Sistemas de adquisición y gestión de datos. Algoritmos Big Data en IoT

Aurora González Vidal

Universidad de Murcia

aurora.gonzalez2@um.es

03/03/2025

Overview

Introducción al Big Data desde el IoT

El Big Data y el internet de las cosas o *Internet of Things* (IoT) son diferentes pero están conectados. Usados de forma conjunta, el IoT provee la información de la cual se puede extraer conocimiento a través del análisis de datos que utiliza herramientas Big Data. El internet de las cosas nos provee de datos que cambian muy rápidamente procedentes de dispositivos en todo el mundo. El reto para muchos investigadores y organizaciones consiste en extraer conocimiento de todos esos datos.

Cada "cosa" (*thing*) o dispositivo inteligente (*smart device*) es un aparato con un sistema electrónico y software embebido que puede actuar como sensor o como actuador (*actuator*).

- Sensor: informa del estado actual del mundo real a su alrededor
- Actuador: alteran el estado del sistema como respuesta de una señal de control

"Un sensor de presencia le indica al sistema de control que una persona está cruzando el pasillo de noche, por lo que hay que activar una serie de luces, para conseguirlo activa un contactor (actuador) que provoca que las bombillas se enciendan".

1. ¿Cuántos tweets se envían al día? (No ha cambiado desde 2014)
   - 1.1 100 millones
   - 1.2 500 millones
   - 1.3 3 millones
2. ¿Qué porcentaje de la población tiene acceso a internet?
   - 2.1 66 %
   - 2.2 30 %
   - 2.3 78 %
3. ¿Cuántos sensores IoT están conectados hoy día?
   - 3.1 Más de 1 billón (1000 millones)
   - 3.2 Más de 10 millones
   - 3.3 Más de 15 billones (10000 millones)

Fuentes: https://thesocialshepherd.com/blog/twitter-statistics
https://www.forbes.com/home-improvement/internet/internet-statistics/
https://explodingtopics.com/blog/number-of-iot-devices

Algunas formas en las que generamos datos diariamente. ¿Ideas?

Algunas formas en las que generamos datos diariamente. ¿Ideas?

- Búsquedas online
- Tomar fotos y vídeos
- Escribir emails
- Escribir documentos
- Usar un mapa o ser localizado.
- Rellenar cuestionarios online
- Andar con el smartphone
- Publicar en las redes sociales
- Enviar un mensaje de texto

El crowdsensing se encarga de recoger la información que ofrecen los seres humanos al interaccionar con diferentes tecnologías.
El crowdsening entre usuarios de teléfonos móbiles y otros aparatos forma parte del IoT. Con la valiosa capacidad sensorial de los smart devices y mobilidad de los usuarios, se pueden proveer varios servicios si se construye una cadena de confianza entre los que demandan y los que proveen.

Formas de crowdsensing (escalabilidad)

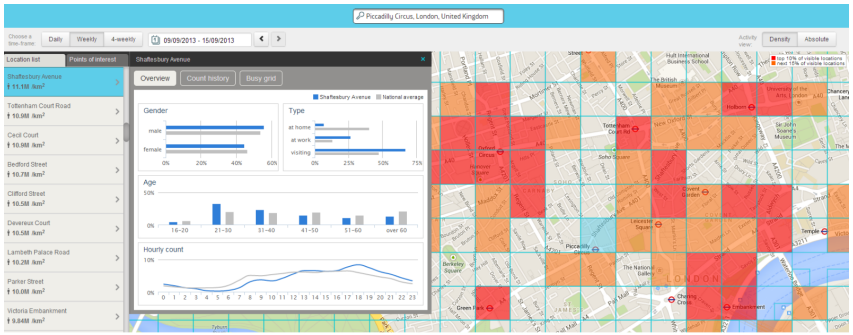
- Recogida separada
- Recogida en grupo (cluster)
- Recogida en comunidad

Formas de crowdsensing (implicación)

- Oportunista
- Participatorio

Ejemplos de crowdsensing

Los datos sociodemográficos de los barrios de Londres permitían predecir en un 62 por ciento la probabilidad de que hubiese un crimen.



En 2014, gracias a "Smart Steps", que es una herramienta estadística que analiza tendencias de grupos de personas, el porcentaje alcanza el 70 por ciento sólo mediante la combinación de esos mismos datos obenidos del Open Data Institute con datos móviles de red (estimación cuánta gente hay en cada barrio).

Fuente: Bogomolov, Andrey, et al. "Once upon a crime: towards crime prediction from demographics and mobile data." Proceedings of the 16th international conference on multimodal interaction. ACM, 2014.

Una investigación reciente estudió la huella de carbón producida tanto por los aparatos electrónicos (smartphones, portátiles, tablets...) como por *data centers* y las redes de comunicación desde 2005.
Para 2030 las TIC podríaN consumir 20-30 % de la energía global.
De mantenerse las tendencias actuales, para 2040, las TIC podrían ser responsables de hasta el 14 % de las emisiones globales de carbono, lo que equivaldría a la mitad de las emisiones generadas por el sector del transporte a nivel mundial.
Fuentes recientes respaldan estas proyecciones y enfatizan la necesidad de una transición hacia una economía digital más verde.

Fuentes:

1. Belkhir, Lotfi, and Ahmed Elmeligi. "Assessing ICT global emissions footprint: Trends to 2040 & recommendations." Journal of Cleaner Production 177 (2018): 448-463.
2. International Energy Agency. (2023, July 11). What the data centre and AI boom could mean for the energy sector. IEA. https://www.iea.org/commentaries/what-the-data-centre-and-ai-boom-could-mean-for-the-energy-sector

El crowdsensing consume muchos recursos que afectan a la calidad de los servicios

- Ancho de banda
- Energía
- Almacenamiento

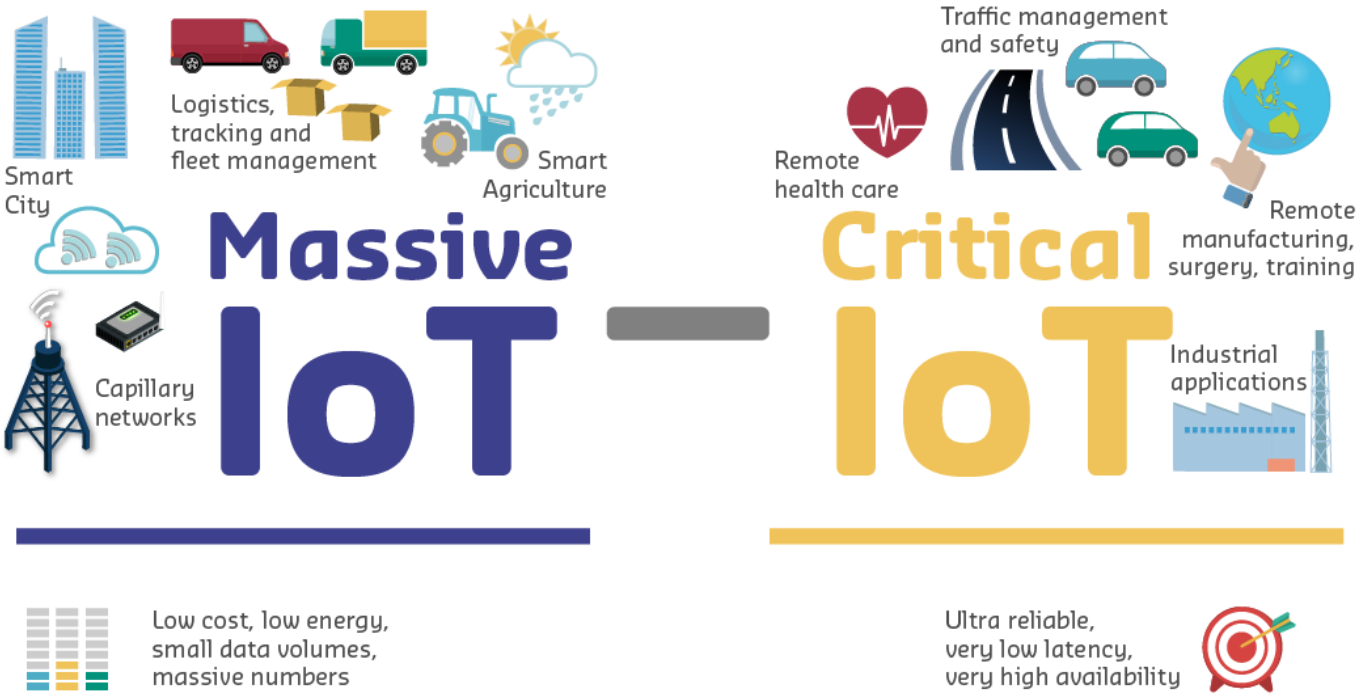
El crowdsensing se enfrenta a los siguientes retos:

- Redundancias y baja calidad de los datos
- Completitud de los datos
- Seguridad y privacidad

Fuente: Liu, Jinwei, et al. "A survey of mobile crowdsensing techniques: A critical component for the internet of things." ACM Transactions on Cyber-Physical Systems 2.3 (2018): 18.

Soluciones a los poblemas de la recogida de datos: uso inteligente del software. Dos conceptos relacionados con el uso inteligente del software para acatar este problema son:

- Preprocesamiento: imputar valores faltantes y otros procesamientos (algoritmos de reducción de datos)
  Ejemplo: datos de vídeo y audio
  Cuanto más cerca movamos el preprocesamiento a los sensores, la cantidad de datos que se envía se verá más reducida. Los sensores ya pueden realizar procesamientos sencillos (p.ej. no enviar datos cuando nada ocurre)
- Crear metadatos: pueden poner los datos guardados en contexto.


Smart City
Logistics, tracking and fleet management
Smart Agriculture
Capillary networks
Massive IoT
Low cost, low energy, small data volumes, massive numbers
Traffic management and safety
Remote health care
Remote manufacturing, surgery, training
Industrial applications
Critical IoT
Ultra reliable, very low latency, very high availability

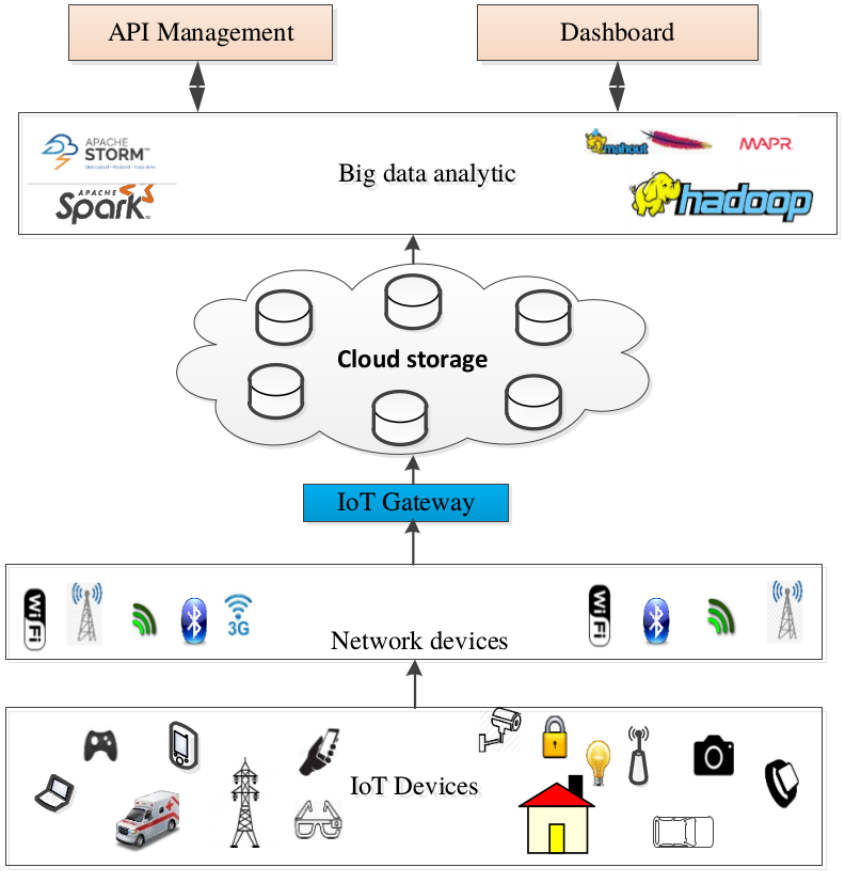
Características de los datos recogidos: espaciotemporalidad
La gran mayoría de los datos que se extraen de dispositivos inteligentes vienen acompañados de una fecha (*timestamp*) y una localización geográfica (longitud, latitud) y en muchas ocasiones una o ambas añaden conocimiento al problema en cuestión: modelos a utilizar.

Datos en el eje (at the edge)

Las arquitecturas IoT tienen muy diversas definiciones basadas en abstracciones e identificaciones de los sensores.
Los 7 niveles del modelo de referencia del IoT.

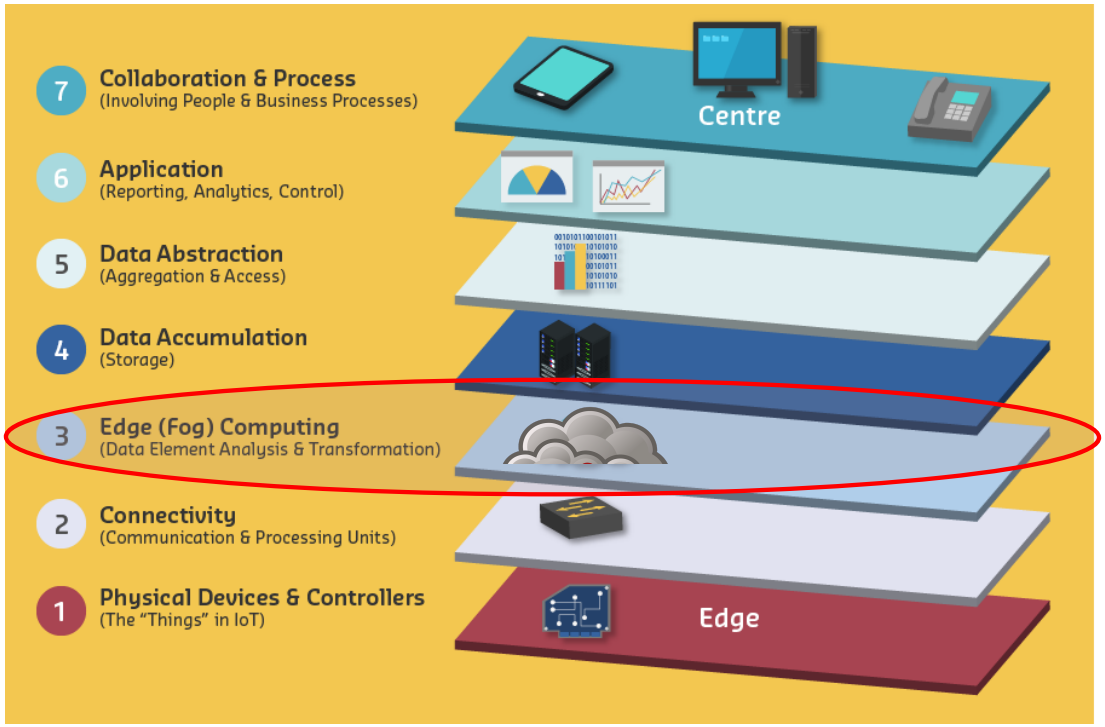
Fuente: Cisco, the Internet of Things Reference Model. White Paper (2014)

Arquitecturas IoT que integran el Big Data.



Fuente: Marjani, Mohsen, et al. "Big IoT data analytics: architecture, opportunities, and open research challenges." IEEE Access 5 (2017): 5247-5261.

En esta sección vamos a ver qué pasa con los datos "at the edge", es decir, ¿qué pasa cerca de los sensores? ¿cómo podemos trabajar con esa información y crear inteligencia? En lugar de enviar toda la información a la nube, podemos hacer procesamiento cerca del sensor (the edge) para conseguir respuestas más rápidas.



Diferencias entre Cloud, Edge y Fog computing.

- Cloud. Compañias: eficiencia (fácil montarlo, variable y provee de soporte)
  Extensión de los servicios:
- Edge
- Fog

Las razones principales por las que se procesa cerca del sensor son:

- Seguridad: menos oportunidad de que intercepten los datos
- Ancho de banda: es costoso.
- Tiempo de respuesta: no es nuestra red, no están bajo nuestro control.

Las arquitecturas no están preparadas para transportar la gran cantidad de datos que se generan a nivel de sensores, por lo que fog y edge computing pueden ser la clave.

Sistema de vigilancia puede ser un buen ejemplo. ¿Cloud, Edge, Fog?

Sistema de vigilancia puede ser un buen ejemplo. ¿Cloud, Edge, Fog?

- Cloud: cámara continuamente conectada a la nube y todo se procesa

Sistema de vigilancia puede ser un buen ejemplo. ¿Cloud, Edge, Fog?

- Cloud: cámara continuamente conectada a la nube y todo se procesa
- Edge: un sensor de movimiento. Cuando se detecta movimiento, se envían las imágenes a la nube

Sistema de vigilancia puede ser un buen ejemplo. ¿Cloud, Edge, Fog?

- Cloud: cámara continuamente conectada a la nube y todo se procesa
- Edge: un sensor de movimiento. Cuando se detecta movimiento, se envían las imágenes a la nube
- Fog: alguien entra y sale (no es emergencia), o es un animal.

Sistema de vigilancia puede ser un buen ejemplo. ¿Cloud, Edge, Fog?

- Cloud: cámara continuamente conectada a la nube y todo se procesa
- Edge: un sensor de movimiento. Cuando se detecta movimiento, se envían las imágenes a la nube
- Fog: alguien entra y sale (no es emergencia), o es un animal.

Avión: ejemplo de Fog node. Produce: 10 terabytes/h. Ancho de banda limitado (vía satélite).

La hegemonía de la nube está llegando a su fin.

La proliferación del IoT es una de las razones por las cuales el análisis está migrando desde lo centralizado (nube) hasta los ejes de las redes y los aparatos inteligentes.

Fuente:
https://www.economist.com/business/2018/01/18/the-era-of-the-clouds-total-dominance-is-drawing-to-a-close

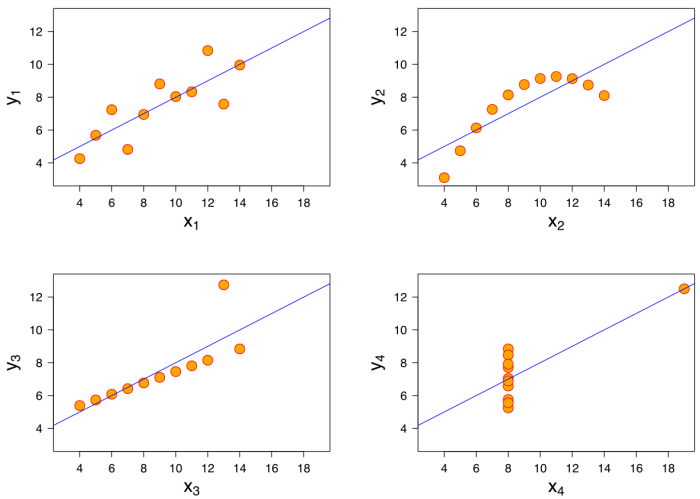
Algoritmos en el eje y fog

Las técnicas analíticas que son apropiadas a nivel de sensor suelen venir definidas como aquellas para:

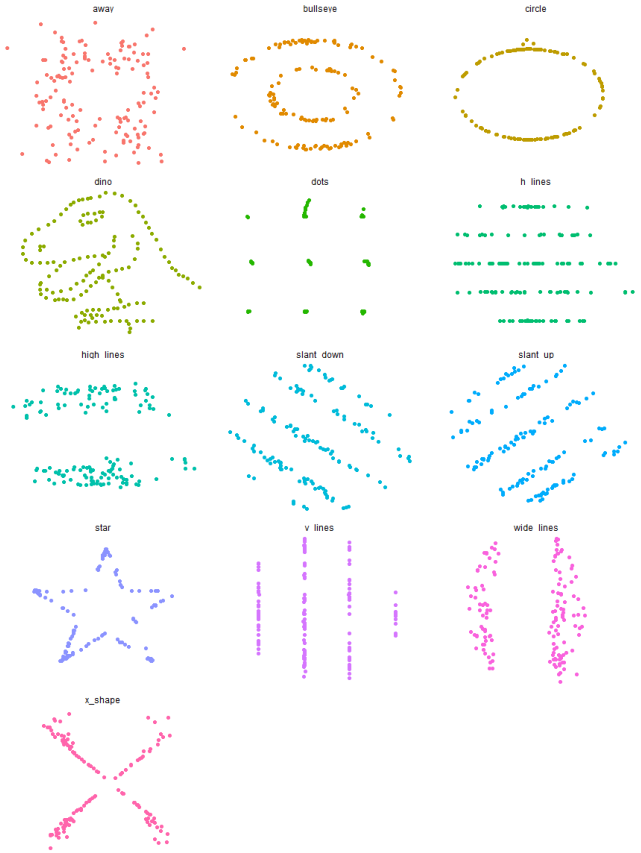
- *Constrained-resource* o recursos restringidos
- *Not compute-intensive* o no exigentes computacionalmente

Más allá de estadísticos básicos (medias, desviaciones estándard, correlaciones...)

Cuarteto de Ascombe



The Datasaurus Dozen



Para reducir el número de puntos de datos en una serie y crear representaciones se pueden utilizar métodos de segmentación como una parte del pre-procesamiento de datos.

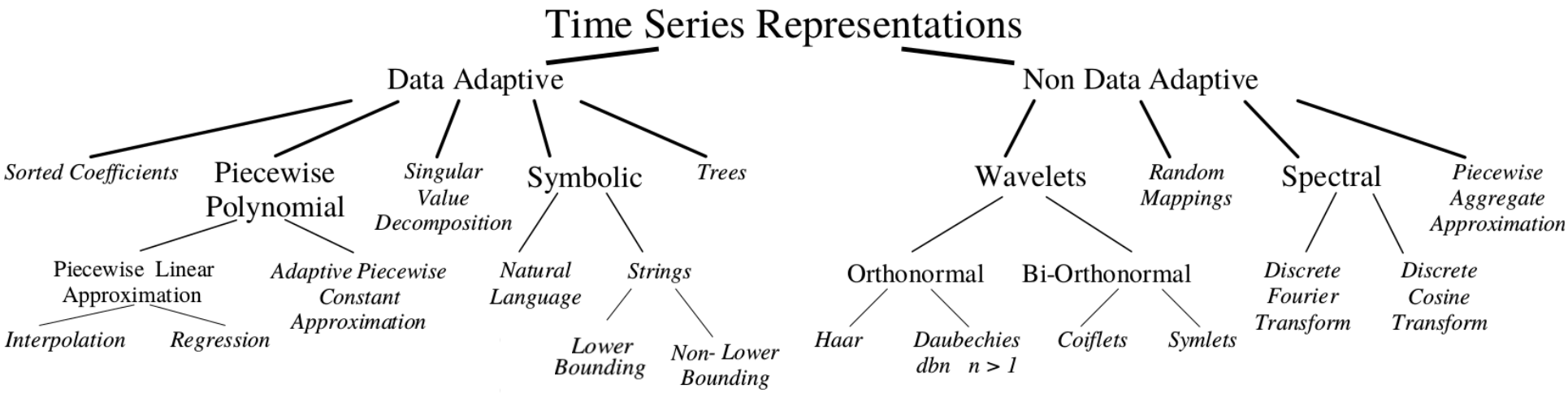
Definition

Segmentación

Dada una serie temporal T que contiene n puntos, la segmentación se define como la construcción de un modelo $\bar{T}$, de l segmentos "a trozos" $(l < n)$ tal que $\bar{T}$ se aproxima a T.

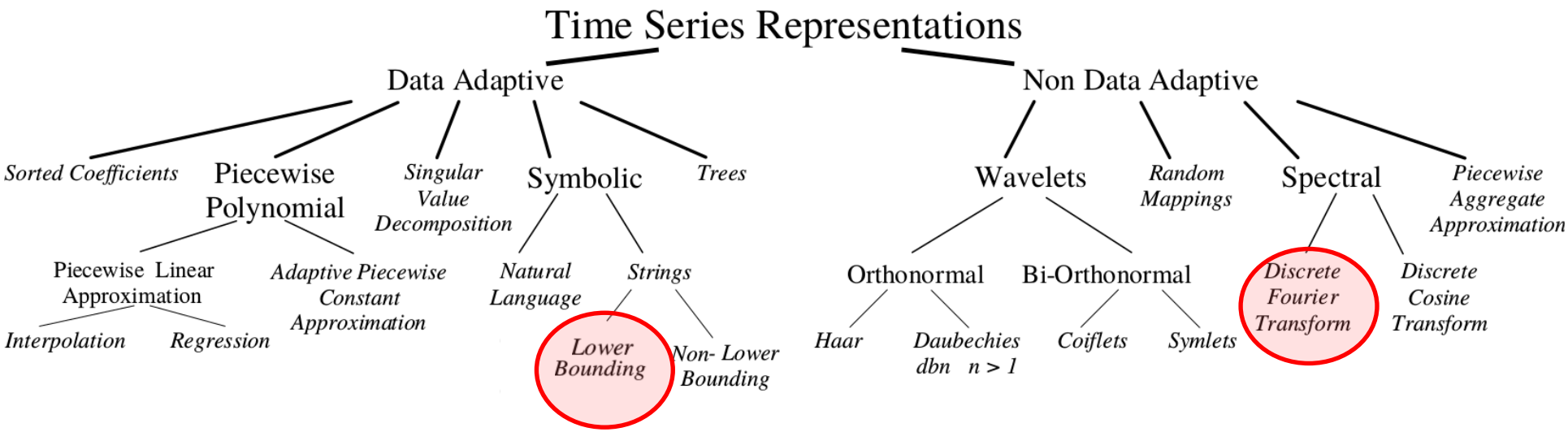
Fuente: Keogh, Eamonn J., and Michael J. Pazzani. "An Enhanced Representation of Time Series Which Allows Fast and Accurate Classification, Clustering and Relevance Feedback." Kdd. Vol. 98. 1998.

Las formas de crear representaciones más compactas de los datos se pueden resumir en:



Fuente: Lin, Jessica, et al., "A symbolic representation of time series, with implications for streaming algorithms." Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery. ACM, 2003

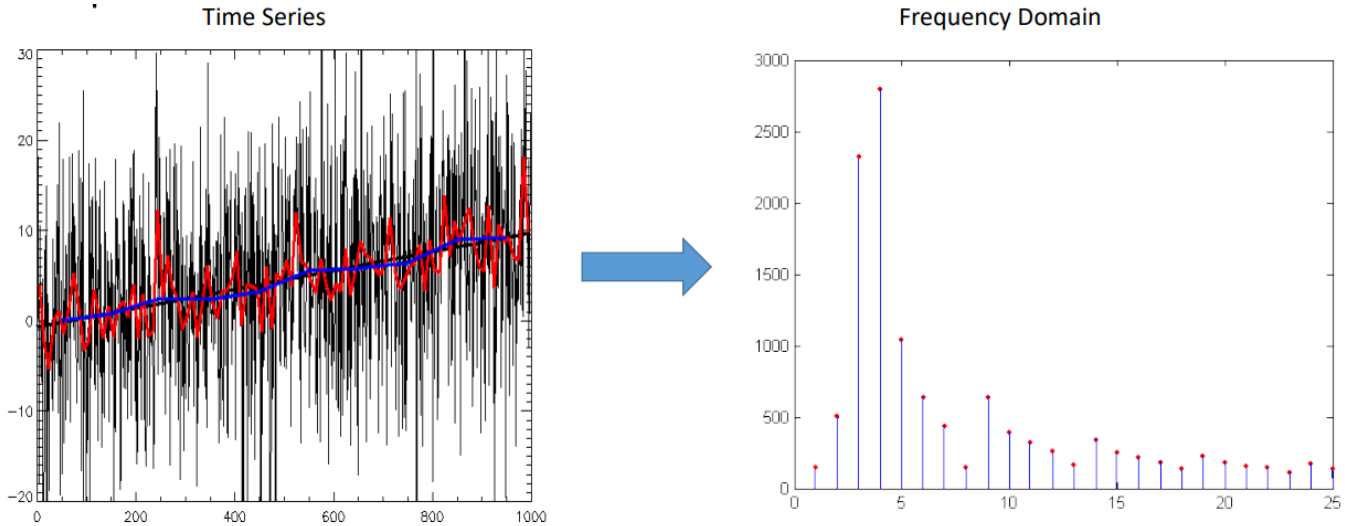
Las formas de crear representaciones más compactas de los datos se pueden resumir en:



Fuente: Lin, Jessica, et al., "A symbolic representation of time series, with implications for streaming algorithms." Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery. ACM, 2003

Dominio de frecuencia

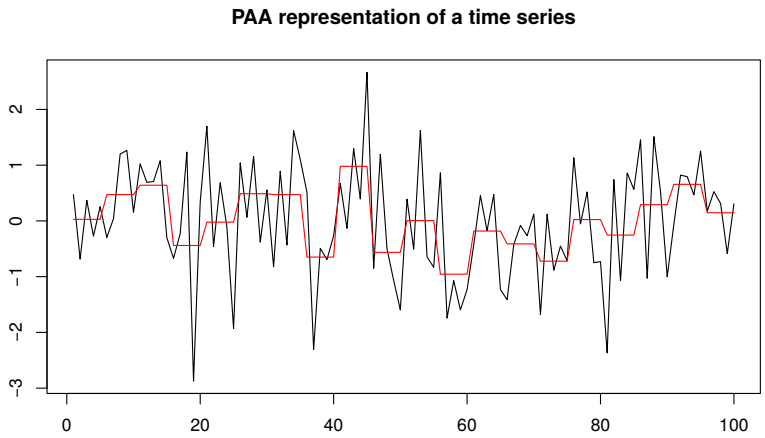
Fast fourier transform: O(nlog(n))

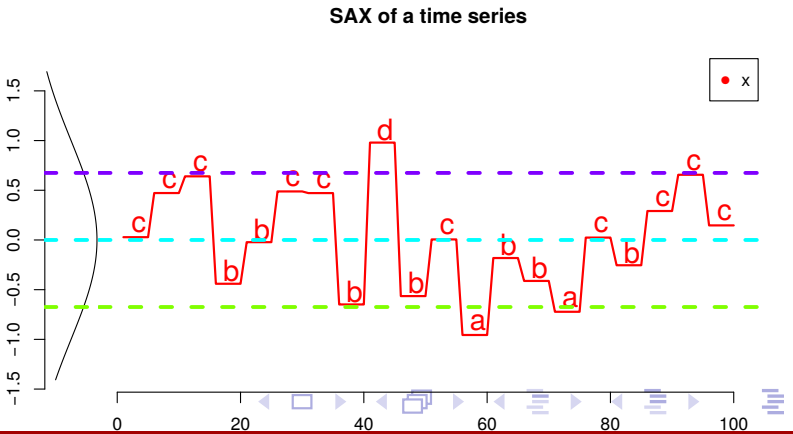


Caso de uso: Monitoreo de las condiciones de máquinas para detectar fallos en los equipos.

SAX: Symbolic Aggregate approXimation

- Normalización
- Piecewise Aggregate Approximation (PAA)
- Simbolización (alfabeto)





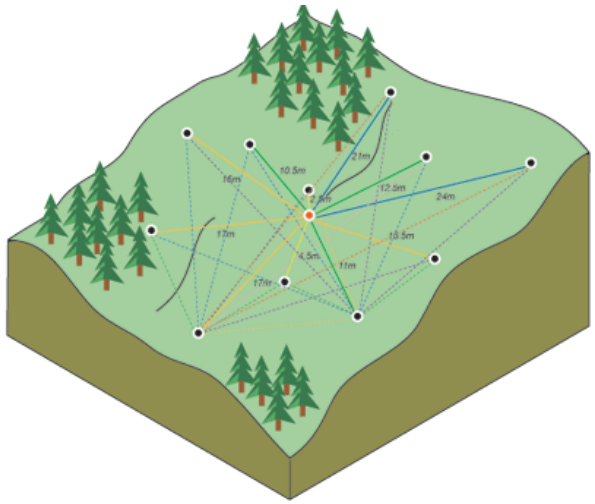
SAX: Symbolic Aggregate approXimation

Usos de SAX en IoT:

- Transmitir datos de aparatos médicos: Edge Real-Time Medical Data Segmentation for IoT Devices with Computational and Memory Constrains
- Identificar y autenticar un aparato electrónico a través de sus emisiones de radiofrecuencia: The Application of the Symbolic Aggregate Approximation Algorithm (SAX) to Radio Frequency Fingerprinting of IoT Devices
- Metadatos para filtrado: Semantic Filtering of IoT Data using Symbolic Aggregate Approximation (SAX)

Valores faltantes

Kriging (método de interpolación). Permite calcular valores que no están muestreados conociendo sus coordenadas y vecinos. Método de geoestadística.

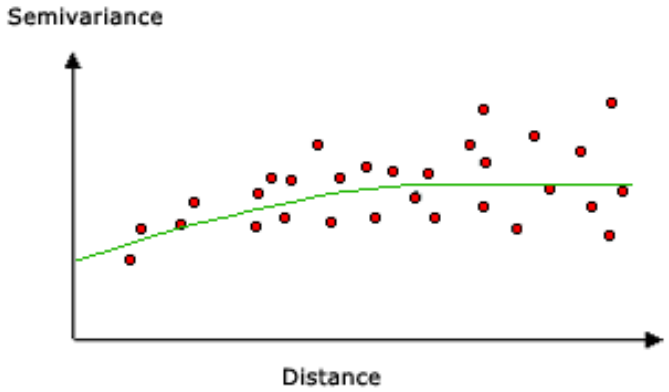


Asume: correlación espacial

Valores faltantes

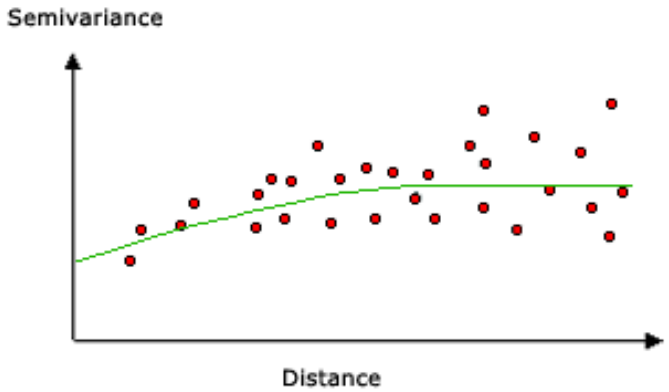
$$Semivariograma(dist(i,j)) = 0.5 * average((valor_i - valor_j)^2)$$

Las distancias se agrupan mediantes *lags*.
El semivariograma se ajusta con un modelo (exponencial, gaussiano...)



Valores faltantes

El semivariograma se utiliza para realizar las estimaciones mediante las covarianzas (fórmula obtenida utilizando los multiplicadores de Lagrange).



Uso: estimar valores faltantes en datos de temperatura

Valores faltantes

```
import numpy as np
# Parámetros del semivariograma ajustado (modelo esférico)
nugget = 0.1   # Varianza mínima (nugget)
sill = 1.5     # Varianza total
range_ = 10.0  # Alcance (distancia donde la variabilidad se estabiliza)

def semivariogram(h):
    """Modelo de semivariograma esférico"""
    if h == 0:
        return 0
    elif h < range_:
        return nugget + (sill - nugget) * (1.5 * (h / range_) - 0.5 * (h / range_)**3)
    else:
        return sill

# Puntos conocidos (x, Z)
points = np.array([
    [2, 5.0],   # (x1, Z1)
    [5, 6.0],   # (x2, Z2)
    [8, 7.5]    # (x3, Z3)
])
x_known = points[:, 0]
z_known = points[:, 1]
```

Valores faltantes

```
# Punto donde queremos estimar
x0 = 6.0

# Construcción de la matriz de covarianza (A) y vector de covarianzas (b)
n = len(points)
A = np.ones((n+1, n+1))  # Matriz aumentada para Lagrange
b = np.ones(n+1)
# Llenar la matriz A con valores de covarianza
for i in range(n):
    for j in range(n):
        A[i, j] = sill - semivariogram(abs(x_known[i] - x_known[j]))  # Covarianza

# Última fila y columna para el multiplicador de Lagrange
A[-1, :-1] = 1
A[:-1, -1] = 1
A[-1, -1] = 0
```

Valores faltantes

```
# Llenar el vector b con las covarianzas entre
#los puntos conocidos y x0

for i in range(n):
    b[i] = sill - semivariogram(abs(x_known[i] - x0))
b[-1] = 1  # Restricción de insesgadez

# Resolver el sistema de ecuaciones
weights = np.linalg.solve(A, b)

# Estimación en el punto x0
z_estimated = np.sum(weights[:-1] * z_known)

print(f"Estimación en x0={x0}: Z* = {z_estimated:.2f}")
```

Detección de anomalías distribuida

A nivel sensor: Cada nodo lleva a cabo clustering hiperelipsoidal utilizando HyCARCE (Hyperellipsoidal Clustering algorithm for Resource-Constrained Environments). Después se utiliza un algoritmo que identifica hiperelipses anómalas (distancia entre elipses).

Propiedades:

- Selección automática del número de clusters
- Bajo coste computacional (O(n))
- Detección de anomalías embebida

Fuentes

1. Moshtaghi, Masud, et al. "An efficient hyperellipsoidal clustering algorithm for resource-constrained environments." Pattern Recognition 44.9 (2011): 2197-2209.
2. Rathore, Punit, et al. "Real-time urban microclimate analysis using internet of things." IEEE Internet of Things Journal 5.2 (2018): 500-511.

Detección de anomalías distribuida

A nivel global: Cada nodo envía un resumen de su resultado al "nodo padre", que los combina. El proceso continúa hasta llegar a la nube donde se aplica el algoritmo para encontrar anomalías. Las anomalías globales se envían de vuelta para encontrar de qué sensor proceden.

Métodos de Big Data Analytics

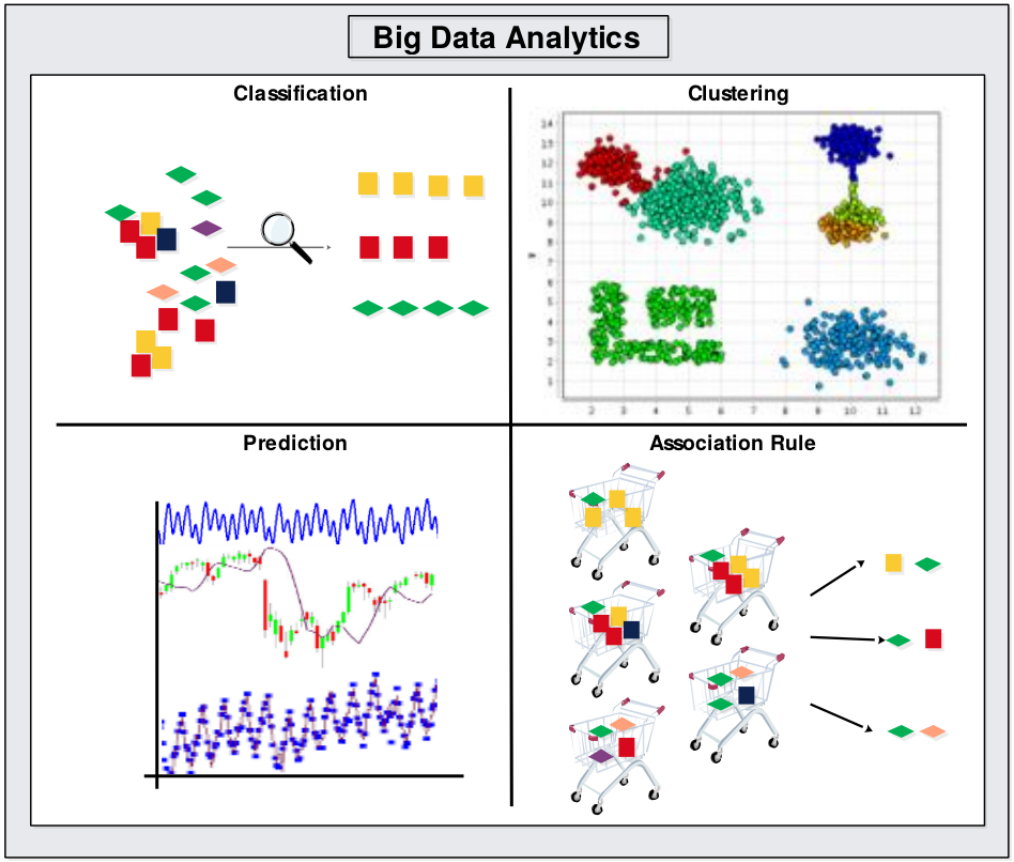
La evolución del Big Data ha cambiado los requerimientos de las técnicas de aprendizaje automático.
Más allá de la gestión Big Data: capturar, guardar y preprocesar, los análisis deben ser tan rápidos o más que usando los métodos tradicionales de análisis de datos y, además, con el mínimo coste posible para tratar una gran volumen a una gran velocidad y una gran variedad de datos.

Conocer las técnicas Big Data de análisis de datos es crucial para poder escoger los métodos adecuados para cada problema.
En esta sección, hablaremos de los métodos que son aconsejables desde el punto de vista Big Data.

Hemos indentificado los siguiente métodos de Big Data Analytics:

- Clasificación
- Clustering
- Reglas de Asociación
- Predicción

Cada categoría es una función de la minería de datos y recoge varios métodos y algoritmos que tienen la intención de extraer información y cumplir los requerimientos de los análisis.

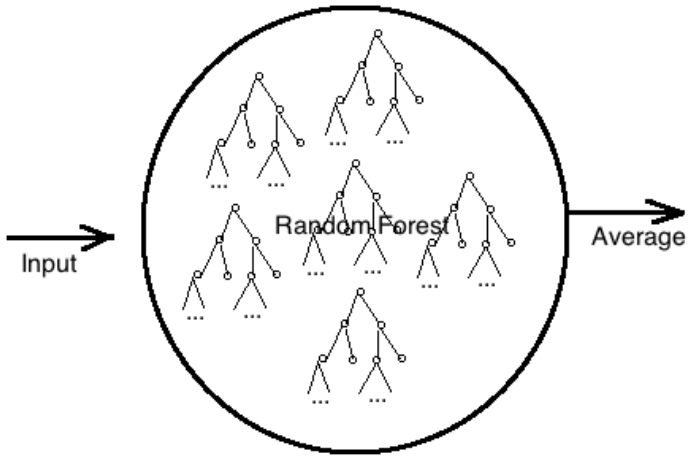


Fuente: Marjani, Mohsen, et al. "Big IoT data analytics: architecture, opportunities, and open research challenges." IEEE Access 5 (2017): 5247-5261.

Clasificación y Predicción

La clasificación es una técnica de aprendizaje supervisado y, por lo tanto, utiliza conocimiento previo (necesita datos con su clase) para agrupar datos. El objetivo es predecir la categoria (o clase) de un sujeto.
Muchas técnicas de machine learning se pueden usar tanto para clasificación como para predicción.

Random forest (*Los bosques aleatorios*) se compone de estructuras "similares" a un bosque formadas por árboles de decisión generados mediante muestreo aleatorio con reemplazamiento. Dichos árboles pueden ser de regresión (predicción) o clasificación. La oportunidad de paralelización de este algoritmo lo hace muy apropiado para el Big Data.



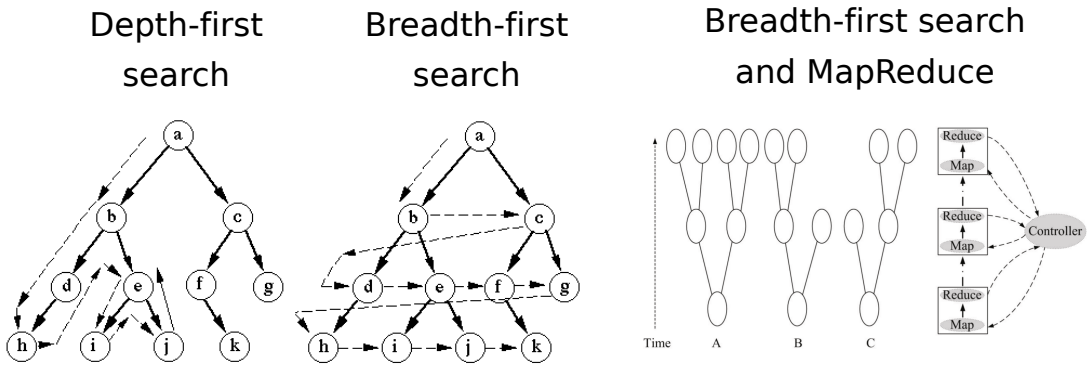
Random forest tiene una estructura **paralelizable** ya que los árboles de decisión de los que se compone se pueden construir de manera aislada. Random Forest utiliza bootstrapping para crear múltiples dominios y se aplican árboles de decisión a cada uno. Esta estructura es clave para el uso de las tecnologías Big Data modernas como el sistema de archivos distribuidos Hadoop y el esquema MapReduce.

Las formas para paralelizar los árboles de decisión se pueden clasificar en 4 categorías principales:

- Horizontal o basada en los datos: procesadores diferentes trabajan en ejemplos diferentes de los datos
- Vertical o basada en las características: cada procesador considera atributos diferentes
- Basada en nodos: los nodos de los árboles son distribuidos a cada procesador
- Híbrido: combina horizontal o vertical al principio con basada en nodos al final.

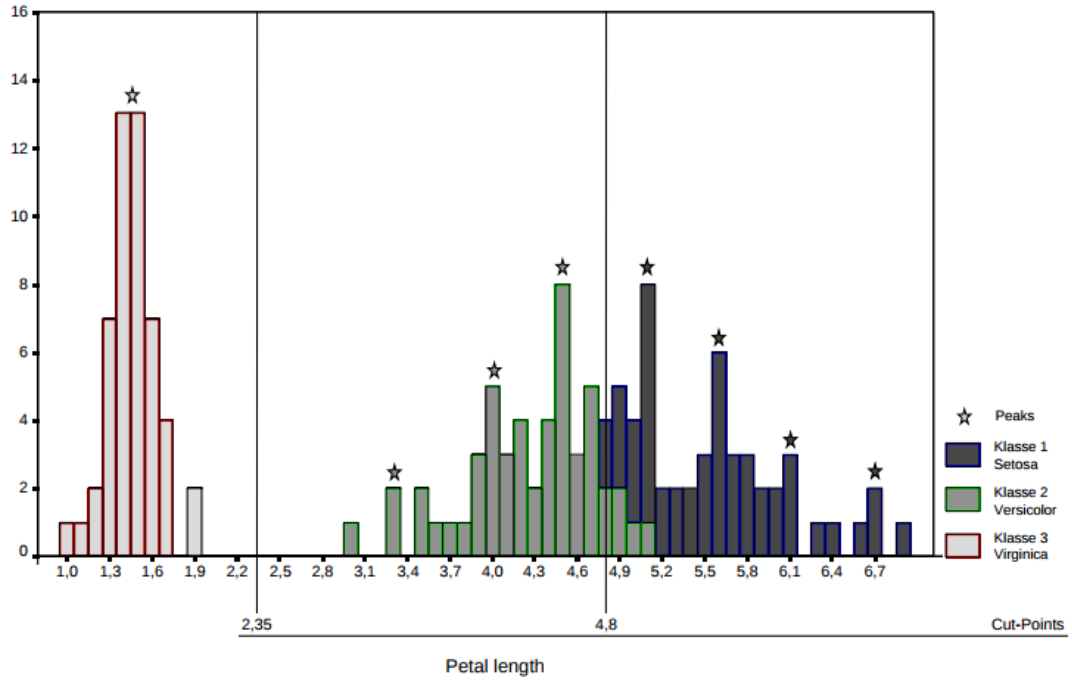
Una propuesta

El algoritmo Random Forest tradicional construye los árboles de decisión usando búsqueda en profundidad (*depth-first*). Sin embargo, de forma más óptima para casos Big Data que sean online los árboles se pueden crear usando búsqueda en anchura (*breadth-first*) porque permite incorporar nuevas características.



Fuente: Li, Bingguo, et al. "Scalable random forests for massive data." Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, Berlin, Heidelberg, 2012.

La parte más costosa: particionar los nodos. Esto se realiza mediante métricas que capturan la distribución de las etiquetas de clase de los datos tras la partición (por ejemplo la entropía).
Una forma más eficaz que se ha mencionado reiteradamente en investigaciones que buscan adaptarse al Big Data es un algoritmo basado en histogramas. No se trabaja con los puntos de las características sino con los histogramas de ellas ya que describen de una forma más compacta la distribución de los datos.



Para indagar más en la partición de nodos basada en histogramas, este es el paper donde se propuso:

Fuente: Ben-Haim, Y., and Tom-Tov, E. (2010). A streaming parallel decision tree algorithm. Journal of Machine Learning Research, 11(Feb), 849-872.

Para cada nivel de los árboles, un par map y reduce se encarga de crear las separaciones de nodos.
El mapper calcula los histogramas locales del subespacio de características correspondientes a cada bloque de datos y se ordenan de acuerdo a los id de los árboles. Los histogramas locales del mismo árbol se envían a los reducers para calcular los histogramas globales de los cuales se extraerán la mejor condición para partirlo.

En el caso de las máquinas de soporte vectorial, se han estudiado principalmente dos formas para distribuirlas:

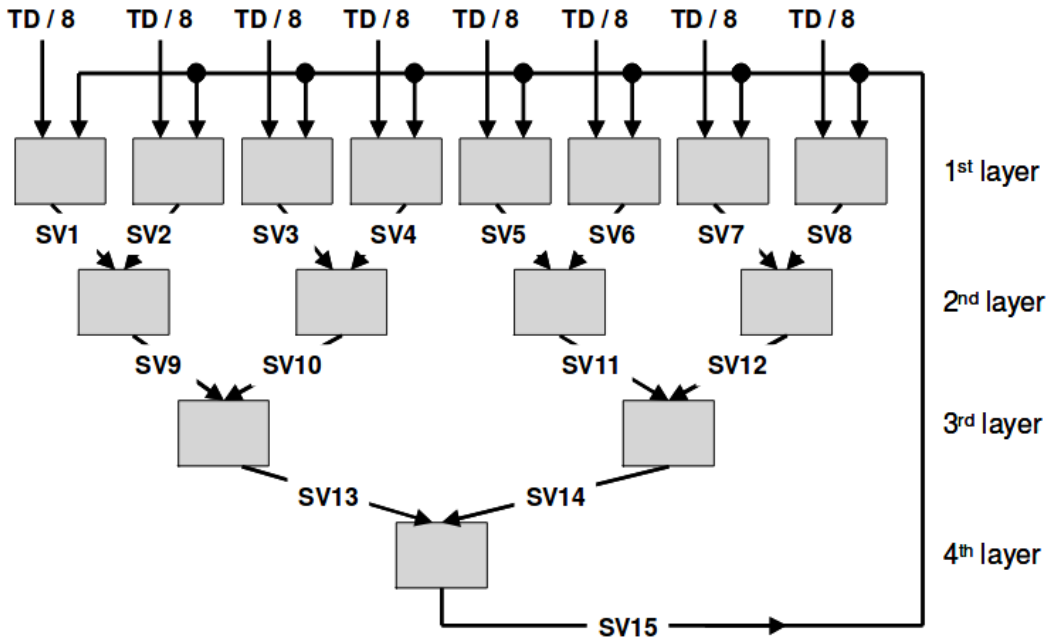
- Horizontal
- Vertical

Los ejemplos de distribución vertical tienen generalmente baja eficiencia en la comunicación y aquellas propuestas que la mejoran lo hacen a costa de bajar la precisión de los algoritmos.

Fuente: Stolpe, Marco, Kanishka Bhaduri, and Kamalika Das. "Distributed support vector machines: An overview." Solving Large Scale Learning Tasks. Challenges and Algorithms. Springer, Cham, 2016. 109-138.

- SVM incremental. Se crean subconjuntos $S_1, ..., S_m$ disjuntos. Para t= 1, se aplica SVM a S_1 y se guardan los vectores de soporte SV_1. A continuación, se aplica SVM a $S_2 \cup SV_1$ y así. Funciona solo si la distribución de los subconjuntos se mantiene en todo el conjunto.
- SVM iterativo. La propuesta se basa en intercambiar los vectores de soporte con un nodo padre. Para cada iteración t y en cada subconjunto j se determina el conjunto de vectores de soporte SV_j^t basado en el conjunto de datos $S_j \cup GSV^{t-1}$ donde lo último es el conjunto global de vectores de soporte. Tras un número finito de iteraciones, converge.

SVM cascada: quita los vectores no soporte.



Fuente: Graf, Hans P., et al. "Parallel support vector machines: The cascade svm." Advances in neural information processing systems. 2005.

Big Data Clustering

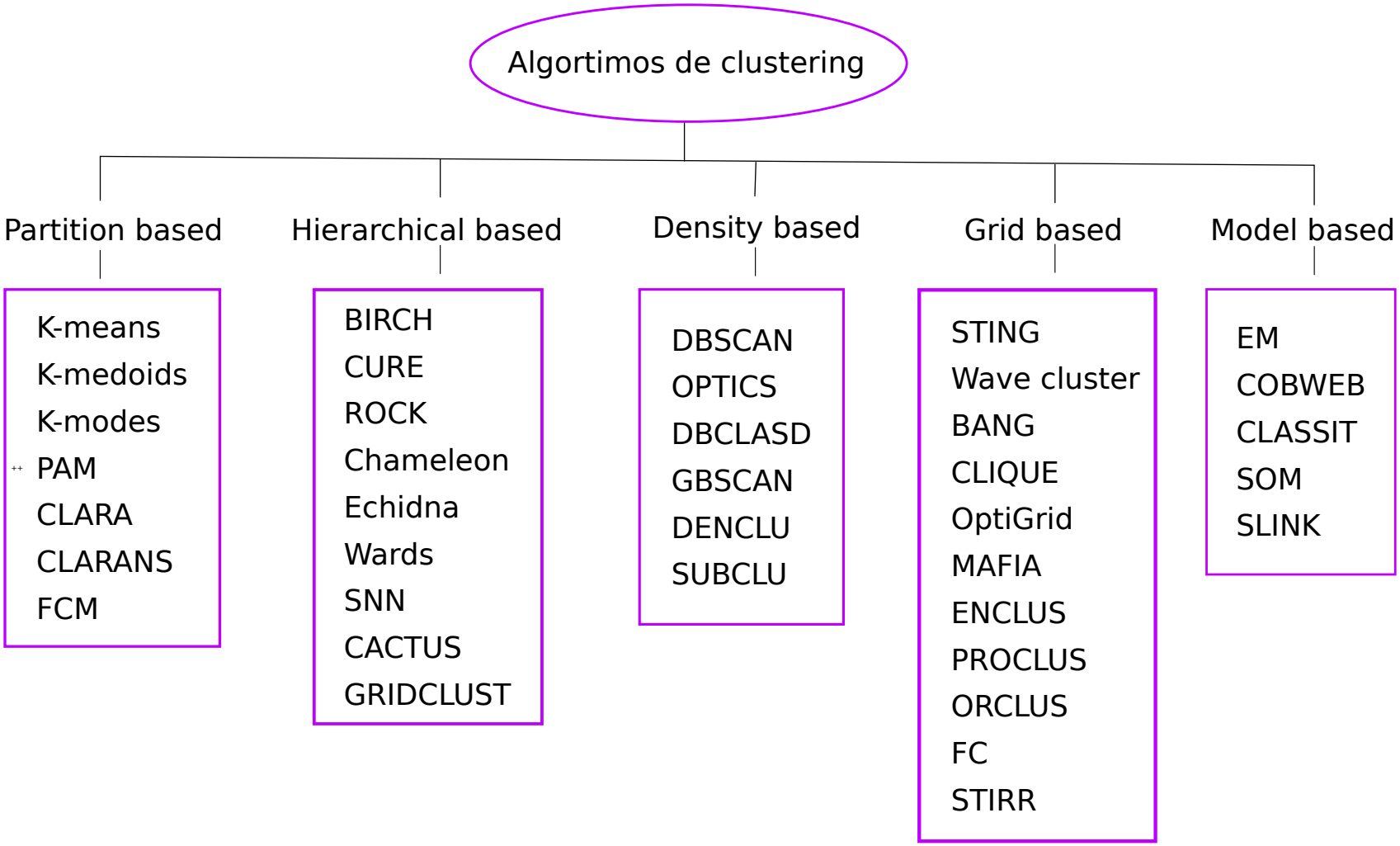
Fuentes:

- Fahad, Adil, et al. " **A survey of clustering algorithms for big data: Taxonomy and empirical analysis**." IEEE transactions on emerging topics in computing 2.3 (2014): 267-279.
- Sajana, T., CM Sheela Rani, and K. V. Narayana. "**A survey on clustering techniques for big data mining**." Indian journal of Science and Technology 9.3 (2016): 1-12.

Los algoritmos de clustering se pueden caracterizar en:

- Partitioning-based
- Hierarchical-based
- Density-based
- Grid-based
- Model-based

Big Data Clustering



Criterios de evaluacion de los métodos de clustering para Big Data:

- Volumen: tamaño del conjunto de datos, dimensionalidad , manejo de outliers y ruido
- Variedad: tipo de datos, forma del cluster
- Velocidad: complejidad del algoritmo y tiempo de ejecución
- Valor: parámetros de entrada

Los mejores de cada tipo para Big Data:

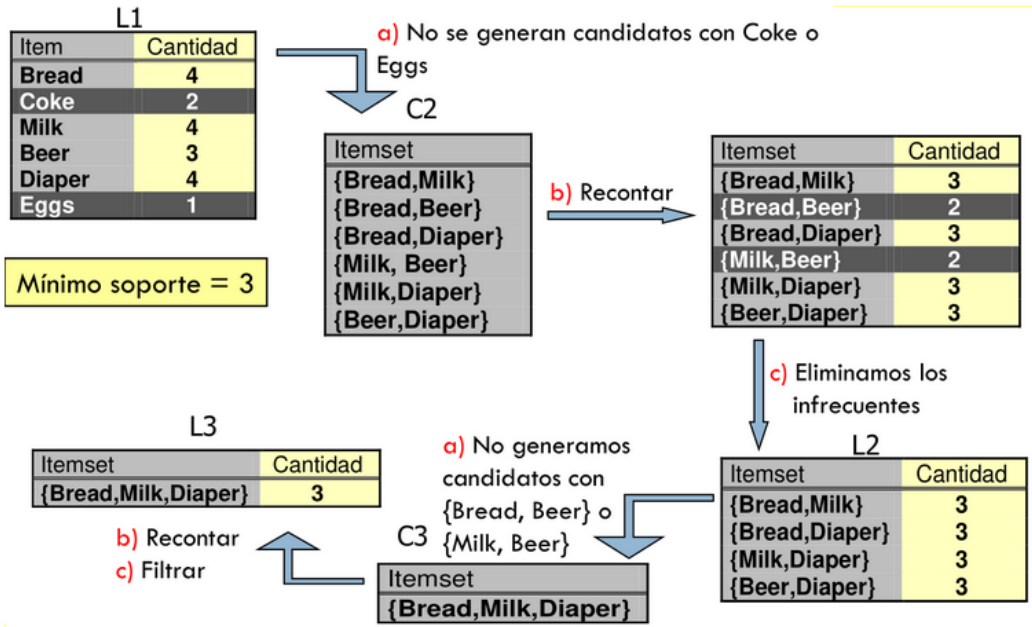
Categoría	Particional	Jerárquico	Densidad	Grid	Basados en modelos
Algoritmo	Fuzzy C-means	BIRCH	DENCLUE	OptiGrid	EM
Tamaño	Grande	Grande	Grande	Grande	Grande
Gran dimensionalidad	No	No	Sí	Sí	Sí
Atípicos y ruidoso	No	No	Sí	Sí	No
Tipo de datos	Numéricos	Numéricos	Numéricos	Espaciales	Espaciales
Forma del cluster	No convexo	No convexo	Arbitrario	Arbitrario	No convexo
Complejidad	O(n)	O(n)	O(log | D|)	(O(nd), O(ndlog(n))	O(knp)
Parámetros de entrada	1	2	2	3	3

Reglas de Asociación

Las reglas de asociación se utilizan para descubrir hechos que ocurren en común dentro de un determinado conjunto de datos.
Referencia: A priori (1994).
Este algoritmo va creando posibles agrupaciones de 1, 2, 3,...,k elementos y va contando las apariciones de los mismos en cada uno de dichos niveles.
Problemas: La búsqueda no es eficiente

Reglas de Asociación

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke



*Ejemplo algoritmo Apriori. *Créd. Rafa Alcalá, Profesor Reglas de Asociación**

Fuente imagen: https://elbauldelprogramador.com/aprendizaje-nosupervisado-reglas/

PARMA: parallel technique for mining Frequent Itemsets and Association Rules. Tiempo de ejecución "casi lineal" y el framework MapReduce.
Observaciones en las que se basa:

- La naturaleza de la minería de datos es exploratoria
- Más importante que obtener resultados exactos, en ocasiones, necesitamos rapidez (si aún tenemos garantías)

Fuente: Riondato, Matteo, et al. "PARMA: a parallel randomized algorithm for approximate association rules mining in MapReduce." Proceedings of the 21st ACM international conference on Information and knowledge management. ACM, 2012.

Objetivos:

- Extraer una muy buena aproximación de los Frequent Itemsets
- Adaptarnos a los recursos (procesadores y memoria)
- Minimizar replicaciones y comunicación entre procesadores
- Alcanzar la paralelización óptima

Crea pequeñas muestras aleatorias de los datos transacionales, las manda a diferentes ordenadores y utiliza algoritmos como "a priori" en cada uno de ellos. Después filtra y agrega los resultados, obteniendo frecuencias y niveles de confianza para valores de precisión predefinidos.
La calidad del resultado se garantiza probabilísticamente ya que el usuario debe especificar los parámetros: precisión y probabilidad del error.
Vídeo sobre el método:
https://www.youtube.com/watch?v=HZ4xyjqFMus

La ¿arrogancia? del Big Data

En 2008, científicos de Google publicaron una investigación en la revista Nature donde decían que podrían predecir en tiempo real la gripe basándose en las búsquedas de la gente y crearon Google Flu Trends.
La investigación demostraba que eran capaces de estimar de forma precisa la prevalencia de la gripe 2 semanas antes que los organismos dedicados para ello, siendo capaces potencialmente de salvar vidas.
En 2013, falló estrepitósamente. Este proyecto ha servido para señalar los problemas que pueden derivar de confiar únicamente en los datos.

Fuente: https://www.wired.com/2015/10/can-learn-epic-failure-google-flu-trends/

Se llevaron a cabo investigaciones sobre el por qué. Es más, este sistema tuvo un buen funcionamiento durante 2 ó 3 años y luego falló porque requería revisión sustancial.
Entre otras cosas, google no tuvo en cuenta cambios en el comportamiento de las búsquedas a través del tiempo, además de sugestionar al público al introducir la herramienta. Por otra parte, había overfitting de los datos de forma estacional con términos que no tenían por qué estar relacionados con la gripe.

Recomendados:

1. Ginsberg, Jeremy, et al. "Detecting influenza epidemics using search engine query data." Nature 457.7232 (2009): 1012.
2. Lazer, David, et al. "The parable of Google Flu: traps in big data analysis." Science 343.6176 (2014): 1203-1205.