



Universidad de Granada

decsai.ugr.es

Inteligencia Artificial en Telecomunicaciones

Máster en Ingeniería de Telecomunicaciones

Tema 6: Soft Computing, Data Mining y Big Data



**Departamento de Ciencias de la
Computación e Inteligencia Artificial**

Soft Computing

Soft Computing es una rama de la Inteligencia Artificial que agrupa técnicas para trabajar sobre información **incompleta, imprecisa y/o incierta**, obteniendo **soluciones válidas aunque sean aproximadas**.

En Soft Computing se usa soluciones inexactas para calcular **problemas NP completos**, que son aquellos para los cuales no existe una solución en tiempo polinomial.

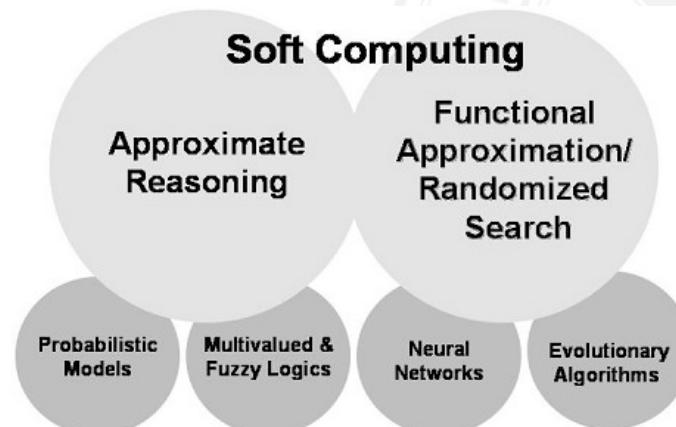
En Soft computing el modelo a seguir en la **inteligencia humana**.

Hard Computing, viene a ser lo contrario y engloba a todas aquellas técnicas que no trabajan con imprecisión, incertidumbre e inexactitud (por ejemplo, cálculo simbólico o análisis numérico).

Técnicas de Soft Computing

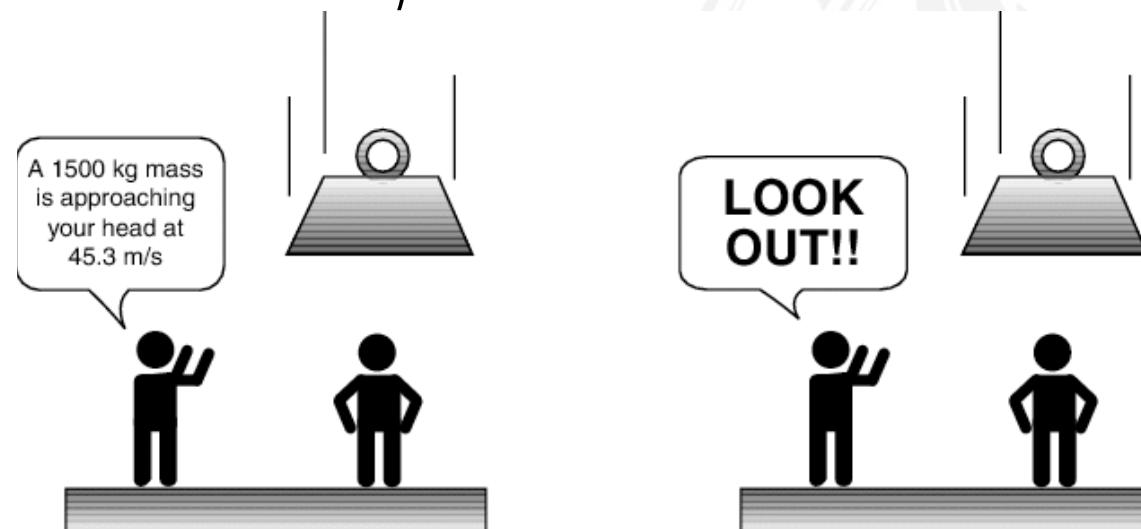
En Soft Computing se utilizan distintas técnicas, algunas de las cuales ya hemos visto:

- ▶ **Redes Neuronales:** Como por ejemplo, el perceptrón multicapa..
- ▶ **Lógica Difusa:** También denominada Fuzzy Logic.
- ▶ **Computación evolutiva (bioinspirada):** Aquí podemos encontrar, por ejemplo, algoritmos genéticos o de inteligencia de enjambre.
- ▶ **Razonamiento probabilístico:** Como son las redes bayesianas o los modelos ocultos de Markov.
- ▶ **Máquinas de vectores de soporte:** También conocidos como Support Vector Machines (SVM).



Lógica Difusa

- ▶ La **lógica difusa** o borrosa (del inglés **Fuzzy Logic**) es una rama de la inteligencia artificial que permite **simular los procedimientos de razonamiento humano** en sistemas basados en el conocimiento.
- ▶ La teoría de la lógica difusa proporciona una **base matemática** que permite **trabajar con la incertidumbre de los procesos cognitivos humanos** de forma que pueda ser tratable por un ordenador.
- ▶ Este tipo de lógica toma dos valores aleatorios, pero contextualizados y referidos entre sí, por ejemplo, una persona que mida dos metros es una persona alta, si previamente se ha tomado el valor de persona baja. Ambos valores están contextualizados a personas y referidos a una medida métrica lineal (metros o centímetros).



Lógica Difusa

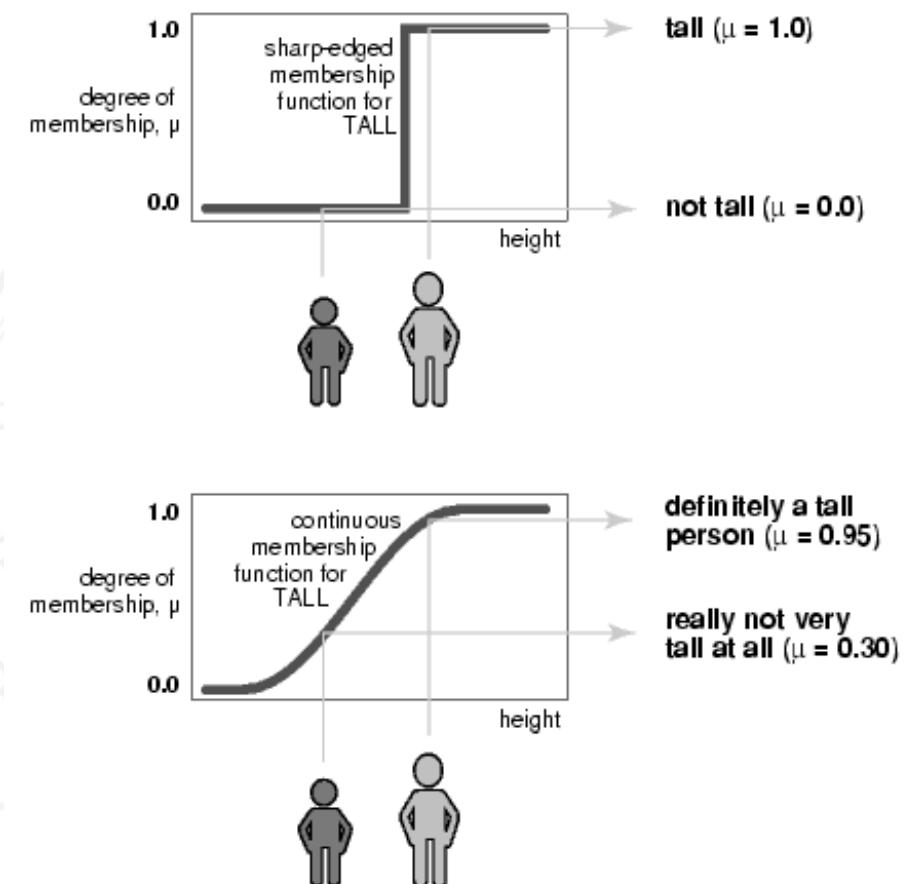
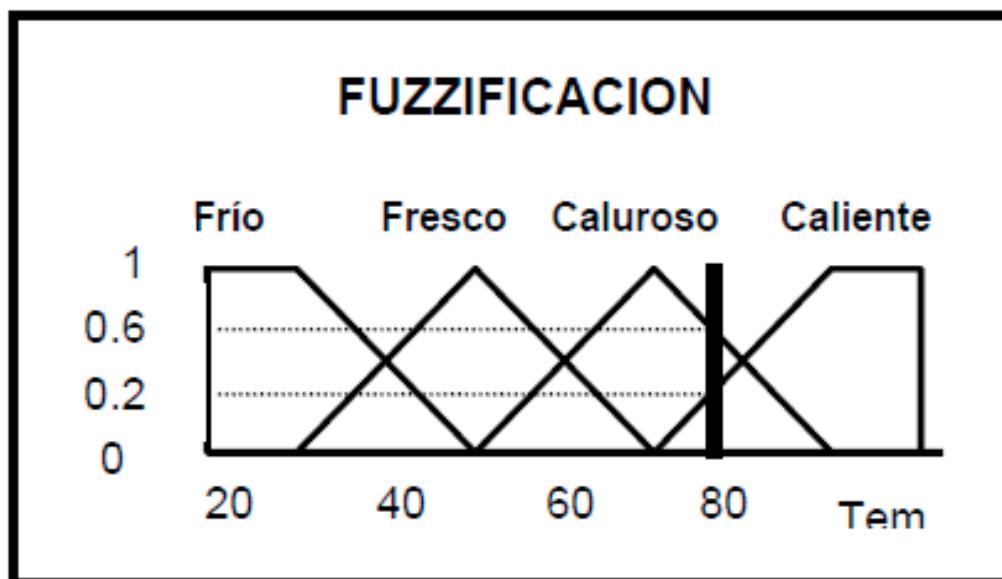
- ▶ El ser humano rara vez piensa en **términos absolutos** y/o perfectos (no todo es blanco o negro). Nuestra forma de razonar conlleva una serie de abstracciones sobre la información que procesamos ya de por si inexacta.
- ▶ La lógica difusa nos permite trabajar con conceptos como, por ejemplo, "hace mucho calor"(qué caló), "un poquito"(una mijitica) o "mucho mayor" (una jartá).
- ▶ En la lógica difusa los **conceptos de verdad o mentira se relajan**, existiendo una **transición suave y progresiva entre estados**.



"I want a computer that does what I want it to do, not what I tell it to do!"

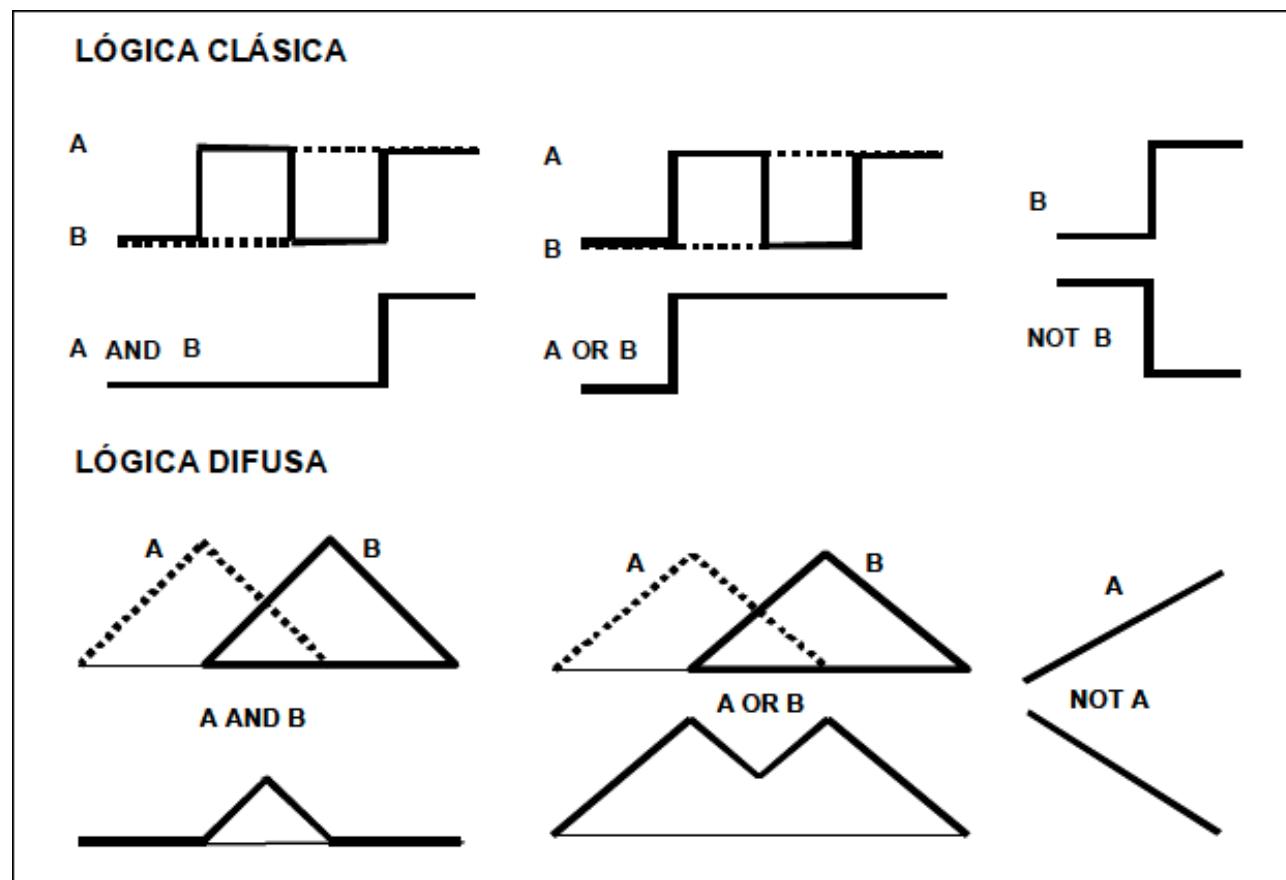
Lógica Difusa

- Referirse a la lógica difusa es referirse al concepto de **conjunto difuso**.
- Para cada conjunto difuso, existe una **función de pertenencia** que indica en qué medida el elemento forma parte de ese conjunto difuso.



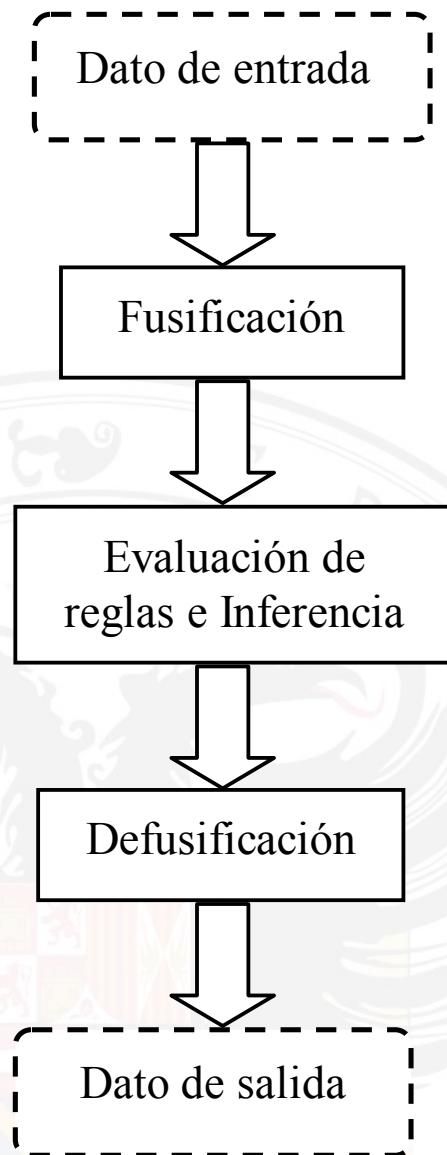
Lógica Difusa

- ▶ En la teoría de conjuntos difusos se definen también las operaciones de unión, intersección, diferencia, negación o complemento.



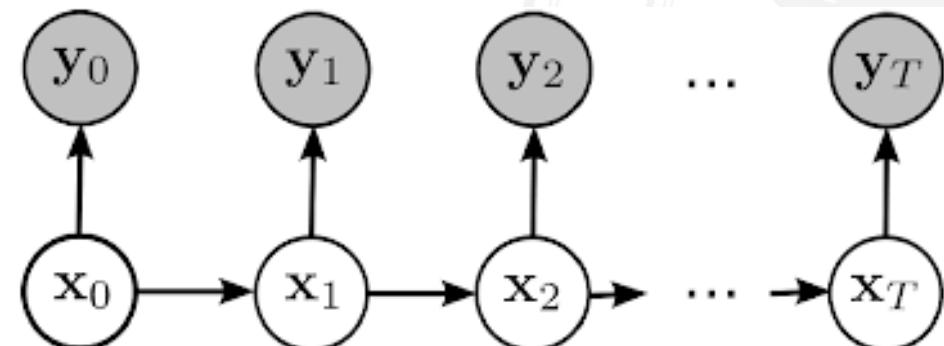
Lógica Difusa

- ▶ En la lógica difusa se utilizan reglas del tipo:
SI (antecedente) **ENTONCES** (consecuente)
- ▶ Donde el **antecedente y el consecuente son conjuntos difusos**, por ejemplo:
 - SI hace mucho calor ENTONCES enciendo el AC.
 - SI voy a mucha velocidad ENTONCES aumento la distancia de seguridad.
- ▶ La **inferencia** en lógica difusa es obtener consecuencias a partir de hechos (premisas), pero a **través de relaciones difusas**.
- ▶ Para escoger una salida concreta se usa el **centroide**: el centro de gravedad del área total resultante.
- ▶ Se aplica principalmente en **sistemas de control** (por ejemplo, AC, lavadoras, enfocado automático, sistemas de control industriales, regulador velocidad coches, sistemas expertos)

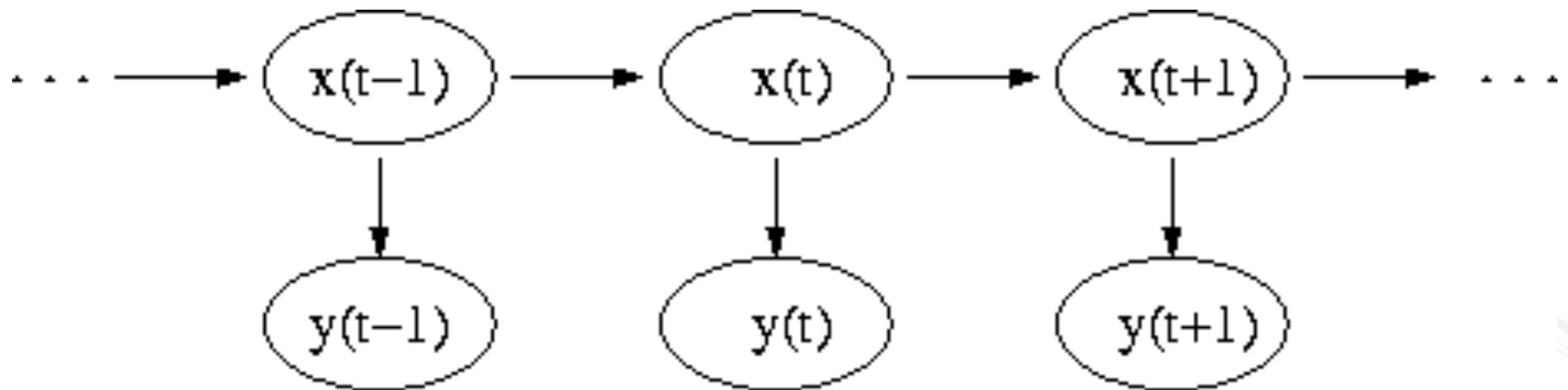


Modelos Ocultos de Markov

- ▶ Un proceso de Markov (o cadena de Markov) es un caso particular de proceso estocástico donde la probabilidad de llegar a un estado $x(t)$ únicamente depende del estado anterior $x(t-1)$ y no del resto de estados que lo preceden (Propiedad de Markov).
- ▶ Una generalización de la cadena de Markov es la de la dependencia de n estados. Donde, la probabilidad de llegar a un estado $x(t)$ está condicionado por los n estados anteriores, siendo $n = 1$ (cadena de Markov de primer orden) el modelo clásico.
- ▶ Un modelo oculto de Markov (o Hidden Markov Model -HMM-) es un proceso de Markov donde el único elemento conocido son los estados, mientras que las transiciones entre éstos y sus probabilidades son desconocidas.



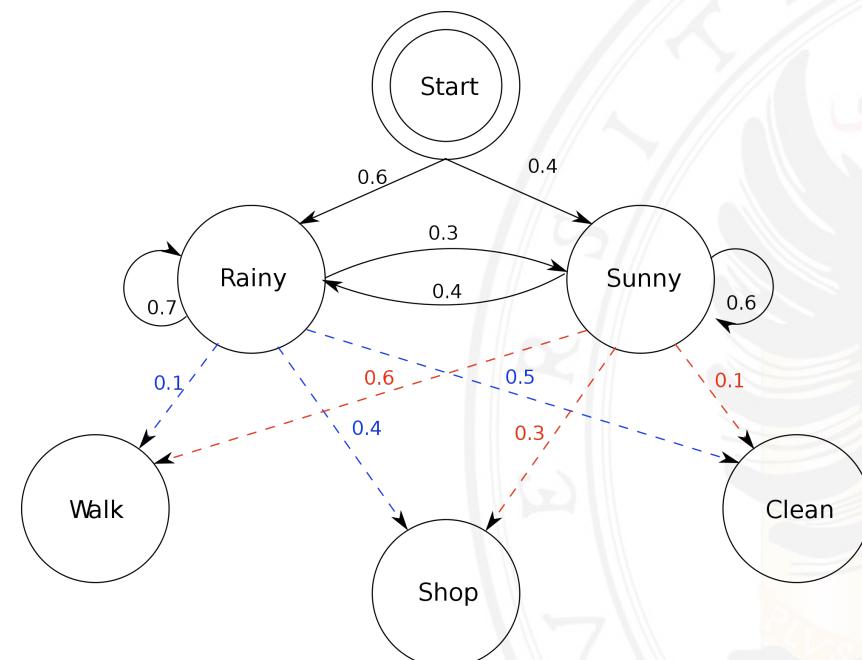
Modelos Ocultos de Markov



- En el anterior diagrama tenemos un HMM, donde cada circulo representa una variable aleatoria y las flechas dependencias condicionales.
- La variable aleatoria $x(t)$ es el valor de la variable oculta en el instante de tiempo t . La variable aleatoria $y(t)$ es el valor de la variable observada en el mismo instante de tiempo t .

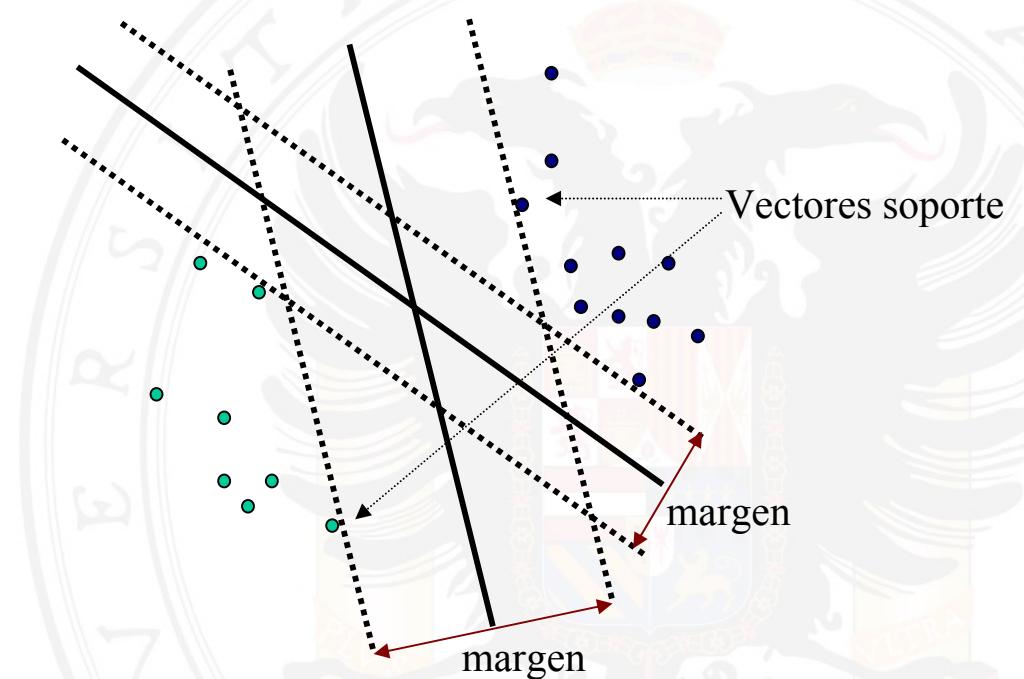
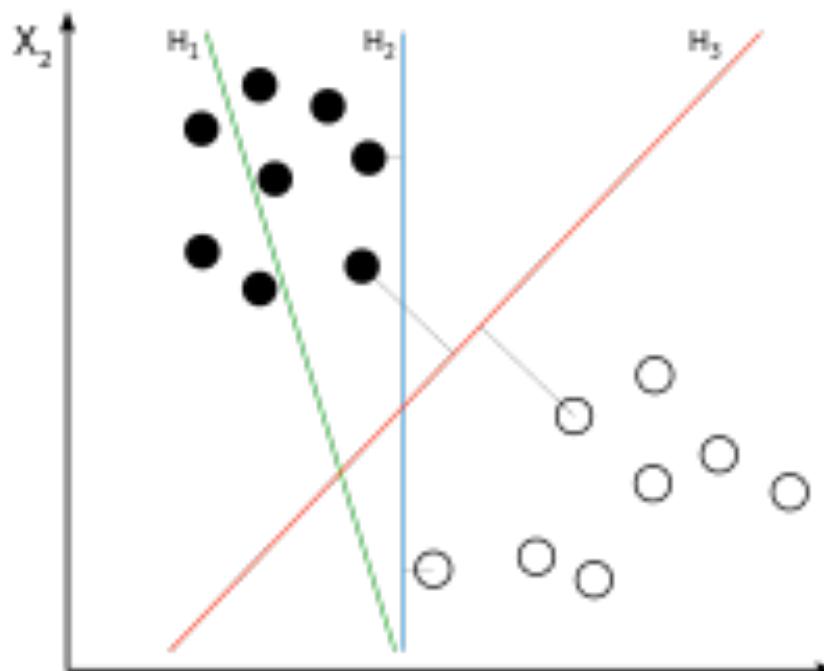
Modelos Ocultos de Markov

- ▶ No hay forma óptima ni analítica de aprender los parámetros.
- ▶ Se puede resolver mediante métodos iterativos de escalada.
- ▶ Algoritmo de avance retroceso.



Máquinas de Vectores de Soporte

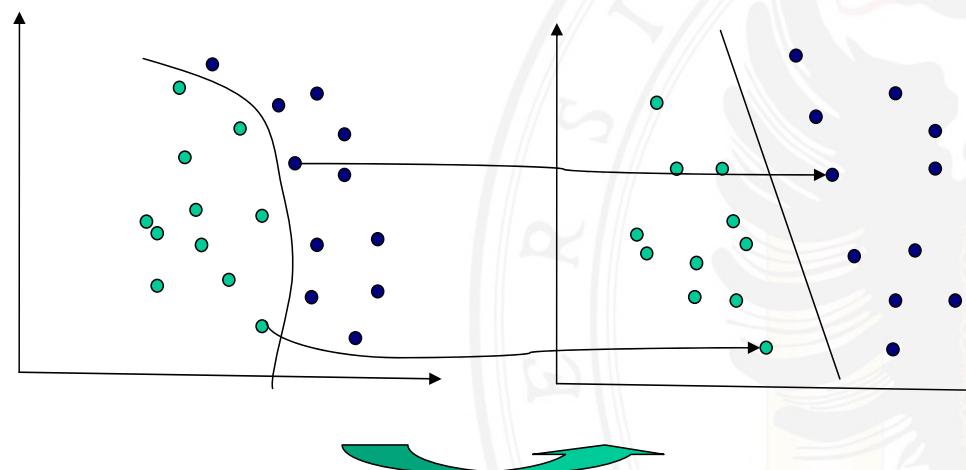
- ▶ La idea que hay detrás de las Máquinas de Vectores de Soporte (Support Vector Machines -SVM-) es la división de los ejemplos usando hiperplanos.
- ▶ Existe un número infinito de posibles divisiones por lo que se define la mejor solución como aquella que permita un margen máximo entre los elementos de las dos categorías.
- ▶ Se denominan vectores de soporte a los puntos que conforman las dos líneas paralelas al hiperplano, siendo la distancia entre ellas (margen) la mayor posible



Máquinas de Vectores de Soporte

Operaciones de una SVM:

- ▶ Transforma los datos a un espacio de dimensión muy alta a través de una función kernel, por lo que se reformula el problema de tal forma que los datos se mapean implícitamente en este espacio.
- ▶ Encuentra el hiperplano que maximiza el “margen” entre dos clases.
- ▶ Si los datos no son linealmente separables encuentra el hiperplano que maximiza el margen y minimiza una función del número de clasificaciones incorrectas (término de penalización de la función).



Encontrar la función $\Phi(x)$ que transforma a un espacio de dimensión muy superior.

Máquinas de Vectores de Soporte

ANNs

- Capas ocultas transforman a espacios de cualquier dimensión
- El espacio de búsqueda tiene múltiples mínimos locales
- El entrenamiento es costoso
- La clasificación es muy eficiente
- Se diseña el número de capas ocultas y nodos
- Muy buen funcionamiento en problemas típicos

SVMs

- Kernels transforman a espacios de dimensión muy superior
- El espacio de búsqueda tiene sólo un mínimo global
- El entrenamiento es muy eficiente
- La clasificación es muy eficiente
- Se diseña la función kernel y el parámetro de coste C
- Muy buen funcionamiento en problemas típicos
- Extremadamente robusto para generalización, menos necesidad de heurísticos para entrenamiento

Data Mining

- ▶ El **Data Mining** o **minería de datos** es el proceso que trata de extraer información o descubrir patrones de un conjunto de datos, normalmente voluminoso.
- ▶ Utiliza los métodos de la **inteligencia artificial, aprendizaje automático, estadística y sistemas de bases de datos.**



Pasos en minería de datos

- ▶ **1. Definir el problema:** se trata de la delimitación de los objetivos que se desean obtener de los datos y qué datos se van a usar.
- ▶ **2. Preprocesamiento de los datos:** Se refiere a la selección de atributos, la limpieza de ruido y anomalías, imputación de datos perdidos, la integración, reducción y transformación de las bases de datos. Esta etapa consume generalmente alrededor del setenta por ciento del tiempo total de un proyecto de data mining.
- ▶ **3. Generar el modelo:** Aquí es donde utilizamos un método de Inteligencia Artificial (si el problema es NP-completo) para extraer conocimiento a partir de los datos, por ejemplo, árboles de decisión.
- ▶ **4. Análisis y validación de los resultados:** Se analizan los resultados obtenidos por un experto o validándolos con el mundo real.

Big Data

- ▶ El **Big Data o Datos Masivos** es un concepto que hace referencia a la acumulación de grandes cantidades de datos y a los procedimientos usados para encontrar patrones repetitivos dentro de esos datos.
- ▶ El fenómeno del Big Data también es llamado **datos a gran escala**.



Dan Ariely

January 6, 2013 ·

[Follow](#)

Big data is like teenage sex: everyone talks about it, nobody really knows how to do it, everyone thinks everyone else is doing it, so everyone claims they are doing it...

Like · Comment · Share

Big Data vs Data Mining

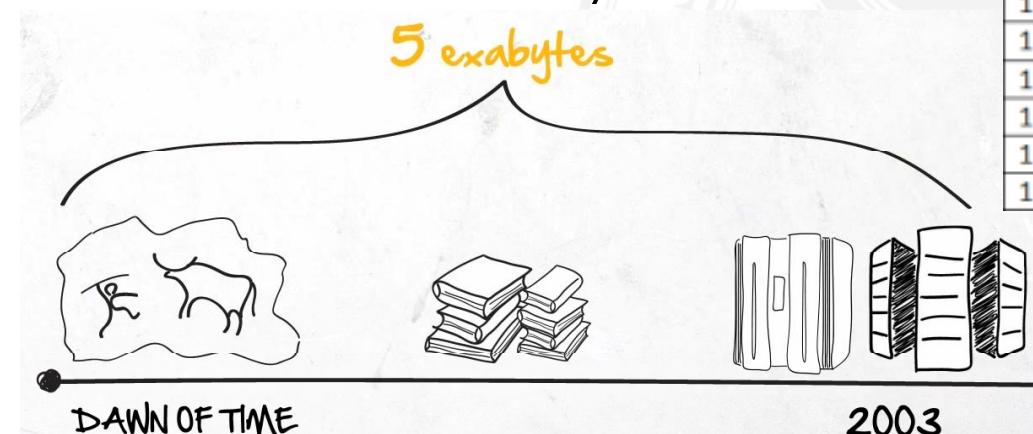
- ▶ El **Big Data** es un conjunto de datos tan grande y/o complejo que no es posible procesarlos con las herramientas tradicionales de minería de datos.
- ▶ Data mining is the old big data

	Statistics	Data Mining	Big Data
Structure	structured	structured	unstructured
Size	small	large	very large
Generation	planned	transactional	behavioral
Aim	understand	optimize business	generate business
Privacy Issues	non	minor	huge
Founded On	concepts & theory	technology & tool	technology & tools
Marketing	bad	good	perfect



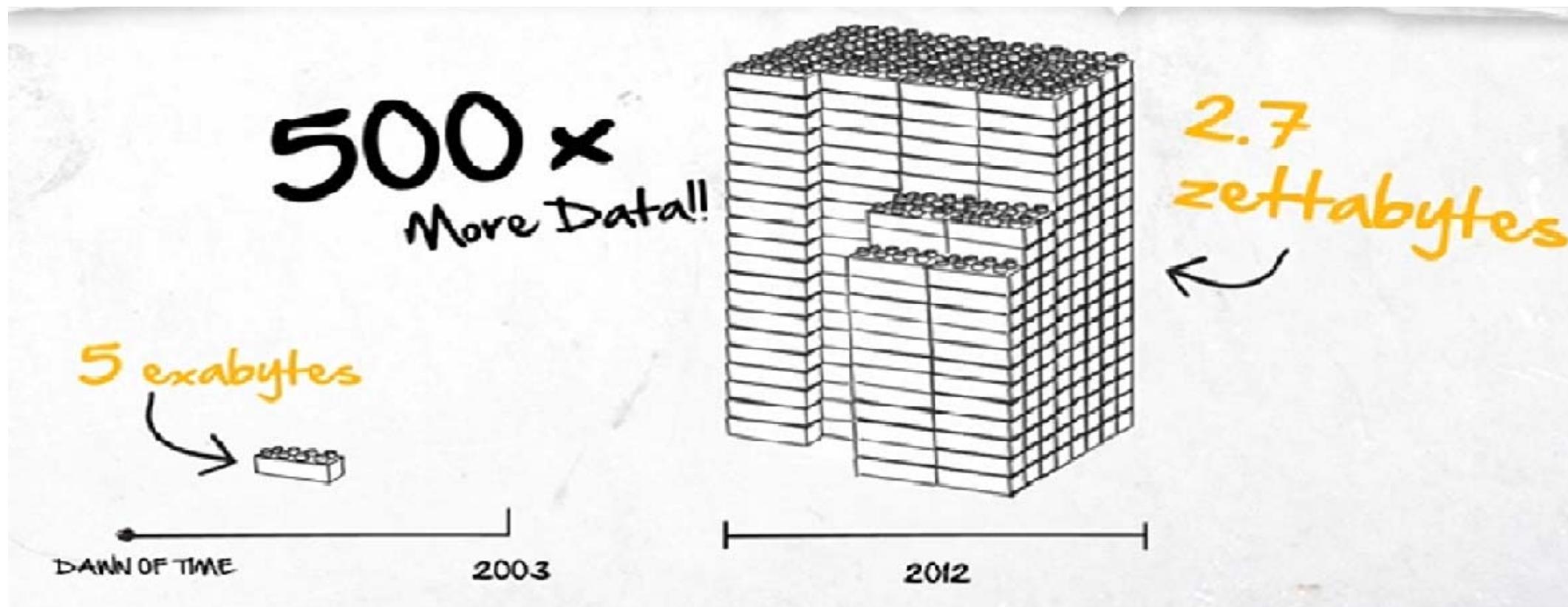
Big Data

- ▶ Eric Schmidt (ex CEO de Google): “entre el nacimiento del mundo y el año 2003, hubo cinco exabytes de información creada. Actualmente creamos cinco exabytes cada dos días”.
- ▶ Aunque otros estudios suponen unos 3,35 EB al día, algo más de lo que decía Eric Schmidt.
- ▶ El 99,9% de la información está en formato digital, o al contrario, sólo el 0,007% de la información del planeta está en papel.
- ▶ <http://www.worldometers.info/>



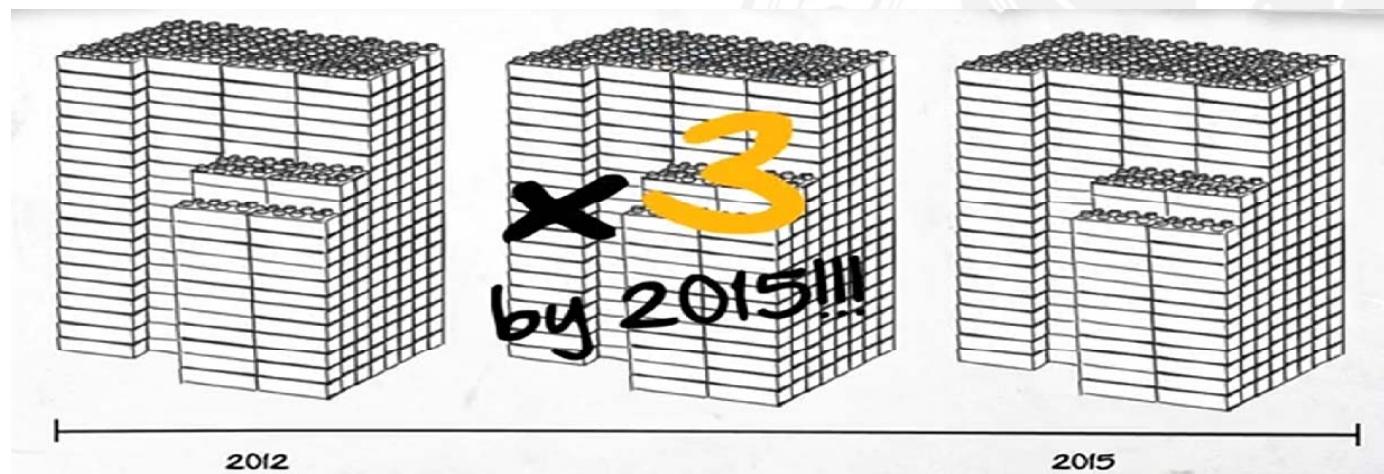
DECIMAL	
1 byte (B)	8 bits
1 kilobyte (KB)	1000 B
1 megabyte (MB)	1000 KB
1 gigabyte (GB)	1000 MB
1 terabyte (TB)	1000 GB
1 petabyte (PB)	1000 TB
1 exabyte (EB)	1000 PB
1 zettabyte (ZB)	1000 EB
1 yottabyte (YB)	1000 ZB

Big Data



Big Data

- El informe Cisco VNI (Visual Networking Index) 2011-2016 destaca distintos factores como los principales responsables de la evolución del tráfico IP: El creciente número de dispositivos conectados (relojes, enchufes, cámaras, etc.), más usuarios de Internet (en 2016 el 45% de la población), mayor velocidad de la banda ancha (9 Mbps de 2011 hasta los 34 Mbps en 2016.), más vídeo, Crecimiento Wi-Fi (el 50% en 2016)..
- El Informe Cisco VNI prevé que el tráfico IP global supere la barrera del Zettabyte y que haya 19.000 millones de dispositivos conectados, casi 2,5 conexiones por cada persona del planeta.
- En España, el tráfico IP se multiplicará por 13 entre 2011 y 2016 y habrá 258 millones de dispositivos conectados (5,1 conexiones por habitante).

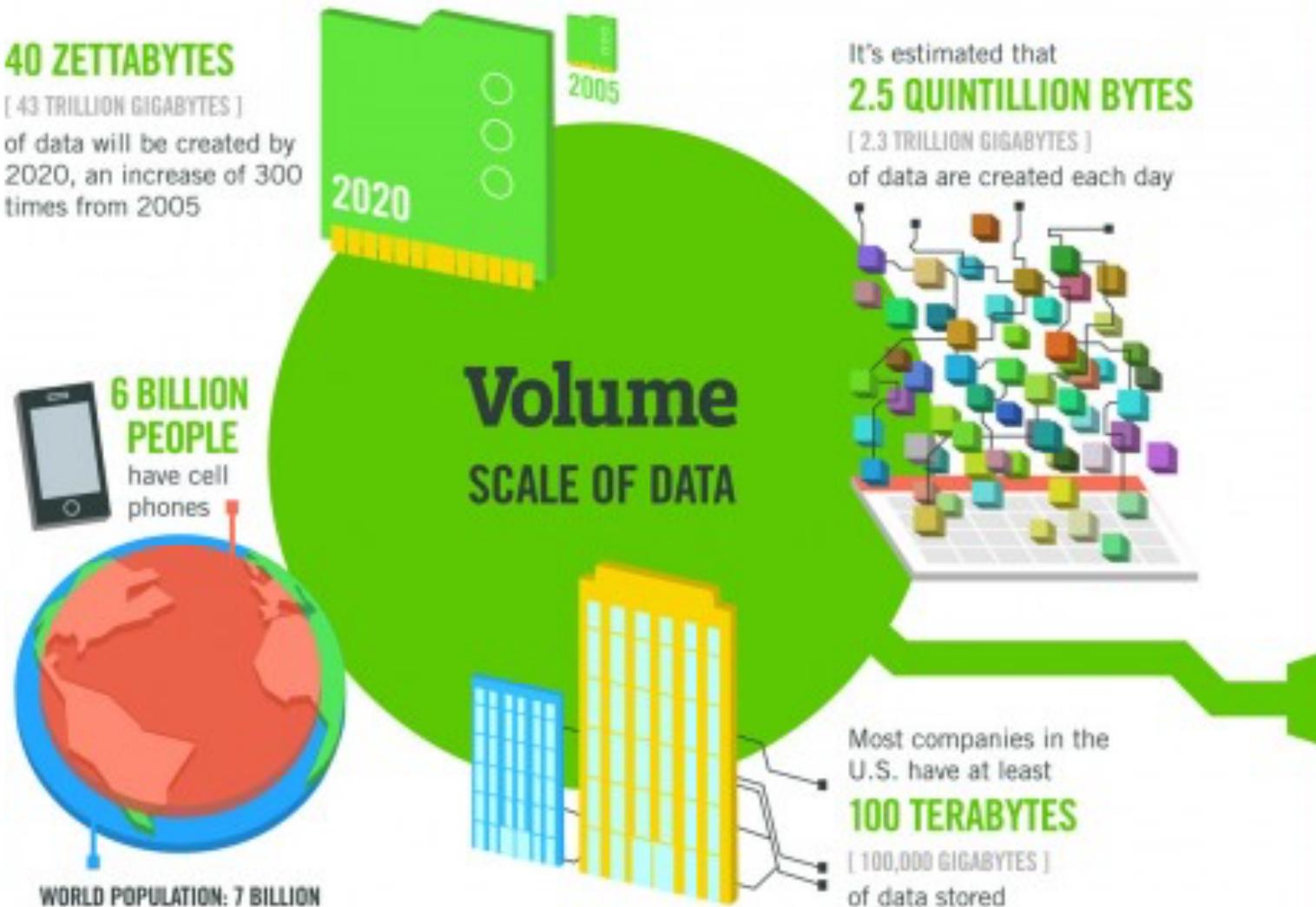


Tipos de datos en Big Data

- ▶ **Datos estructurados:** Datos que tienen bien definidos su longitud y su formato, como las fechas, los números o las cadenas de caracteres. Se almacenan en tablas (BD y hojas de cálculo).
- ▶ **Datos no estructurados:** Datos que carecen de un formato específico. No se pueden almacenar dentro de una tabla ya que no se puede desgranar su información a tipos básicos de datos. Algunos ejemplos son los PDF, documentos multimedia, e-mails o documentos de texto.
- ▶ **Datos semiestructurados:** Datos que no se limitan a campos determinados, pero que contiene marcadores para separar los diferentes elementos. Por ejemplo HTML, XML o JSON.

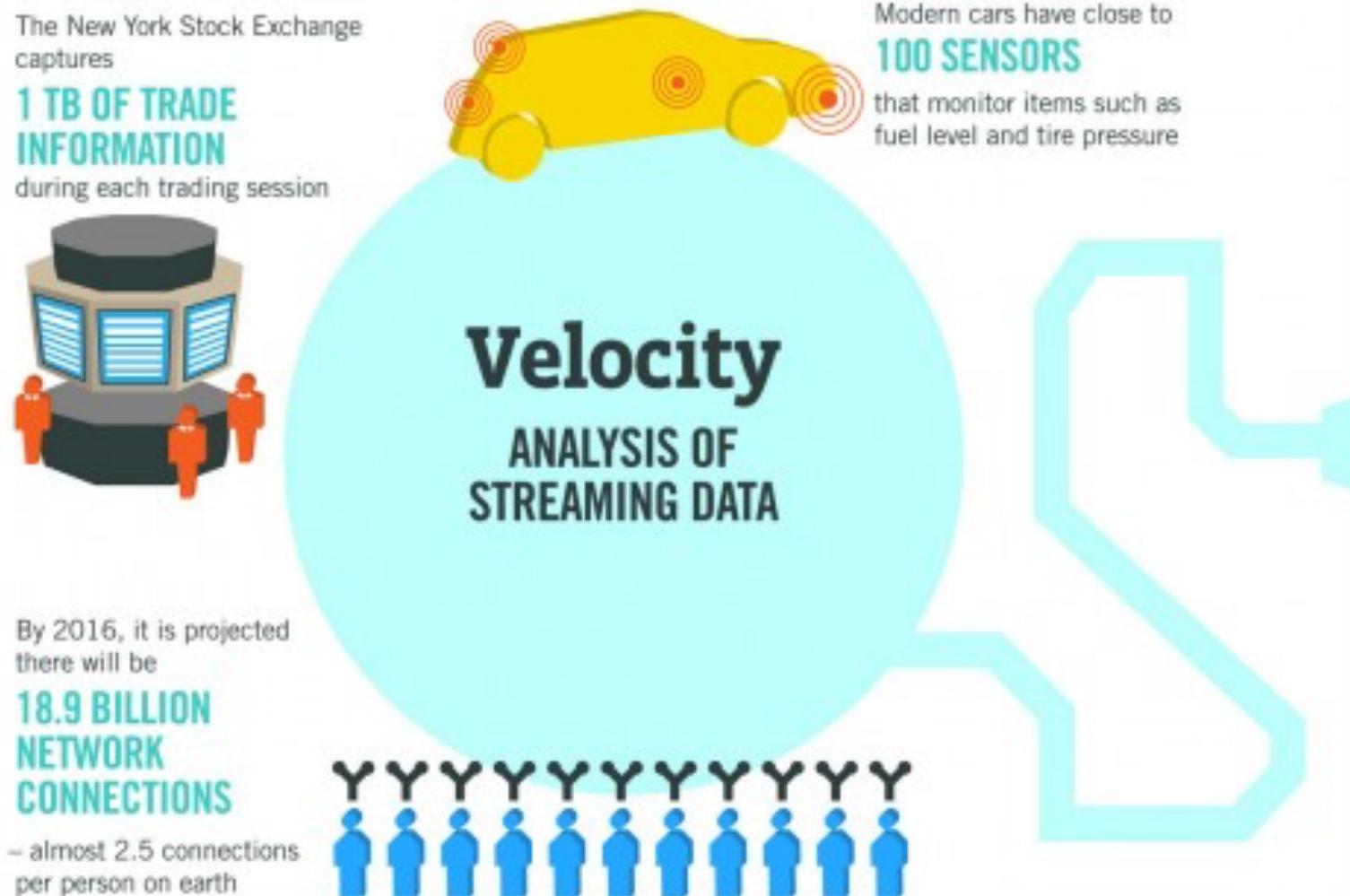
¿Qué es big data? Las 4 Vs del Big Data

► El **Volumen** de los datos crece exponencialmente.



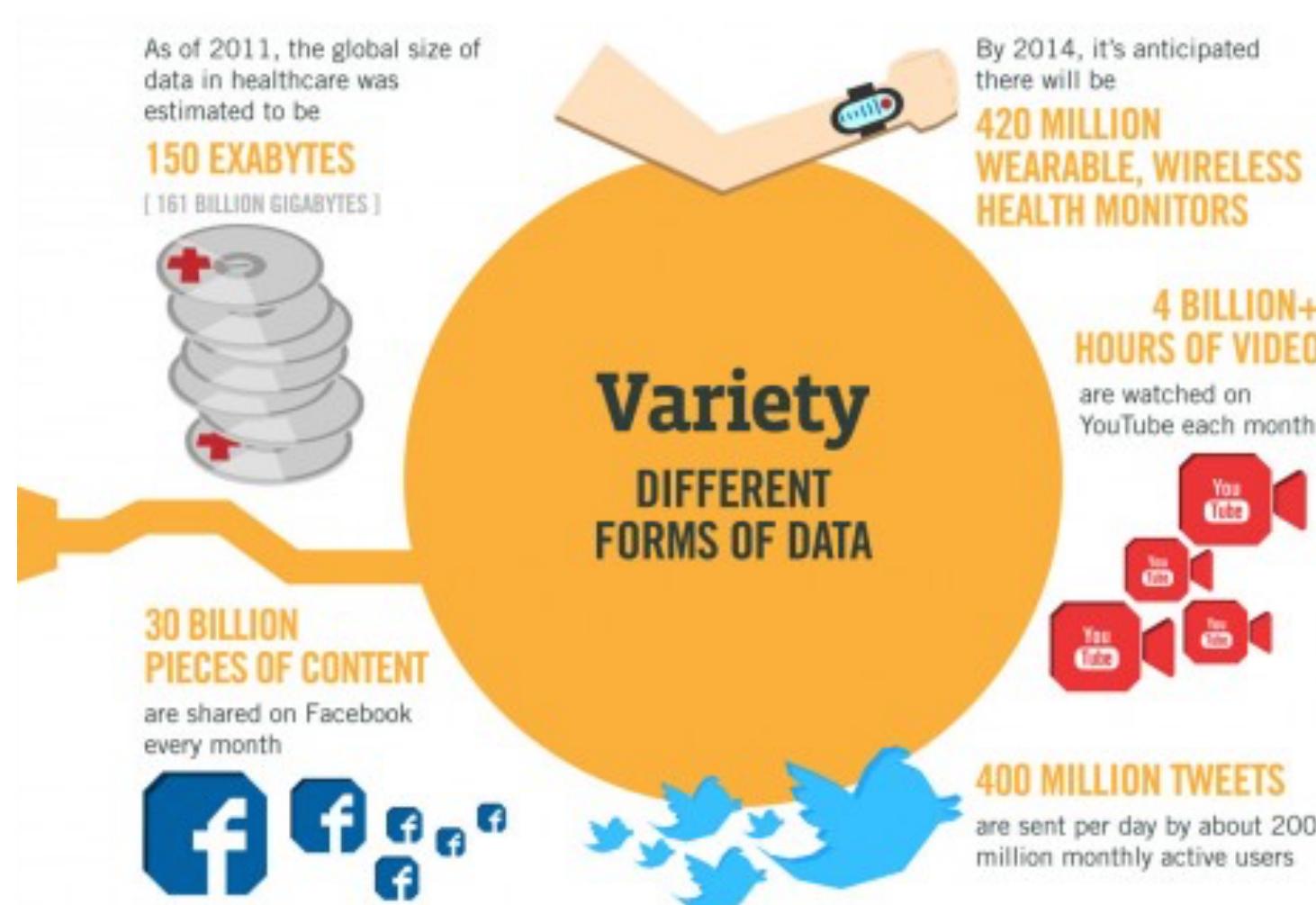
¿Que es big data? Las 4 Vs del Big Data

- ▶ Los datos se generan y deben ser procesados a gran **Velocidad**, por ejemplo, publicidad basada en la localización.



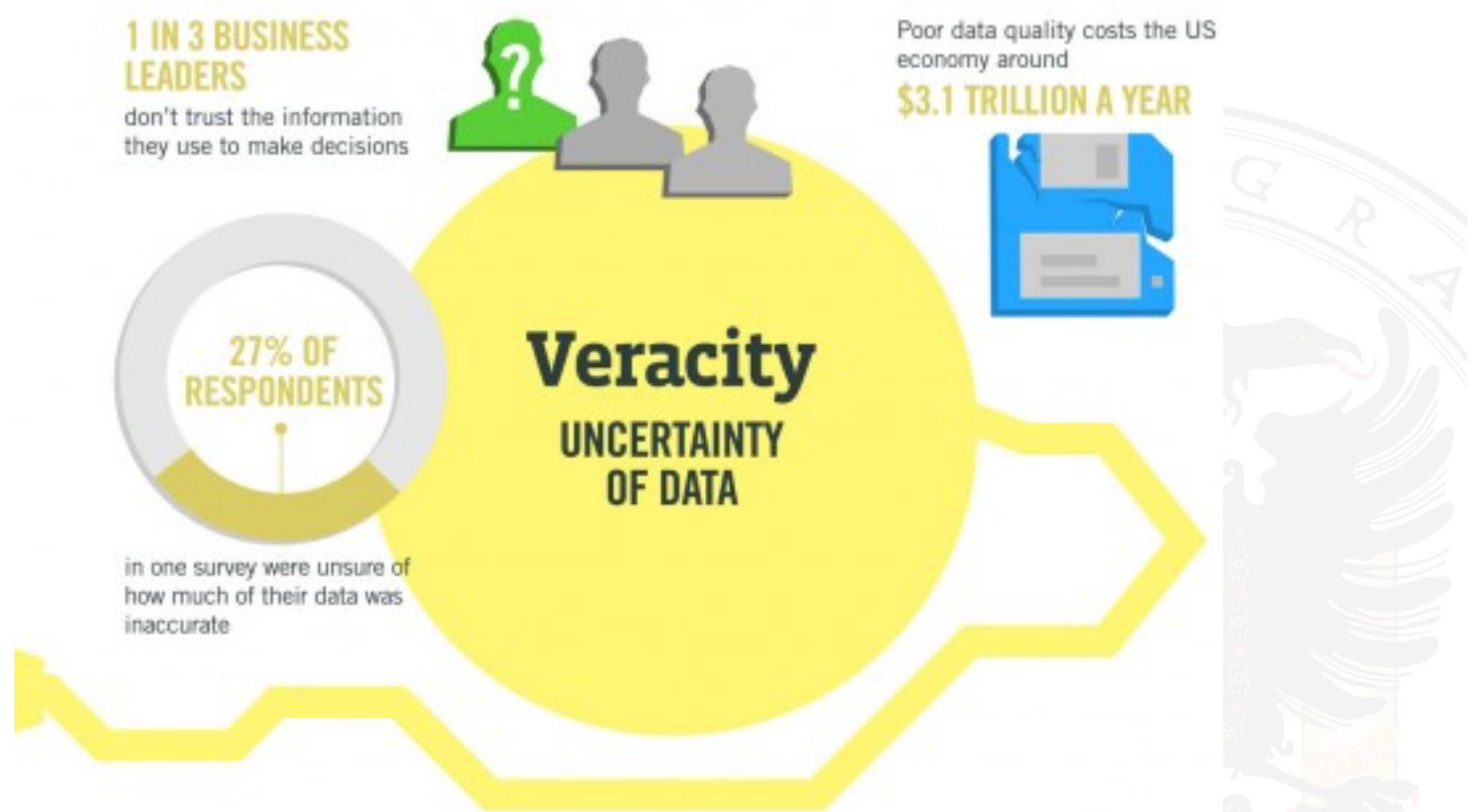
¿Que es big data? Las 4 Vs del Big Data

- Tenemos gran **Variedad** de datos (una misma aplicación puede generar distintos tipos de datos) y de fuentes.



¿Que es big data? Las 4 Vs del Big Data

- ▶ Veracidad: Tenemos incertidumbre asociada a los datos debido a datos inconsistentes, incompletos, ambiguos o erróneos.



Origen del Big Data

- ▶ **Generados por las personas:** Se estima que cada minuto al día se envían más de 200 millones de e-mails y se realizan 2 millones de búsquedas en Google. También se generan nuevos registros en BD u hojas de cálculo, mensajes por whatsapp, llamadas telefónicas, registros médicos, notas de voz, contestar encuestas, etc.
- ▶ **Redes Sociales y páginas web:** Cada minuto se comparten más de 700.000 contenidos en FaceBool o se editan 48h de vídeo en Youtube. En la web se utilizan muchas herramientas (por ejemplo, Google Analytics) de tracking utilizadas en su mayoría con fines de marketing y análisis de negocio. Los movimientos de ratón quedan grabados en mapas de calor y queda registro de cuánto pasamos en cada página y cuándo las visitamos.
- ▶ **Transacciones de datos:** La facturación, los registros detallados de las llamadas, las transacción entre cuentas, compra por internet o movimientos en la bolsa.

Origen del Big Data

- ▶ **Machine to machine (M2M)**: información generada por dispositivos que se la envían a otros. M2M utiliza dispositivos como sensores o medidores que capturan algún evento en particular (telecopios, velocidad, temperatura, posicionamiento, movimiento, presión, variables meteorológicas, variables químicas como la calidad del agua, etc.) Existen desde hace décadas, pero la llegada de las comunicaciones inalámbricas (Wi-Fi, Bluetooth, RFID...) ha revolucionado el mundo de los sensores. Algunos ejemplos son los GPS en la automoción o los sensores de signos vitales en la medicina.
- ▶ **Biométrica**: Información biométrica en la que se incluye huellas digitales, escaneo de la retina, reconocimiento facial, genética, etc. Un ejemplo de aplicación es el cruce de ADN entre una muestra de un crimen y una muestra en nuestra base de datos.

Metodología en el Big Data

Hemos visto que el **Big Data** es un conjunto de datos tan grande y/o complejo que no es posible procesarlos con las herramientas tradicionales de minería de datos. Veamos qué tenemos que cambiar:

- ▶ **Almacenamiento de la información:** Las BD tradicionales no nos permiten trabajar de forma eficiente con tanta información, por tanto, tendremos que recurrir a BD NoSQL
- ▶ **Acceso y procesamiento de la información:** Es inviable que todo el trabajo con los datos se haga en sólo computador, tenemos que parallelizar el trabajo, por ejemplo, con la metodología MapReduce.
- ▶ **Minería de datos:** La minería de datos tradicional puede ser demasiado dificultosa para tal cantidad de datos, además debe poder realizarse de forma distribuida.

Bases de Datos NoSQL

- ▶ El término NoSQL se refiere a Not Only SQL y son sistemas de almacenamiento que no cumplen con el esquema entidad-relación. Por lo que sería más acertado llamarlas BD No Relacionales.
- ▶ Proveen un sistema de almacenamiento mucho más flexible y concurrente y permiten manipular grandes cantidades de información de manera mucho más rápida que las bases de datos relacionales.
- ▶ Aunque las BD relacionales son bastante escalables, con el big data la complejidad y la eficiencia no son tan buenas (no son altamente escalables).
- ▶ Los sistemas NoSQL son más rápidas pero a costa de perder ciertas funcionalidades como las transacciones que engloban operaciones en más de una colección de datos, o la incapacidad de ejecutar el producto cartesiano de dos tablas (también llamado JOIN) teniendo que recurrir a la desnormalización de datos.

Bases de Datos NoSQL

Distinguimos cuatro grandes grupos de bases de datos NoSQL:

- ▶ **Almacenamiento Clave-Valor (Key-Value):** Los datos se almacenan de forma similar a las tablas hash o diccionarios de datos, donde se accede al dato a partir de una clave única. este sistema de almacenamiento carece de una estructura de datos clara y establecida, por lo que no requiere un formateo de los datos muy estricto
- ▶ **Almacenamiento Documental:** Las bases de datos documentales guardan un gran parecido con las bases de datos Clave-Valor, diferenciándose en que en este caso guardamos datos semiestructurados. Estos datos pasan a llamarse documentos, y pueden estar formateados en XML, JSON.
- ▶ **Almacenamiento en Grafo:** Las bases de datos en grafo establecen que la información son los nodos y las relaciones entre la información son las aristas, ejemplo Facebook (nodos usuarios, aristas amistades)
- ▶ **Almacenamiento Orientado a Columnas:** Su modelo de datos es definido como “un mapa de datos multidimensional poco denso, distribuido y persistente”. Permite guardar diferentes atributos y objetos bajo una misma Clave.

MapReduce

► **MapReduce** es un modelo de programación utilizado por Google para dar soporte a la **computación paralela** sobre Big Data. Por ejemplo:

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

► Existe una implementación OpenSource denominada **Hadoop** cuyo desarrollo lo realiza el proyecto Apache.

► MapReduce se emplea en la resolución práctica de algunos algoritmos susceptibles de ser paralelizados. No obstante MapReduce **no es adecuado para todos los problemas**, pero cuando funciona es muy eficiente.

► Por regla general se abordan problemas con datasets de gran tamaño, alcanzando los **petabytes de tamaño**. Es por esta razón por la que este framework suele ejecutarse en **sistema de archivos distribuidos** (HDFS).

► Muchos sistemas **NoSQL permiten utilizar consultas del tipo Map-Reduce**, las cuales pueden ejecutarse en todos los nodos a la vez (cada uno operando sobre una porción de los datos) y reunir luego los resultados antes de devolverlos al cliente.

MapReduce

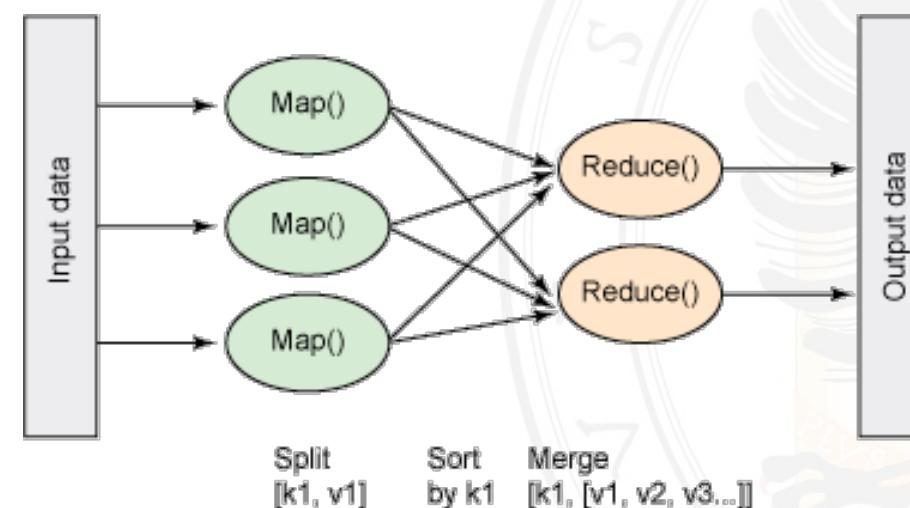
MapReduce se basa en dos procedimientos: Map() y Reduce():

- ▶ Map() toma pares Clave, Valor y devuelve una lista de pares en un dominio diferente:

$\text{Map}(k_1, v_1) \rightarrow \text{list}(k_2, v_2)$

- ▶ Reduce() es aplicada en paralelo para cada grupo, produciendo una colección de valores para cada dominio:

$\text{Reduce}(k_2, \text{list } (v_2)) \rightarrow \text{list}(v_3)$

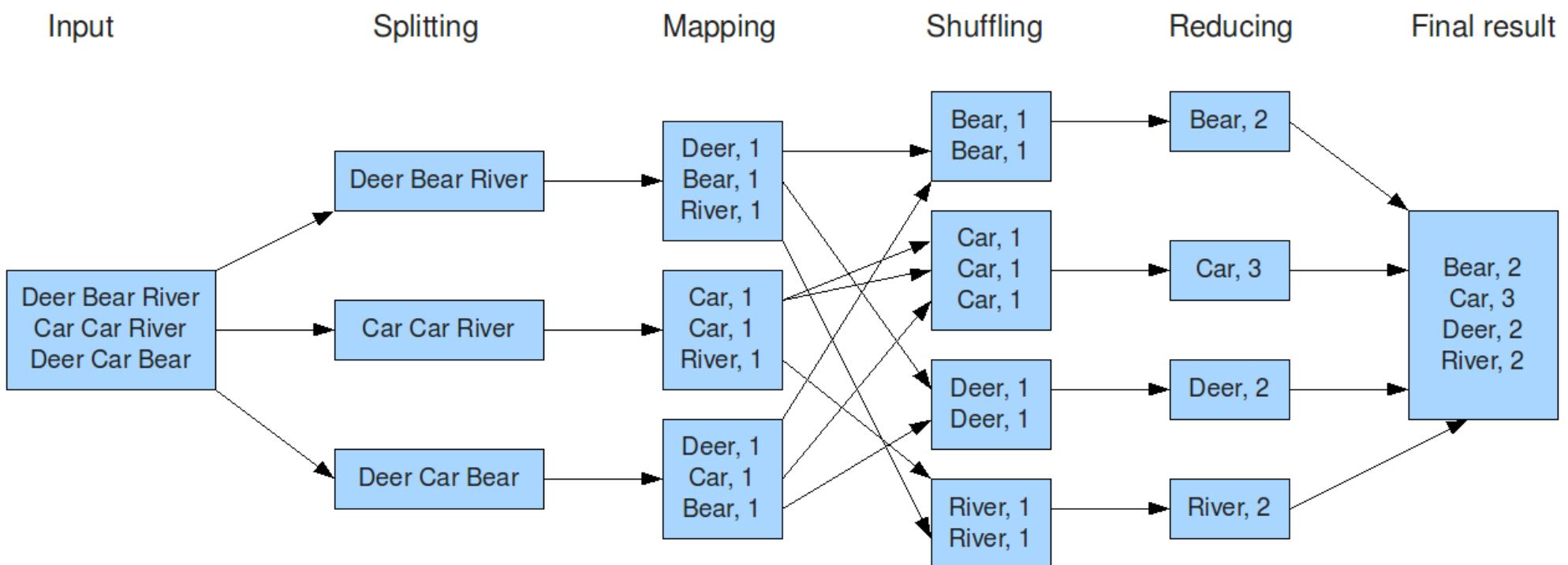


MapReduce: Ejemplo

- ▶ Ejemplo contar las apariciones de cada palabra en un conjunto de documentos
- ▶ La función map() en este caso divide un documento en palabras (es decir lo tokeniza) mediante el empleo de un simple analizador léxico, y emite una serie de tuplas de la forma (clave, valor) donde la clave es la palabra y el valor es "1". Por ejemplo, del documento "La casa de la pradera" la función map retornaría: ("la", "1"), ("casa", "1"), ("de", "1"), ("la", "1"), ("pradera", "1").
- ▶ Se reúne todos los pares con la misma clave y se pasa a Reduce, por lo tanto, esta función sólo necesita la suma de todos los valores de su entrada para encontrar el total de las apariciones de esa palabra. En el ejemplo anterior ("la", "1") aparece dos veces debido a que la clave "la" tiene dos ocurrencias, el resto de claves sólo aparece una vez.

MapReduce: Ejemplo

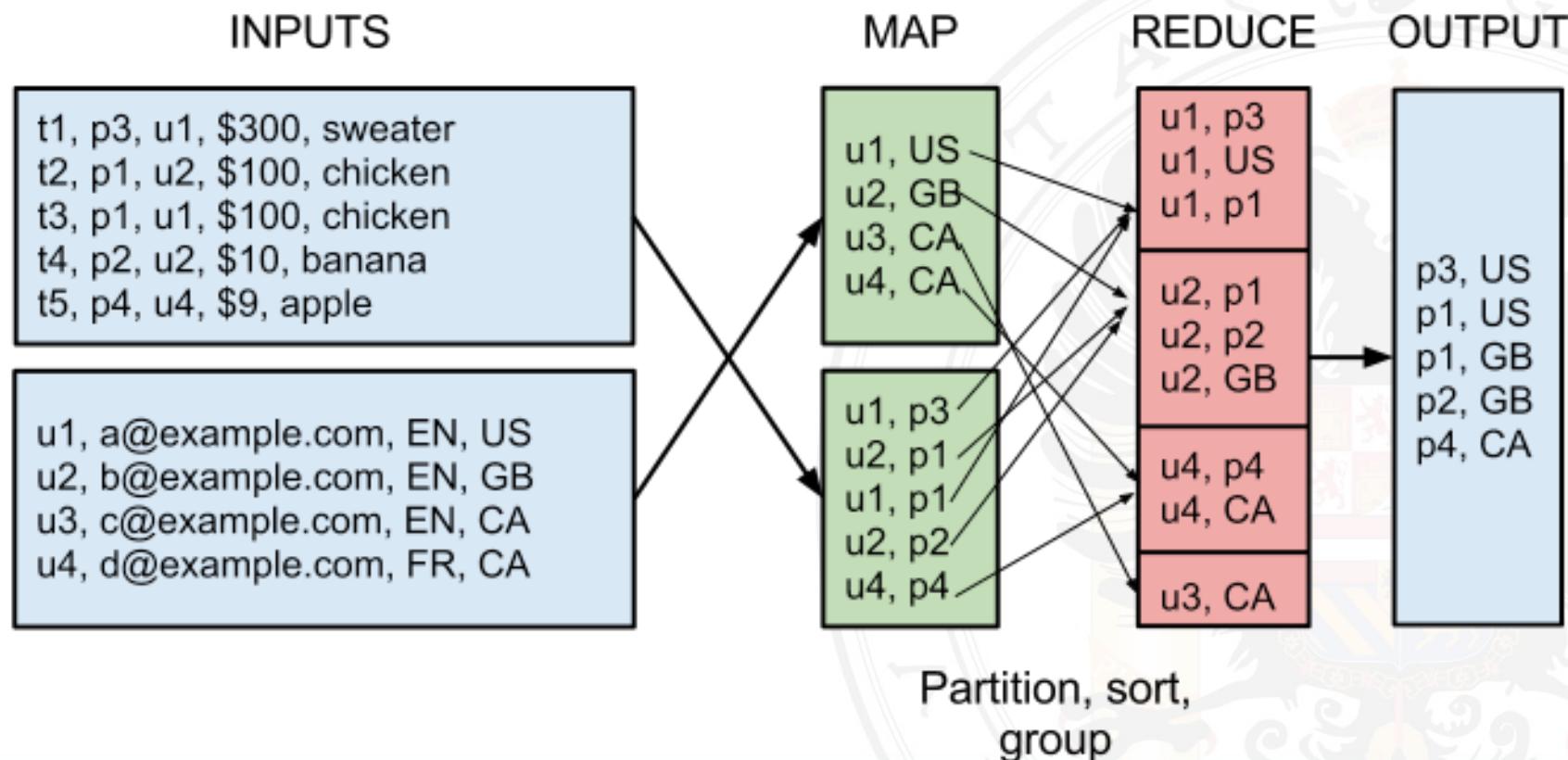
The overall MapReduce word count process



MapReduce: Ejemplo

Quiero saber la localización de los productos vendidos y tengo dos bases de datos:

- ▶ Usuarios (id, email, language, location)
- ▶ Transacción (transaction-id, product-id, user-id, purchase-amount, item-description)



MapReduce: Características

► Paralelización automática:

- Dependiendo del tamaño de entrada de datos, se crean múltiples tareas Map.
- Dependiendo del número intermedio de particiones <clave,valor> se crean tareas Reduce.

► Escalabilidad:

- Funciona sobre cualquier cluster de procesadores.
- Puede trabajar desde 2 a 10.000 máquinas

► Transparencia programación:

- Manejo de fallos de la máquina
- Gestión de comunicación entre máquina

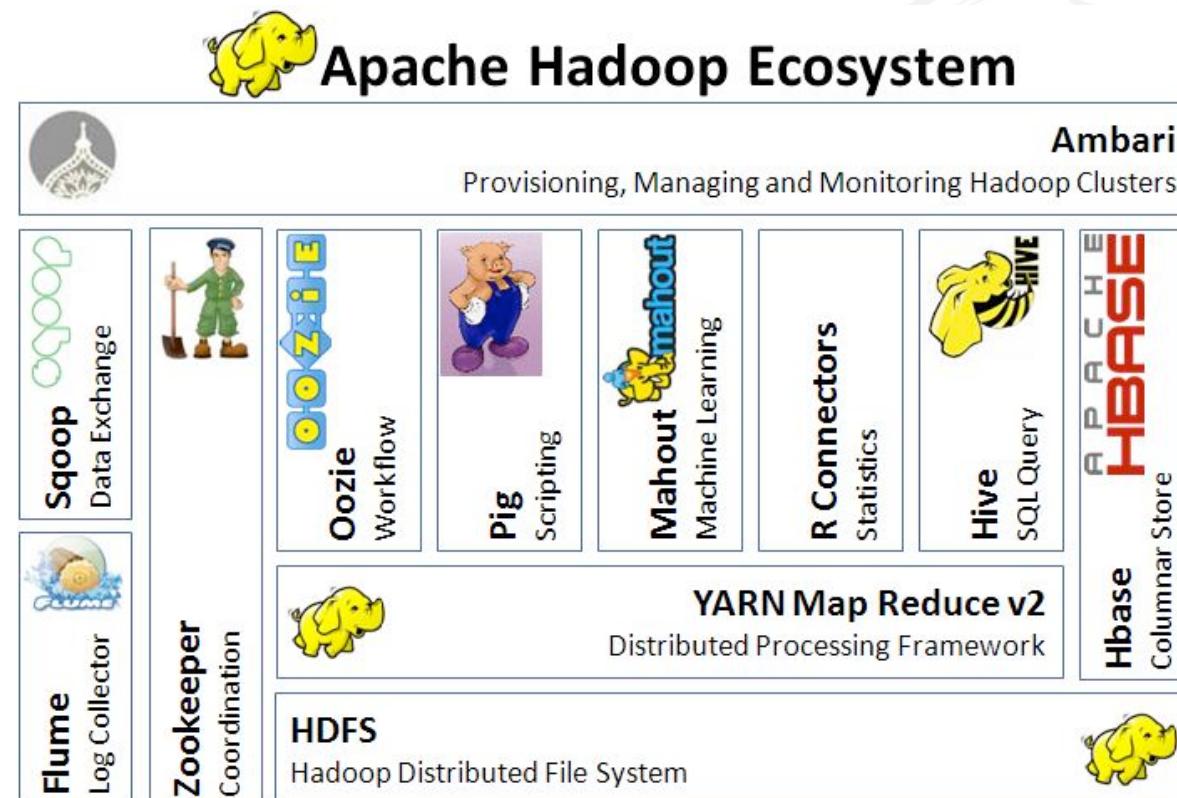
MapReduce: Ventajas

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
- Es muy escalable
- Más barato: se ahorran costes en hardware, programación y administración.

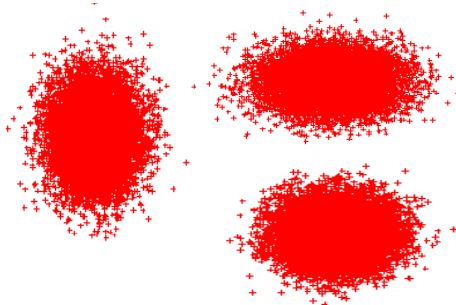
Hadoop

- ▶ Hadoop es una implementación OpenSource del modelo MapReduce.
- ▶ Su desarrollo fue liderado inicialmente por Yahoo y actualmente lo realiza la fundación Apache.
- ▶ <http://hadoop.apache.org/>

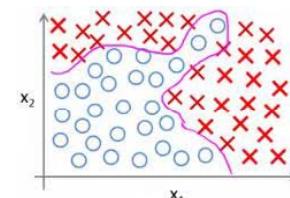


Mahout

- Apache™ Mahout es una librería para algoritmos de aprendizaje automático escalables.
- Está implementado en Hadoop y utiliza el paradigma MapReduce.
- <http://mahout.apache.org/>
- <https://mahout.apache.org/users/basics/algorithms.html>

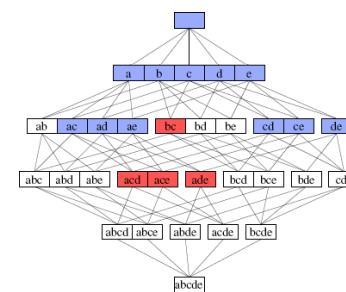


Agrupamiento



Clasificación

Asociación



Sistemas de Recomendaciones



Mahout: Ejemplo KddCup99

- Random Forest para KddCup99 usando un cluster de 12 nodos, cada nodo 6cores/12hebras

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

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