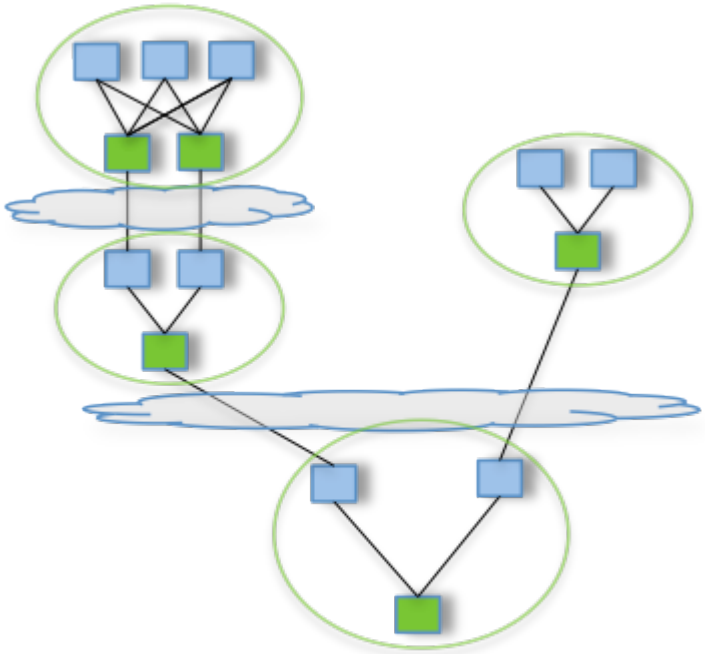
Ecosistema Apache Hadoop incluye más de 150 proyectos:

Avro: Un sistema de serialización de datos.
Cassandra: Una base de datos escalable multi-master sin puntos individuales y fallo
Chukwa: Un sistema de recogida de datos para la gestión de grandes sistemas distribuidos.
Hbase: Una base de datos distribuida, escalable que soporta estructurado de almacenamiento de datos para tablas de gran tamaño.
Hive: Un almacén de datos que proporciona el Resumen de datos para tablas de gran tamaño.
Pig: Lenguaje para la ejecución de alto nivel de flujo de datos para computación paralela.
Tez: Sustituye al modelo "MapShuffleReduce" por un flujo de ejecución con grafos acíclico dirigido (DAG)
Giraph: Procesamiento iterativo de grafos
Mahout: Aprendizaje automático escalable (biblioteca de minería de datos)

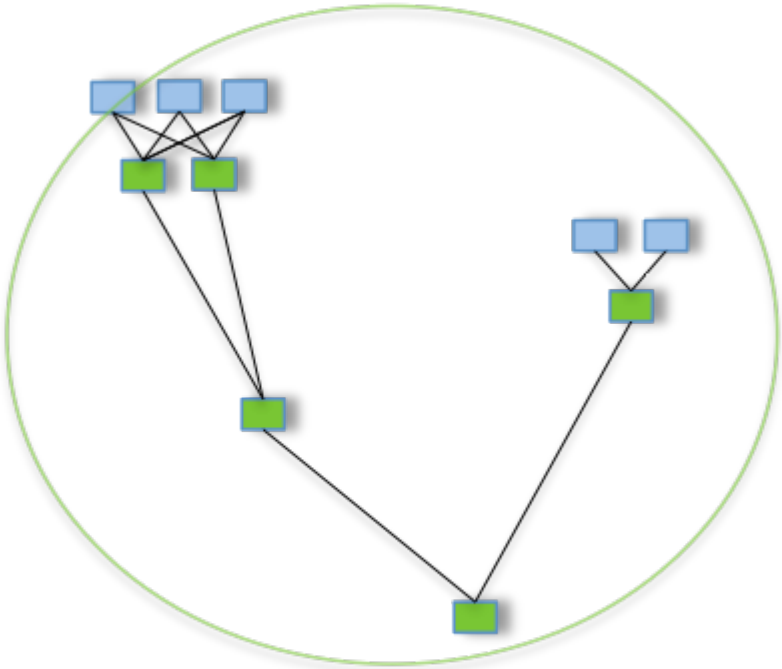
Recientemente: Apache Spark

Limitaciones de MapReduce

Procesos con flujos acíclicos de procesamiento de datos



Pig/Hive - MR

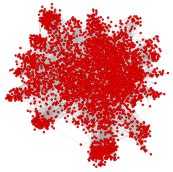


Pig/Hive - Tez

http://tez.apache.org/

Limitaciones de MapReduce: Nuevas herramientas

GIRAPH (APACHE Project)
(http://giraph.apache.org/)
Procesamiento iterativo de grafos



GPS - A Graph Processing System, (Stanford)
http://infolab.stanford.edu/gps/
para Amazon's EC2

Distributed GraphLab (Carnegie Mellon Univ.)
https://github.com/graphlab-code/graphlab
Amazon's EC2

Spark (UC Berkeley)
(Apache Foundation)
http://spark.incubator.apache.org/research.html

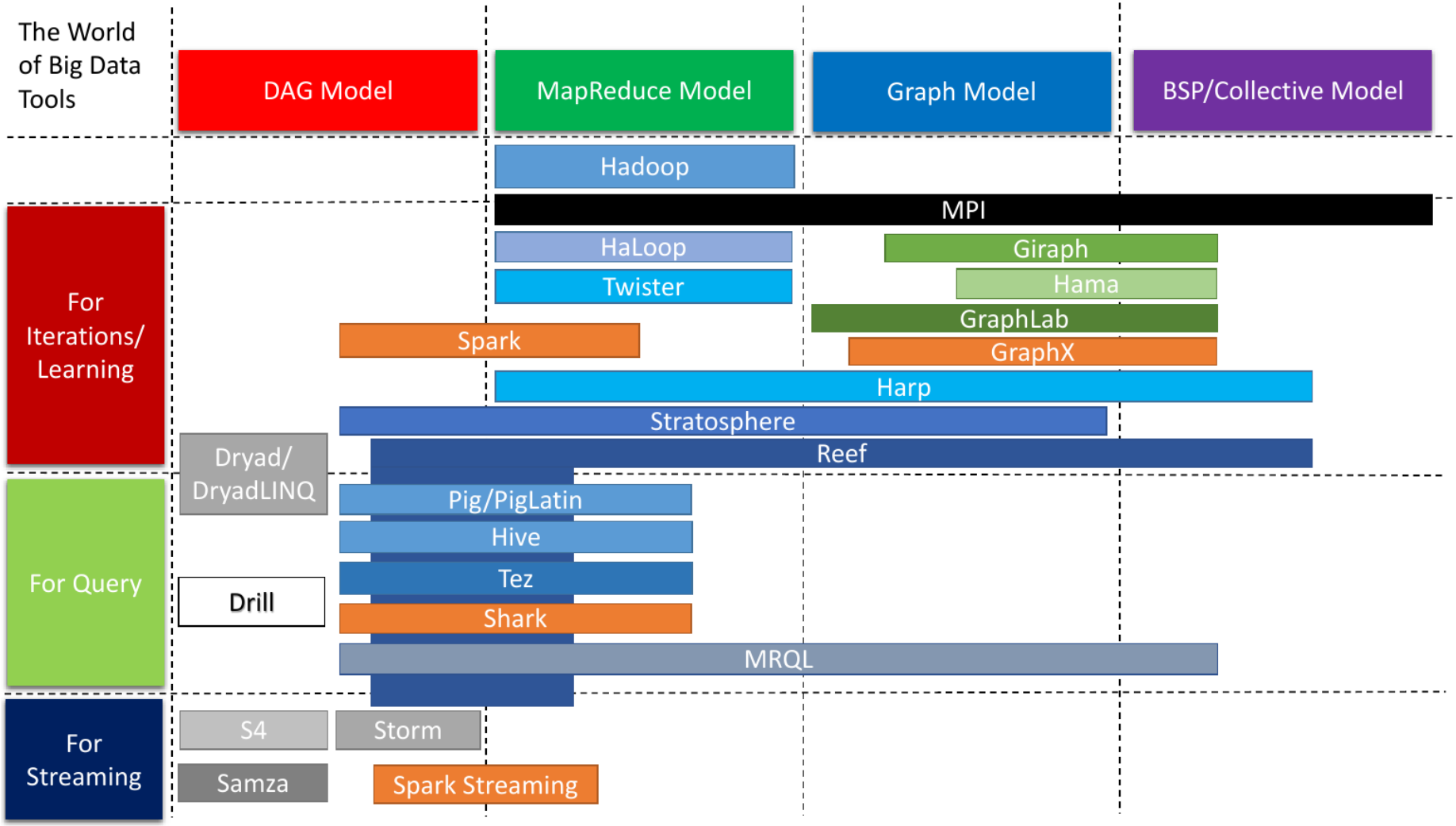
Twister (Indiana University)
http://www.iterativemapreduce.org/
Clusters propios

PrIter (University of Massachusetts Amherst, Northeastern University-China)
http://code.google.com/p/priter/
Cluster propios y Amazon EC2 cloud

HaLoop (University of Washington)
http://clue.cs.washington.edu/node/14
http://code.google.com/p/haloop/
Amazon's EC2

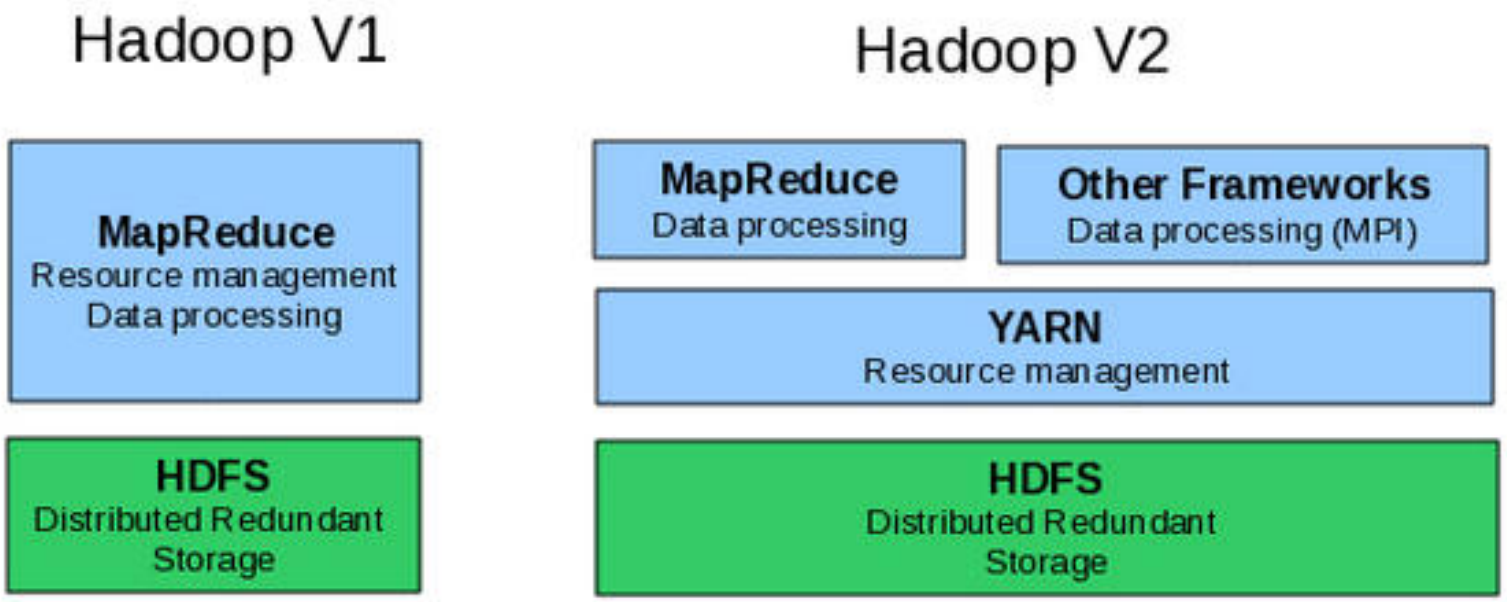
GPU based platforms
Mars
Grex
GPMR

Limitaciones de MapReduce: Nuevas herramientas

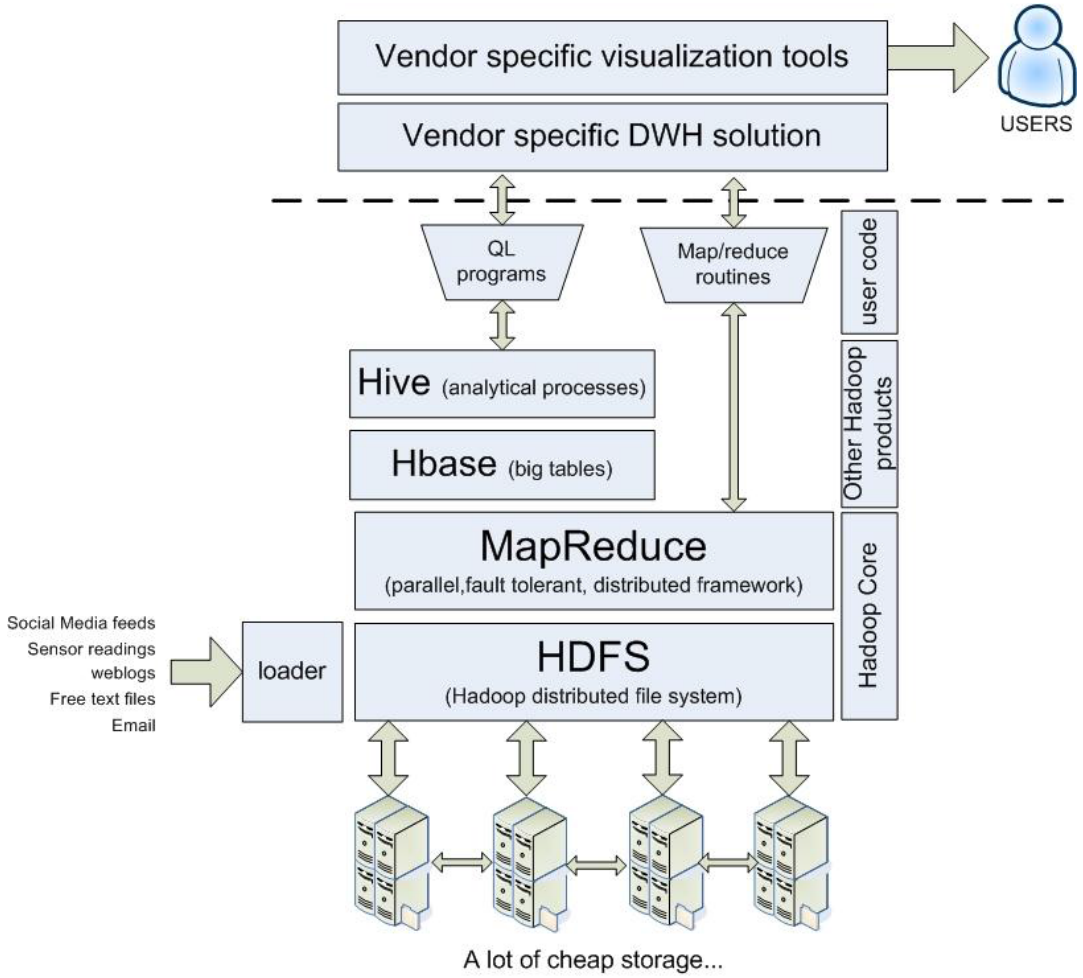


Evolución de Hadoop

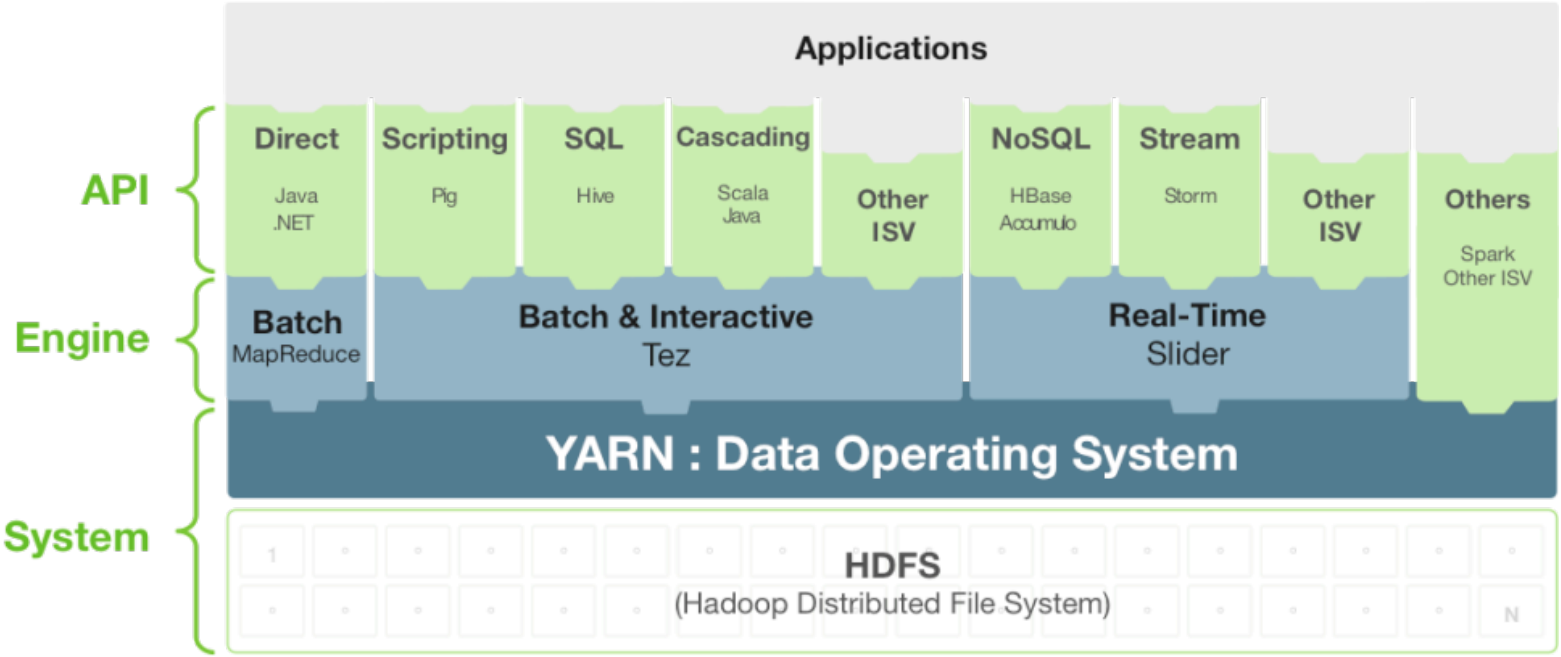
Evolución de Hadoop



Evolución de Hadoop

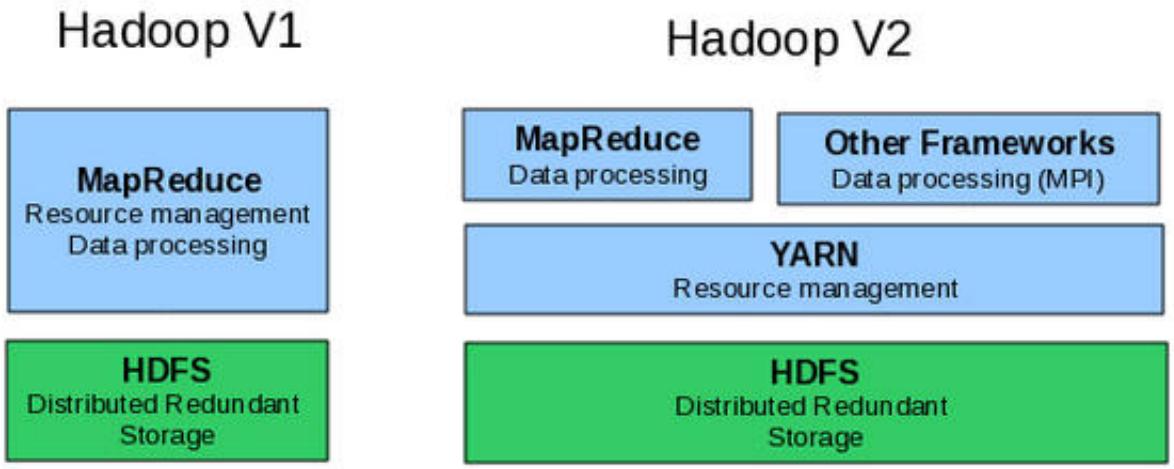


Apache Hadoop YARN



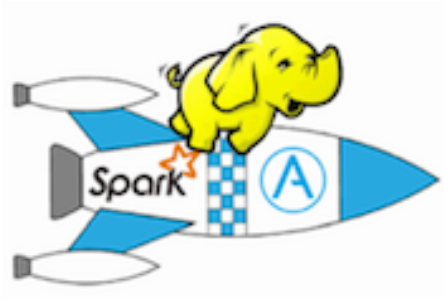
Apache Hadoop YARN es el sistema operativo de datos de Hadoop 2, responsable de la gestión del acceso a los recursos críticos de Hadoop. YARN permite al usuario interactuar con todos los datos de múltiples maneras al mismo tiempo, haciendo de Hadoop una verdadera plataforma de datos multi-uso y lo que le permite tomar su lugar en una arquitectura de datos moderna.

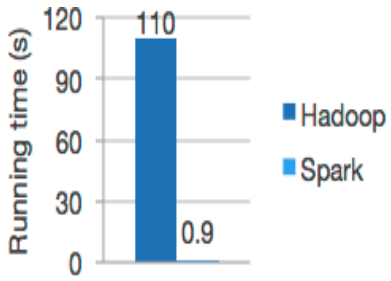
Apache Spark



https://spark.apache.org/

Enfoque InMemory
HDFS Hadoop + SPARK





Fast and expressive cluster computing system compatible with Apache Hadoop

Apache Spark (Birth 2009-2010)

Fast and Expressive Cluster Computing Engine Compatible with Apache Hadoop

Up to 10× faster on disk, 100× in memory

2-5× less code

Efficient

- General execution graphs
- In-memory storage

Usable

- Rich APIs in Java, Scala, Python
- Interactive shell

http://spark.apache.org/talks/pydata-july-2013.pdf

Apache Spark

Spark Programming Model

KEY Concept: RDD (Resilient Distributed Datasets)

Write programs in terms of operations on distributed data sets

- Collection of objects spread across a cluster, stored in RAM or on Disk
- Built through parallel transformations on distributed datasets
- An RDD is a fault-tolerant collection of elements that can be operated on in parallel.
- There are two ways to create RDDs:

  Parallelizing an existing collection in your driver program

  Referencing a dataset in an external storage system, such as a shared filesystem, HDFS, Hbase.
- *Can be cached for future reuse*
- Built through parallel transformations on distributed datasets
- RDD operations: transformations and actions

  Transformations (e.g. map, filter, groupBy)…

  (Lazy operations to build RDSs from other RDSs)

  Actions (eg. Count, collect, save …)

  (Return a result or write it a storage)

Zaharia-2012- Zaharia M, Chowdhury M, Das T, Dave A, Ma J, McCauley M, Franklin MJ, Shenker S, Stoica I. **Resilient distributed datasets: a fault-tolerant abstraction for in-memory cluster computing.** In: *9th USENIX Conference on Networked Systems Design and Implementation, San Jose, CA, 2012*, 1–14.

Apache Spark

Spark Operations

Transformations (define a new RDD)	map filter sample groupByKey reduceByKey sortByKey	flatMap union join cogroup cross mapValues
Actions (return a result to driver program)	collect reduce count save lookupKey	

Zaharia-2012- Zaharia M, Chowdhury M, Das T, Dave A, Ma J, McCauley M, Franklin MJ, Shenker S, Stoica I.
Resilient distributed datasets: a fault-tolerant abstraction for in-memory cluster computing.
In: *9th USENIX Conference on Networked Systems Design and Implementation, San Jose, CA, 2012,* 1–14.

Apache Spark

- RDDs allow us to express different programming models:
  - MapReduce.
  - Iterative MapReduce. It can implement HaLoop or Twister systems.
  - Stream processing.
  - Iterative graph applications (Google's Pregel).
  - SQL.
  - ...
- It provides us more flexibility to design scalable ML tools.

Zaharia-2012- Zaharia M, Chowdhury M, Das T, Dave A, Ma J, McCauley M, Franklin MJ, Shenker S, Stoica I.
Resilient distributed datasets: a fault-tolerant abstraction for in-memory cluster computing.
In: *9th USENIX Conference on Networked Systems Design and Implementation, San Jose, CA, 2012,* 1–14.

Apache Spark

DataFrames

A DataFrame is a distributed collection of data organized into named columns. It is conceptually equivalent to a table in a relational database or a data frame in R/Python, but with richer optimizations under the hood. DataFrames can be constructed from a wide array of sources such as: structured data files, tables in Hive, external databases, or existing RDDs.

The DataFrame API is available in Scala, Java, Python, and R.

Datasets

A Dataset is a new experimental interface added in Spark 1.6 that tries to provide the benefits of RDDs (strong typing, ability to use powerful lambda functions) with the benefits of Spark SQL's optimized execution engine. A Dataset can be constructed from JVM objects and then manipulated using functional transformations (map, flatMap, filter, etc.).

The unified Dataset API can be used both in Scala and Java. Python does not yet have support for the Dataset API, but due to its dynamic nature many of the benefits are already available (i.e. you can access the field of a row by name naturally `row.columnName`). Full python support will be added in a future release.

Zaharia-2012- Zaharia M, Chowdhury M, Das T, Dave A, Ma J, McCauley M, Franklin MJ, Shenker S, Stoica I.
Resilient distributed datasets: a fault-tolerant abstraction for in-memory cluster computing.
In: *9th USENIX Conference on Networked Systems Design and Implementation, San Jose, CA, 2012,* 1–14.

Apache Spark

	Hadoop World Record	Spark 100 TB *
Data Size	102.5 TB	100 TB
Elapsed Time	72 mins	23 mins
# Nodes	2100	206
# Cores	50400	6592
# Reducers	10,000	29,000
Rate	1.42 TB/min	4.27 TB/min
Rate/node	0.67 GB/min	20.7 GB/min
Sort Benchmark Daytona Rules	Yes	Yes
Environment	dedicated data center	EC2 (i2.8xlarge)

October 10, 2014

Using Spark on 206 EC2 nodes, we completed the benchmark in 23 minutes. This means that Spark sorted the same data 3X faster using 10X fewer machines. All the sorting took place on disk (HDFS), without using Spark's in-memory cache.

http://databricks.com/blog/2014/10/10/spark-petabyte-sort.html

Apache Spark

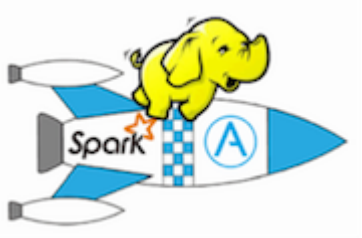
November 5, 2014

	Daytona
Gray	2013, 1.42 TB/min **Hadoop** 102.5 TB in 4,328 seconds 2100 nodes x (2 2.3Ghz hexcore Xeon E5-2630, 64 GB memory, 12x3TB disks) Thomas Graves Yahoo! Inc.

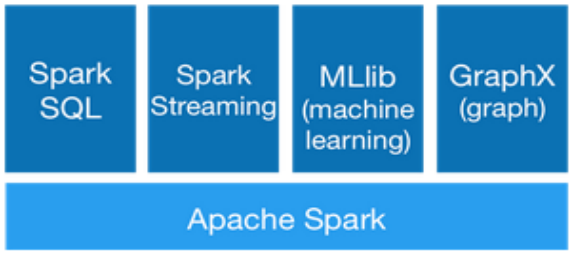
	Daytona
Gray	2-way tie: 2014, 4.35 TB/min **TritonSort** 100 TB in 1,378 seconds 186 Amazon EC2 i2.8xlarge nodes x (32 vCores - 2.50Ghz Intel Xeon E5-2670 v2, 244GB memory, 8x800 GB SSD) Michael Conley, Amin Vahdat, George Porter University of California, San Diego 2014, 4.27 TB/min **Apache Spark** 100 TB in 1,406 seconds 207 Amazon EC2 i2.8xlarge nodes x (32 vCores - 2.5Ghz Intel Xeon E5-2670 v2, 244GB memory, 8x800 GB SSD) Reynold Xin, Parviz Deyhim, Xiangrui Meng, Ali Ghodsi, Matei Zaharia Databricks

http://sortbenchmark.org/

Apache Spark

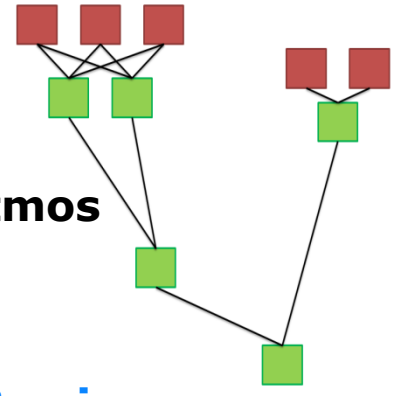


Ecosistema Apache Spark



• **Big Data "in-memory".** Spark permite realizar trabajos paralelizados totalmente en memoria, lo cual reduce mucho los tiempos de procesamiento. Sobre todo si se trata de unos procesos iterativos. En el caso de que algunos datos no quepan en la memoria, Spark seguirá trabajando y usará el disco duro para volcar aquellos datos que no se necesitan en este momento (Hadoop **"commodity hardware")**.

KEY Concept: RDD (Resilient Distributed Datasets)
Write programs in terms of operations on distributed data sets



• **Esquema de computación más flexible que MapReduce.**
Permite la flujos acíclicos de procesamiento de datos, algoritmos iterativos

• **Spark ofrece una API para Java, Python y Scala**

• Spark / Wiki Homepage
Powered By Spark
Creado por Andy Konwinski, modificado por última vez por Reynold Xin el sep 08, 2015

Databricks, Groupon, eBay inc., Amazon, Hitachi, Nokia, Yahoo!, …

https://cwiki.apache.org/confluence/display/SPARK/Powered+By+Spark

Flink

https://flink.apache.org/

Flink | Overview | Features | Downloads | FAQ | Quickstart | Documentation

Apache Flink is an open source platform for distributed stream and batch data processing.

Flink's core is a streaming dataflow engine that provides data distribution, communication, and fault tolerance for distributed computations over data streams.

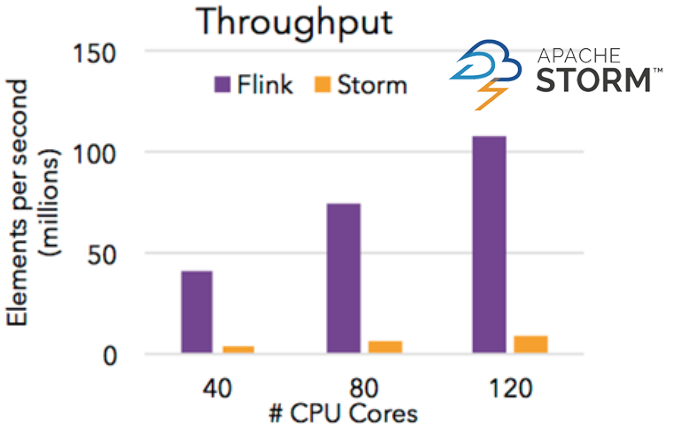
Flink includes **several APIs** for creating applications that use the Flink engine:

1. DataStream API for unbounded streams embedded in Java and Scala, and
2. DataSet API for static data embedded in Java, Scala, and Python,
3. Table API with a SQL-like expression language embedded in Java and Scala.

Flink also bundles **libraries for domain-specific use cases**:

1. CEP, a complex event processing library,
2. Machine Learning library, and
3. Gelly, a graph processing API and library.

You can **integrate** Flink easily with other well-known open source systems both for data input and output as well as deployment.



Batch on Streaming

Batch processing applications run efficiently as special cases of stream processing applications.

Streaming First

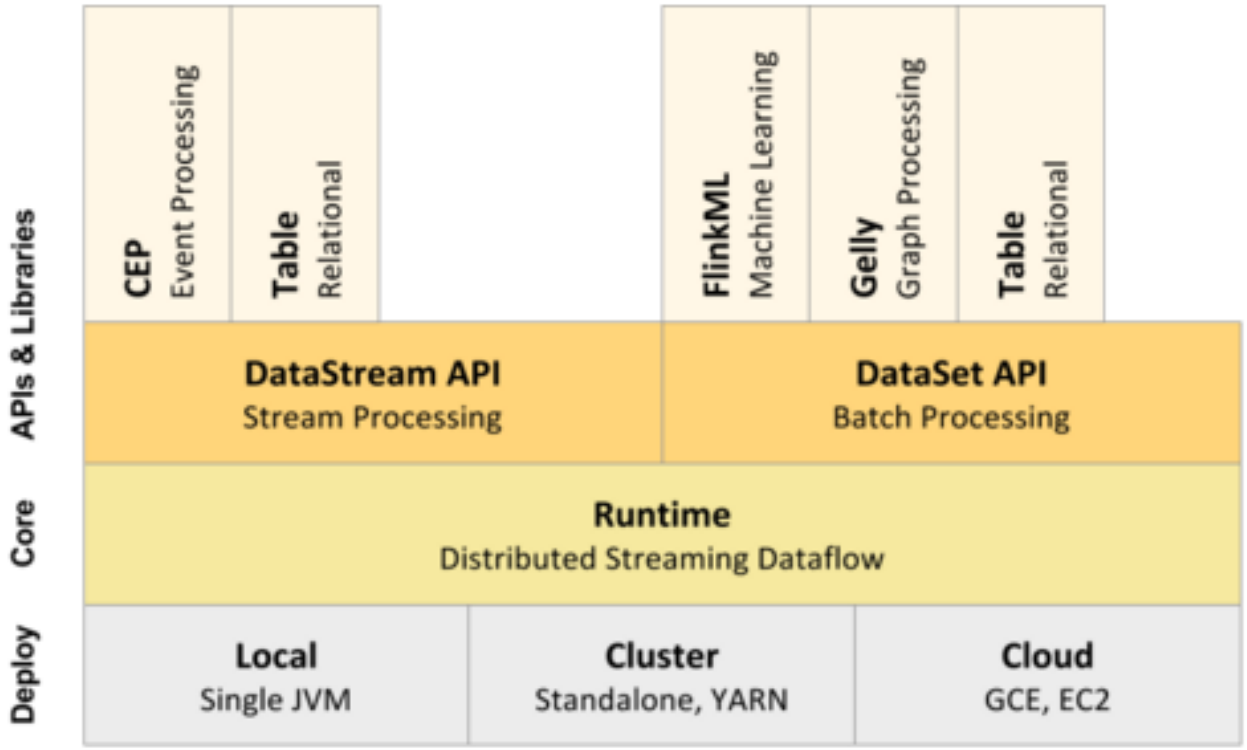
High throughput and low latency stream processing with exactly-once guarantees.

APIs, Libraries, and Ecosystem

DataSet, DataStream, and more. Integrated with the Apache Big Data stack.

Flink

https://flink.apache.org/



¿Cómo accedo a una plataforma Hadoop?

Plataformas Cloud con instalación de Hadoop

Amazon Elastic Compute Cloud (Amazon EC2)
http://aws.amazon.com/es/ec2/

Windows Azure
http://www.windowsazure.com/

Instalación en un cluster
Ejemplo ATLAS, infraestructura del grupo SCI²S

Cluster ATLAS: 4 super servers from Super Micro Computer Inc. (4 nodes per server)
The features of each node are:

- **Microprocessors: 2 x Intel Xeon E5-2620 (6 cores/12 threads, 2 GHz, 15 MB Cache)**
- **RAM 64 GB DDR3 ECC 1600MHz, Registered**
- **1 HDD SATA 1TB, 3Gb/s; (system)**
- **1 HDD SATA 2TB, 3Gb/s; (distributed file system)**

Distribución que ofrece Cloudera para Hadoo

http://www.cloudera.com/content/cloudera/en/why-cloudera/hadoop-and-big-data.html

Tecnologías para Big Data: Ecosistema Hadoop (Hadoop, Spark, …) (Una instantánea)

Big Data is high-volume, high-velocity and high-variety information assets that demand cost-effective, innovative forms of information processing for enhanced insight and decision making.

There are two main issues related to Big Data:

1. **Database/storage frameworks: to write, read, and manage data.**
2. **Computational models: to process and analyze data.**

Recently, there are the following Big Data frameworks:

1. **Storage frameworks: Google File System (GFS), Hadoop Distributed File Systems (HDFS).**
2. **Computational models: MapReduce (Apache Hadoop), Resilient Distributed Datasets (RDD by Apache Spark, DataFrames, Dataset API (Spark 1.6)).**

2001-2010

2010-2017

Big Data: Technology and Chronology

The Apache Software Foundation

BIG DATA

2001
3V's Gartner
Doug Laney

Google

MapReduce

2004
MapReduce
Google
Jeffrey Dean

Hadoop Ecosystem

2009-2013 Flink
TU Berlin
Flink Apache (Dec. 2014) Volker Markl

Flink

Big Data

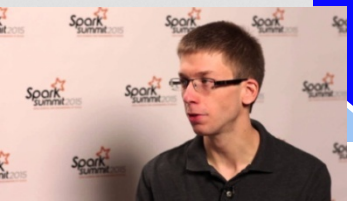
YAHOO!

hadoop

2008
Hadoop
Yahoo!
Doug Cutting

2010 Spark
U Berckeley
Apache Spark
Feb. 2014
Matei Zaharia

Spark
Lightning-Fast Cluster Computing



2010-2017:

Big Data Analytics: Mahout, MLLib, …

Hadoop Ecosystem

Applications New Technology

Índice

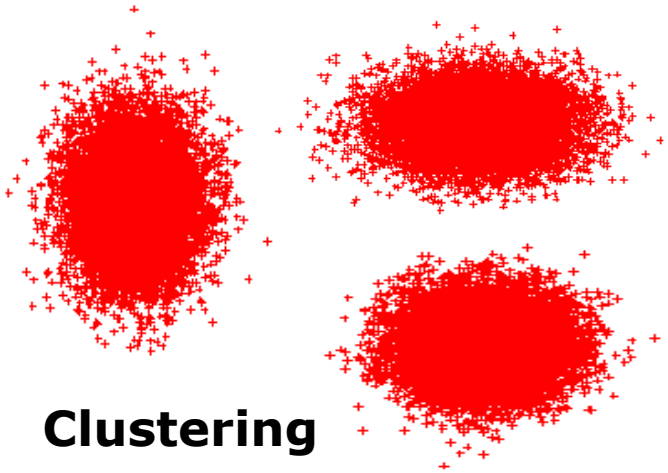
- Big Data. Big Data Science
- ¿Por qué Big Data? Google crea el Modelo de Programación MapReduce
- Tecnologías para Big Data: Ecosistema Hadoop (Hadoop, Spark, …)
- **Big Data Analytics: Librerías para Analítica de Datos en Big Data.**
- Casos de estudio: Random Forest, Clustering
- Algunas aplicaciones: Salud, Social Media, Identificación
- Imbalanced Big Data: Caso de estudio
- Comentarios Finales

Big Data Analytics

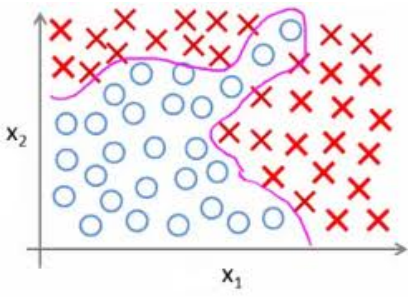
- **Big Data Analytics: Escenario**
- **Big Data Analytics: Tools**
  (Mahout, MLlib, Spark.ml, FlinkML, H2O)
- **Caso de estudio: Random Forest**
- **Aprendizaje no supervisado: Caso de estudio K-Means**
- **Big Data Analytics: 3 Comentarios finales.**
  - Without Analytics, Big data is just noise: Smart Data
  - Big data preprocessing: Es necesario
  - Los expertos en Ciencia de Datos son necesarios en el uso de herramientas de Analytics y Big Data.

Big Data Analytics

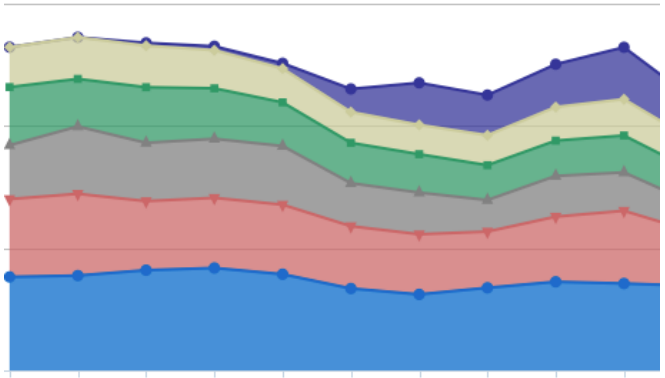
Potentiales escenarios:



Clustering

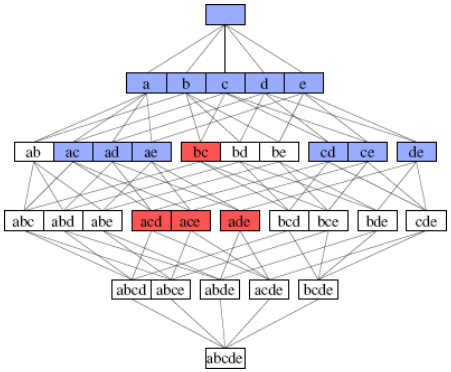


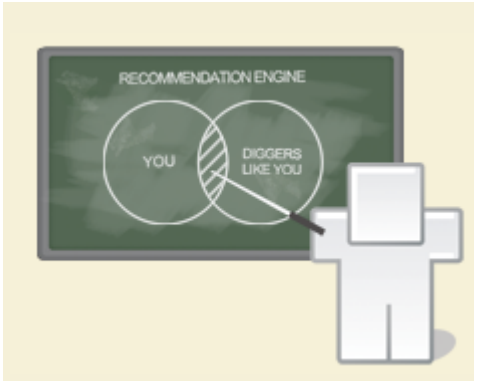
Classification



Real Time Analytics/
Big Data Streams

Association





Recommendation
Systems

Social Media Mining
Social Big Data

Big Data Analytics: Tools

Generation	1st Generation	2nd Generation	3nd Generation
Examples	SAS, R, Weka, SPSS, KEEL	Mahout, Pentaho, Cascading	Spark, Haloop, GraphLab, Pregel, Giraph, ML over Storm
Scalability	Vertical	Horizontal (over Hadoop)	Horizontal (Beyond Hadoop)
Algorithms Available	Huge collection of algorithms	Small subset: sequential logistic regression, linear SVMs, Stochastic Gradient Decendent, k-means clustsering, Random forest, etc.	Much wider: CGD, ALS, collaborative filtering, kernel SVM, matrix factorization, Gibbs sampling, etc.
Algorithms Not Available	Practically nothing	Vast no.: Kernel SVMs, Multivariate Logistic Regression, Conjugate Gradient Descendent, ALS, etc.	Multivariate logistic regression in general form, k-means clustering, etc. – Work in progress to expand the set of available algorithms
Fault-Tolerance	Single point of failure	Most tools are FT, as they are built on top of Hadoop	FT: HaLoop, Spark Not FT: Pregel, GraphLab, Giraph

Big Data Analytics: Tools

Mahout

Classification	Single Machine	MapReduce
Logistic Regression - trained via SGD	x	
Naive Bayes / Complementary Naive Bayes		x
Random Forest		x
Hidden Markov Models	x	
Multilayer Perceptron	x	

MLlib

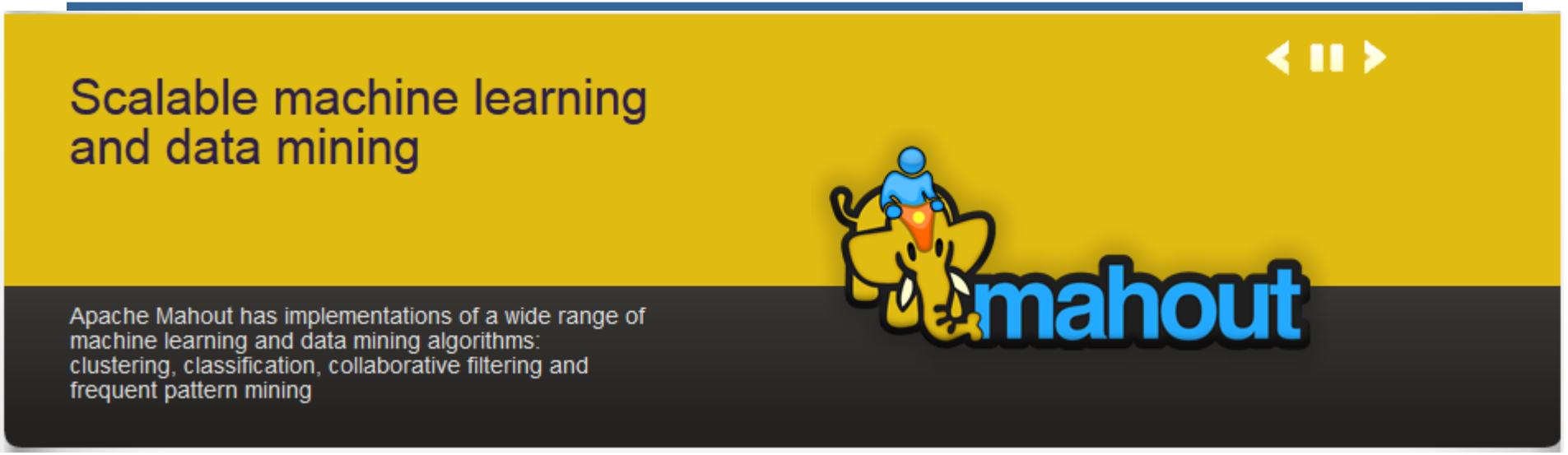
MLlib types, algorithms and utilities

This lists functionality included in `spark.mllib`, the main MLlib API.

- Data types
- Basic statistics
  - summary statistics
  - correlations
  - stratified sampling
  - hypothesis testing
  - random data generation
- Classification and regression
  - linear models (SVMs, logistic regression, linear regression)
  - naive Bayes
  - decision trees
  - ensembles of trees (Random Forests and Gradient-Boosted Trees)
  - isotonic regression
- Collaborative filtering
  - alternating least squares (ALS)
- Clustering
  - k-means
  - Gaussian mixture
  - power iteration clustering (PIC)
  - latent Dirichlet allocation (LDA)
  - streaming k-means
- Dimensionality reduction
  - singular value decomposition (SVD)
  - principal component analysis (PCA)
- Feature extraction and transformation
- Frequent pattern mining
  - FP-growth
- Optimization (developer)
  - stochastic gradient descent
  - limited-memory BFGS (L-BFGS)
- PMML model export

https://spark.apache.org/mllib/

Mahout



Biblioteca de código abierto

Currently Mahout supports mainly three use cases: Recommendation mining takes users' behavior and from that tries to find items users might like. Clustering takes e.g. text documents and groups them into groups of topically related documents. Classification learns from existing categorized documents what documents of a specific category look like and is able to assign unlabelled documents to the (hopefully) correct category.

http://mahout.apache.org/

Latest release version 0.9 has

- User and Item based recommenders
- Matrix factorization based recommenders
- K-Means, Fuzzy K-Means clustering
- Latent Dirichlet Allocation
- Singular Value Decomposition
- Logistic regression classifier
- (Complementary) Naive Bayes classifier
- Random forest classifier
- High performance java collections
- A vibrant community

Mahout

Historia

25 July 2013 - Apache Mahout 0.8 released

Visit our release notes page for details.

16 June 2012 - Apache Mahout 0.7 released

Visit our release notes page for details.

6 Feb 2012 - Apache Mahout 0.6 released

Visit our release notes page for details.

9 Oct 2011 - Mahout in Action released

The book Mahout in Action is available in print. Sean Owen, Robin Anil, Ted Dunning and Ellen Friedman thank the community (especially those who were reviewers) for input during the process and hope it is enjoyable.

Find it at your favorite bookstore, or order print and eBook copies from Manning -- use discount code "mahout37" for 37% off.

Mahout

Historia

1 February 2014 - Apache Mahout 0.9 released

Apache Mahout has reached version 0.9. All developers are encouraged to begin using version 0.9. Highlights include:

- New and improved Mahout website based on Apache CMS - MAHOUT-1245
- Early implementation of a Multi Layer Perceptron (MLP) classifier - MAHOUT-1265
- Scala DSL Bindings for Mahout Math Linear Algebra. See this blogpost and MAHOUT-1297
- Recommenders as Search. See [https://github.com/pferrel/solr-recommender] and MAHOUT-1288
- Support for easy functional Matrix views and derivatives - MAHOUT-1300
- JSON output format for ClusterDumper - MAHOUT-1343
- Enabled randomised testing for all Mahout modules using Carrot RandomizedRunner - MAHOUT-1345
- Online Algorithm for computing accurate Quantiles using 1-dimensional Clustering - See this pdf and MAHOUT-1361
- Upgrade to Lucene 4.6.1 - MAHOUT-1364

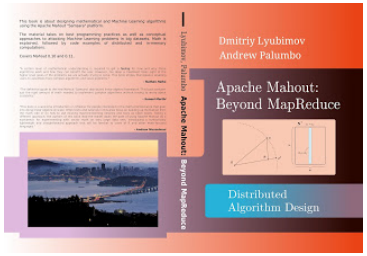
Changes in 0.9 are detailed in the release notes.

The following algorithms that were marked deprecated in 0.8 have been removed in 0.9:

- Switched LDA implementation from Gibbs Sampling to Collapsed Variational Bayes
- Meanshift - removed due to lack of actual usage and support
- MinHash - removed due to lack of actual usage and support
- Winnow - removed due to lack of actual usage and support
- Perceptron - removed due to lack of actual usage and support
- Slope One - removed due to lack of actual usage
- Distributed Pseudo recommender - removed due to lack of actual usage
- TreeClusteringRecommender - removed due to lack of actual usage

Mahout

Historia



12 March 2016 - Apache Mahout 0.11.2 released¶

23 February 2016 - New Apache Mahout Book - "Apache Mahout: Beyond MapReduce" by D.Lyubimov and A.Palumbo released. See Mahout "Samsara" Book Is Out

Mahout News

25 April 2014 - Goodbye MapReduce

The Mahout community decided to move its codebase onto modern data processing systems that offer a richer programming model and more efficient execution than Hadoop MapReduce. **Mahout will therefore reject new MapReduce algorithm implementations from now on**. We will however keep our widely used MapReduce algorithms in the codebase and maintain them.

We are building our future implementations on top of a DSL for linear algebraic operations which has been developed over the last months. Programs written in this DSL are automatically optimized and executed in parallel on Apache Spark.

Furthermore, there is an experimental contribution undergoing which aims to integrate the h20 platform into Mahout.

Mahout

Versión 0.12.2

- **Apache Mahout software** provides three major features:
- A simple and extensible programming environment and framework for building scalable algorithms
- A wide variety of premade algorithms for Scala + Apache Spark, H2O, Apache Flink
- Samsara, a vector math experimentation environment with R-like syntax which works at scale

Mahout

Versión 0.12.2

Latest release version 0.12.2 has
Apache Mahout Samsara Environment includes
Distributed Algebraic optimizer
R-Like DSL Scala API
Linear algebra operations
Ops are extensions to Scala
IScala REPL based interactive shell
Integrates with compatible libraries like MLLib
Runs on distributed Spark, H2O, and Flink
fastutil to speed up sparse matrix and vector computations
Matrix to tsv conversions for integration with Apache Zeppelin
Apache Mahout Samsara Algorithms included
Stochastic Singular Value Decomposition (ssvd, dssvd)
Stochastic Principal Component Analysis (spca, dspca)
Distributed Cholesky QR (thinQR)
Distributed regularized Alternating Least Squares (dals)
Collaborative Filtering: Item and Row Similarity
Naive Bayes Classification
Distributed and in-core

http://mahout.apache.org/

Spark Libraries

https://spark.apache.org/

Lightning-fast cluster computing

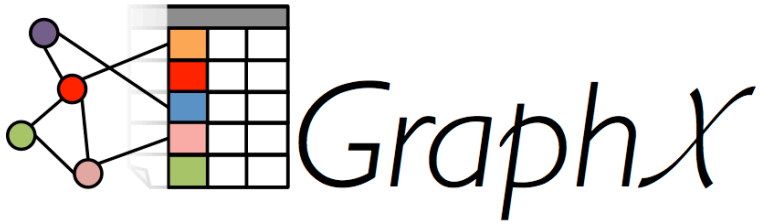
Download | Libraries ▾ | Documentation ▾ | Examples | Community ▾ | FAQ

- APIs: RDD, DataFrame and SQL
- Backend Execution: DataFrame and SQL
- Integrations: Data Sources, Hive, Hadoop, Mesos and Cluster Management
- R Language
- Machine Learning and Advanced Analytics
- Spark Streaming
- Deprecations, Removals, Configs, and Behavior Changes
  - Spark Core
  - Spark SQL & DataFrames
  - Spark Streaming
  - MLlib
- Known Issues
  - SQL/DataFrame
  - Streaming
- Credits

MLlib

Spark
MLlib

https://spark.apache.org/docs/latest/mllib-guide.html



MLlib

Spark
Lightning-Fast Cluster Computing

Machine Learning Library (MLlib) Guide

MLlib is Spark's machine learning (ML) library. Its goal is to make practical machine learning scalable and easy. At a high level, it provides tools such as:

- ML Algorithms: common learning algorithms such as classification, regression, clustering, and collaborative filtering
- Featurization: feature extraction, transformation, dimensionality reduction, and selection
- Pipelines: tools for constructing, evaluating, and tuning ML Pipelines
- Persistence: saving and load algorithms, models, and Pipelines
- Utilities: linear algebra, statistics, data handling, etc.

Announcement: DataFrame-based API is primary API

The MLlib RDD-based API is now in maintenance mode.

As of Spark 2.0, the RDD-based APIs in the `spark.mllib` package have entered maintenance mode. The primary Machine Learning API for Spark is now the DataFrame-based API in the `spark.ml` package.

https://spark.apache.org/mllib/

MLlib

Spark
MLlib

http://spark.apache.org/mllib/

MLlib is Apache Spark's scalable machine learning library.

Ease of Use

Usable in Java, Scala and Python.

MLlib fits into Spark's APIs and interoperates with NumPy in Python (starting in Spark 0.9). You can use any Hadoop data source (e.g. HDFS, HBase, or local files), making it easy to plug into Hadoop workflows.

Performance

High-quality algorithms, 100x faster than MapReduce.

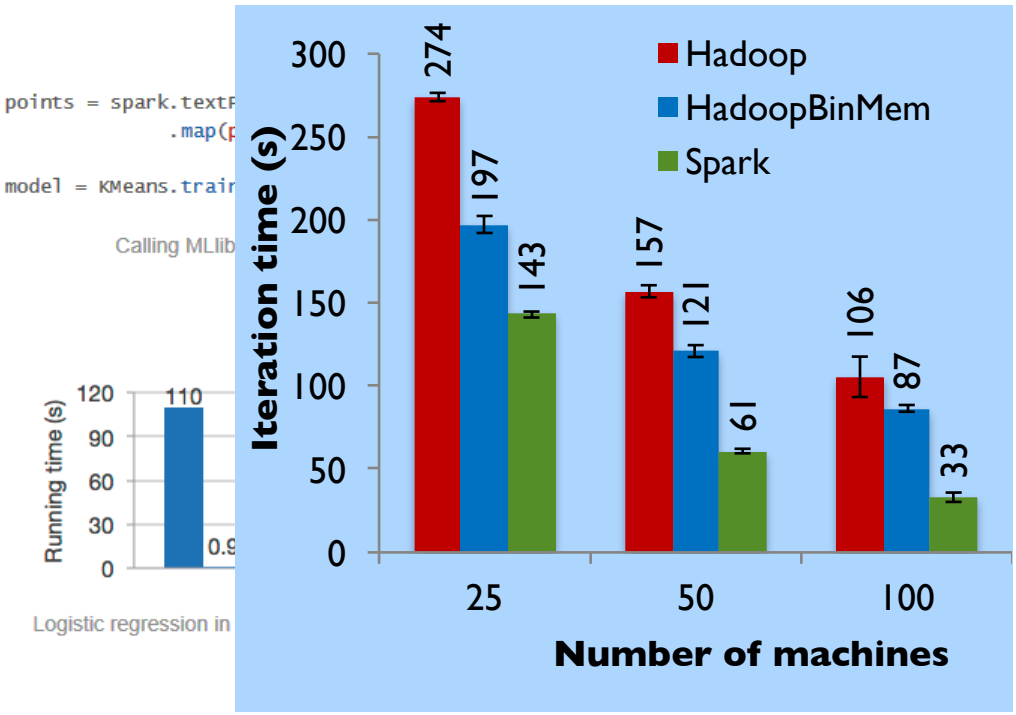
Spark excels at iterative computation, enabling MLlib to run fast. At the same time, we care about algorithmic performance: MLlib contains high-quality algorithms that leverage iteration, and can yield better results than the one-pass approximations sometimes used on MapReduce.

Easy to Deploy

Runs on existing Hadoop clusters and data.

If you have a Hadoop 2 cluster, you can run Spark and MLlib without any pre-installation. Otherwise, Spark is easy to run standalone or on EC2 or Mesos. You can read from HDFS, HBase, or any Hadoop data source.

K-Means



[Zaharia et. al, NSDI'12]

hadoop

MLlib

Spark MLlib

https://spark.apache.org/docs/latest/mllib-guide.html

MLlib: RDD-based API

This page documents sections of the MLlib guide for the RDD-based API (the `spark.mllib` package). Please see the MLlib Main Guide for the DataFrame-based API (the `spark.ml` package), which is now the primary API for MLlib.

- Data types
- Basic statistics
  - summary statistics
  - correlations
  - stratified sampling
  - hypothesis testing
  - streaming significance testing
  - random data generation
- Classification and regression
  - linear models (SVMs, logistic regression, linear regression)
  - naive Bayes
  - decision trees
  - ensembles of trees (Random Forests and Gradient-Boosted Trees)
  - isotonic regression
- Collaborative filtering
  - alternating least squares (ALS)
- Clustering
  - k-means
  - Gaussian mixture
  - power iteration clustering (PIC)
  - latent Dirichlet allocation (LDA)
  - bisecting k-means
  - streaming k-means
- Dimensionality reduction
  - singular value decomposition (SVD)
  - principal component analysis (PCA)
- Feature extraction and transformation
- Feature extraction and transformation
- Frequent pattern mining
  - FP-growth
  - association rules
  - PrefixSpan
- Evaluation metrics
- PMML model export
- Optimization (developer)
  - stochastic gradient descent
  - limited-memory BFGS (L-BFGS)

http://spark-packages.org/

Spark Packages

A community index of packages for Apache Spark.

FlinkML

https://ci.apache.org/projects/flink/flink-docs-master/apis/batch/libs/ml/

- Home
- Concepts
- Quickstart
- Examples
- Project Setup
- **Application Development**
  - Basic API Concepts
  - Streaming (DataStream API)
  - Batch (DataSet API)
  - Data Types & Serialization
  - Managing Execution

</> Application Development / Libraries / Machine Learning

FlinkML - Machine Learning for Flink

FlinkML is the Machine Learning (ML) library for Flink. It is a new effort in the Flink community, with a growing list of algorithms and contributors. With FlinkML we aim to provide scalable ML algorithms, an intuitive API, and tools that help minimize glue code in end-to-end ML systems. You can see more details about our goals and where the library is headed in our vision and roadmap here.

FlinkML

https://ci.apache.org/projects/flink/flink-docs-master/apis/batch/libs/ml/

Supported Algorithms

FlinkML currently supports the following algorithms:

Supervised Learning

- SVM using Communication efficient distributed dual coordinate ascent (CoCoA)
- Multiple linear regression
- Optimization Framework

Unsupervised Learning

- k-Nearest neighbors join

Data Preprocessing

- Polynomial Features
- Standard Scaler
- MinMax Scaler

Recommendation

- Alternating Least Squares (ALS)

Outlier selection

- Stochastic Outlier Selection (SOS)

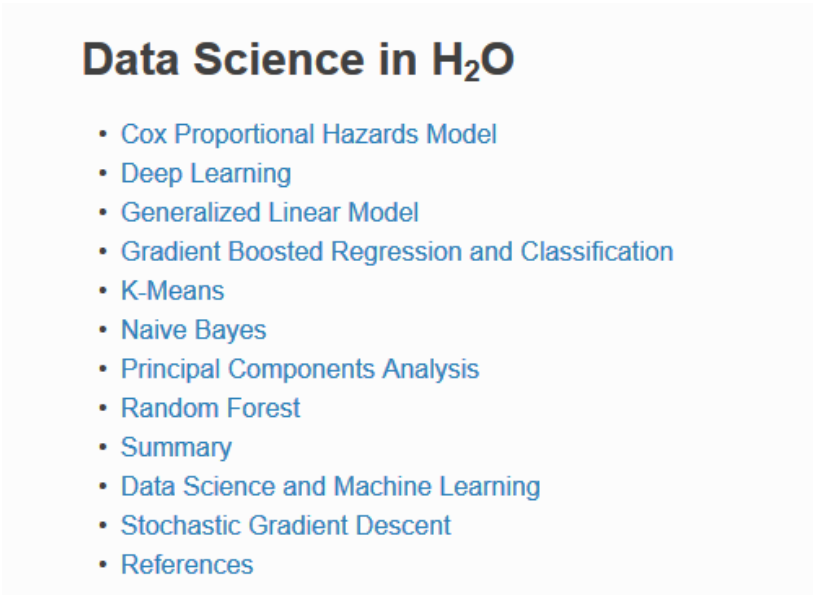
Utilities

- Distance Metrics
- Cross Validation

Librería H_2O

H_2O

H_2O **https://www.h2o.ai/**



Soporte para R, Python, Hadoop y Spark

Funcionamiento: Crea una máquina virtual con Java en la que optimiza el paralelismo de los algoritmos

Librería H_2O

H_2O

http://www.h2o.ai/resources/

H_2O APIs

Overview and walkthroughs for the different APIs to H_2O.

- R On H_2O
- Tableau on H_2O

Spark + H_2O SPARKLING WATER

http://h2o-release.s3.amazonaws.com/h2o/rel-turan/4/docs-website/h2o-r/h2o_package.pdf

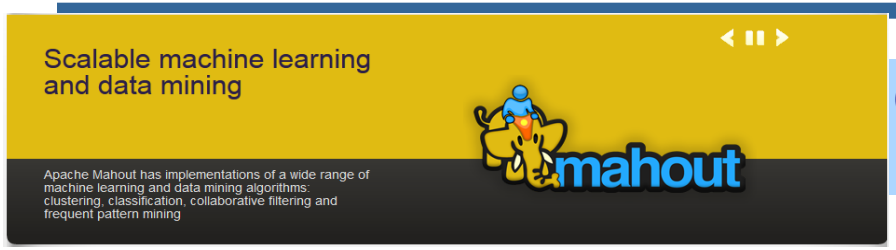
Machine Learning with Sparkling Water: H2O + Spark

Sparkling Water allows users to combine the fast, scalable machine learning algorithms of H2O with the capabilities of Spark. With Sparkling Water, users can drive computation from Scala/R/Python and utilize the H2O Flow UI, providing an ideal machine learning platform for application developers.

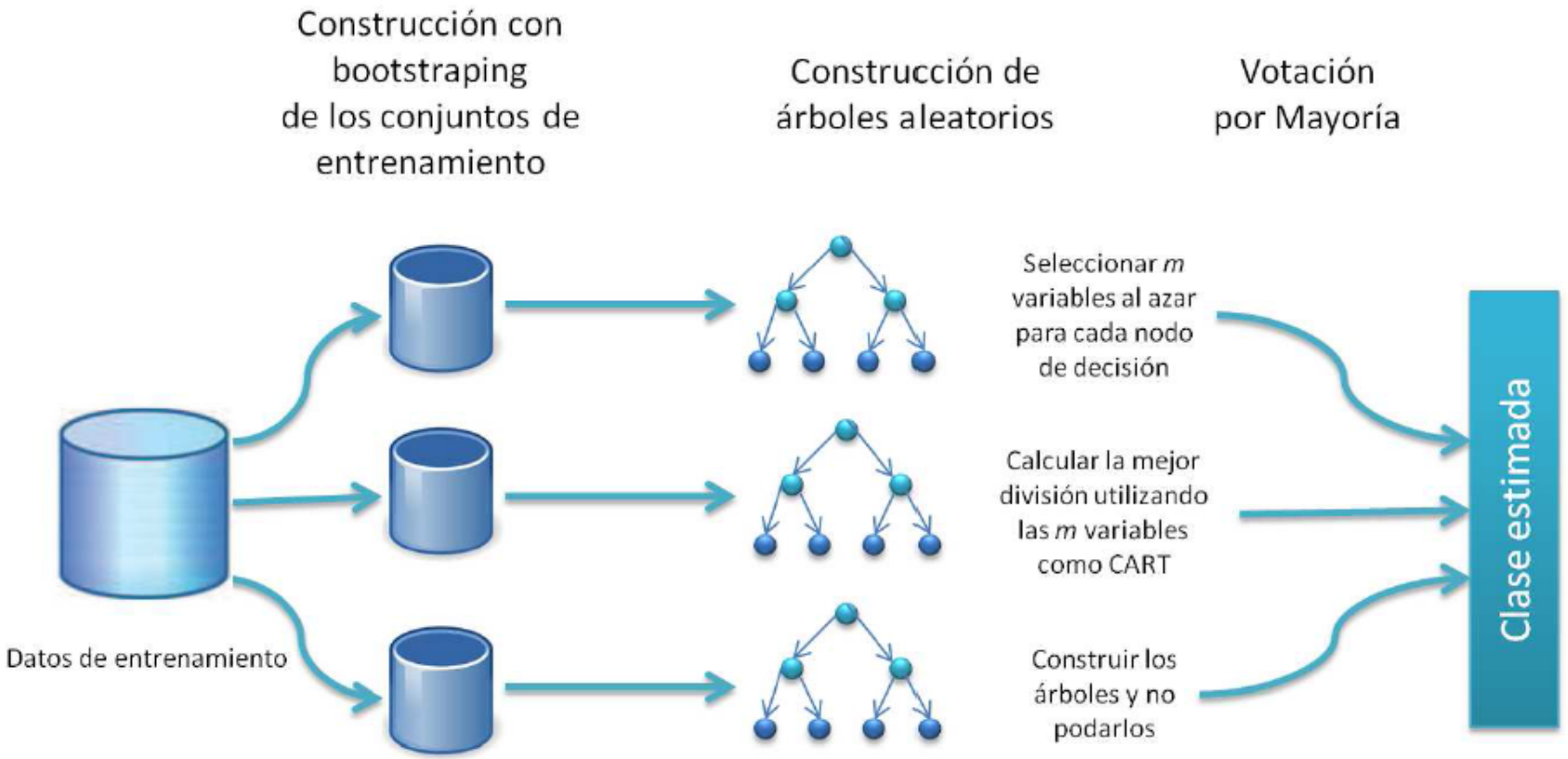
Índice

- Big Data. Big Data Science
- ¿Por qué Big Data? Google crea el Modelo de Programación MapReduce
- Tecnologías para Big Data: Ecosistema Hadoop (Hadoop, Spark, …)
- Big Data Analytics: Librerías para Analítica de Datos en Big Data.
- **Casos de estudio: Random Forest, Clustering**
- Algunas aplicaciones: Salud, Social Media, Identificación
- Imbalanced Big Data: Caso de estudio
- Comentarios Finales

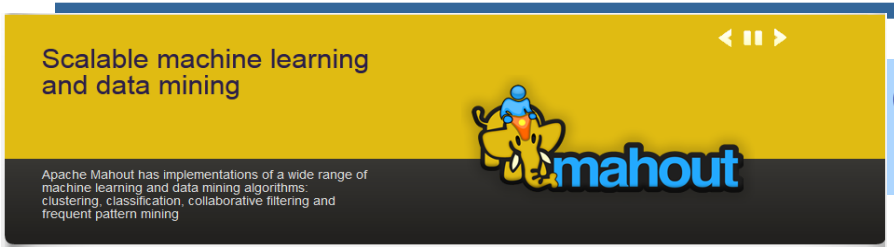
Mahout. Caso de estudio



Caso de estudio:
Random Forest para KddCup99



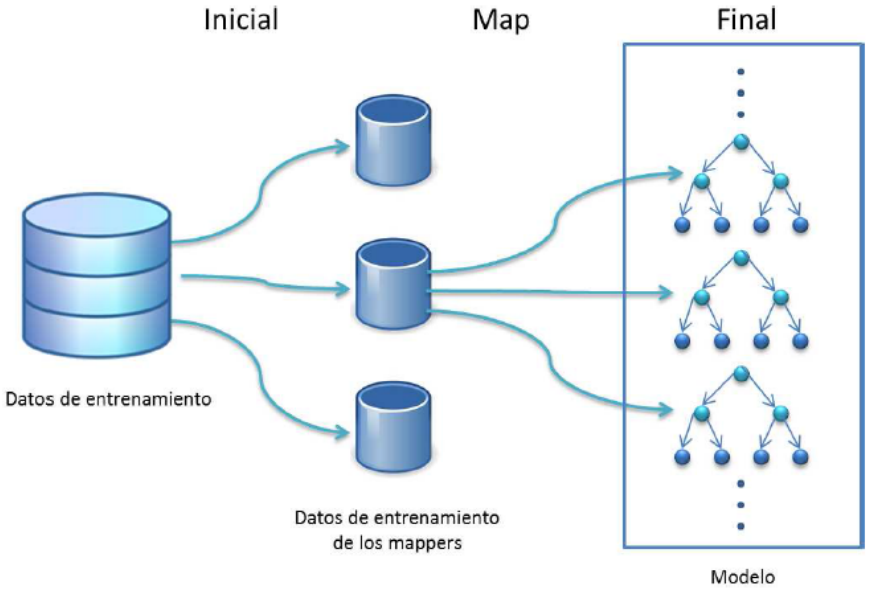
Mahout. Caso de estudio



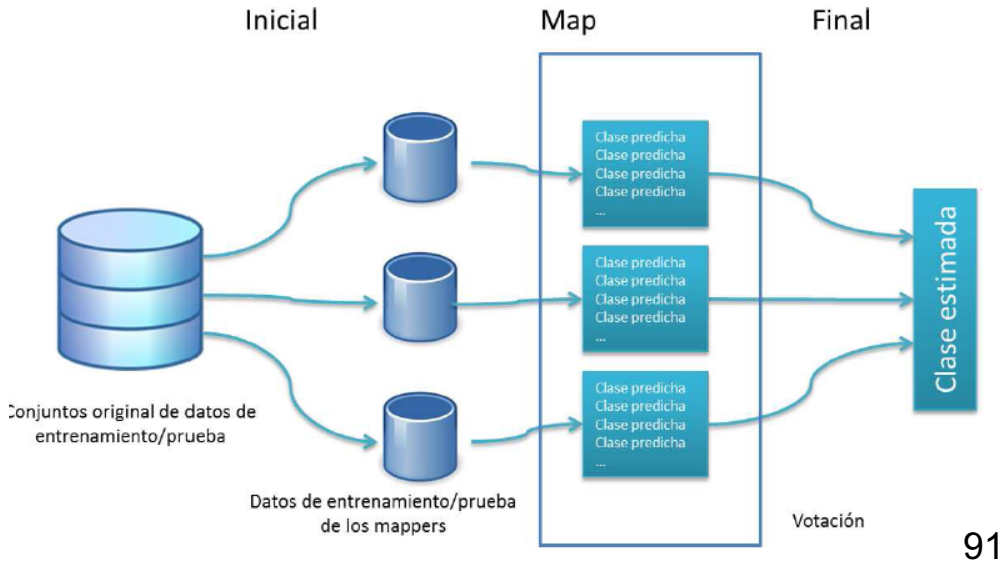
Caso de estudio:
Random Forest para KddCup99

Implementación RF Mahout Partial: Es un algoritmo que genera varios árboles de diferentes partes de los datos (maps). Dos fases:

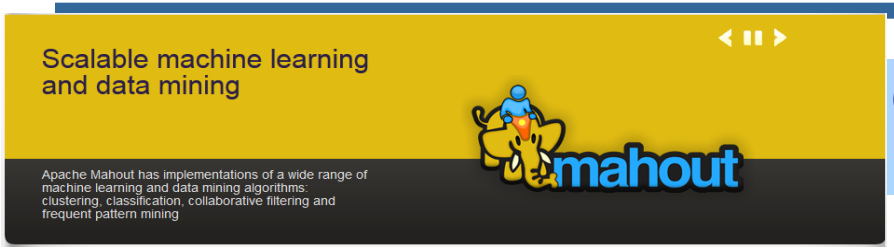
Fase de Construcción



Fase de Clasificación



Mahout. Caso de estudio



Caso de estudio:
Random Forest para KddCup99

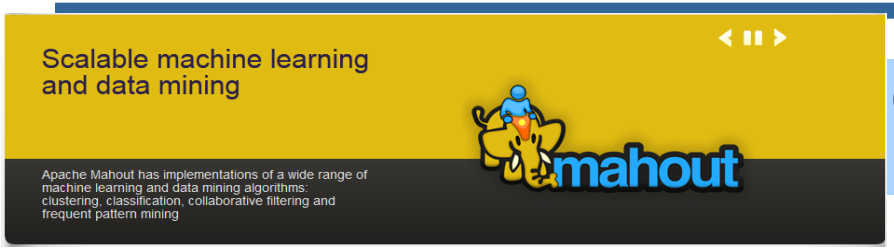
Class	Instance Number
normal	972.781
DOS	3.883.370
PRB	41.102
R2L	1.126
U2R	52

Tiempo en segundos para ejecución secuencial

Datasets	RF		
	10%	50%	full
DOS_versus_normal	6344.42	49134.78	NC
DOS_versus_PRB	4825.48	28819.03	NC
DOS_versus_R2L	4454.58	28073.79	NC
DOS_versus_U2R	3848.97	24774.03	NC
normal_versus_PRB	468.75	6011.70	NC
normal_versus_R2L	364.66	4773.09	14703.55
normal_versus_U2R	295.64	4785.66	14635.36

Cluster ATLAS: 16 nodos
-Microprocessors: 2 x Intel E5-2620 (6 cores/12 threads, 2 GHz)
- RAM 64 GB DDR3 ECC 1600MHz
- Mahout version 0.8

Mahout. Caso de estudio



Caso de estudio:
Random Forest para KddCup99

Class	Instance Number
normal	972.781
DOS	3.883.370
PRB	41.102
R2L	1.126
U2R	52

	10%	50%	full
DOS_versus_normal	6344.42	49134.78	NC
DOS_versus_PRB	4825.48	28819.03	NC

Tiempo en segundos para Big Data con 20 particiones

Datasets	RF-BigData		
	10%	50%	full
DOS_versus_normal	98	221	236
DOS_versus_PRB	100	186	190
DOS_versus_R2L	97	157	136
DOS_versus_U2R	93	134	122
normal_versus_PRB	94	58	72
normal_versus_R2L	92	39	69
normal_versus_U2R	93	52	64

Cluster ATLAS: 16 nodos
-Spark Random Forest: 43.50 seconds (20 partitions)

Aprendizaje no supervisado

Mahout

Clustering	Single Machine	MapReduce
Canopy Clustering	*deprecated*	*deprecated*
k-Means Clustering	x	x
Fuzzy k-Means	x	x
Streaming k-Means	x	x
Spectral Clustering		x

MLlib

- Frequent pattern mining
  - FP-growth

- Clustering
  - k-means
  - Gaussian mixture
  - power iteration clustering (PIC)
  - latent Dirichlet allocation (LDA)
  - streaming k-means

K-means

- **Mahout: The K-Means algorithm**
- **Input**
  - Dataset (set of points in 2D) –Large
  - Initial centroids (K points) –Small
- **Map Side**
  - Each map reads the K-centroids + one block from dataset
  - Assign each point to the closest centroid
  - Output <centroid, point>

R. M. Esteves, C. Rong, R. Pais, **K-means Clustering in the Cloud – A Mahout Test**. IEEE Workshops of International Conference on Advanced Information Networking and Applications, pp.514,519, 22-25 March 2011.

K-means

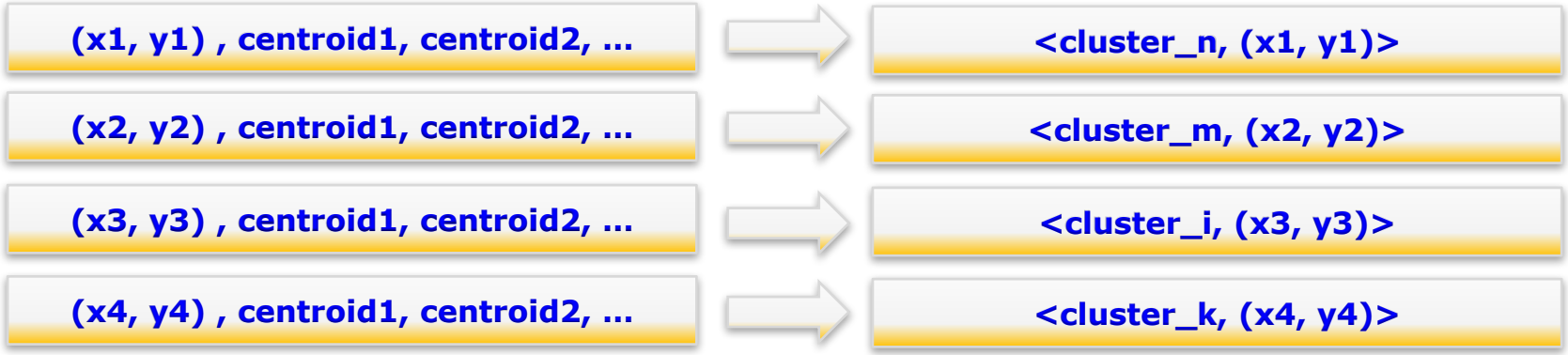
Mahout: K-means clustering

- **Reduce Side**
  - Gets all points for a given centroid
  - Re-compute a new centroid for this cluster
  - Output: <new centroid>
- **Iteration Control**
  - Compare the old and new set of K-centroids If similar or max iterations reached then Stop Else Start another Map-Reduce Iteration

- **THIS IS AN ITERATIVE MAP-REDUCE ALGORITHM**

K-means

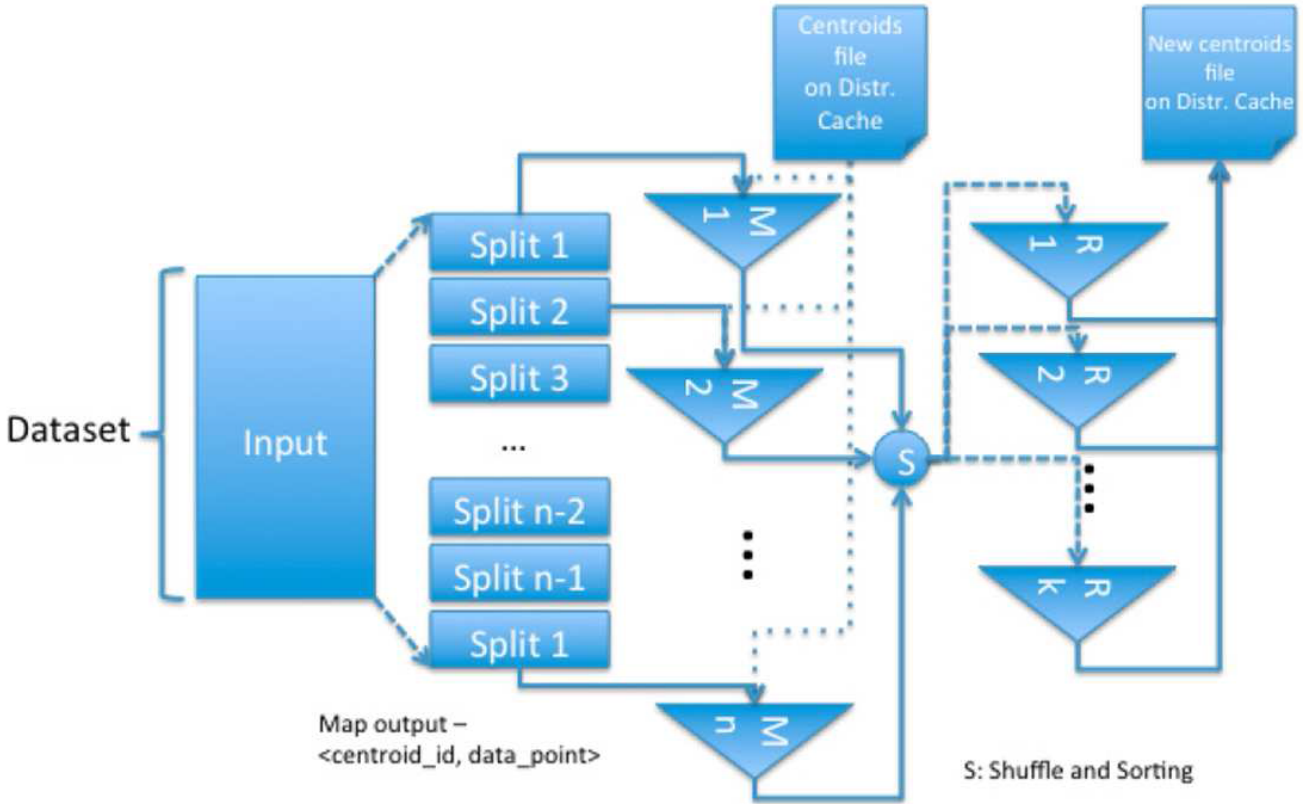
- **Map phase: assign cluster IDs**



- **Reduce phase: reset centroids**

K-means

Mahout: K-means clustering



R. M. Esteves, C. Rong, R. Pais, **K-means Clustering in the Cloud – A Mahout Test**. IEEE Workshops of International Conference on Advanced Information Networking and Applications, pp.514,519, 22-25 March 2011.

K-means

K-Means: An example of limitation of MapReduce

- **What's wrong with these iterative approaches?**
  - Iterative algorithms in MapReduce chain multiple jobs together.
  - The standard MapReduce is not ready to do this.
  - Hadoop offers some snippets (Counters) to determine the stopping criteria.

- **Main issues:**
  - MapReduce jobs have high startup costs.
  - Repetitive Shuffle.
  - Results are serialized to HDFS.

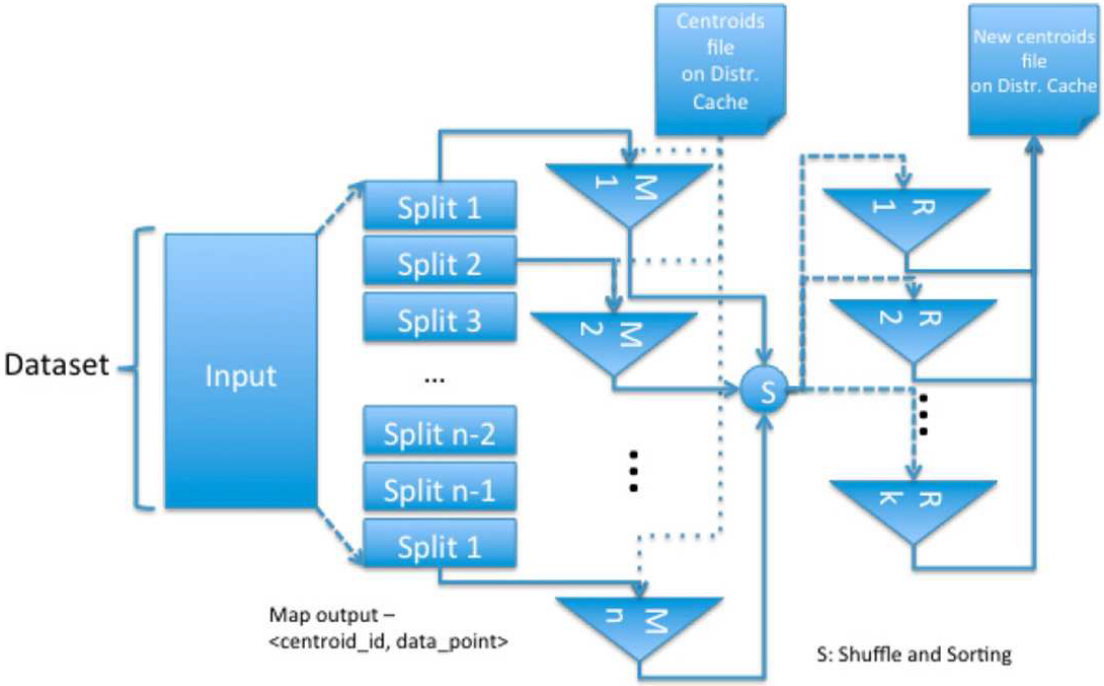
K-means

K-Means Clustering using Spark

Focus: Implementation and Performance

K-means

Mahout: K-means clustering



- Iterative algorithms in MapReduce chain multiple jobs together.
- The standard MapReduce is not ready to do this.

R. M. Esteves, C. Rong, R. Pais, **K-means Clustering in the Cloud – A Mahout Test**. IEEE Workshops of International Conference on Advanced Information Networking and Applications, pp.514,519, 22-25 March 2011.

K-means

MLlib: K-means clustering

k-means pseudo-code:

- Initialize K cluster centers
- Repeat until convergence:
  - Assign each data point to the cluster with the closest center.
  - Assign each cluster center to be the mean of its cluster's data points.

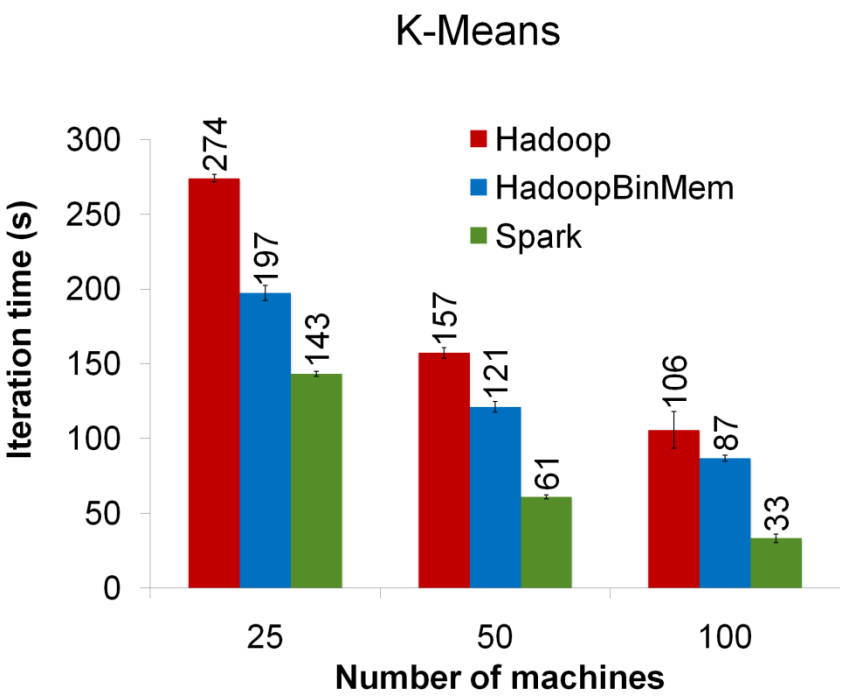
k-means MLlib source

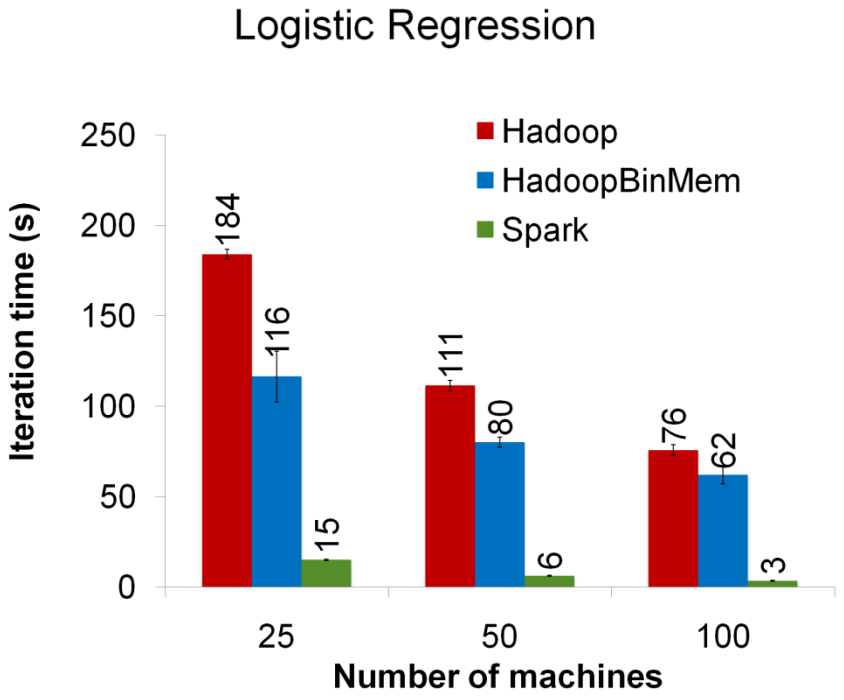
```
centers = data.takeSample(
    false, K, seed)
while (d > ε)
{
 closest = data.map(p =>
     (closestPoint(p,centers),p))
 pointsGroup =
     closest.groupByKey()
 newCenters =pointsGroup.mapValues(
     ps => average(ps))
 d = distance(centers, newCenters)
 centers = newCenters.map(_)
}
```

K-Means and Logistic Regression Performance

Spark MLlib

Performance

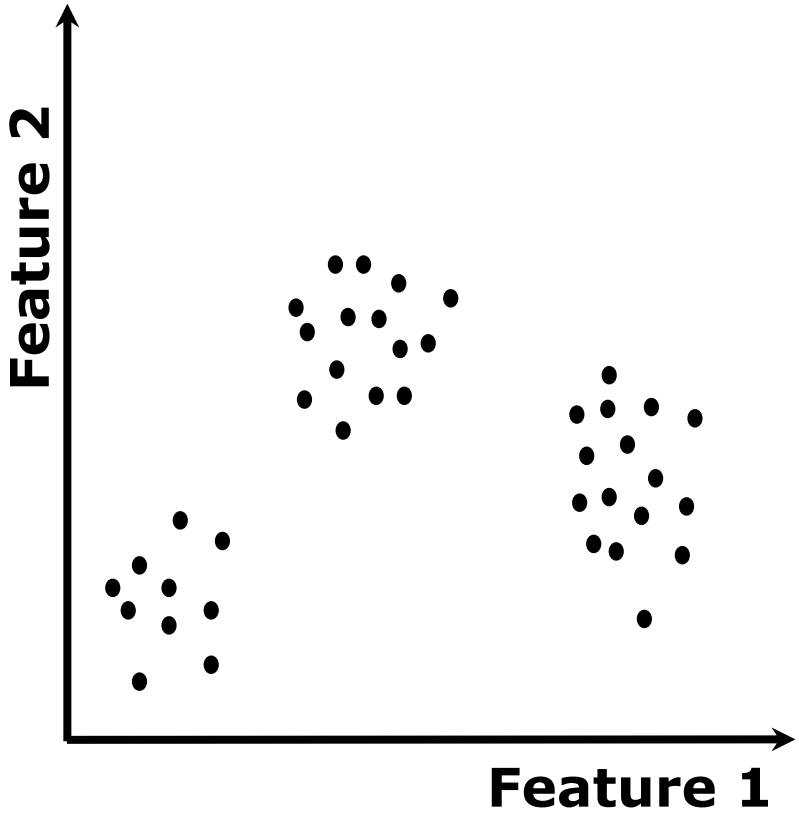




M. Zaharia et al. **Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing**. NSDI 2012.

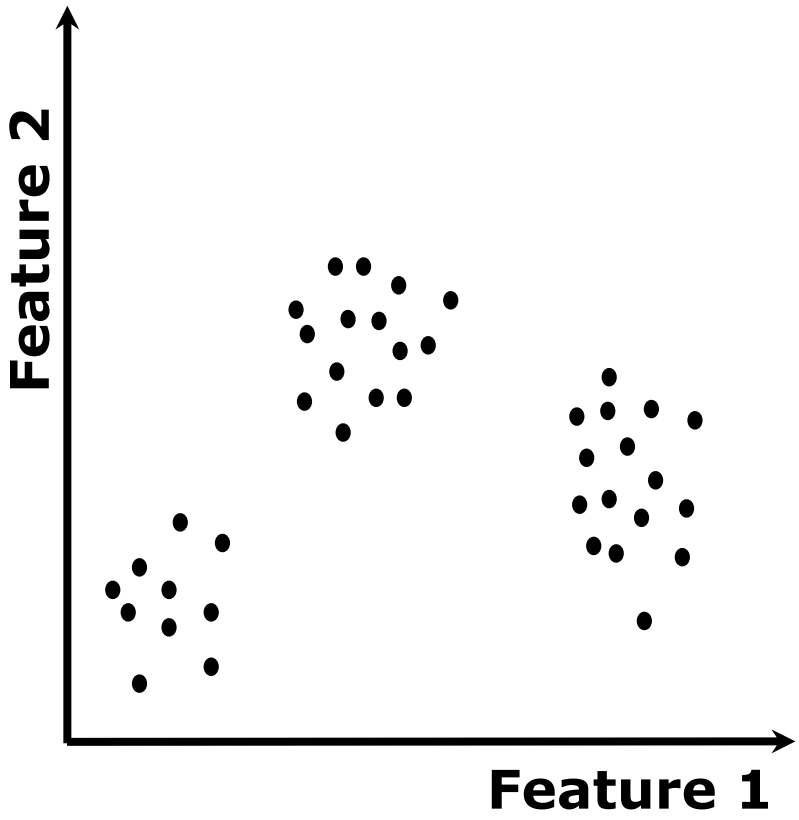
K-Means Algorithm

- Initialize K cluster centers
- Repeat until convergence:
  - Assign each data point to the cluster with the closest center.
  - Assign each cluster center to be the mean of its cluster's data points.



K-Means Algorithm

- **Initialize K cluster centers**
- **Repeat until convergence:**
  - **Assign each data point to the cluster with the closest center.**
  - **Assign each cluster center to be the mean of its cluster's data points.**

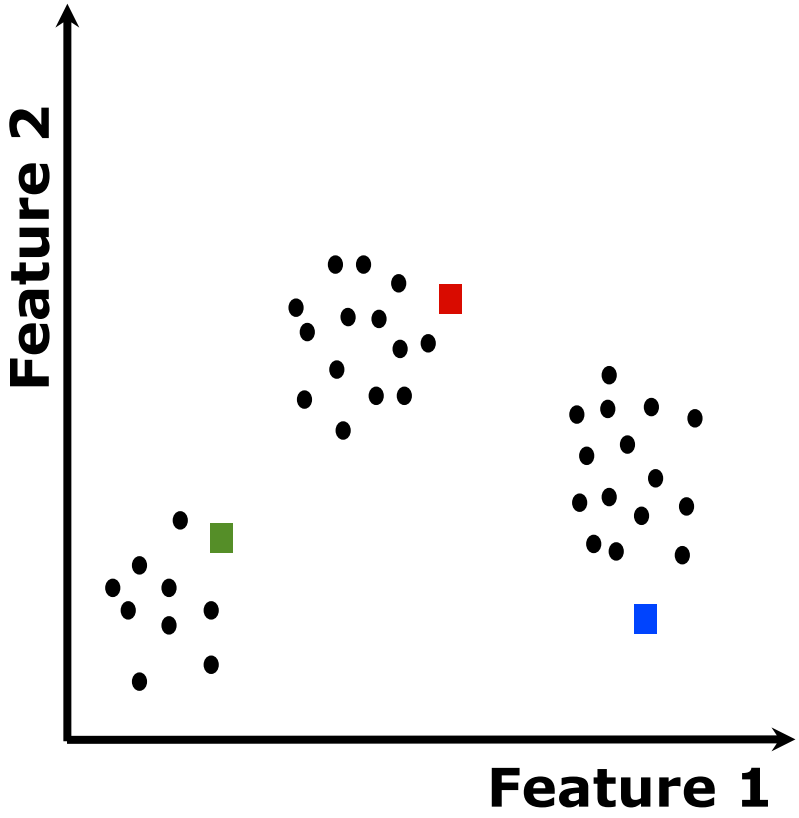


K-Means Algorithm

- **Initialize K cluster centers**

```
centers = data.takeSample(
    false, K, seed)
```

- **Repeat until convergence:**
  - **Assign each data point to the cluster with the closest center.**
  - **Assign each cluster center to be the mean of its cluster's data points.**

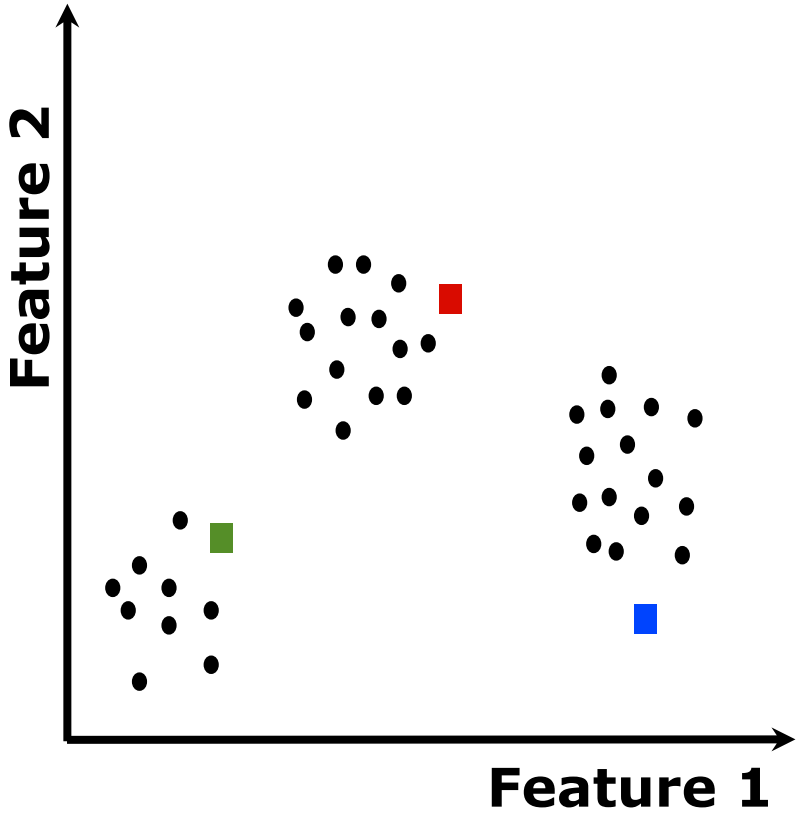


K-Means Algorithm

- Initialize K cluster centers

```
centers = data.takeSample(
    false, K, seed)
```

- **Repeat until convergence:**
  - Assign each data point to the cluster with the closest center.
  - Assign each cluster center to be the mean of its cluster's data points.

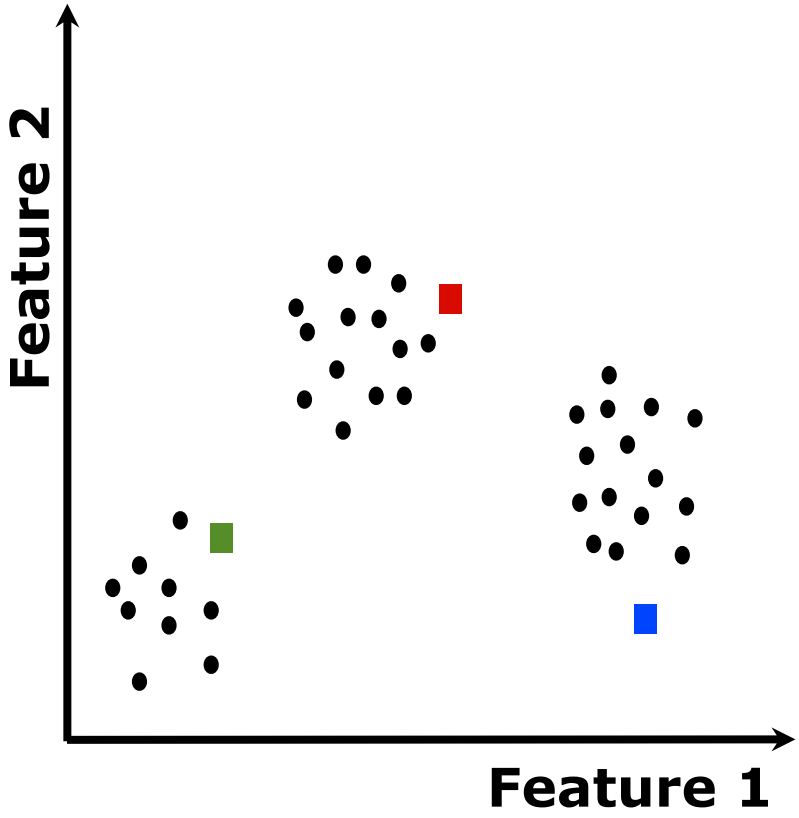


K-Means Algorithm

- Initialize K cluster centers

```
centers = data.takeSample(
    false, K, seed)
```

- Repeat until convergence:
  - **Assign each data point to the cluster with the closest center.**
  - Assign each cluster center to be the mean of its cluster's data points.



K-Means Algorithm

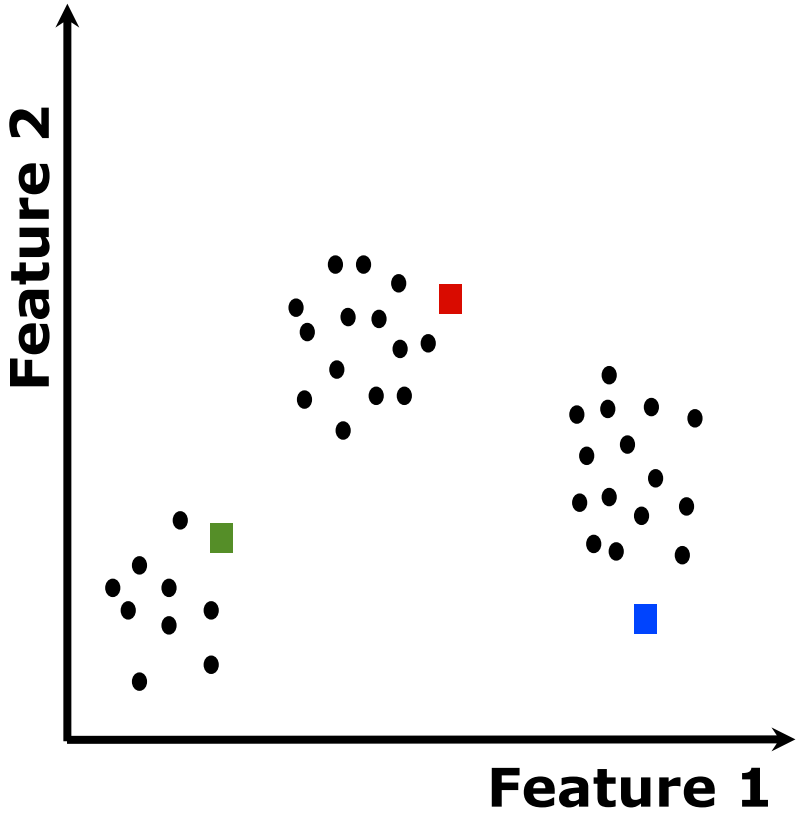
- Initialize K cluster centers

```
centers = data.takeSample(
    false, K, seed)
```

- Repeat until convergence:

```
closest = data.map(p =>

(closestPoint(p,centers),p))
```

Assign each cluster center to be the mean of its cluster's data points.



K-Means Algorithm

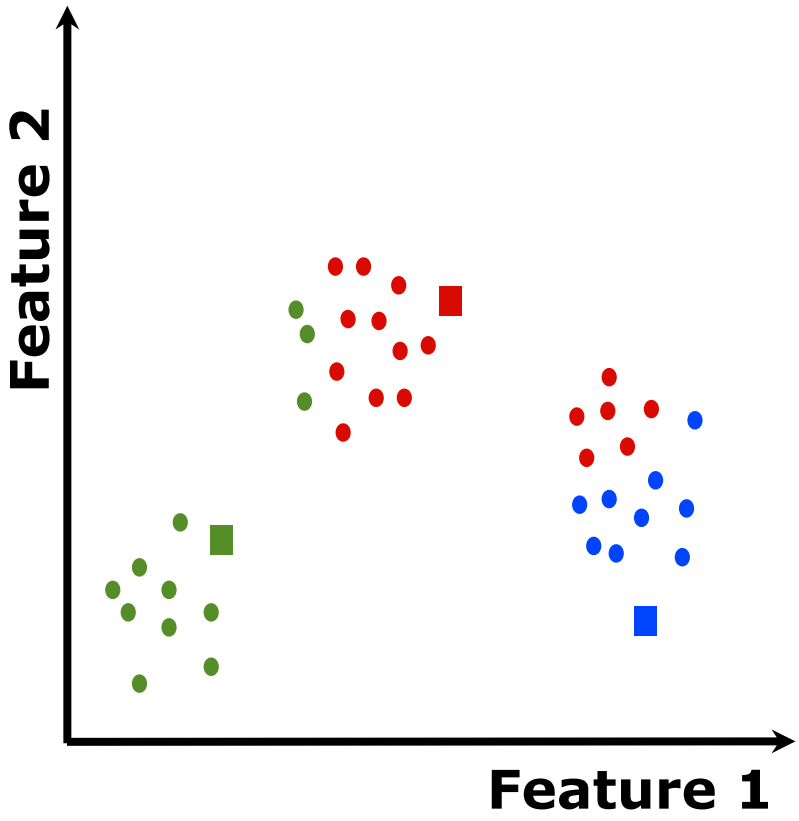
- Initialize K cluster centers

```
centers = data.takeSample(
    false, K, seed)
```

- Repeat until convergence:

```
closest = data.map(p =>

(closestPoint(p,centers),p))
```

Assign each cluster center to be the mean of its cluster's data points.



K-Means Algorithm

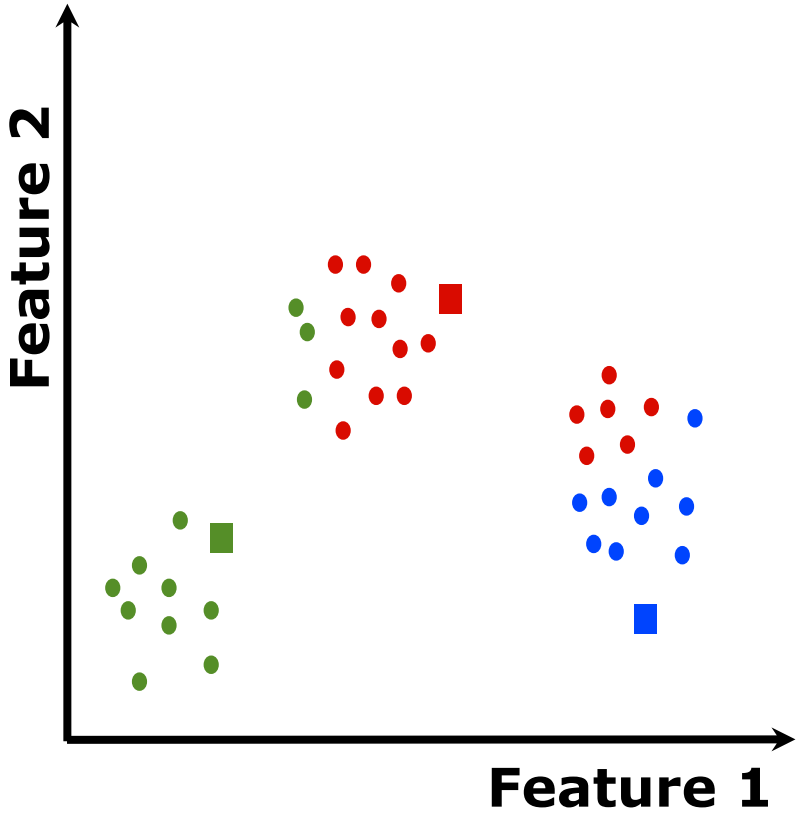
- Initialize K cluster centers

```
centers = data.takeSample(
    false, K, seed)
```

- Repeat until convergence:

```
closest = data.map(p =>

(closestPoint(p,centers),p))
```

Assign each cluster center to be the mean of its cluster's data points.



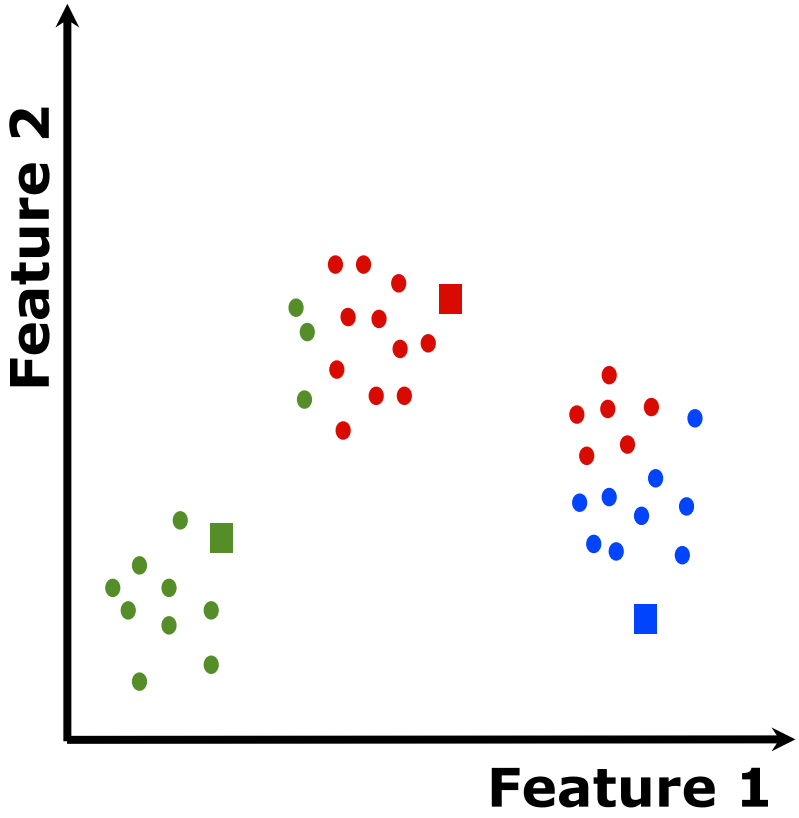
K-Means Algorithm

- Initialize K cluster centers

```
centers =data.takeSample(
    false, K, seed)
```

- Repeat until convergence:

```
closest = data.map(p =>

(closestPoint(p,centers),p))

pointsGroup =
    closest.groupByKey()
```



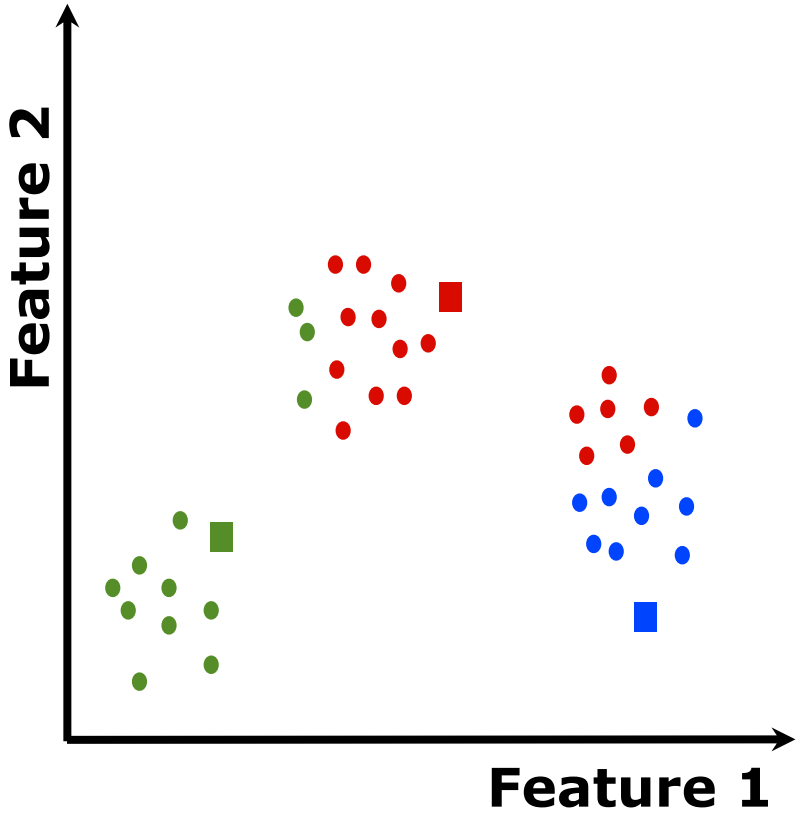
K-Means Algorithm

- Initialize K cluster centers

```
centers = data.takeSample(
    false, K, seed)
```

- Repeat until convergence:

```
closest = data.map(p =>

 (closestPoint(p,centers),p))
pointsGroup =
    closest.groupByKey()

newCenters =
pointsGroup.mapValues(
    ps => average(ps))
```



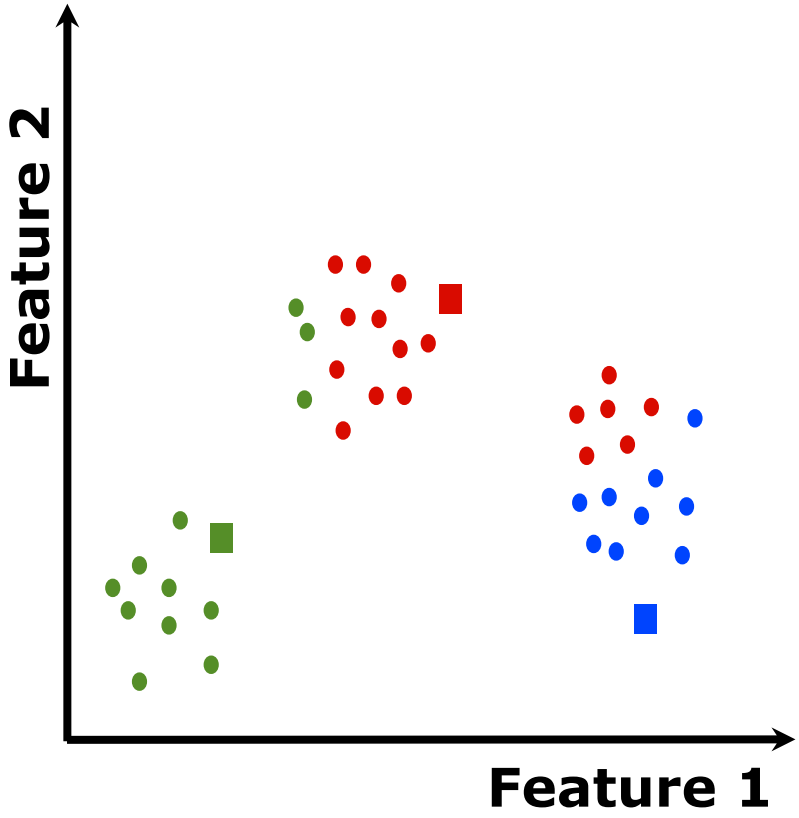
K-Means Algorithm

- Initialize K cluster centers

```
centers = data.takeSample(
    false, K, seed)
```

- Repeat until convergence:

```
closest = data.map(p =>

(closestPoint(p,centers),p))

pointsGroup =
    closest.groupByKey()

newCenters =
pointsGroup.mapValues(
    ps => average(ps))
```



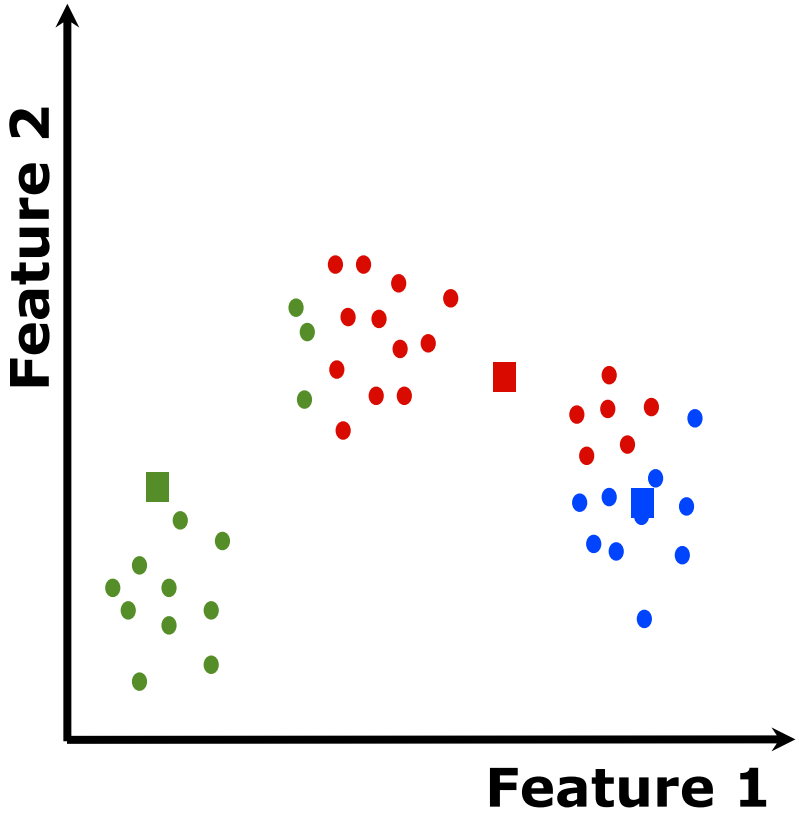
K-Means Algorithm (proceso interativo utilizando los bloques en memoria)

- Initialize K cluster centers

```
centers = data.takeSample(
    false, K, seed)
```

- **Repeat until convergence:**

```
closest = data.map(p =>

(closestPoint(p,centers),p))

pointsGroup =
    closest.groupByKey()

newCenters =
pointsGroup.mapValues(
    ps => average(ps))
```



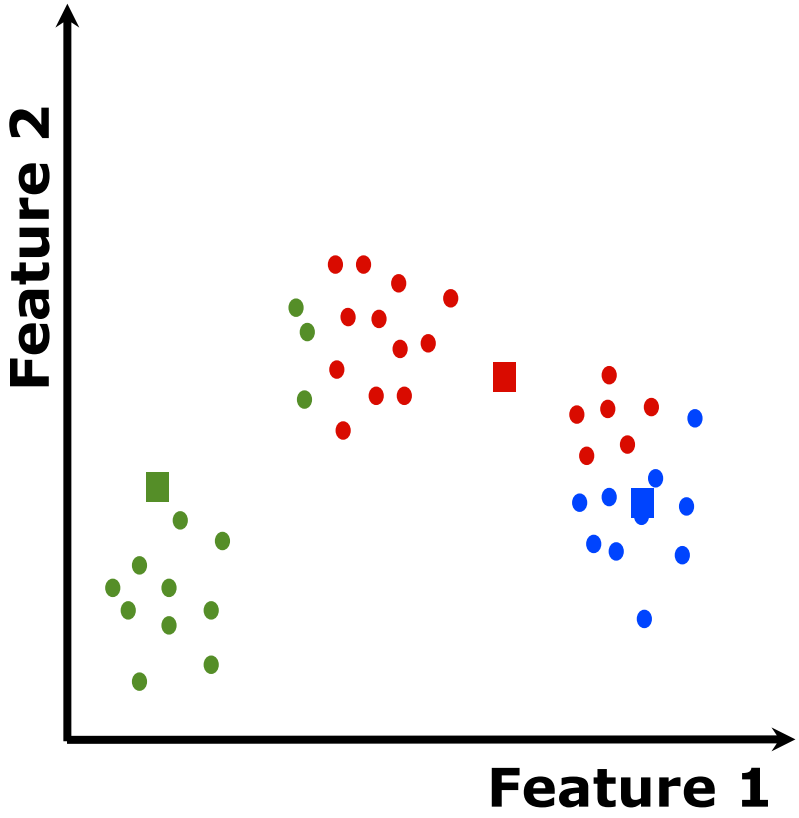
K-Means Algorithm

- Initialize K cluster centers

```
centers = data.takeSample(
    false, K, seed)
```

- **Repeat until convergence:**

```
while (dist(centers, newCenters)
> ε)

closest = data.map(p =>

(closestPoint(p,centers),p))

pointsGroup =
    closest.groupByKey()

newCenters
=pointsGroup.mapValues(
    ps => average(ps))
```



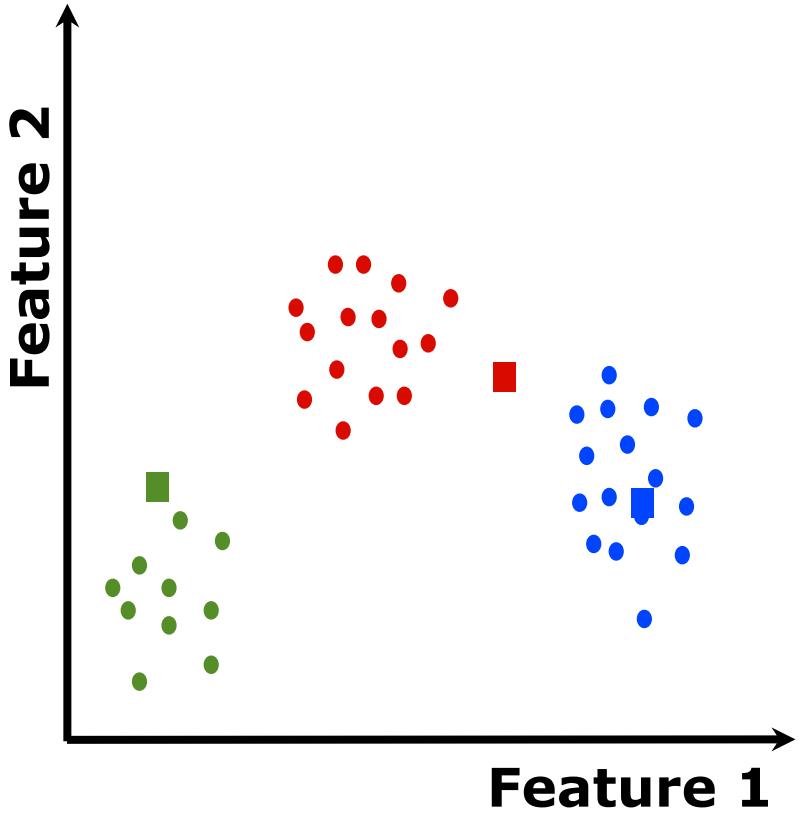
K-Means Algorithm

- Initialize K cluster centers

```
centers = data.takeSample(
    false, K, seed)
```

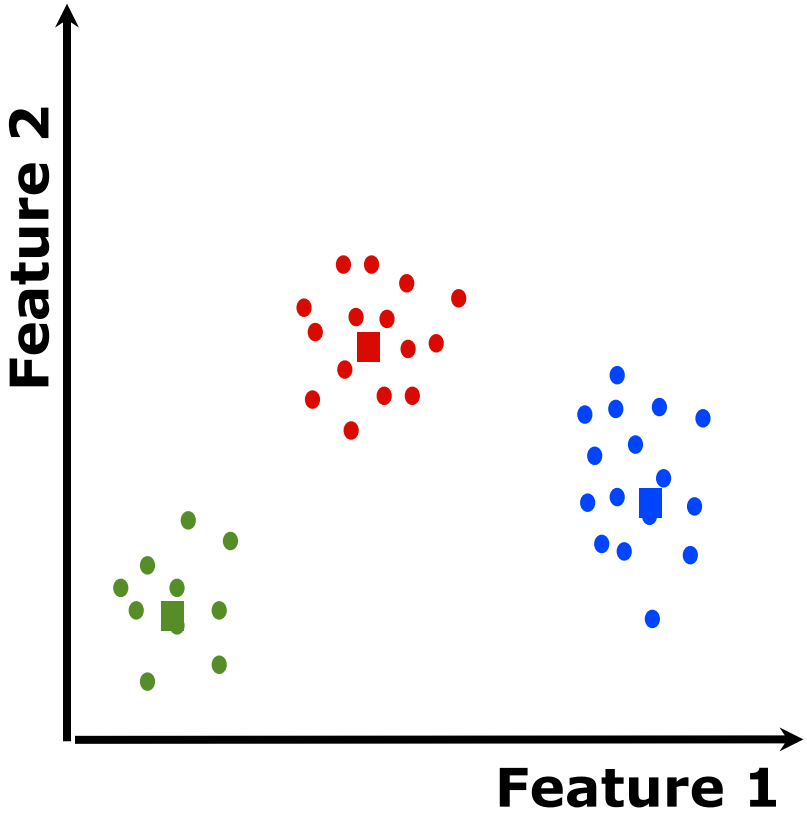
- **Repeat until convergence:**

```
while (dist(centers, newCenters)
> ε)

closest = data.map(p =>

(closestPoint(p,centers),p))

pointsGroup =
    closest.groupByKey()

newCenters
=pointsGroup.mapValues(
    ps => average(ps))
```



K-Means Source

```
centers = data.takeSample(
    false, K, seed)

while (d > ε)
{
  closest = data.map(p =>

   (closestPoint(p,centers),p))

  pointsGroup =
      closest.groupByKey()

  d = distance(centers, newCenters)

  newCenters
  =pointsGroup.mapValues(
      ps => average(ps))

  centers = newCenters.map(_)

 }
```

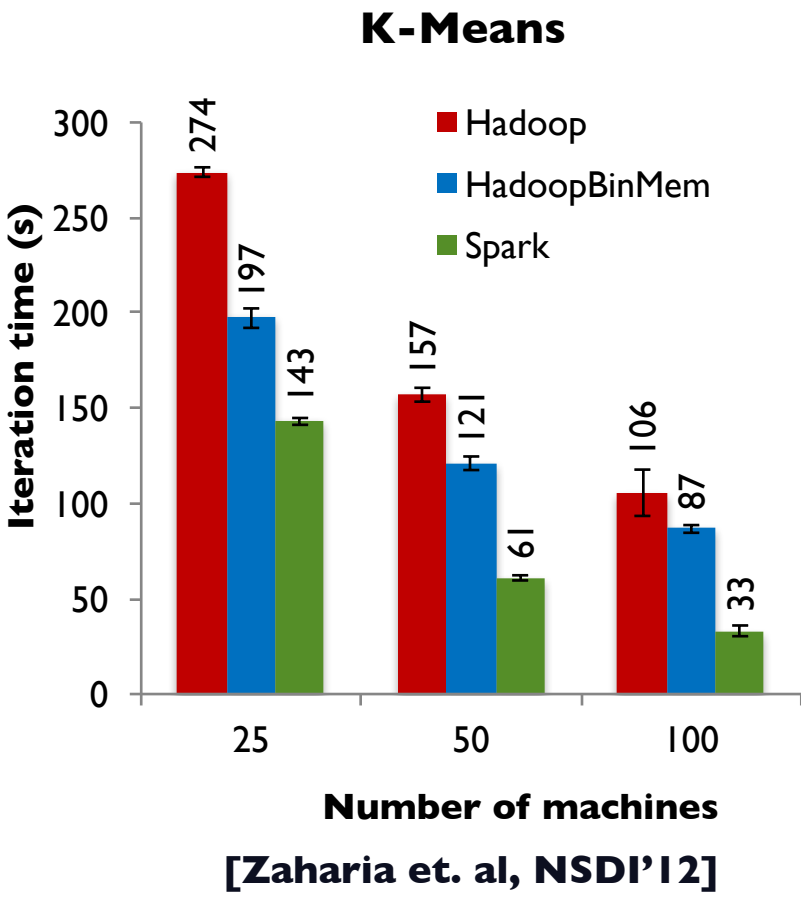


K-Means Performance

Lines of code for K-Means

Spark ~ 90 lines –

Hadoop ~ 4 files, > 300 lines



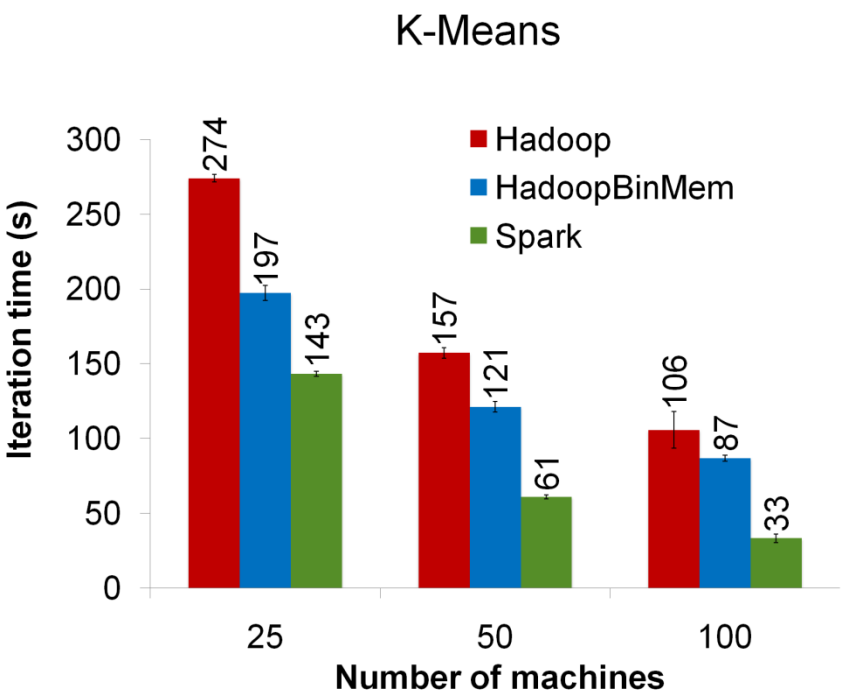
[Zaharia et. al, NSDI'12]

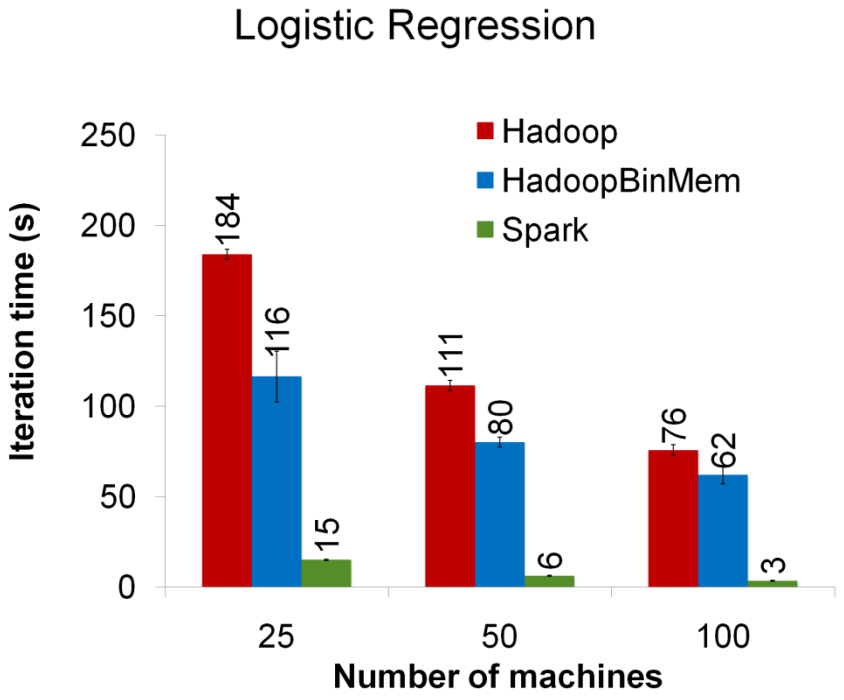
M. Zaharia et al. **Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing**. NSDI 2012.

K-Means and Logistic Regression Performance

Spark MLlib

Performance





M. Zaharia et al. **Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing**. NSDI 2012.

Unsupervised Learning

MLlib: FP-growth

Mllib implements a parallel version of FP-growth called PFP, as described in

Li et al., PFP: Parallel FP-growth for query recommendation. RecSys'08 Proceedings of the 2008 ACM conference on Recommender systems Pages 107-114

PFP distributes the work of growing FP-trees based on the suffices of transactions, and hence more scalable than a single-machine implementation.

Big Data Analytics: 2 Comentarios finales

Image Credit: Shutterstock

Without Analytics, Big Data is Just Noise

Guest post by Eric Schwartzman, founder and CEO of Comply Socially

http://www.briansolis.com/2013/04/without-analytics-big-data-is-just-noise/

Big Data Analytics: 2 Comentarios finales

Mahout

Classification	Single Machine	MapReduce
Logistic Regression - trained via SGD	x	
Naive Bayes / Complementary Naive Bayes		x
Random Forest		x
Hidden Markov Models	x	
Multilayer Perceptron	x	

Casi ausencia de algoritmos para big data preprocessing

MLlib

MLlib types, algorithms and utilities

This lists functionality included in `spark.mllib`, the main MLlib API.

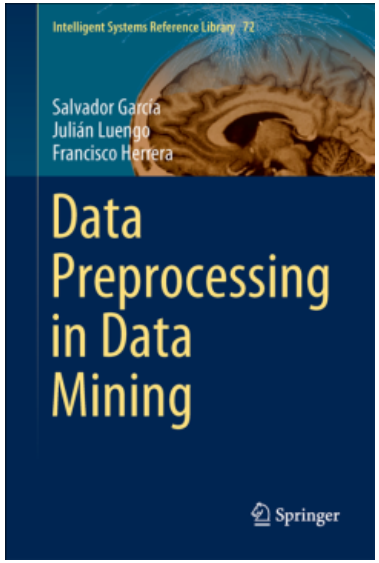
- Data types
- Basic statistics
  - summary statistics
  - correlations
  - stratified sampling
  - hypothesis testing
  - random data generation
- Classification and regression
  - linear models (SVMs, logistic regression, linear regression)
  - naive Bayes
  - decision trees
  - ensembles of trees (Random Forests and Gradient-Boosted Trees)
  - isotonic regression
- Collaborative filtering
  - alternating least squares (ALS)
- Clustering
  - k-means
  - Gaussian mixture
  - power iteration clustering (PIC)
  - latent Dirichlet allocation (LDA)
  - streaming k-means
- Dimensionality reduction
  - singular value decomposition (SVD)
  - principal component analysis (PCA)
- Feature extraction and transformation
- Frequent pattern mining
  - FP-growth
- Optimization (developer)
  - stochastic gradient descent
  - limited-memory BFGS (L-BFGS)
- PMML model export

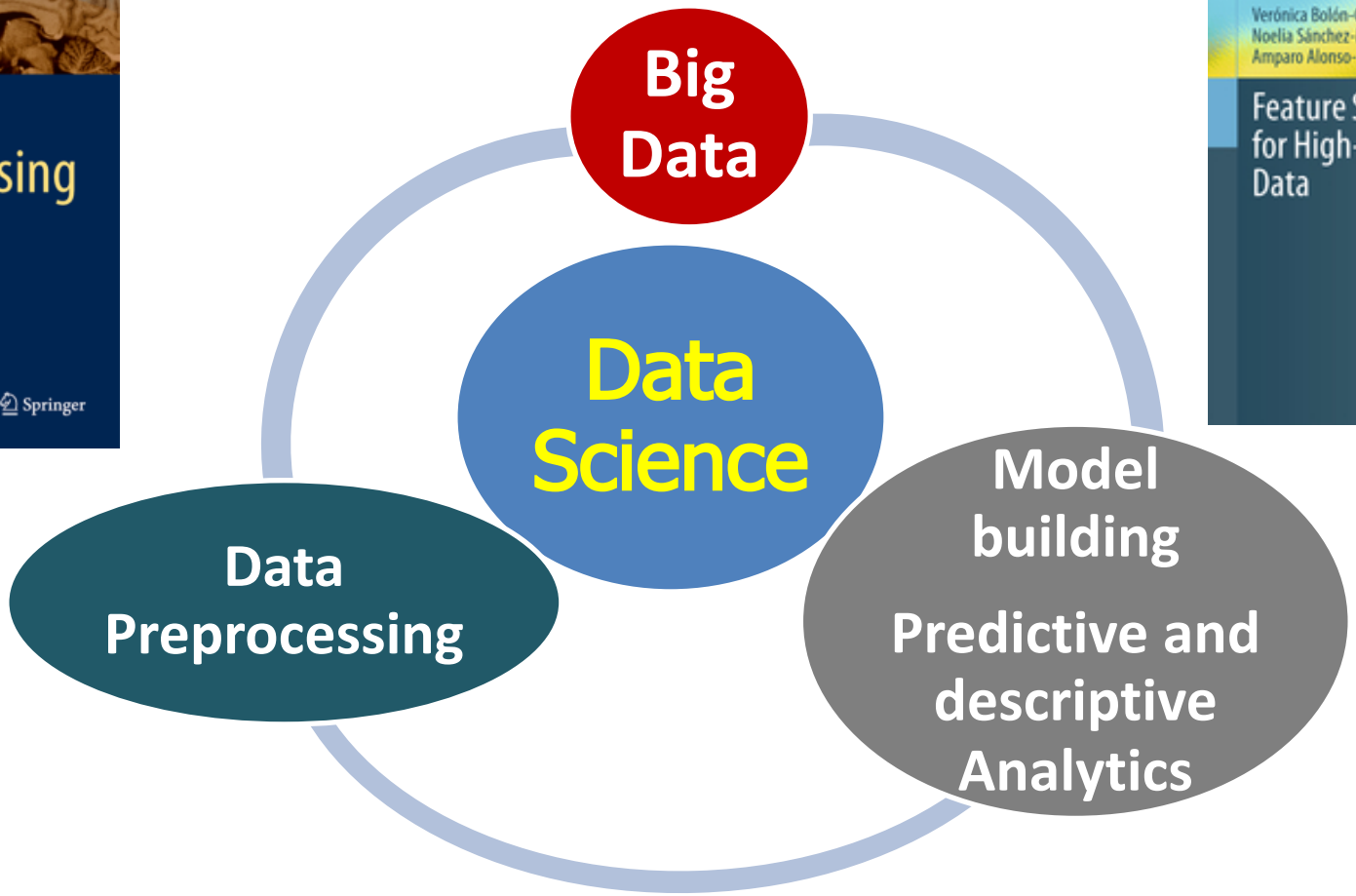
https://spark.apache.org/mllib/
Version 1.4.1

Big Data Analytics:
Big Data Preprocessing

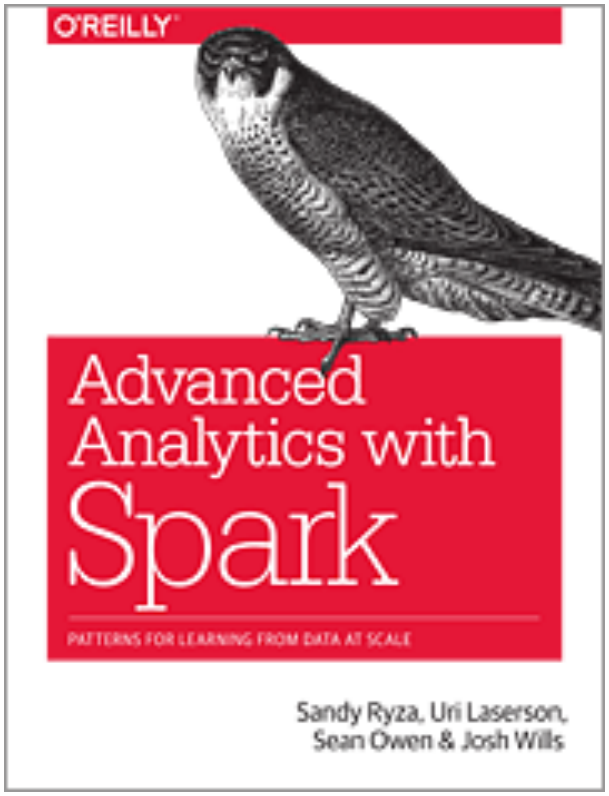
BIG DATA Preprocessing

¡Se requieren datos de calidad para diseñar modelos de calidad!.

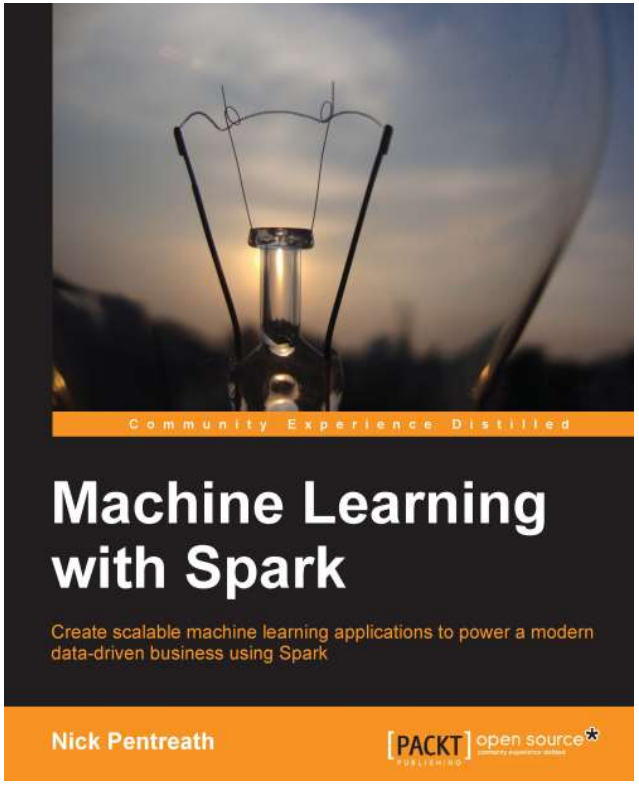




Big Data Analytics: 2 libros

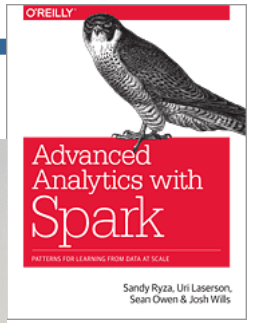


9 cases of study

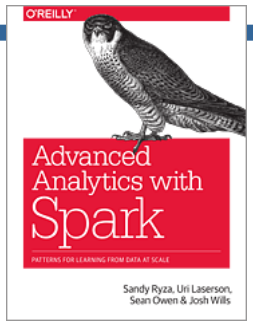


10 chapters giving a quick glance on Machine Learning with Spark

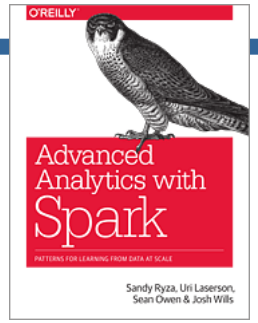
Big Data Analytics: Introducción



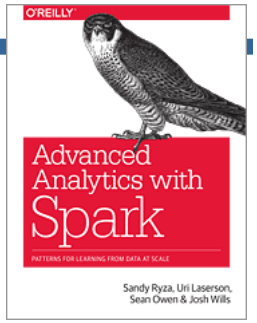
Big Data Analytics: Casos de estudio



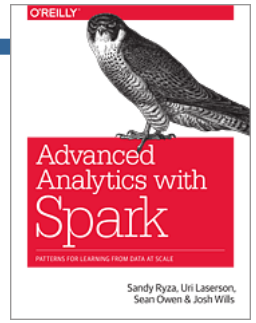
Big Data Analytics: Casos de estudio



Big Data Analytics: Casos de estudio



Big Data Analytics: Casos de estudio



Índice

- Big Data. Big Data Science
- ¿Por qué Big Data? Google crea el Modelo de Programación MapReduce
- Tecnologías para Big Data: Ecosistema Hadoop (Hadoop, Spark, …)
- Big Data Analytics: Librerías para Analítica de Datos en Big Data.
- Casos de estudio: Random Forest, Clustering
- **Algunas aplicaciones: Salud, Social Media, Identificación**
- Big Data en el grupo de investigación SCI^{2}S
- Comentarios Finales

Big Data: Google Flu

Aplicación de Google - 2009

Google Flu

Detect pandemic risk in real time

- 2009, nuevo virus gripe A: cepa H1N1
- Sanidad pública temía una pandemia similar a la de la gripe española de 1918
  - 500 millones de afectados
  - Decenas de millones de fallecidos
- No hay vacuna, hay que ralentizar la propagación
- Solución:
  - Los centros de control y prevención de enfermedades (CDC) recopilan datos de los médicos
  - ¡Se consigue un panorama de la pandemia con un desfase, retraso de 2 semanas!

Big Data: Google Flu

Google Flu

Detect pandemic risk in real time

"Google puede predecir la propagación de la gripe (…) analizando lo que la gente busca en internet"

+ de 3.000M de búsquedas a diario

J. Ginsberg, M.H. Mohebbi, R.S. Patel, L. Brammer, M.S. Smolinski, L. Brilliant.
Detecting influenza epidemics using search engine query data.
Nature 475 (2009) 1012-1014

Big Data: Google Flu

Google Flu

Detect pandemic risk in real time

"Google puede predecir la propagación de la gripe (...) analizando lo que la gente busca en internet"

- Google utilizó:
  - 50 M de términos de búsqueda más utilizados
  - Comparó esta lista con los datos de los CDC sobre propagación de gripe entre 2003 y 2008
    - Identificar a los afectados en base a sus búsquedas
    - Buscaron correlaciones entre frecuencia de búsquedas de información y propagación de la gripe en tiempo y espacio

Big Data: Google Flu

Google Flu

Detect pandemic risk in real time

- Encontraron una combinación de 45 términos de búsqueda que al usarse con un modelo matemático presentaba una correlación fuerte entre su predicción y las cifras oficiales de la enfermedad

Podían decir, como los CDC, a dónde se había propagado la gripe pero casi en tiempo real, no una o dos semanas después

Con un método basado en Big Data

- Se ha extendido a 29 países

Big Data: Google Flu

Google Flu

Detect pandemic risk in real time

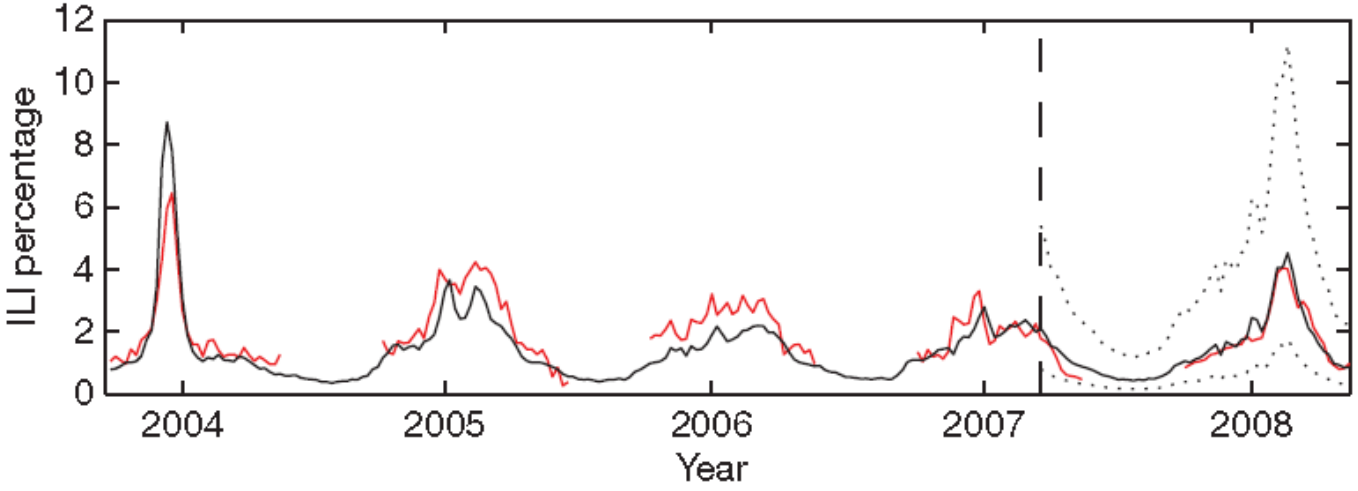


Figure 2 | A comparison of model estimates for the mid-Atlantic region (black) against CDC-reported ILI percentages (red), including points over which the model was fit and validated. A correlation of 0.85 was obtained over 128 points from this region to which the model was fit, whereas a correlation of 0.96 was obtained over 42 validation points. Dotted lines indicate 95% prediction intervals. The region comprises New York, New Jersey and Pennsylvania.

J. Ginsberg, M.H. Mohebbi, R.S. Patel, L. Brammer, M.S. Smolinski, L. Brilliant. Detecting influenza epidemics using search engine query data. Nature 475 (2009) 1012-1014

Big Data: Google Flu

Google Flu

Detect pandemic risk in real time

- En 2013 sobreestimó los niveles de gripe (x2 la estimación CDC)
  - La sobreestimación puede deberse a la amplia cobertura mediática de la gripe que puede modificar comportamientos de búsqueda
  - Los modelos se van actualizando anualmente

POLICYFORUM

BIG DATA

The Parable of Google Flu: Traps in Big Data Analysis

David Lazer,[1,2]* Ryan Kennedy,[1,3]* Gary King,[2] Alessandro Vespignani[3,4]

Large errors in flu prediction were largely avoidable, which offers lessons for the use of big data.

www.sciencemag.org SCIENCE VOL 343 14 MARCH 2014

Big Data: Google Flu

Google Flu

https://www.google.org/flutrends/about/

El modelo se ha actualizado hasta 2014.

El modelo se ha extendido al análisis del Dengue:

Nota reciente:

Google

Thank you for stopping by.

Google Flu Trends and Google Dengue Trends are no longer publishing current estimates of Flu and Dengue fever based on search patterns. The historic estimates produced by Google Flu Trends and Google Dengue Trends are available below. It is still early days for nowcasting and similar tools for understanding the spread of diseases like flu and dengue – we're excited to see what comes next. Academic research groups interested in working with us should fill out this form.

Sincerely,

The Google Flu and Dengue Trends Team.

Presente y futuro de big data analytics
Éxitos y limitaciones de una disciplina todavía muy joven

Big Data: Google Applications

MapReduce inside Google

Google

Googlers' hammer for 80% of our data crunching

- Large-scale web search indexing
- Clustering problems for Google News
- Produce reports for popular queries, e.g. Google Trend
- Processing of satellite imagery data
- Language model processing for statistical machine translation
- Large-scale machine learning problems
- Just a plain tool to reliably spawn large number of tasks
  - e.g. parallel data backup and restore

Algunas aplicaciones: Salud

HEALTHCARE ORGANIZATIONS HAVE COLLECTED EXABYTES OF DATA LAST YEARS

BIG DATA AND DISEASE PREVENTION:

From Quantified Self to Quantified Communities

Meredith A. Barrett,[1,2] Olivier Humblet,[1,2*] Robert A. Hiatt,[3] and Nancy E. Adler[1]*

Big data can facilitate action on the modifiable risk factors that contribute to a large fraction of the chronic disease burden, such as physical activity, diet, tobacco use, and exposure to pollution.

Big Data, Vol.1, 3, 2013

Wikipedia y la detección de la gripe

Wikipedia is better than Google at tracking flu trends?

Detect pandemic risk in real time

Wikipedia traffic could be used to provide real time tracking of flu cases, according to the study published by John Brownstein.

Wikipedia Usage Estimates Prevalence of Influenza-Like Illness in the United States in Near Real-Time
David J. McIver , John S. Brownstein**,** Harvard Medical School, Boston Children's Hospital, Plos Computational Biology, 2014.

Redes sociales, big data y medidas de salud pública

Studies: Health, Social Media and Big Data

You Are What You Tweet: Analyzing Twitter for Public Health

Discovering Health Topics in Social Media Using Topic Models
Michael J. Paul, Mark Dredze, Johns Hopkins University, Plos One, 2014

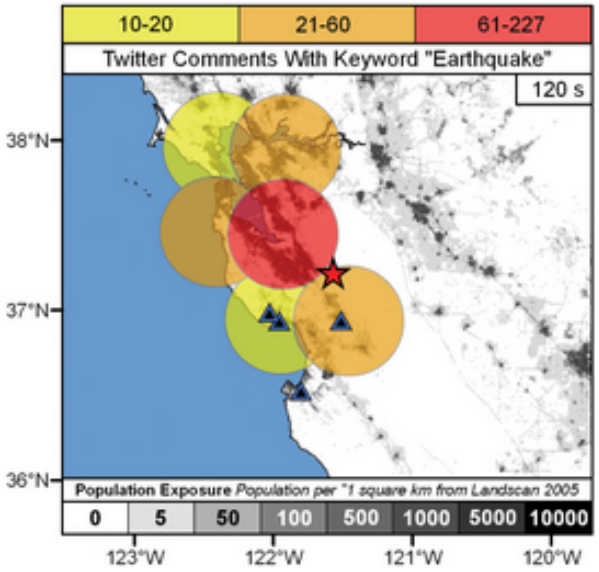
Se obtienen 13 grupos coherentes de mensajes correlacionados

- **Gripe estacional (r= 0.689) y alergias (r = 0.810)**
- **Ejercicio y obesidad relacionados con datos geográficos, ..**

Algunas aplicaciones

Earthquakes and Social Media

U.S. Geological Survey: Twitter Earthquake Detector (TED)



The United States Geological Survey Twitter searches increases in the volume of messages on earthquake and has been able to locate earthquakes with 90% accuracy.

Algunas aplicaciones: La banca es un ámbito de aplicación importante

Expansión.com
Domingo, 08.12.13. Actualizado a las 11:02

BBVA da un paso al frente en 'big data'

El banco ha actualizado todo su sistema de almacenamiento de datos corporativo para tener acceso desde un único punto a todas las 'islas de información' de la entidad en el mundo y consolidarse como un referente mundial en innovación tecnológica.

BBVA

Innova Challenge API

Available statistics services

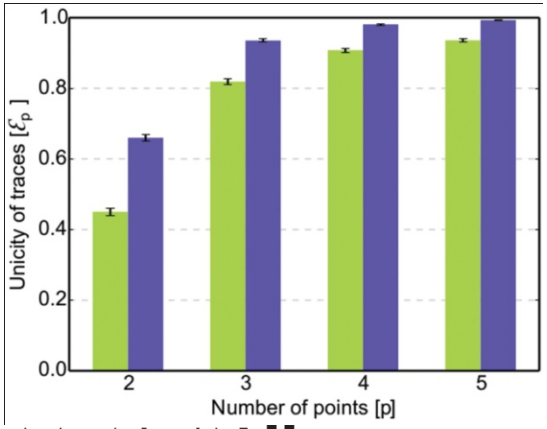
Algunas aplicaciones

Science
AAAS

PRIVACIDAD EN INTERNET »

Cuatro compras con la tarjeta bastan para identificar a cualquier persona

- Los patrones de uso de las tarjetas permiten descubrir la identidad del 90% de una muestra de 1,1 millones de personas anónimas, según demuestra un estudio del MIT

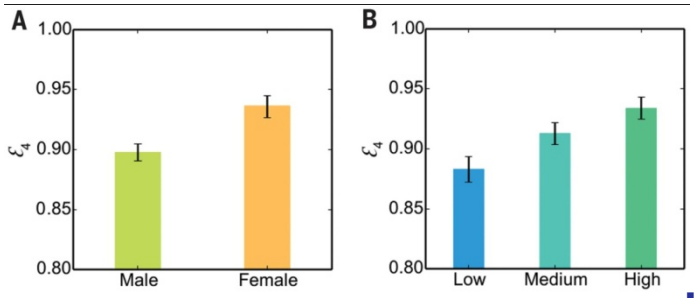


IDENTITY AND PRIVACY

Unique in the shopping mall: On the reidentifiability of credit card metadata

Yves-Alexandre de Montjoye,[1]* Laura Radaelli,[2] Vivek Kumar Singh,[1,3] Alex "Sandy" Pentland[1]

Large-scale data sets of human behavior have the potential to fundamentally transform the way we fight diseases, design cities, or perform research. Metadata, however, contain sensitive information. Understanding the privacy of these data sets is key to their broad use and, ultimately, their impact. We study 3 months of credit card records for 1.1 million people and show that four spatiotemporal points are enough to uniquely reidentify 90% of individuals. We show that knowing the price of a transaction increases the risk of reidentification by 22%, on average. Finally, we show that even data sets that provide coarse information at any or all of the dimensions provide little anonymity and that women are more reidentifiable than men in credit card metadata.



http://www.sciencemag.org/content/347/6221/536

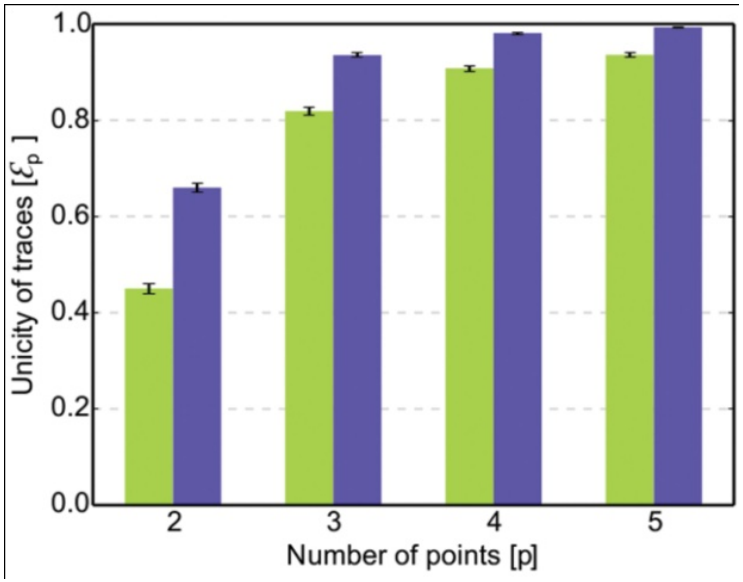
http://elpais.com/elpais/2015/01/29/ciencia/1422520042_066660.html

Banca: Identificación de personas con las compras de tarjetas de crédito

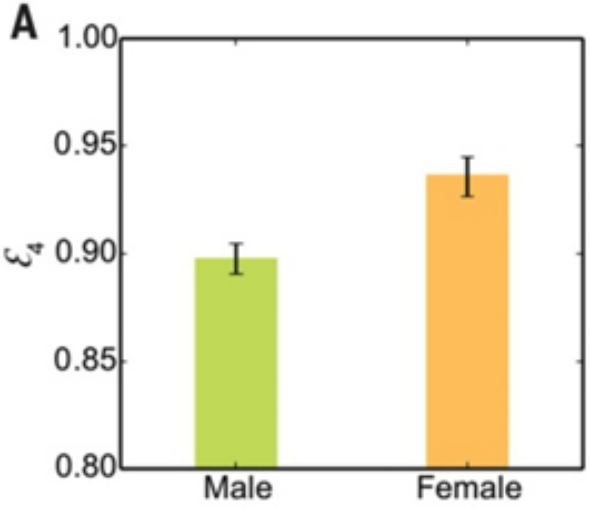
PRIVACIDAD EN INTERNET »

Cuatro compras con la tarjeta bastan para identificar a cualquier persona

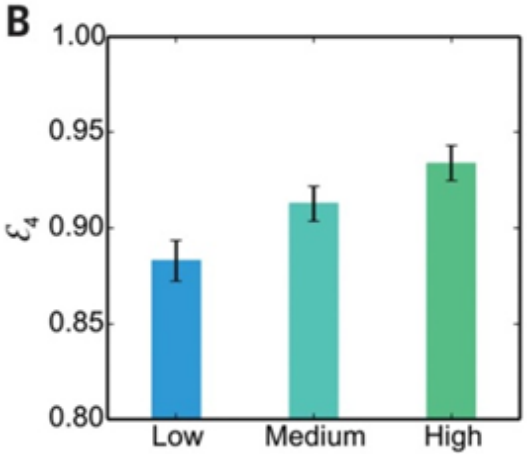
- Los patrones de uso de las tarjetas permiten descubrir la identidad del 90% de una muestra de 1,1 millones de personas anónimas, según demuestra un estudio del MIT



Identificación por el número de compras



Identificación por el género



Identificación por el poder adquisitivo

Análisis de transacciones

TARGET

Modelo de predicción de clientes embarazadas por medio de sus patrones de compra.

Descubrimiento: Cremas sin perfume al tercer mes. Dos docenas de productos

Predicción de fecha parto

Acción: Envío de cupones para cada fase del embarazo

Enfado de un padre: Su Hija recibe publicidad de productos para embarazadas

Disculpas Empresa. Disculpas del padre, confirmación del embarazo de la hija

Fuente: Big Data. La revolución de los datos masivos. Pag. 77

Análisis de transacciones

Target (cadena de grandes almacenes) que utiliza el análisis de transacciones y asociaciones.

Unos días después el director llamó al padre para disculparse.

Respuesta conciliadora del padre:

"He estado hablando con mi hija –dijo el padre– Resulta que en mi casa han tenido lugar ciertas actividades de las que yo no estaba del todo informado. Mi hija sale de cuentas en agosto. Soy yo el que les debe una disculpa".

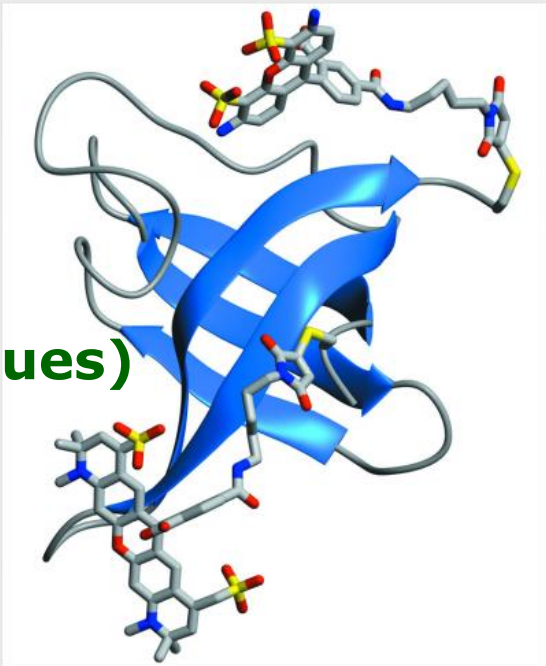
Fuente: Big Data. La revolución de los datos masivos. Pag. 77

Índice

- Big Data. Big Data Science
- ¿Por qué Big Data? Google crea el Modelo de Programación MapReduce
- Tecnologías para Big Data: Ecosistema Hadoop (Hadoop, Spark, …)
- Big Data Analytics: Librerías para Analítica de Datos en Big Data. Casos de estudio
- Algunas aplicaciones: Salud, Social Media, Identificación
- **Imbalanced Big Data: Caso de estudio**
- Comentarios Finales

ECBDL'14 Big Data Competition Vancouver, 2014

ECBDL'14 Big Data Competition 2014: Self-deployment track



Objective: Contact map prediction

Details:

- 32 million instances
- 631 attributes (539 real & 92 nominal values)
- 2 classes
- 98% of negative examples
- About 56.7GB of disk space

Evaluation:

True positive rate · True negative rate

TPR · TNR

http://cruncher.ncl.ac.uk/bdcomp/index.pl?action=data

J. Bacardit et al, Contact map prediction using a large-scale ensemble of rule sets and the fusion of multiple predicted structural features, Bioinformatics 28 (19) (2012) 2441-2448

Evolutionary Computation for Big Data and Big Learning Workshop

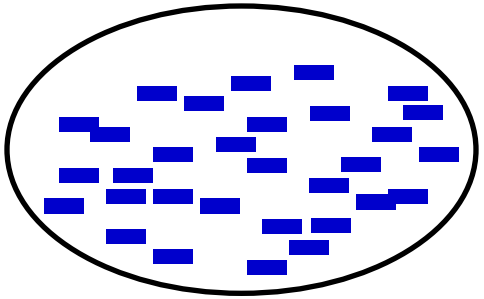
ECBDL'14 Big Data Competition 2014: Self-deployment track

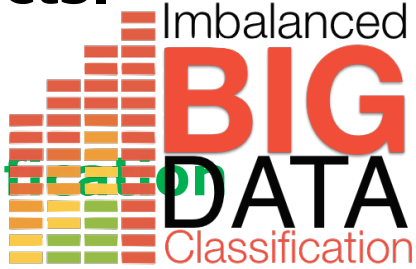
The challenge:

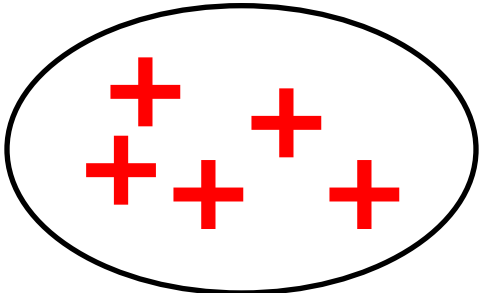
- Very large size of the training set
  - Does not fit all together in memory.
- Even large for the test set (5.1GB, 2.9 million instances)
- Relatively high dimensional data.
- Low ratio (<2%) of true contacts.

Imbalance rate: > 49

  - Imbalanced problem!
  - Imbalanced Big Data Classification







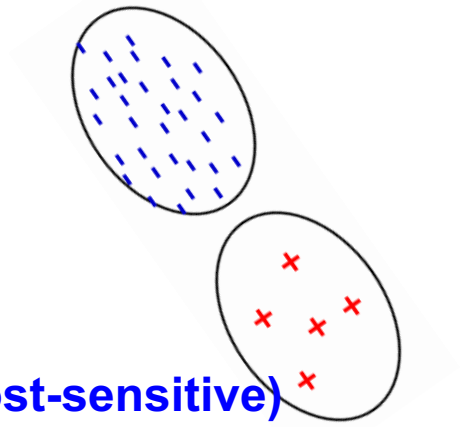
A MapReduce Approach

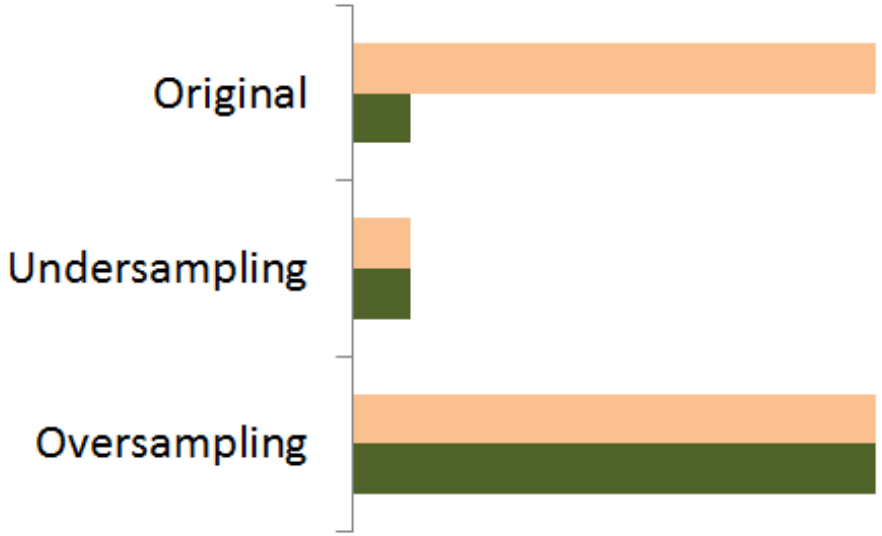
32 million instances, 98% of negative examples

Low ratio of true contacts (<2%). Imbalance rate: > 49. Imbalanced problem!

Previous study on extremely imbalanced big data: S. Río, V. López, J.M. Benítez, F. Herrera, On the use of MapReduce for Imbalanced Big Data using Random Forest. *Information Sciences 285 (2014) 112-137.*

- **Over-Sampling**
  - Random
  - Focused
- **Under-Sampling**
  - Random
  - Focused
- **Cost Modifying (cost-sensitive)**
- **Boosting/Bagging approaches (with preprocessing)**





ECBDL'14 Big Data Competition 2014

Our approach:

1. Balance the original training data
   - ❑ Random Oversampling
   - ❑ (As first idea)

2. Learning a model.
   - ❑ Random Forest

We initially focused on

- **Oversampling rate: 100%**

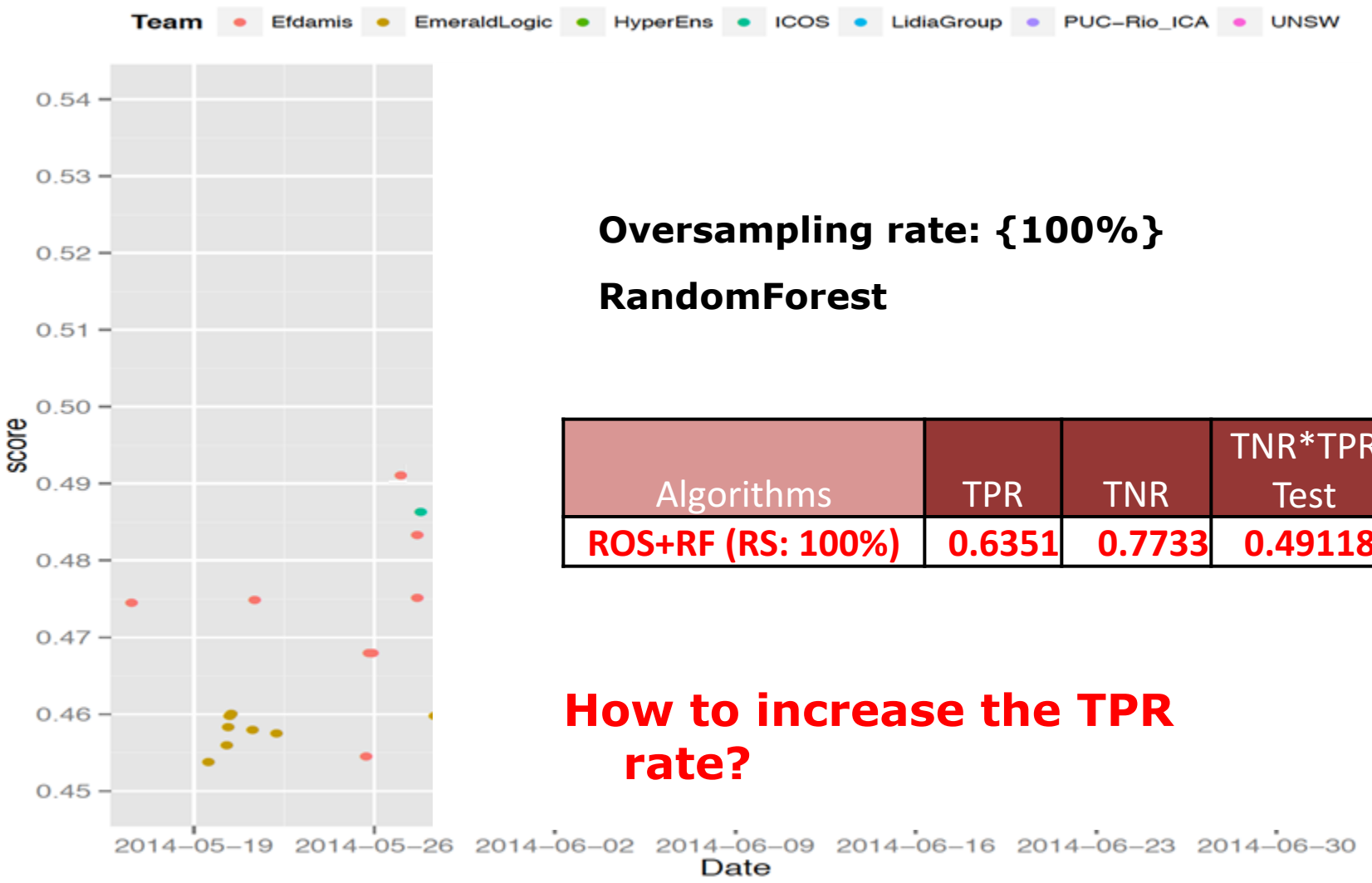
RandomForest:

- **Number of used features: 10 (log n +1); Number of trees: 100**
- **Number of maps: {64, 190, 1024, 2048}**

Nº mappers	TPR_tst	TNR_tst	TNR*TPR Test
64	0,601723	0,806269	0,485151

Very low TPR (relevant!)

ECBDL'14 Big Data Competition



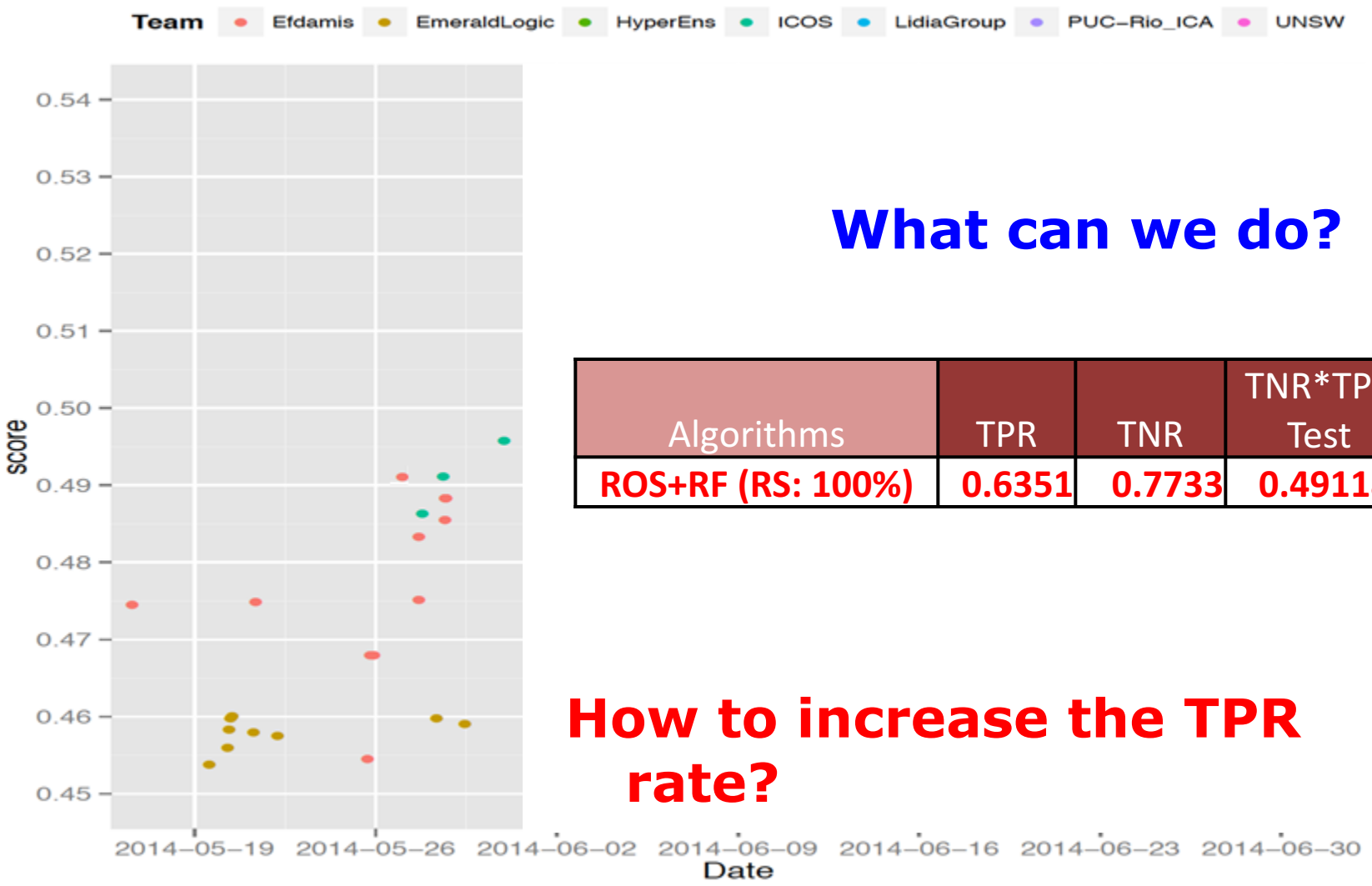
Oversampling rate: {100%}

RandomForest

Algorithms	TPR	TNR	TNR*TPR Test
ROS+RF (RS: 100%)	0.6351	0.7733	0.491186

How to increase the TPR rate?

ECBDL'14 Big Data Competition



What can we do?

Algorithms	TPR	TNR	TNR*TPR Test
ROS+RF (RS: 100%)	0.6351	0.7733	0.491186

How to increase the TPR rate?

ECBDL'14 Big Data Competition 2014

Our approach:

1. Balance the original training data
   - ❑ Random Oversampling increasing the ROS percentage
   - ❑ (As first idea, it was extended)

2. Learning a model.
   - ❑ Random Forest

How to increase the TPR rate?

Idea: To increase the ROS percentaje

- **Oversampling rate: {100, 105, 110, 115, 130}**

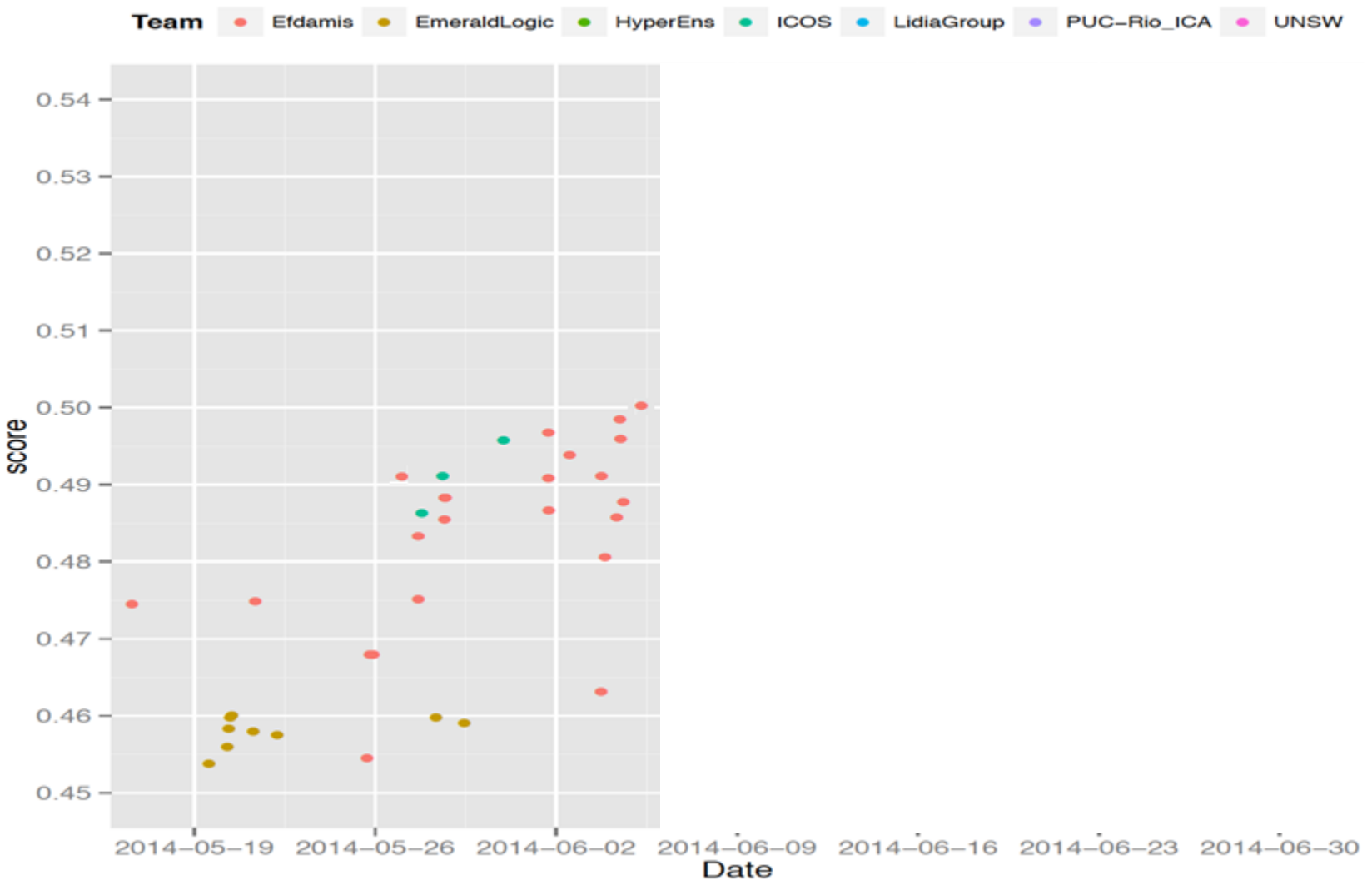
RandomForest:

- **Number of used features:10; Number of trees: 100**

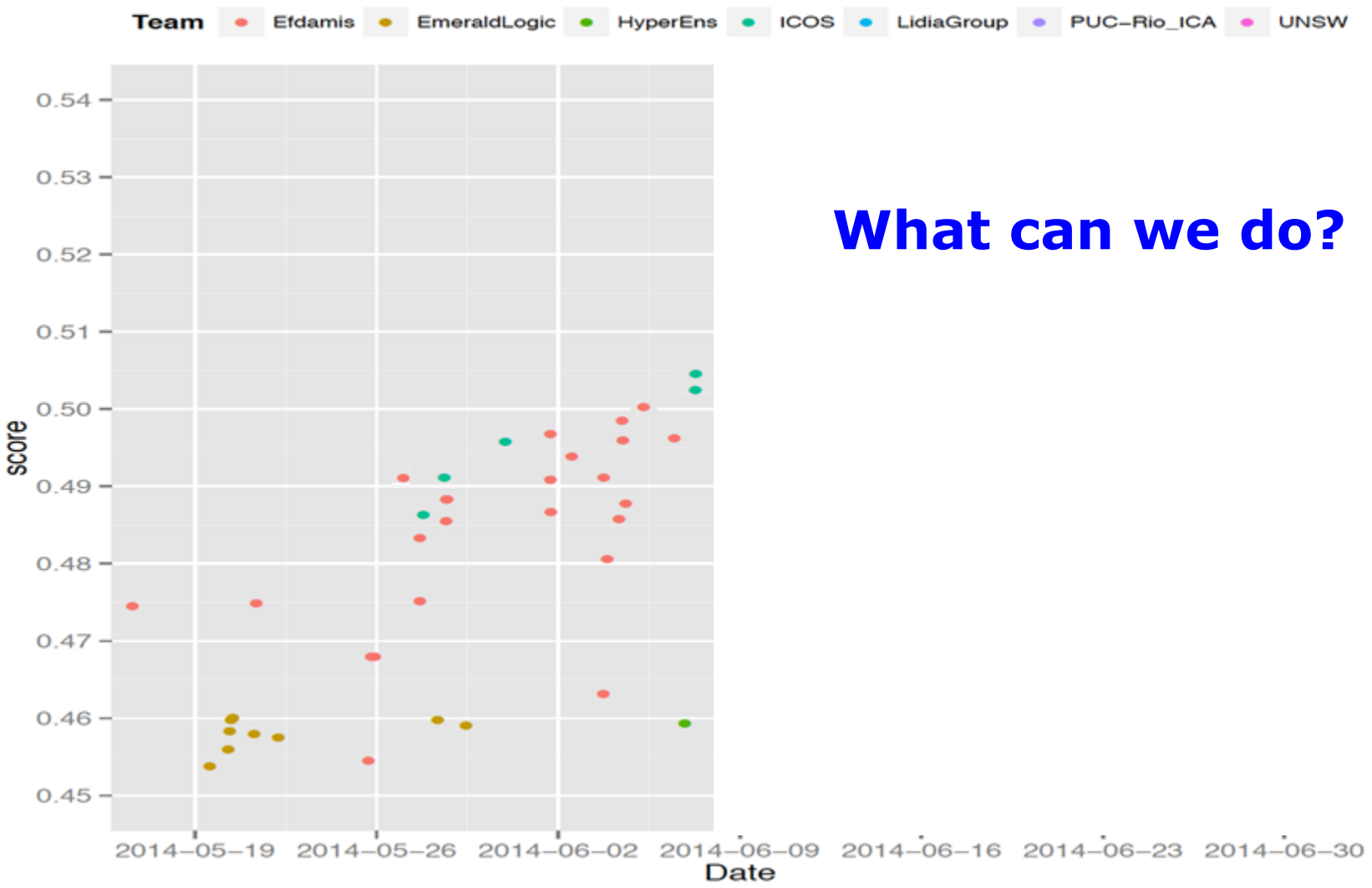
Algorithms	TPR	TNR	TNR*TPR Test
ROS+RF (RS: 100%)	**0.6351**	**0.7733**	**0.491186**
ROS+RF (RS: 105%)	0.6568	0.7555	0.496286
ROS+RF (RS: 110%)	0.6759	0.7337	0.495941
ROS+RF (RS: 115%)	**0.7041**	**0.7103**	**0.500175**
ROS+RF (RS: 130%)	0.7472	0.6609	0.493913

The higher ROS percentage, the higher TPR and the lower TNR

ECBDL'14 Big Data Competition



ECBDL'14 Big Data Competition

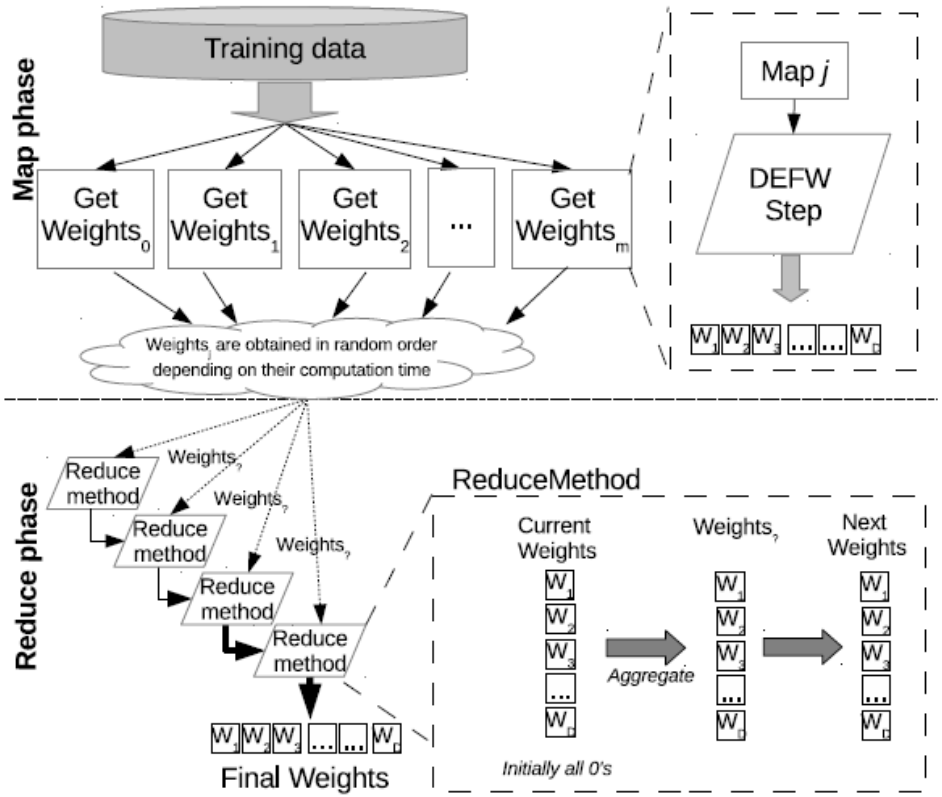


What can we do?

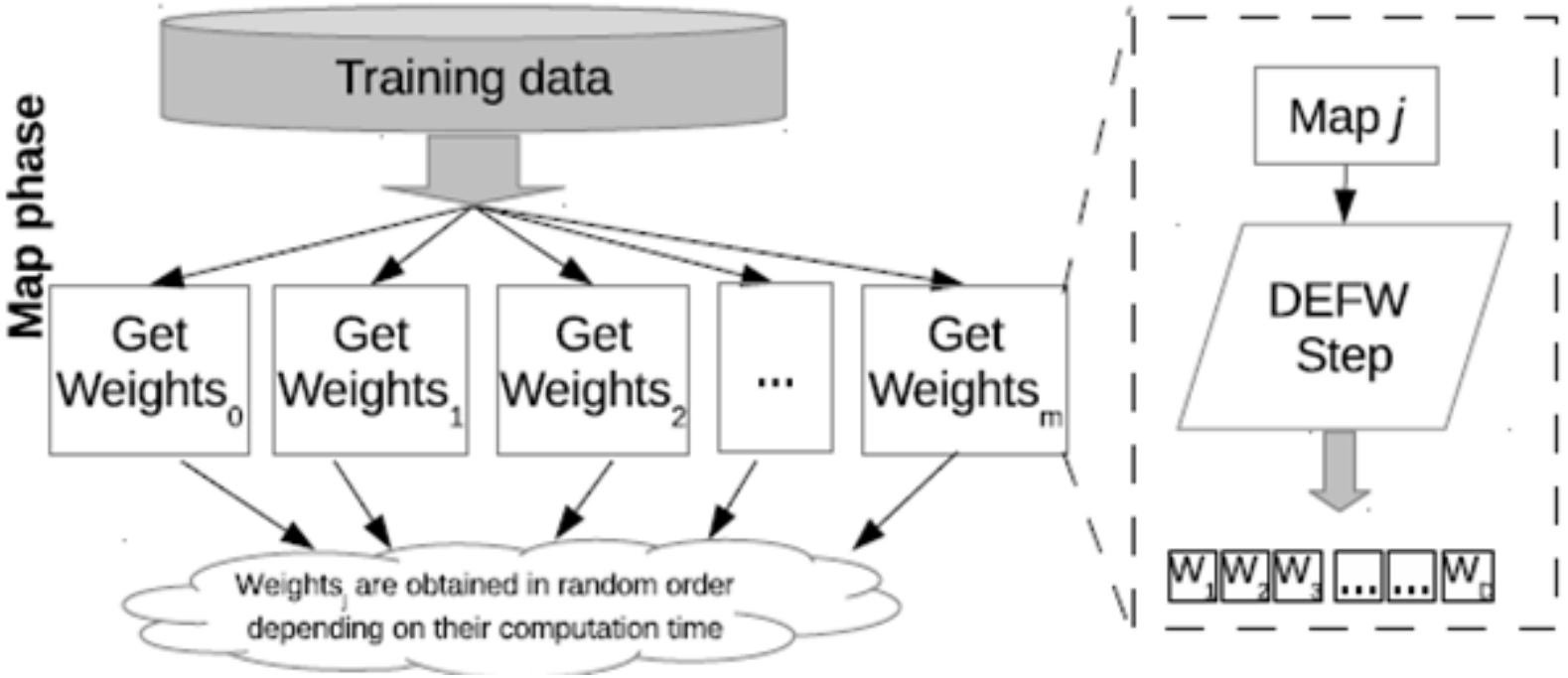
How to increase the performance?

Third component: MapReduce Approach for Feature Weighting

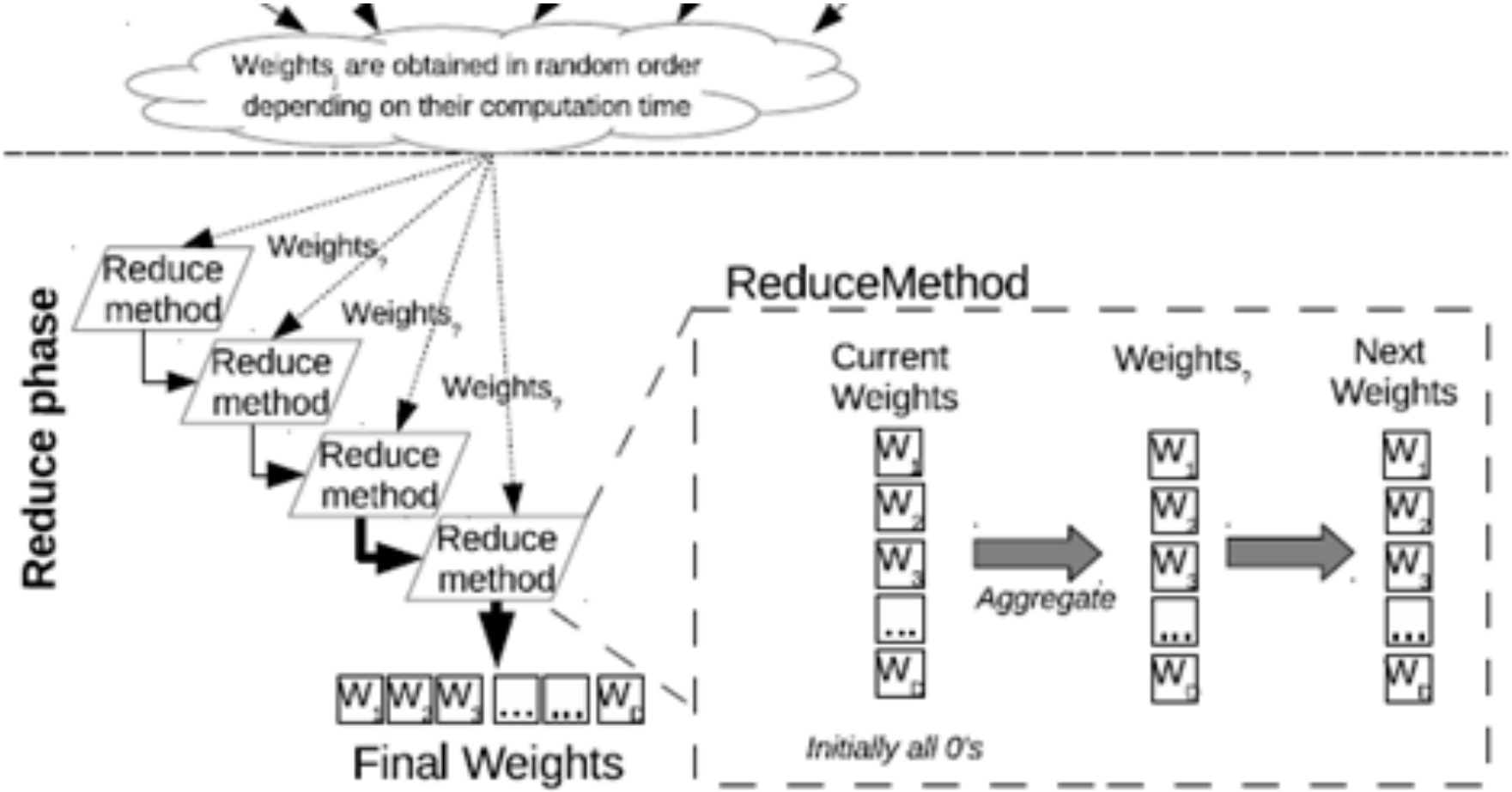
for getting



ECBDL'14 Big Data Competition



ECBDL'14 Big Data Competition



ECBDL'14 Big Data Competition 2014

Our approach:

1. Balance the original training data
   - ❑ Random Oversampling increasing the ROS percentage
   - ❑ (As first idea, it was extended)

2. Learning a model.
   - ❑ Random Forest

3. Detect relevant features.
   1. Evolutionary Feature Weighting

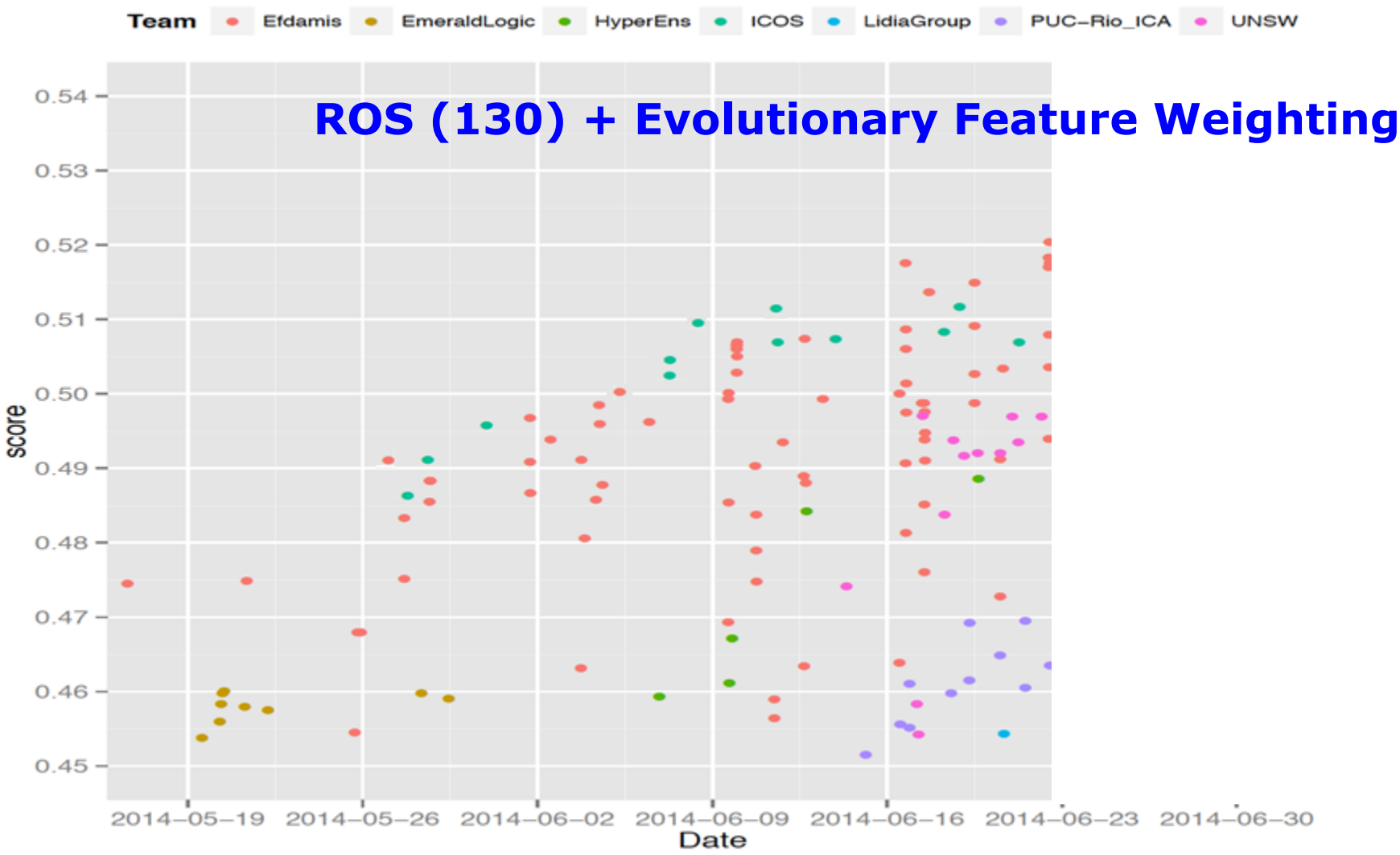
Classifying test set.

Evolutionary Feature Weighting.

It allows us to construct several subset of features (changing the threshold).

	64 mappers			
Algorithms	TNR*TPR Training	TPR	TNR	TNR*TPR Test
ROS+RF (130% - Feature Weighting 63)	0.726350	0.66949	0.775652	0.519292
ROS+RF (115% - Feature Weighting 63)	0.736596	0.652692	0.790822	0.516163
ROS+RF (100% - Feature Weighting 63)	0.752824	0.626190	0.811176	0.507950

ECBDL'14 Big Data Competition



We decided to investigate:

a) **On the Random Forest: the influence of the Random Forest's parameters (internal features and number of trees)**
b) **Higher number of features (90) and ROS with 140%**

	190 mappers			
Algorithms	TNR*TPR Training	TPR	TNR	TNR*TPR Test
ROS+ RF (130%+ FW 63+6f+100t)	0.604687	0.698152	0.742462	0.518351
ROS+ RF (130%+ FW 63+6f+200t)	0.632078	0.700064	0.745225	0.521705
ROS+ RF (140%+ FW 63+15f+200t)	0.627409	0.719678	0.728912	0.524582
ROS+ RF (140%+ FW 90+15f+200t)	0.635855	0.722639	0.726397	0.524923
ROS+ RF (140%+ FW 90+25f+200t)	0.629273	0.721652	0.729740	0.526618

Correct decisions with FW 90 and RF with 25f and 200 trees.
Good trade off between TPR and TNR

The less number of maps, the less TPR and the high TNR

	190 mappers			
Algorithms	TNR*TPR Training	TPR	TNR	TNR*TPR Test
ROS+ RF (140%+ FW 90+25f+200t)	0.629273	0.721652	0.729740	0.526618

64 mappers and we got 0.53

	64 mappers			
Algorithms	TNR*TPR Training	TPR	TNR	TNR*TPR Test
ROS+ RF (130%+ FW 90+25f+200t)	0.736987	0.671279	0.783911	0.526223
ROS+ RF (140%+ FW 90+25f+200t)	0.717048	0.695109	0.763951	0.531029

ROS 130 – 65 (140 – 68) replications of the minority instances

4 days to finish the competion:

Can we take decisions for improving the model?

Last decision: We investigated to increase ROS until 180% with 64 mappers

	64 mappers			
Algorithms	TNR*TPR Training	TPR	TNR	TNR*TPR Test
ROS+ RF (130%+ FW 90+25f+200t)	0.736987	0.671279	0.783911	0.526223
ROS+ RF (140%+ FW 90+25f+200t)	0.717048	0.695109	0.763951	0.531029
ROS+ RF (150%+ FW 90+25f+200t)	0.706934	0.705882	0.753625	0.531971
ROS+ RF (160%+ FW 90+25f+200t)	0,698769	0.718692	0.741976	0.533252
ROS+ RF (170%+ FW 90+25f+200t)	0.682910	0.730432	0.730183	0.533349
ROS+ RF (180%+ FW 90+25f+200t)	0,678986	0.737381	0.722583	0.532819

To increase ROS and reduce the mappers number lead us to get a trade-off with good results

ROS 170 – 85 replications of the minority instances

ECBDL'14 Big Data Competition

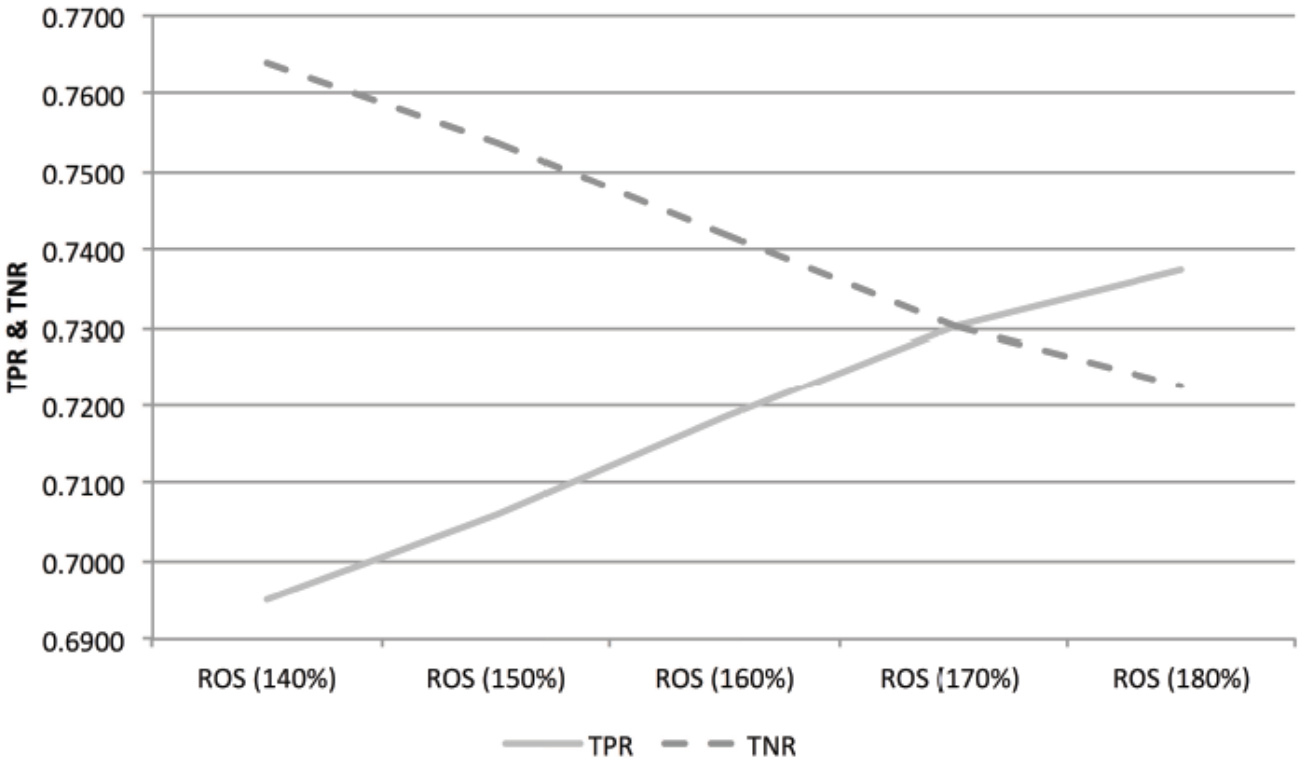


Figure 8: TPR vs. TNR varying the ROS percentage

ROS 170 – 85 replications of the minority instances
Experiments with 64 maps

Evolutionary Computation for Big Data and Big Learning Workshop

Results of the competition: Contact map prediction

Team Name	TPR	TNR	Acc	TPR · TNR
Efdamis	**0.730432**	**0.730183**	**0.730188**	**0.533349**
ICOS	**0.703210**	**0.730155**	**0.729703**	**0.513452**
UNSW	**0.699159**	**0.727631**	**0.727153**	**0.508730**
HyperEns	0.640027	0.763378	0.761308	0.488583
PUC-Rio_ICA	0.657092	0.714599	0.713634	0.469558

EFDAMIS team ranked first in the ECBDL'14 big data competition

http://cruncher.ncl.ac.uk/bdcomp/index.pl?action=ranking

ECBDL'14 Big Data Competition

ECBDL'14: Evolutionary Computation for Big Data and Big Learning Workshop
July 13th, 2014
GECCO-2014, Vancouver, Canada

This is to certify that team EFDAMIS, formed by Isaac Triguero, Sara del Río, Victoria López, José Manuel Benítez and Francisco Herrera, ranked **first** in the ECBDL'14 big data competition

Jaume Bacardit, organizer
ECBDL'14 big data competition

ECBDL'14 Big Data Competition

Final comments

Team Name	Learning strategy	Computational Infrastructure
Efdamis	Oversampling+EFW+Random Forest	MapReduce
ICOS	Oversampling+Ensemble of Rule sets	Batch HPC
UNSW	Ensemble of Deep Learning classifiers	Parallel HPC
HyperEns	SVM	Parallel HPC
PUC-Rio_ICA	Linear GP	GPUs
EmeraldLogic	~Linear GP	GPUs
LidiaGroup	1-layer NN	Spark

At the beginning ROS+RF (RS: 100%)

Nº mappers	TPR_tst	TNR_tst	TNR*TPR Test
64	0,601723	0,806269	0,485151

At the end

	64 mappers			
Algorithms	TNR*TPR Training	TPR	TNR	TNR*TPR Test
ROS+ RF (160%+ FW 90+25f+200t)	0,698769	0.718692	0.741976	0.533252
ROS+ RF (170%+ FW 90+25f+200t)	0.682910	0.730432	0.730183	0.533349
ROS+ RF (180%+ FW 90+25f+200t)	0,678986	0.737381	0.722583	0.532819

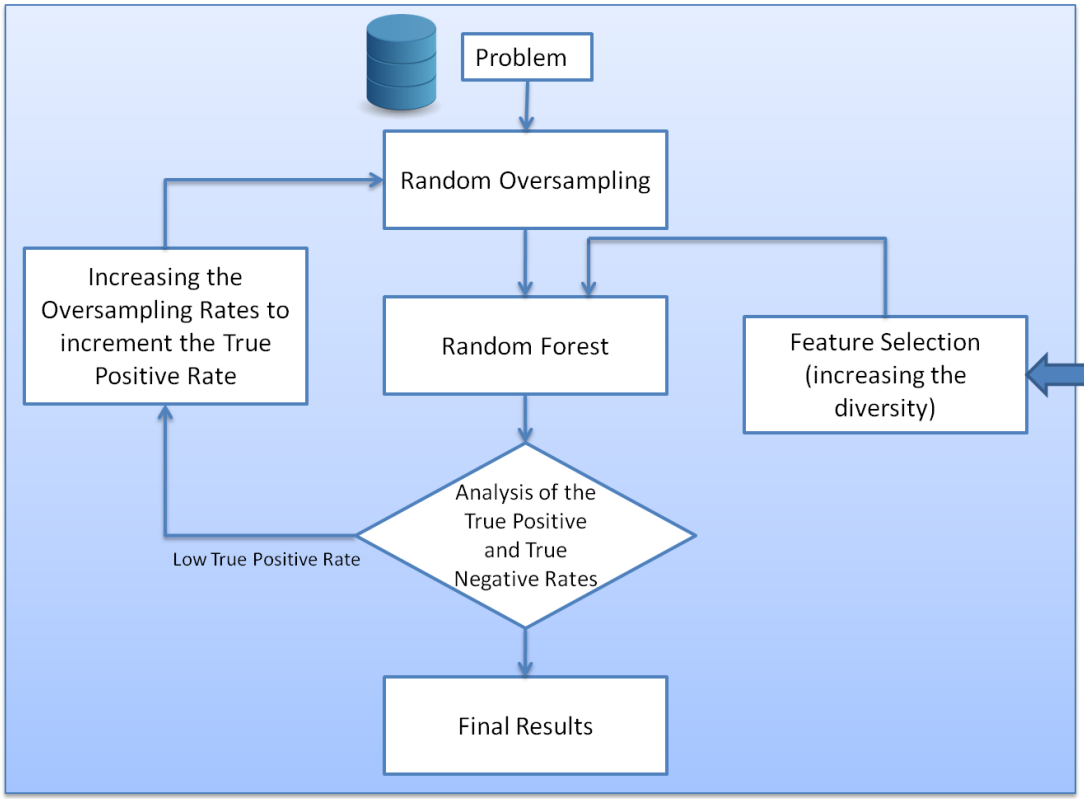
BIG DATA Preprocessing

Quality models based on quality data!

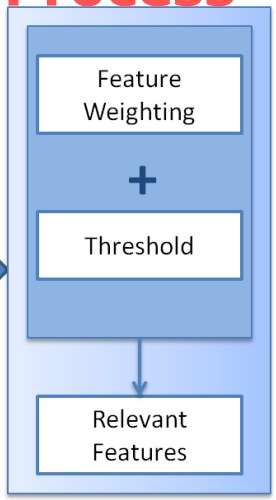
ECBDL'14 Big Data Competition

Our algorithm: ROSEFW-RF

Iterative MapReduce Process



MapReduce Process



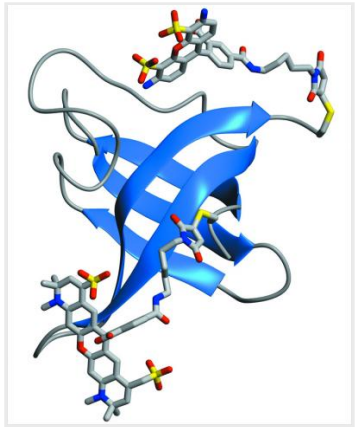
I. Triguero, S. del Río, V. López, J. Bacardit, J.M. Benítez, F. Herrera. ROSEFW-RF: The winner algorithm for the ECBDL'14 Big Data Competition: An extremely imbalanced big data bioinformatics problem. Knowledge-Based Systems, Volume 87, **October 2015, Pages 69–79**

https://github.com/triguero/ROSEFW-RF

Big Data Preprocessing

Bird's eye view SCI²S website http://sci2s.ugr.es/BigData

Two applications in Bionformatics



Knowledge-Based Systems
Volume 87, October 2015, Pages 69–79
Computational Intelligence Applications for Data Science

ROSEFW-RF: The winner algorithm for the ECBDL'14 big data competition: An extremely imbalanced big data bioinformatics problem

Isaac Triguero[a, b], Sara del Río[c], Victoria López[c], Jaume Bacardit[d], José M. Benítez[c], Francisco Herrera[c]

doi:10.1016/j.knosys.2015.05.027

BioMed Research International
Volume 2015 (2015), Article ID 748681, 12 pages
http://dx.doi.org/10.1155/2015/748681

Research Article

An Effective Big Data Supervised Imbalanced Classification Approach for Ortholog Detection in Related Yeast Species

Deborah Galpert,[1] Sara del Río,[2] Francisco Herrera,[2] Evys Ancede-Gallardo,[3] Agostinho Antunes,[4,5] and Guillermin Agüero-Chapin[3,4]

http://dx.doi.org/10.1155/2015/748681

CIMAR/CIIMAR, Centro Interdisciplinar de Investigação Marinha e Ambiental,

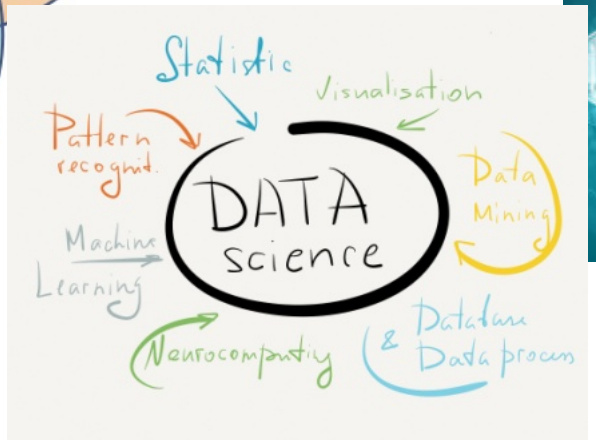
Índice

- Big Data. Big Data Science
- ¿Por qué Big Data? Google crea el Modelo de Programación MapReduce
- Tecnologías para Big Data: Ecosistema Hadoop (Hadoop, Spark, …)
- Big Data Analytics: Librerías para Analítica de Datos en Big Data.
- Casos de estudio: Random Forest, Clustering
- Algunas aplicaciones: Salud, Social Media, Identificación
- Imbalanced Big Data: Caso de estudio
- Comentarios Finales

Comentarios Finales

Ámbito del conocimiento que engloba las habilidades asociadas al análisis inteligente de datos, incluyendo Big Data

Científico de datos
(actualmente se menciona la demanda en big data profesionalmente, entendida como el global).



Comentarios Finales

Oportunidades en Big Data

Big Data es un área emergente y en expansión. Las posibilidades de desarrollo de algoritmos para nuevos datos, aplicaciones reales ... es un nicho de investigación y desarrollo en los próximos años.

Comentarios Finales

http://elpais.com/elpais/2015/03/26/buenavida/1427382655_646798.html

¿Qué es eso del 'big data'?

- Lo mencionan en conferencias, charlas y facultades. Aclaramos el concepto de moda. O lo que es lo mismo: lo que todas las empresas quieren saber de usted

EVA VAN DEN BERG | 31 MAR 2015 - 12:28 CEST

Tiene continua repercusión en la prensa

Comentarios Finales

Big Data: Gran Impacto en la Sociedad y presencia en los medios de comur

ELMUNDO.es
Líder mundial en español | Miércoles 04/09/2013. Actualizado 16:27h.

INTERNET | Campus Party Europa 2013

'Es la década de los ahí vendrá la revolu

Alex 'Sandy' Pentland, durante su ponencia. | M. Sáinz

http://www.elmundo.es/elmundo/201

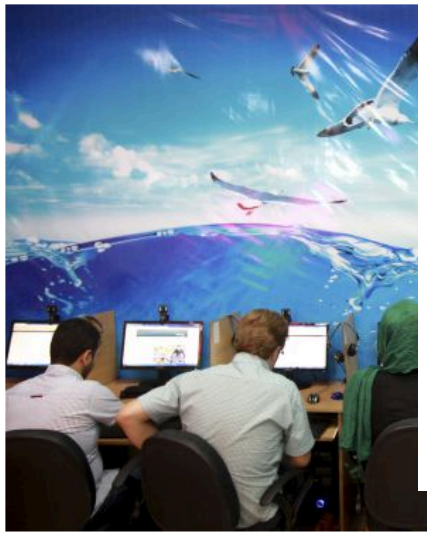
EL PAÍS
ECONOMÍA
ECONOMÍA EMPRESAS MERCADOS BOLSA

El maná de los datos

- La conversión de datos en información útil para la millones de dólares en 2015. La herramienta 'big

SUSANA BLÁZQUEZ | Madrid | 29 SEP 2013 - 01:00 C

Archivado en: Citigroup Cap Gemini Sogeti SAP IBM Telefónica Aplicaciones informáticas Tecnología



EL PAÍS
ECONOMÍA
ECONOMÍA EMPRESAS MERCADOS BOLSA MIS AHORROS VIVIENDA TECNOLOGÍA OP

EMPRENDEDORES »

El Big Data echa una mano al campo

- Una empresa española recoge miles de datos para predecir las cosechas

MARÍA FERNÁNDEZ | 30 NOV 2014 - 00:00 CET

Archivado en: Bases datos Emprendedores Aplicaciones informáticas Empresas Programas informáticos Economía Informática Industria

Comentarios Finales

- La paralelización de los algoritmos de aprendizaje automático junto al particionamiento de datos pueden proporcionar algoritmos de calidad con MapReduce.
  - Paticionando datos y aplicando el algoritmo a cada parte.
  - Centrando la atención en la fase de combinacion (**reduce**). La combinación de modelos es un reto en el diseño de cada algoritmo.
- Data Mining, Machine learning and data preprocessing: Inmensa colección de algoritmos frente a los pocos algoritmos en big data analytics.

Comentarios Finales

Machine learning: Huge collection of algorithms
Big data: A small subset of algorithms

Para el diseño y/o adaptación de cualquier algoritmo es necesario diseñar de forma adecuada una fase de fusión de información por cuanto siempre será necesario utilizar funciones Map y Reduce cuando la base de datos sea muy grande.

Igualmente los procedimientos iterativos requieren de un diseño adecuado para optimizar la eficiencia.

Todavía se está en una fase muy temprana de diseño de algoritmos de aprendizaje automático para big data.

El preprocesamiento de datos es esencial para mejorar el comportamiento de los algoritmos de aprendizaje. El diseño de estos algoritmos para big data está en una fase muy incipiente.

Comentarios Finales

Data Mining, Machine learning and data preprocessing: Inmensa colección de algoritmos

Big Data Analytics

Big Data: Un pequeño conjunto de algoritmos

Big Data Preprocessing:

Unos pocos métodos de preprocesamiento.

Deep learning:

Redes neuronales para procesamiento de señales/imágenes en grandes volúmenes.

Comentarios Finales

Big data and analytics:
Un gran reto que ofrece múltiples oportunidades

- **Pequeño conjunto de algoritmos**
  Es necesario rediseñar nuevos algoritmos.
- **Modelo de Computación**
  - **Precisión y aproximación**
  - **Requiere "eficiencia" en los algoritmos.**

- **Datos de calidad para modelos de calidad en big data**
  Modelos/Decisiones de calidad están basados en datos de calidad.
- **Preprocesamiento en Big Data**
  - **Análisis del ruido en datos**
    Métodos automáticos de limpieza
  - **Procesamiento de valores perdidos**
  - **Big Data Reduction**

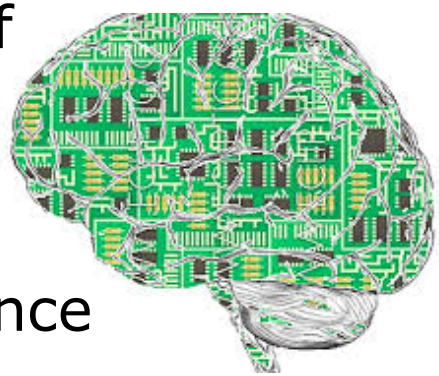
Comentarios Finales

¿Hacia donde vamos?: 3 etapas de Big Data

http://www.kdnuggets.com/2013/12/3-stages-big-data.html
By Gregory Piatetsky, Dec 8, 2013.

Big Data 3.0: Intelligent
Google Now, Watson (IBM) ...

Big Data 3.0 would be a combination of data, with huge knowledge bases and a very large collection of algorithms, perhaps reaching the level of true Artificial Intelligence (Singularity?).



Big Data 1.0:
Transactional

Big Data 2.0:
Networked

Big Data 3.0:
Intelligent

Comentarios Finales

Una demanda creciente de profesionales en "Big Data" y "Ciencia de Datos"

Oportunidades en Big Data (en España)

http://www.revistacloudcomputing.com/2013/10/espana-necesitara-60-000-profesionales-de-big-data-hasta-2015/?goback=.gde_4377072_member_5811011886832984067#!

España necesitará 60.000 profesionales de Big Data hasta 2015

22 octubre, 2013 Eventos 18

España necesitará 60.000 profesionales de Big Data hasta 2015

"España va a necesitar alrededor de sesenta mil profesionales del Big Data de aquí a 2015", así lo ha asegurado Francisco Javier Antón, Subdirector General de Tecnologías del Ministerio de Educación, Cultura y Deportes en una mesa redonda sobre beneficio y aplicación de Big Data en pymes, moderada por Daniel Tapias de Sigma Technologies, celebrada durante el 4º Congreso Nacional de CENTAC de Toledo.

"Existe una demanda mundial para formar a 4,4 millones de profesionales de la gestión Big Data desde ingenieros, gestores y científicos de datos", comenta Antón. Sin embargo, "las empresas todavía no ven en el Big Data un modelo de negocio", lamenta. "Solo se extrae un 1% de los datos disponibles en la red", añade. "Hace falta formación y concienciación.

Big Data at SCI²S - UGR

http://sci2s.ugr.es/BigData

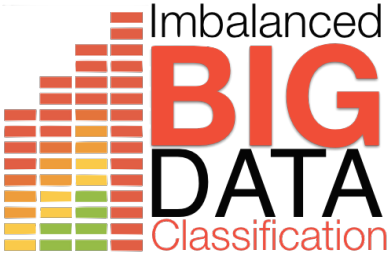
SCI²S website

Home » Thematic Sites » Big Data: Algorithms for Data Preprocessing, Computational Intelligence, and Imbalanced Classes

Big Data: Algorithms for Data Preprocessing, Computational Intelligence, and Imbalanced Classes

The web is organized according to the following summary:

1. Introduction to Big Data
2. Big Data Technologies: Hadoop ecosystem and Spark
3. Big Data preprocessing
4. Imbalanced Big Data classification
5. Big Data classification with fuzzy models
6. Classification Algorithms: k-NN
7. Big Data Applications
8. Dataset Repository
9. Literature review: surveys and overviews
10. Keynote slides
11. Links of interest



This **Website** contains SCI²S research material on algorithms for data preprocessing, computational intelligence and classification with imbalanced datasets in the scenario of Big Data. All information shown here is related to the following SCI²S journal papers and algorithms developed:

Comentarios Finales

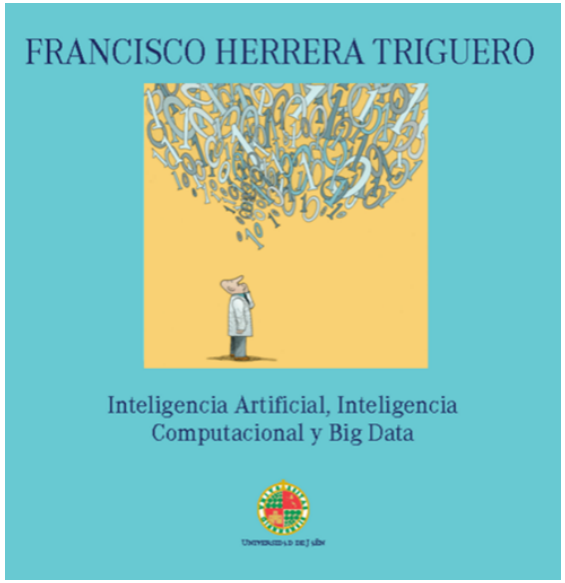
Nubes de datos
Sergio García y Lola Moral
Noviembe 2014

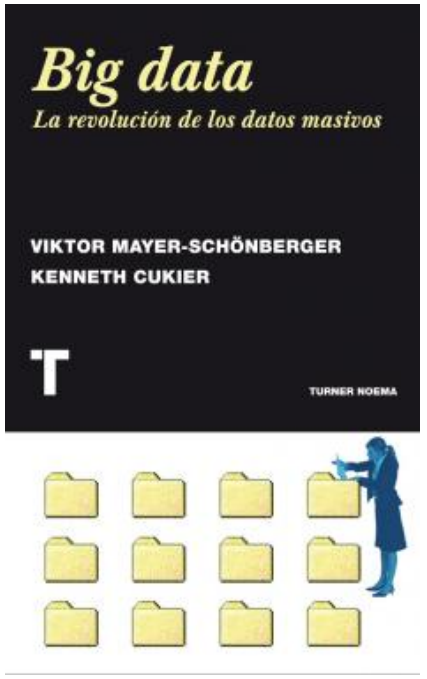
Las grandes colecciones de datos nos van a conducir a nuevas innovaciones (en la industria, ciencia, sociedad)

Ben Chams - Fotolia

Comentarios Finales

2 Lecturas rápidas:





Capítulo 3.

http://issuu.com/secacult_uja/docs/libro_francisco_herrera.indd

A. Fernandez, S. Río, V. López, A. Bawakid, M.J. del Jesus, J.M. Benítez, F. Herrera, **Big Data with Cloud Computing: An Insight on the Computing Environment, MapReduce and Programming Frameworks.** *WIREs Data Mining and Knowledge Discovery 4:5 (2014) 380-409*


¡Gracias!