



DECSAI

Departamento de Ciencias de la Computación e I.A.

Universidad de Granada



Aprendizaje automático

© Fernando Berzal, berzal@acm.org

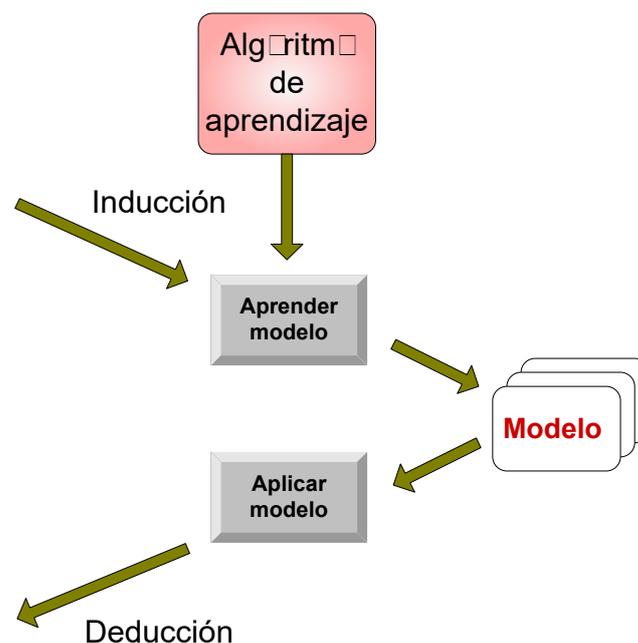
Aprendizaje automático

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Conjunto de entrenamiento

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Conjunto de prueba





Aprendizaje

Supervisado vs. No Supervisado

- **Aprendizaje supervisado** (**clasificación & regresión**):
Los casos del conjunto de entrenamiento aparecen etiquetados con la clase a la que corresponden.
- **Aprendizaje no supervisado** (**asociación & clustering**):
No se conocen las clases de los casos del conjunto de entrenamiento (ni siquiera su existencia).



Ciencia de datos



- **Análisis de datos [data analysis]**
1962 John Tukey (Princeton & Bell Labs)
- **Ciencia de datos [data science]**
1985 C.F. Jeff Wu (Georgia Institute of Technology)
- **KDD [Knowledge Discovery in Databases]**
1989 Gregory Piatetsky-Shapiro (KDnuggets)
- **Minería de datos [data mining]**
1990's @ Bases de datos





Más denominaciones afines...

- **Inteligencia de negocio [Business Intelligence]**

1958 Hans Peter Luhn (IBM)

→ 1989 Howard Dresner (Gartner)

- **Aprendizaje automático [ML: Machine Learning]**

1959 Arthur Samuel (IBM)

@ IA

- **Reconocimiento de formas [pattern recognition]**

1967 k-NN (término prestado de psicología)

@ Visión artificial [CV: Computer Vision]

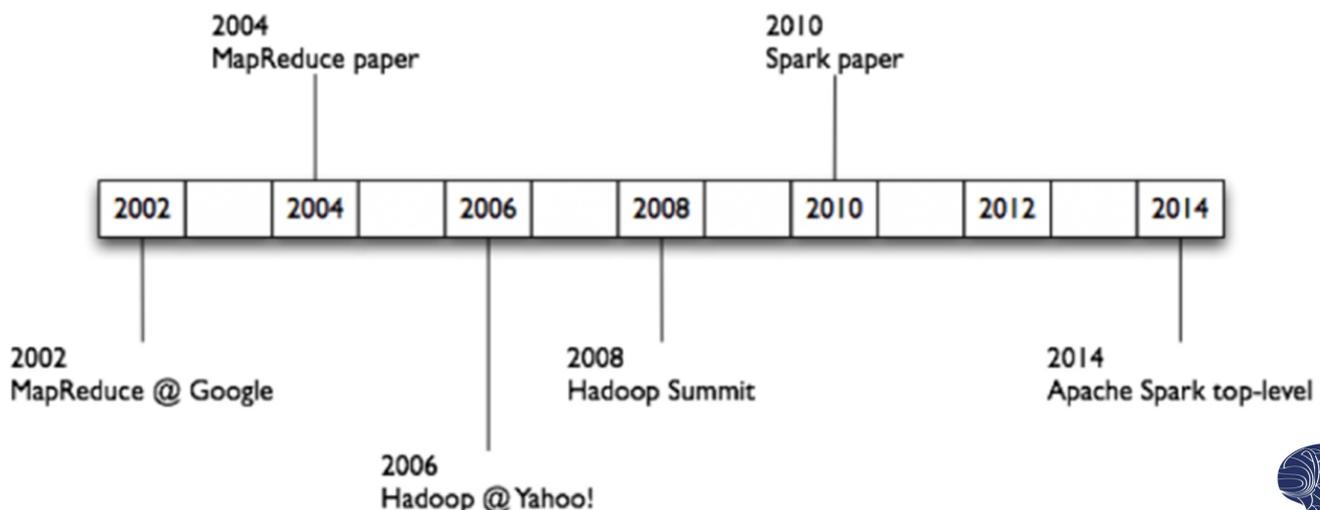


Big Data



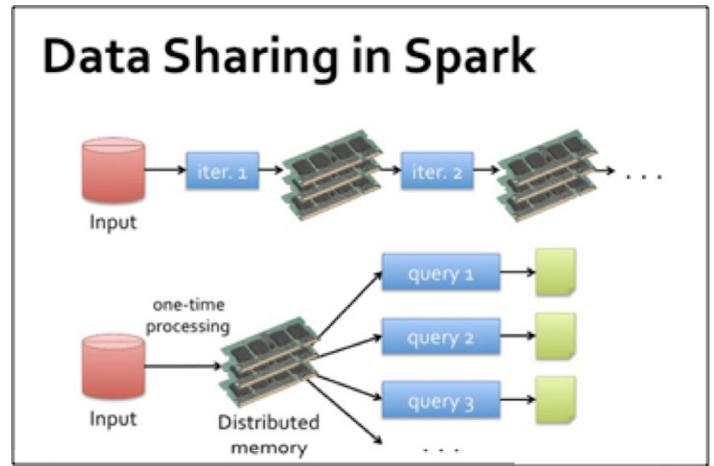
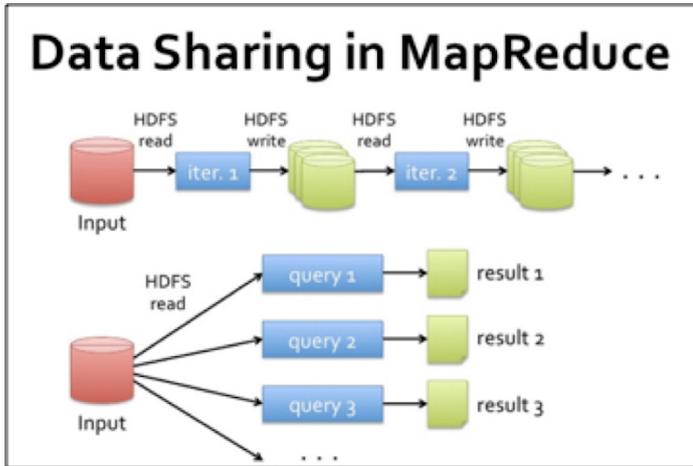
¿Cuándo pasa a ser “big data”?

Cuando hacen falta técnicas paralelas para procesar los datos...





Hadoop vs. Spark



La importancia de la eficiencia de un algoritmo...

n	10	100	1000	10000	100000
O(n)	10ms	0.1s	1s	10s	100s
O(n·log₂ n)	33ms	0.7s	10s	2 min	28 min
O(n²)	100ms	10s	17 min	28 horas	115 días
O(n³)	1s	17min	12 días	31 años	32 milenios





Clasificación de los modelos de minería de datos

En función de su propósito general:

- **Modelos descriptivos**

(describen el comportamiento de los datos de forma que sea interpretable por un usuario experto).

- **Modelos predictivos**

(además de describir los datos, se utilizan para predecir el valor de algún atributo desconocido).



Ejemplos

- Reglas de asociación (modelo descriptivo)

Los compradores de pañales también suelen comprar cerveza.



- Clustering (modelo descriptivo)

Segmentación de los clientes de un hipermercado:

- Clientes ocasionales que gastan mucho.
- Clientes habituales con presupuesto limitado.
- Clientes ocasionales con presupuesto limitado.

- Clasificación (modelo predictivo):

- Datagramas que corresponden a intentos de intrusión.
- Perfil de un cliente de alto riesgo para préstamos bancarios.





DECSAI

Departamento de Ciencias de la Computación e I.A.

Universidad de Granada



Aprendizaje no supervisado

© Fernando Berzal, berzal@acm.org



DECSAI

Departamento de Ciencias de la Computación e I.A.

Universidad de Granada



Reglas de asociación

© Fernando Berzal, berzal@acm.org

Reglas de asociación



Problema

Dado un conjunto de transacciones, encontrar reglas que describen tendencias en los datos:



Detectar cuándo la ocurrencia de un artículo está asociada a la ocurrencia de otros artículos en la misma transacción.



Reglas de asociación



“Market-basket analysis”

Transacciones

TID	Artículos
1	Pan, leche, huevos
2	Pan, pañales, cerveza
3	Leche, pañales, cerveza
4	Pan, leche, pañales, cerveza
5	Pan, leche, huevos, cerveza

Reglas de asociación

{pañales} → {cerveza}

{leche, pan} → {huevos}

{cerveza, pan}
→ {leche, huevos}

¡OJO! → implica co-ocurrencia, no causalidad.





“Product placement”: Colocación de productos en las estanterías de un supermercado

Objetivo

Identificar artículos que muchos clientes compran conjuntamente.

Solución

Procesar los datos de los terminales de punto de venta proporcionados por los escáneres de códigos de barras.

Ejemplo

Si un cliente compra pañales, es muy probable que compre cerveza (¡no se sorprenda si ve las cervezas colocadas al lado de los pañales en el súper!)



Promociones y ofertas

Si se identifica una regla del tipo: {impresora} → {tónor}

- **Tónor en el consecuente**
=> Puede determinarse cómo incrementar sus ventas.
- **Impresora en el antecedente**
=> Puede determinarse qué productos se verían afectados si dejamos de vender impresoras.
- **Impresora en el antecedente y tónor en el consecuente**
=> Puede utilizarse para ver qué productos deberían venderse con impresoras para promocionar las ventas de tónor.





Gestión de inventarios

Problema

Una empresa de reparación de electrodomésticos quiere anticipar la naturaleza de las reparaciones que tendrá que realizar y mantener a sus vehículos equipados con las piezas que permitan reducir el número de visitas a casa de sus clientes.

Solución

Procesar los datos sobre herramientas y piezas utilizadas en reparaciones previas para descubrir patrones de co-ocurrencia.



Extracción de reglas de asociación Formulación del problema



Dado un conjunto de transacciones T ,
encontrar todas las reglas de asociación...

- cuyo soporte sea mayor o igual que un umbral mínimo de soporte, MinSupp :

$$\text{supp}(X \rightarrow Y) \geq \text{MinSupp}$$

- cuya confianza sea mayor o igual que un umbral mínimo de confianza, MinConf :

$$\text{conf}(X \rightarrow Y) \geq \text{MinConf}$$





Solución en dos etapas



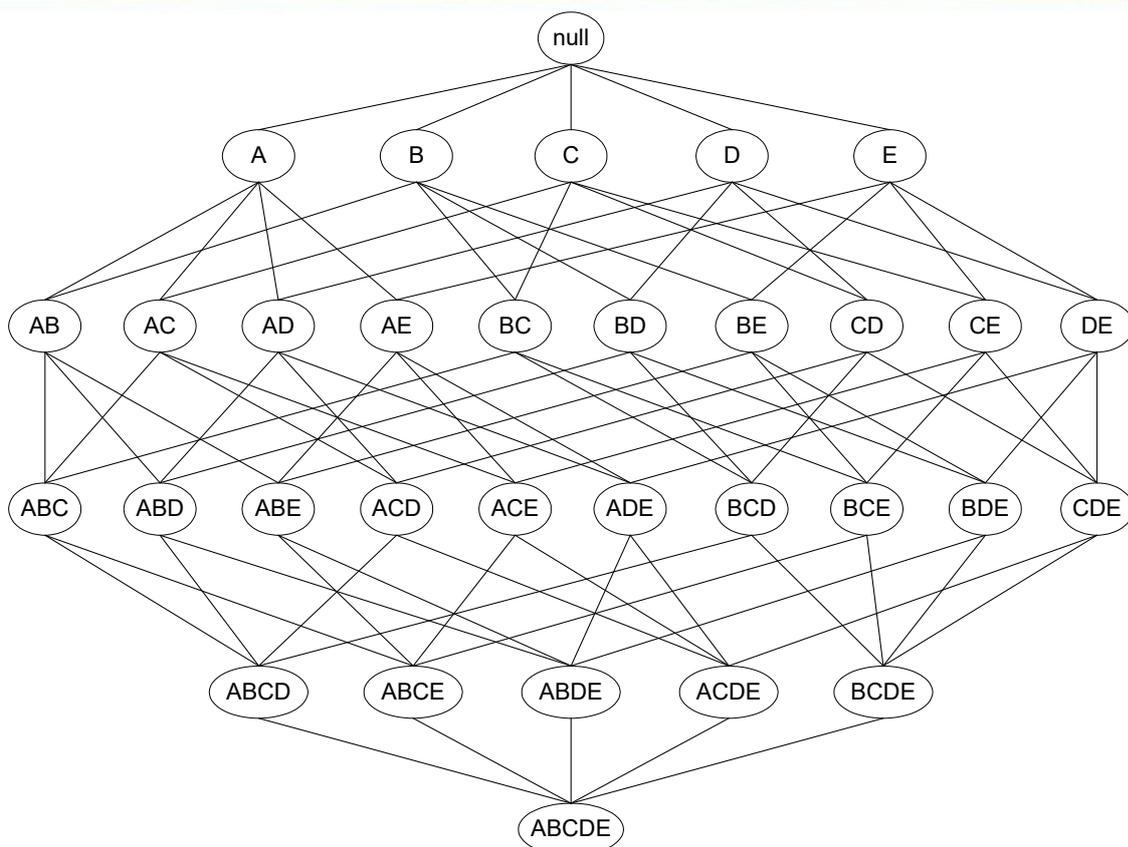
1. Generación de itemsets frecuentes:
Identificar los itemsets con soporte \geq **MinSupp**.
2. Generación de reglas de asociación:
Obtener reglas de asociación con una confianza elevada a partir de cada itemset frecuente, donde cada regla es una partición binaria del itemset.

Nota: La generación de itemsets frecuentes sigue siendo computacionalmente costosa.



Extracción de reglas de asociación

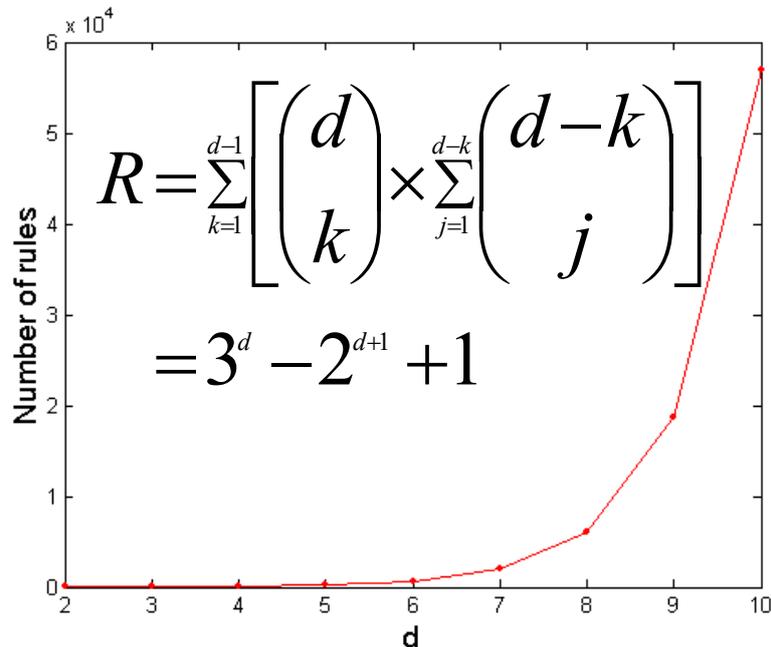
Itemsets frecuentes





Complejidad computacional

Dados d items, tenemos 2^d itemsets y R posibles reglas:



Estrategias

- Reducir el número de candidatos (M)
 - Uso de técnicas de poda.
Ejemplo: Algoritmos Apriori y DHP [Direct Hashing and Pruning]
- Reducir el número de transacciones (N)
 - Reducir N conforme aumenta el tamaño del itemset.
Ejemplo: Algoritmo AprioriTID
- Reducir el número de comparaciones (NM)
 - Uso de estructuras de datos eficientes para almacenar los candidatos o las transacciones, de forma que no haya que comparar cada candidato con todas las transacciones.





Reducción del número de candidatos

La propiedad Apriori

Si un itemset es frecuente,
también lo son todos sus subconjuntos

¿Por qué? Porque el soporte de un itemset nunca puede ser mayor que el de cualquiera de sus subconjuntos:

$$\forall X, Y : (X \subseteq Y) \Rightarrow s(X) \geq s(Y)$$

Formalmente, esta propiedad se conoce con el nombre de anti-monotonía del soporte.



Extensiones y variaciones



- Reglas de asociación **cuantitativas**
(atributos continuos)
- Reglas **multinivel**
(a.k.a. reglas de asociación generalizadas)
- Variaciones en función del **tipo de patrones**:
 - Análisis de secuencias
 - Análisis de árboles
 - Análisis de grafos (e.g. motif discovery)
 - Detección de anomalías



Extensiones y variaciones

Detección de anomalías



Regla de asociación anómala

Regla con una confianza elevada que representa una desviación homogénea del comportamiento habitual.

```
if WORKCLASS: Local-gov
```

```
then CAPGAIN: [99999.0, 99999.0]
```

```
when not CAPGAIN: [0.0, 20051.0]
```

"Anomalía"



(7 out of 7)

Consecuente habitual



DECSAI

Departamento de Ciencias de la Computación e I.A.

Universidad de Granada

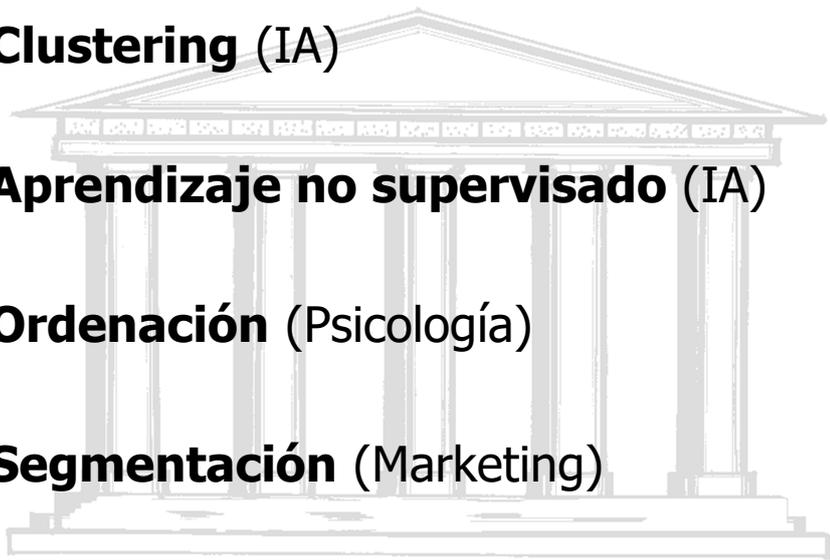


Clustering [agrupamiento]

© Fernando Berzal, berzal@acm.org

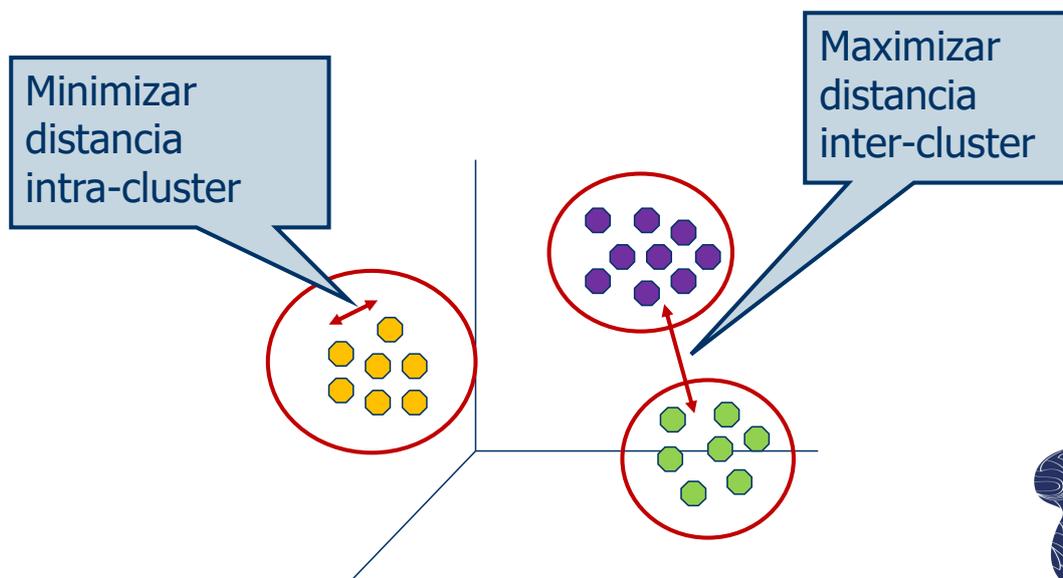
“Sinónimos” según el contexto...

- **Clustering** (IA)
- **Aprendizaje no supervisado** (IA)
- **Ordenación** (Psicología)
- **Segmentación** (Marketing)

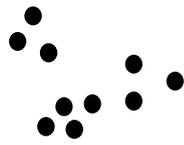
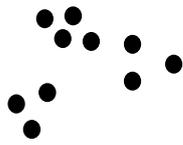


Objetivo

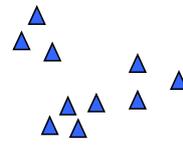
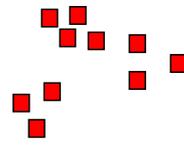
Encontrar agrupamientos de tal forma que los objetos de un grupo sean similares entre sí y diferentes de los objetos de otros grupos [*clusters*].



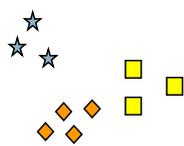
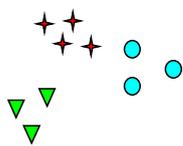
Clustering



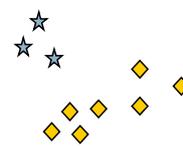
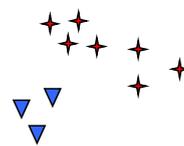
¿Cuántos agrupamientos?



¿Dos?



¿Seis?



¿Cuatro?



Clustering



- **Aprendizaje no supervisado:**
No existen clases predefinidas.
- Los resultados obtenidos dependerán de:
 - ❖ El algoritmo de agrupamiento seleccionado.
 - ❖ El conjunto de datos disponible.
 - ❖ La medida de similitud utilizada para comparar objetos (usualmente, definida como medida de distancia).



- Reconocimiento de formas.
- Mapas temáticos (GIS)
- Marketing: Segmentación de clientes
- Clasificación de documentos
- Análisis de web logs (patrones de acceso similares)
- ...

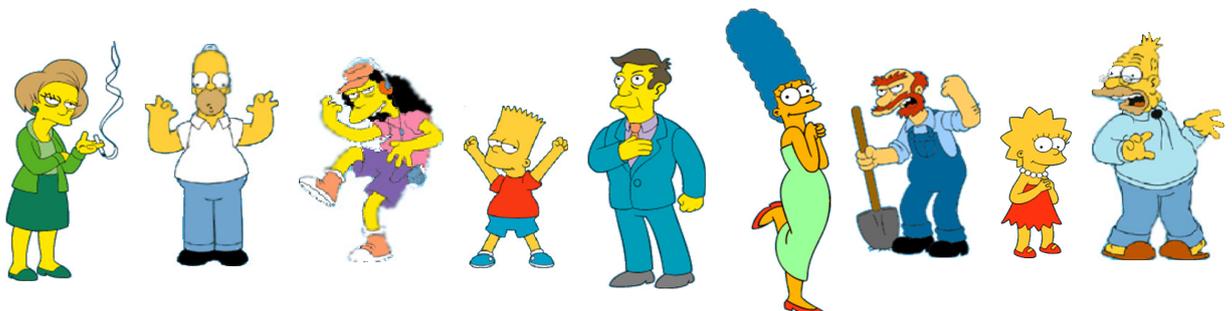
También se usa como paso previo a otras técnicas de minería de datos:

- Exploración de datos (segmentación & outliers)
- Preprocesamiento (p.ej. reducción de datos)

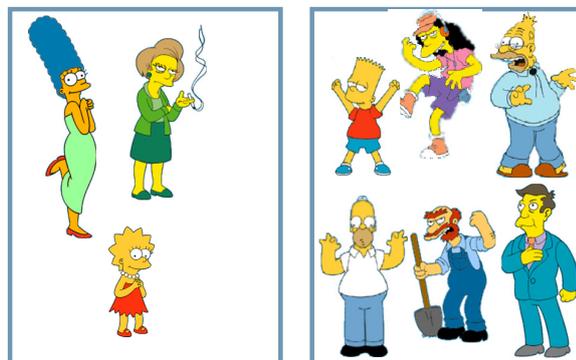


Medidas de similitud

¿Cuál es la forma natural de agrupar los personajes?



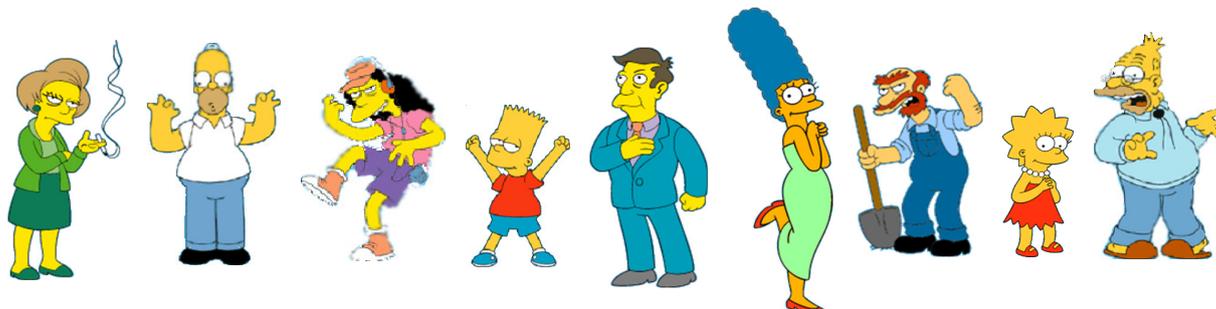
Mujeres
vs.
Hombres



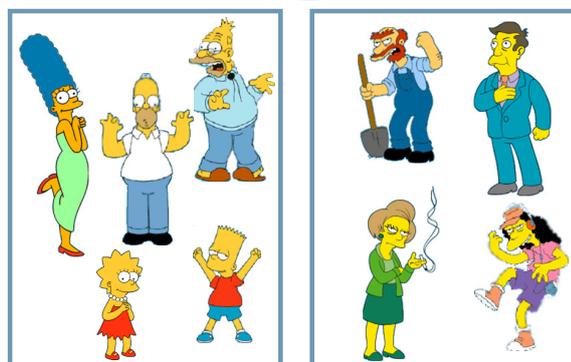
Medidas de similitud



¿Cuál es la forma natural de agrupar los personajes?



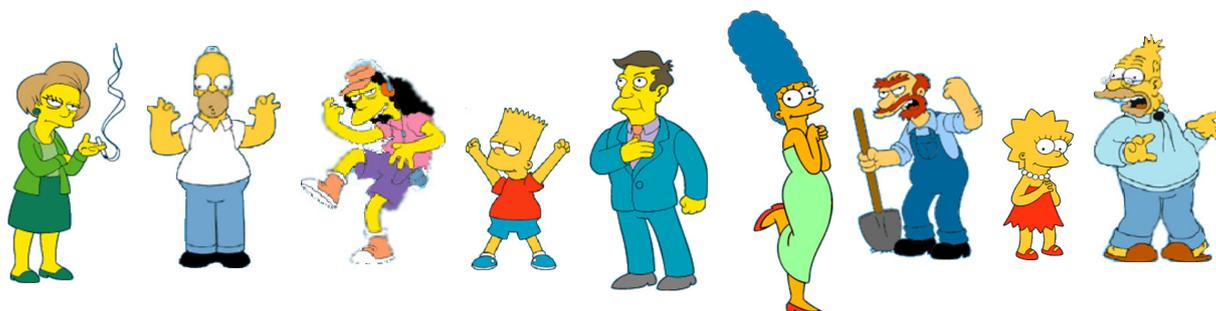
Simpsons
vs.
Empleados de
la escuela de
Springfield



Medidas de similitud



¿Cuál es la forma natural de agrupar los personajes?



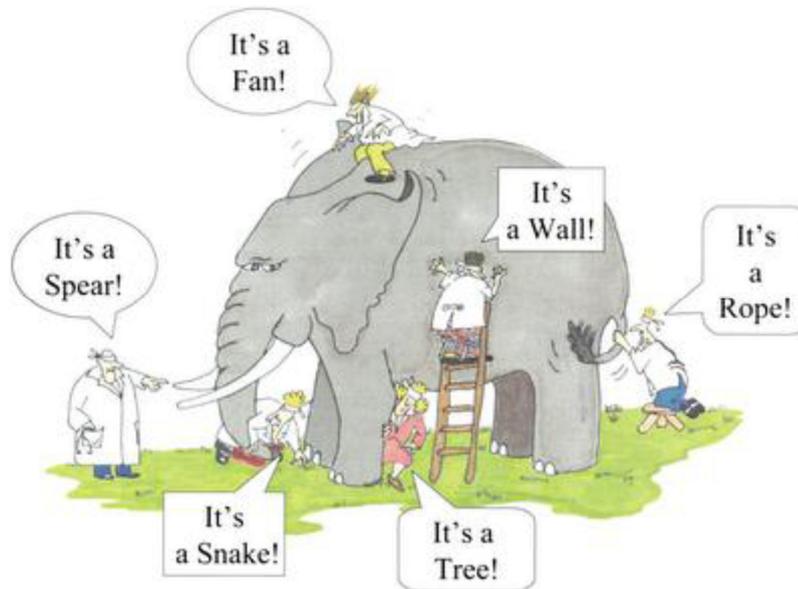
iii El clustering es subjetivo !!!



Medidas de similitud



Diferentes perspectivas...



En la práctica, hay que ofrecer alternativas (identificando sus posibles pros y contras)

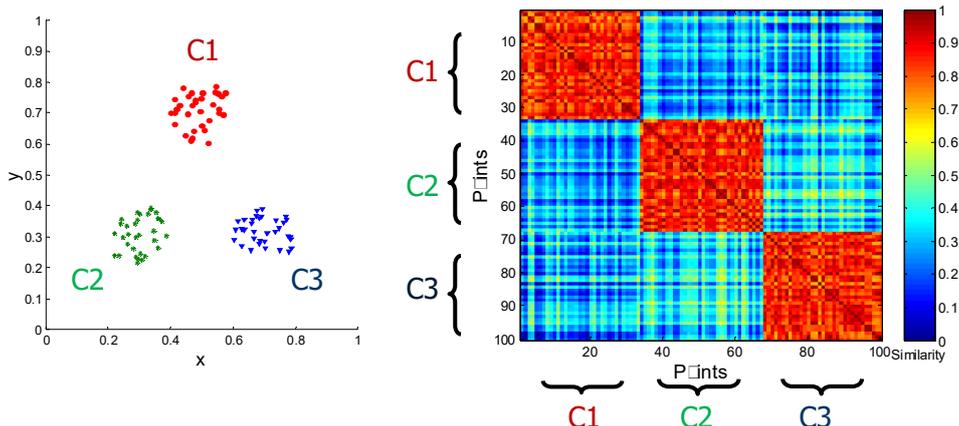


Evaluación de resultados



Matriz de similitud

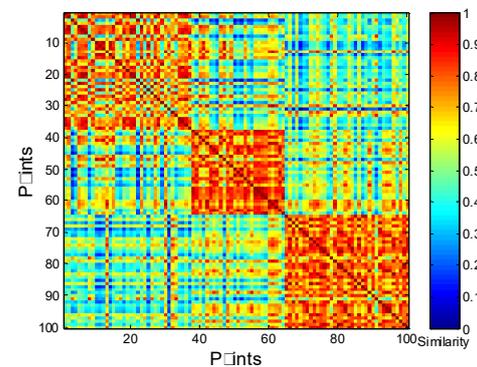
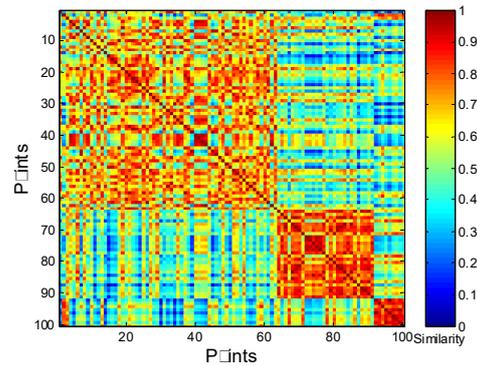
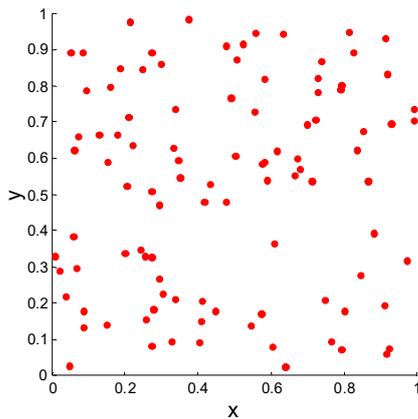
Ordenamos los datos en la matriz de similitud con respecto a los clusters en los que quedan los datos e inspeccionamos visualmente...





Problema

Incluso en datos aleatorios, si nos empeñamos, encontramos clusters: DBSCAN (arriba) y k-Means (abajo)



Métodos de agrupamiento



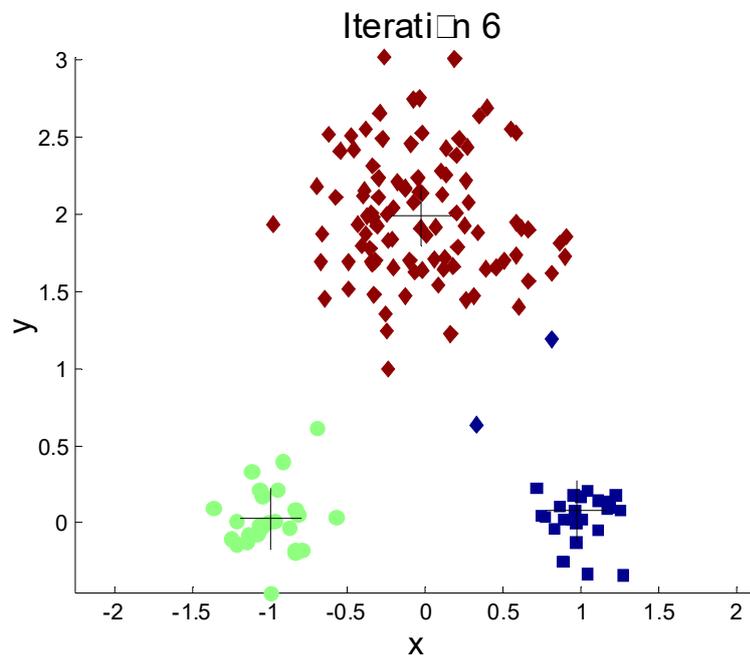
Tipos de algoritmos de clustering:

- **Agrupamiento por particiones**
k-Means, CLARANS, BFR
- **Métodos basados en densidad**
DBSCAN, Optics, DenClue, SNN
- **Clustering jerárquico**
BIRCH, CURE, ROCK, CHAMELEON
- **Clustering en subespacios**
CLIQUE, SUBCLU, RSC



Agrupamiento por particiones

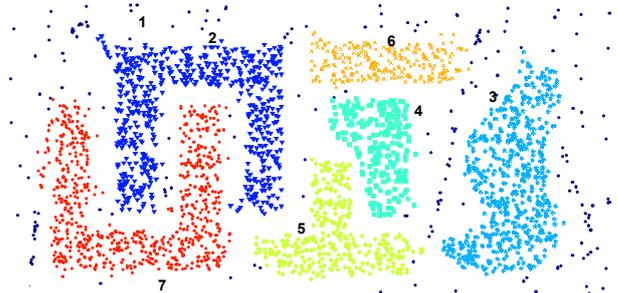
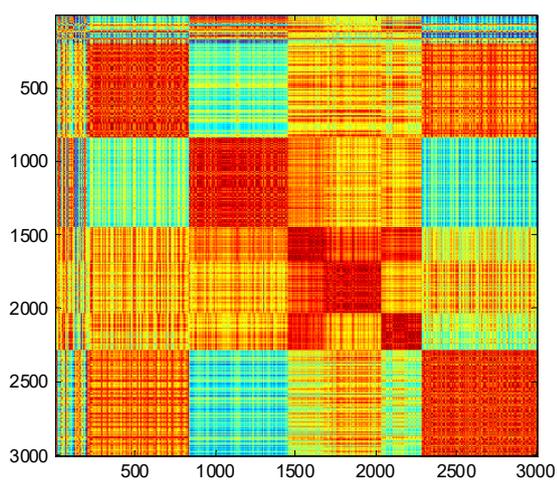
k-Means



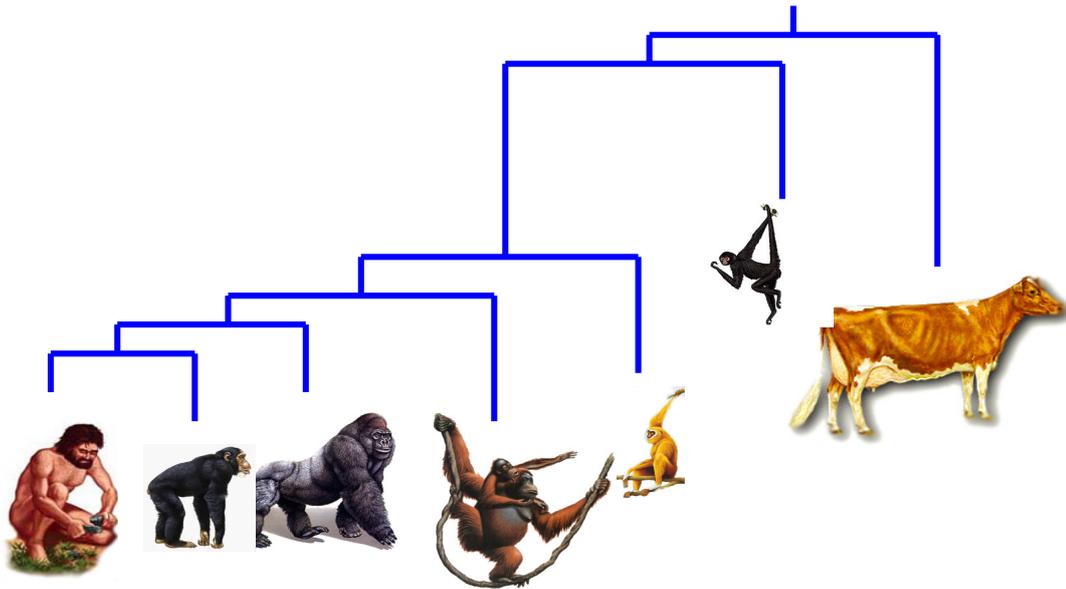
Métodos basados en densidad

Matriz de similitud

DBSCAN



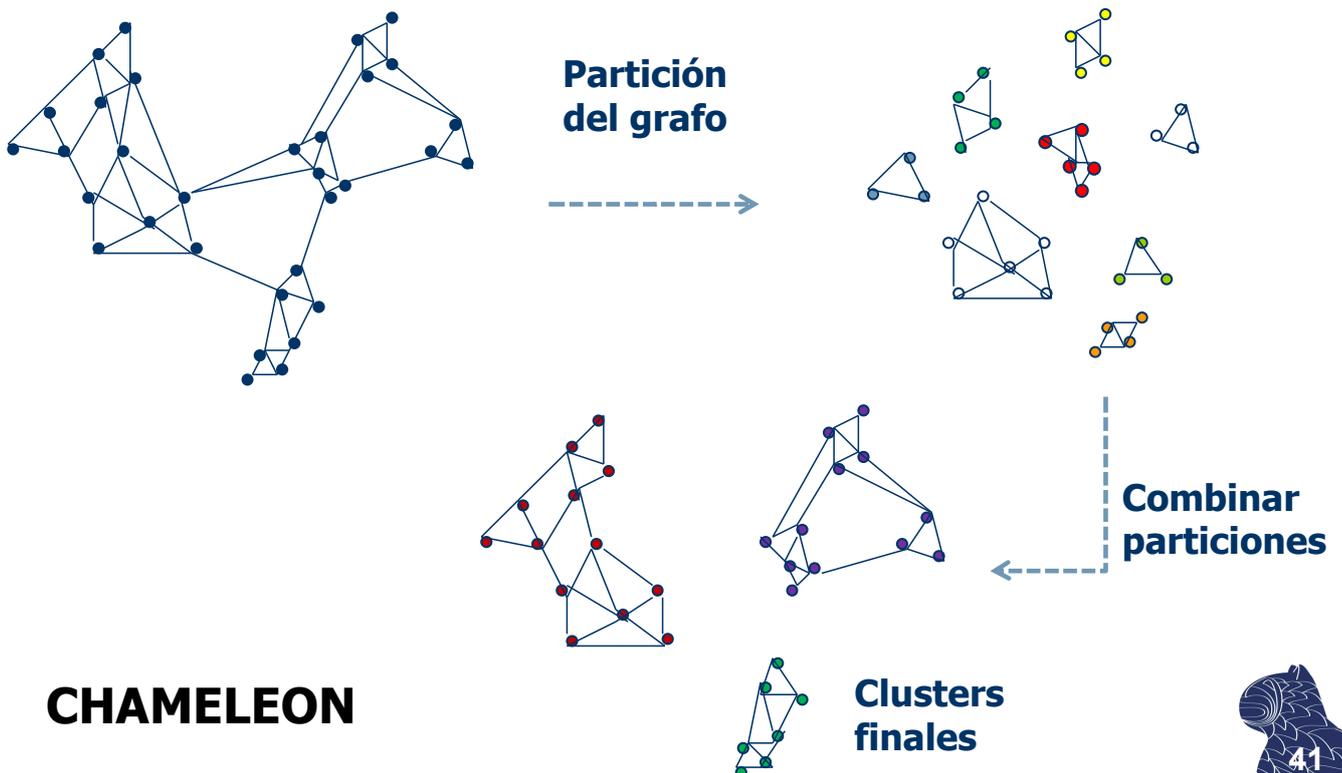
Clustering jerárquico



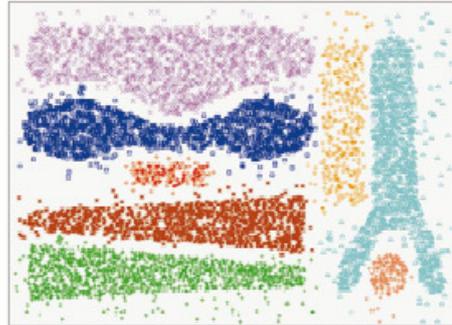
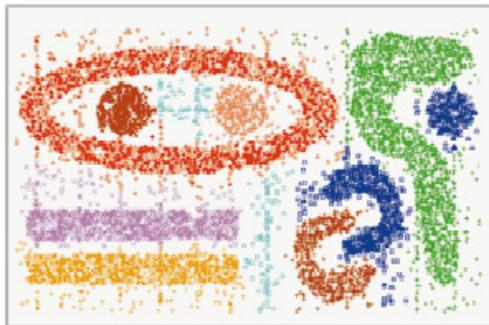
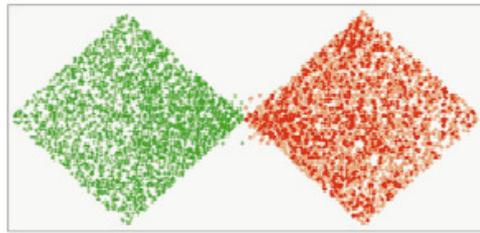
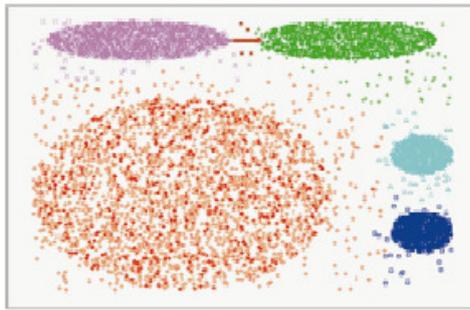
DENDROGRAMA: La similitud entre dos objetos viene dada por la "altura" del nodo común más cercano.



Clustering jerárquico



Clustering jerárquico



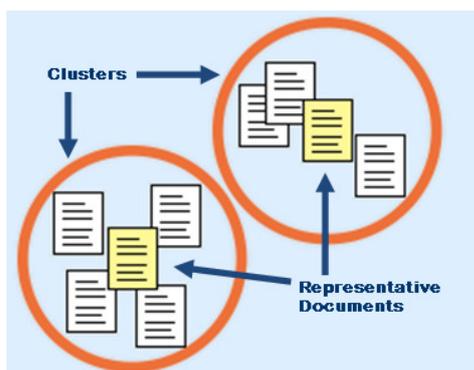
CHAMELEON



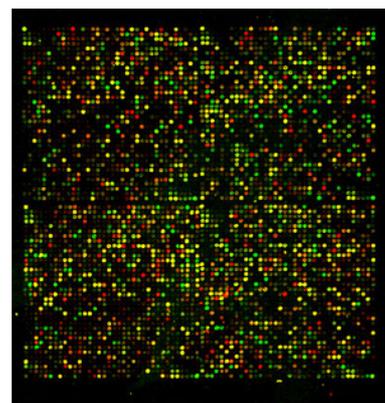
Clustering en subespacios



En muchas aplicaciones,
la dimensionalidad de los datos es elevada.



Clustering de documentos



DNA Microarrays



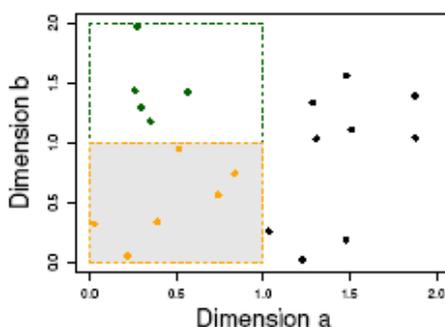
Clustering en subespacios



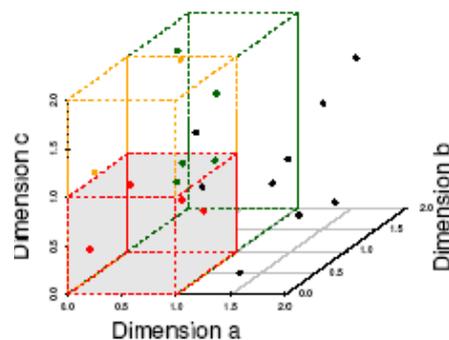
La dimensionalidad de los datos

¿Por qué es un problema?

- Los datos en una dimensión están relativamente cerca
- Al añadir una nueva dimensión, los datos se alejan.
- Cuando tenemos muchas dimensiones, las medidas de distancia no son útiles ("equidistancia").



(b) 6 Objects in One Unit Bin



(c) 4 Objects in One Unit Bin

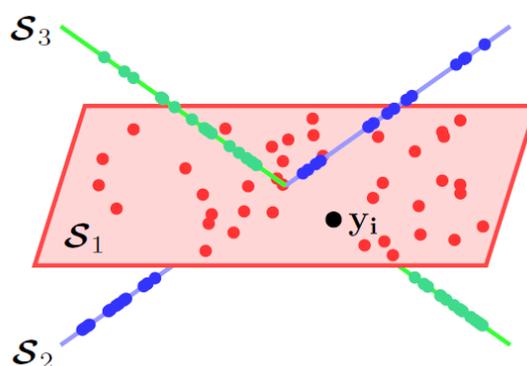


44

Clustering en subespacios



- La existencia de dimensiones irrelevantes puede enmascarar la presencia de clusters en el conjunto de datos.
- Los clusters puede que existan sólo en algunos subespacios.



45

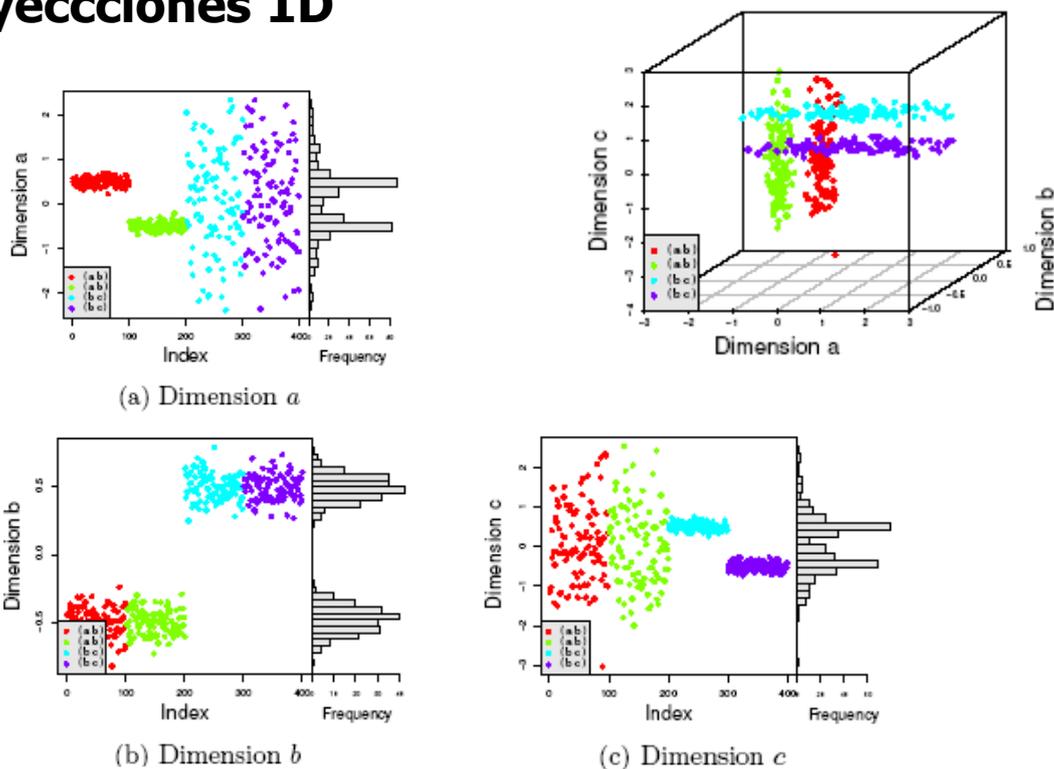


Posibles soluciones

- **Transformación de características** (PCA, SVD) para reducir la dimensionalidad de los datos, útil sólo si existe correlación/redundancia.
- **Selección de características** (wrapper/filter) útil si se pueden encontrar clusters en subespacios.
- **“Subspace clustering”**
Buscar clusters usando distintas combinaciones de atributos, vg. CLIQUE, PROCLUS, SUBCLU, RSC...

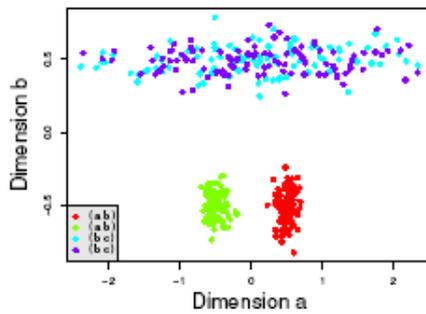


Proyecciones 1D

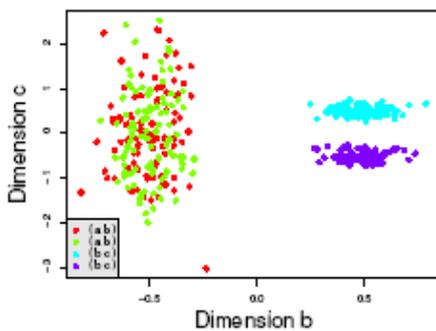
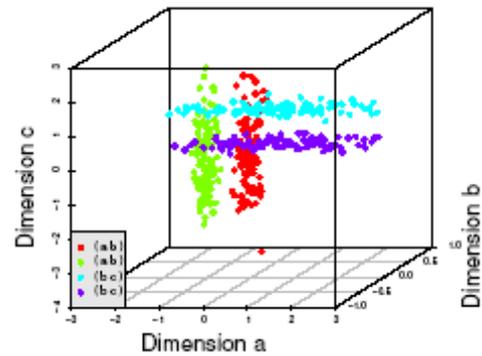




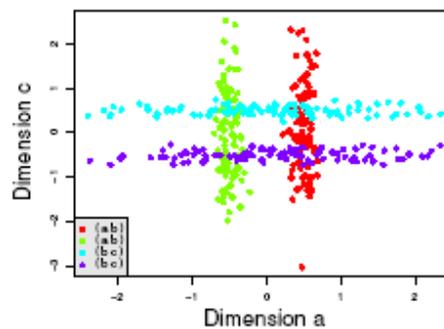
Proyecciones 2D



(a) Dims a & b



(b) Dims b & c



(c) Dims a & c



Algoritmos de clustering



Requisitos del algoritmo "perfecto"

- Escalabilidad.
- Manejo de distintos tipos de datos.
- Identificación de clusters con formas arbitrarias.
- Número mínimo de parámetros.
- Tolerancia frente a ruido y outliers.
- Independencia con respecto al orden de presentación de los patrones de entrenamiento.
- Posibilidad de trabajar en espacios con muchas dimensiones diferentes.
- Capacidad de incorporar restricciones especificadas por el usuario ("domain knowledge").
- Interpretabilidad / Usabilidad.

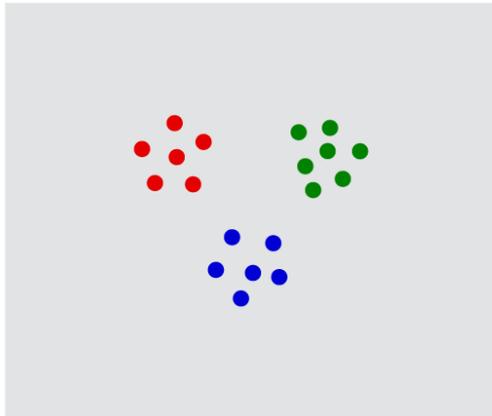


Teorema de imposibilidad

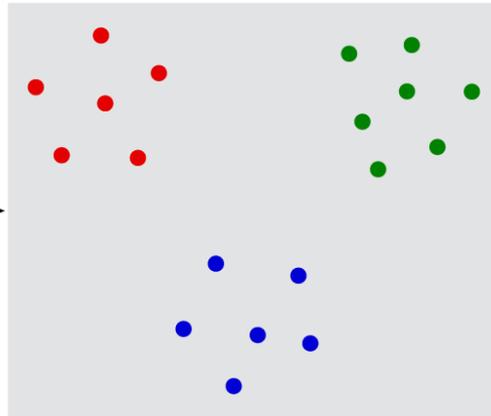


Invarianza frente a cambios de escala

Initial data and clustering



Same clustering after transformation

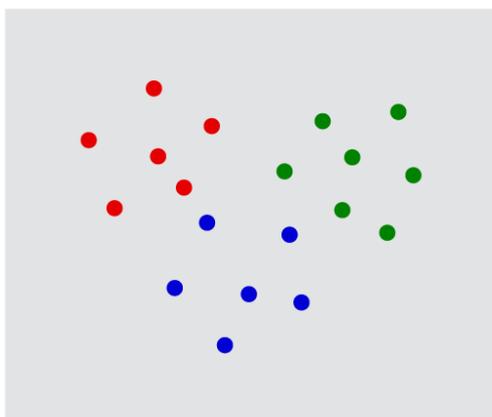


Teorema de imposibilidad

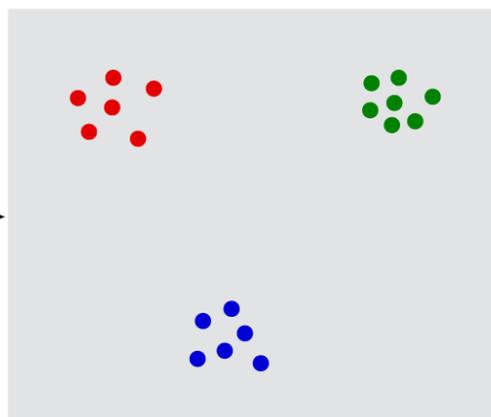


Consistencia

Initial data and clustering



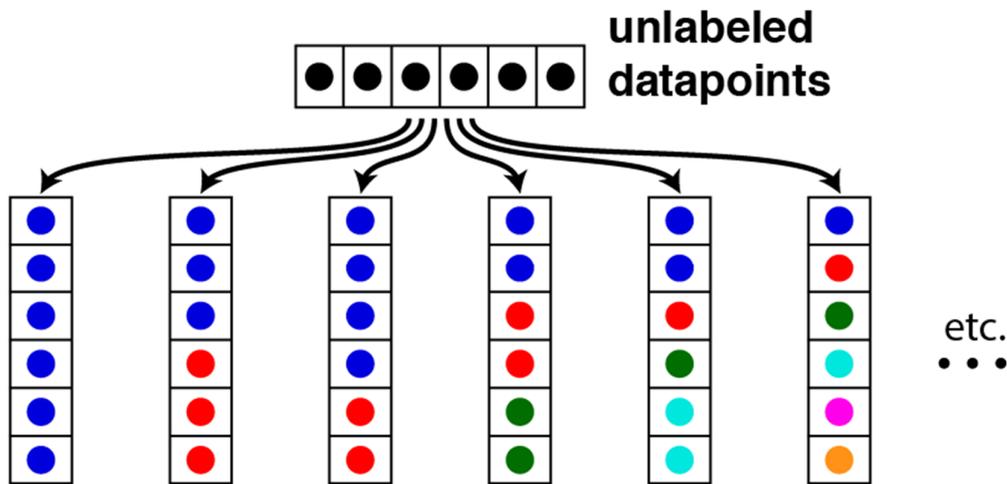
Same clustering after transformation



Teorema de imposibilidad



Riqueza



Teorema de imposibilidad



Ningún algoritmo de agrupamiento puede cumplir simultáneamente las 3 propiedades siguientes:

- **Invarianza frente a cambios de escala.**
- **Consistencia.**
- **Riqueza.**

Jon Kleinberg: "An Impossibility Theorem for Clustering"
15th International Conference on Neural Information
Processing Systems, NIPS'02, pages 463–470, 2002

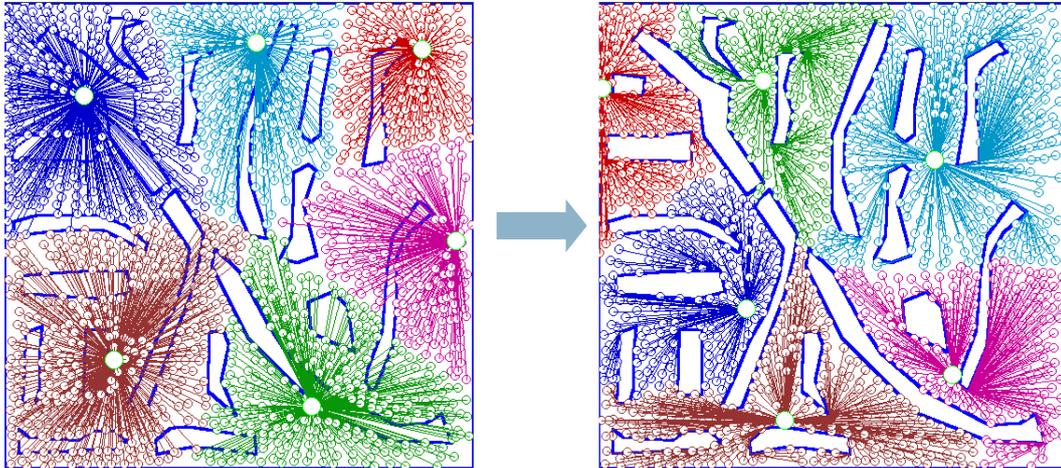


Extensiones y variaciones



Clustering con restricciones

p.ej. Clustering con obstáculos



Posibles aplicaciones:

Distribución de cajeros automáticos/supermercados...

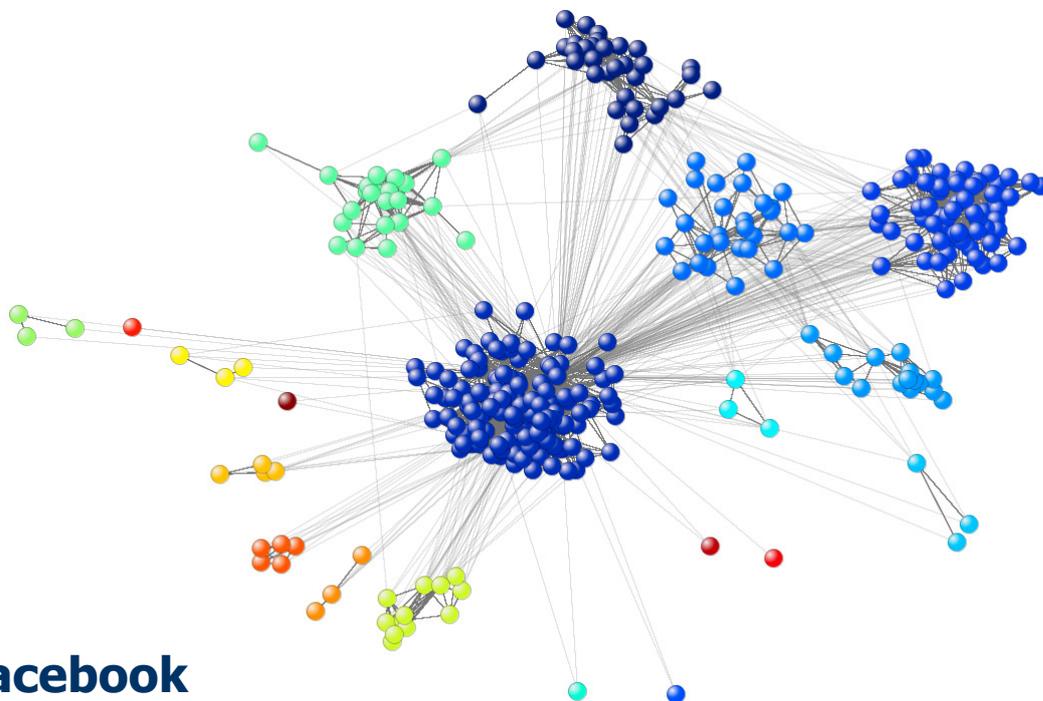


Extensiones y variaciones



Detección de comunidades

Redes sociales



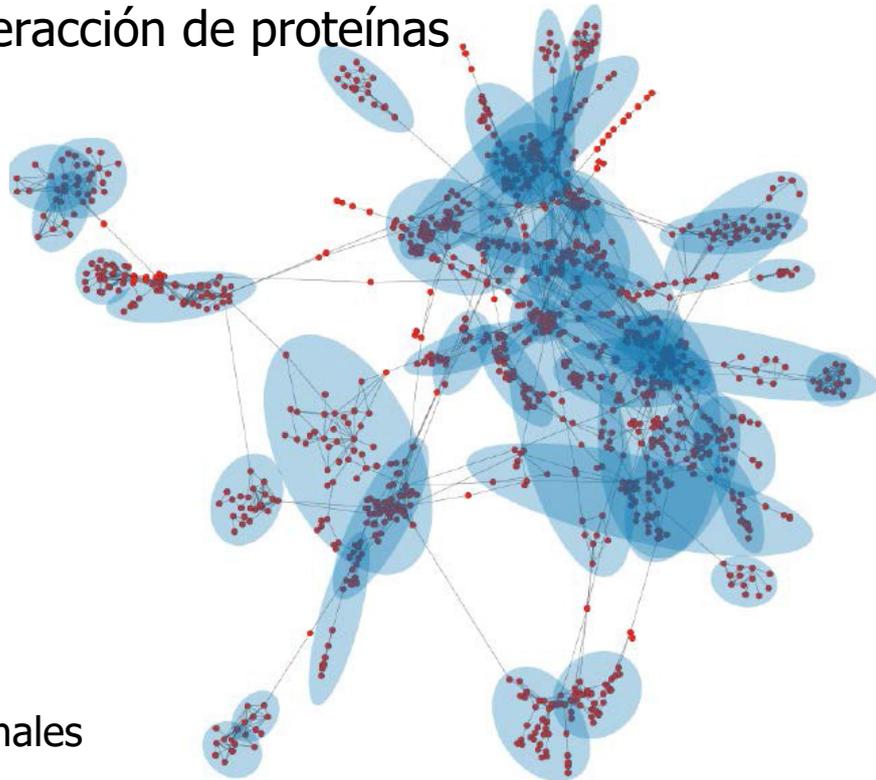
Facebook





Detección de comunidades

Redes de interacción de proteínas



Nodos
Proteínas

Enlaces
Interacciones

Comunidades
Módulos funcionales



DECSAI

Departamento de Ciencias de la Computación e I.A.

Universidad de Granada



Aprendizaje supervisado



■ Clasificación:

Para predecir el valor de un atributo categórico (discreto o nominal).

■ Regresión:

Para modelar funciones que toman valores continuos (esto es, predecir valores numéricos desconocidos).

Aplicaciones

Concesión de créditos, campañas de marketing dirigido, diagnóstico médico, detección de fraudes...



Aprendizaje supervisado

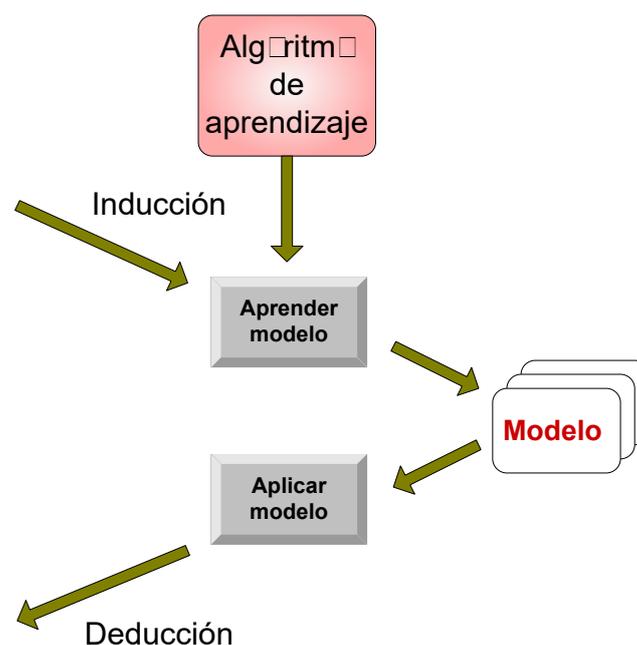


Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Conjunto de entrenamiento

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Conjunto de prueba





Construcción del modelo

- El conjunto de datos utilizado para construir el modelo de clasificación se denomina **conjunto de entrenamiento**.
- Cada caso/tupla/muestra corresponde a una clase predeterminada: los casos de entrenamiento vienen etiquetados por su atributo de clase.

Uso del modelo

- El modelo construido a partir del conjunto de entrenamiento se utiliza para clasificar nuevos datos.



Estimación de la precisión del modelo

Antes de construir el modelo de clasificación, se divide el conjunto de datos disponible en

- un **conjunto de entrenamiento** (para construir el modelo) y
- un **conjunto de prueba** (para evaluar el modelo).



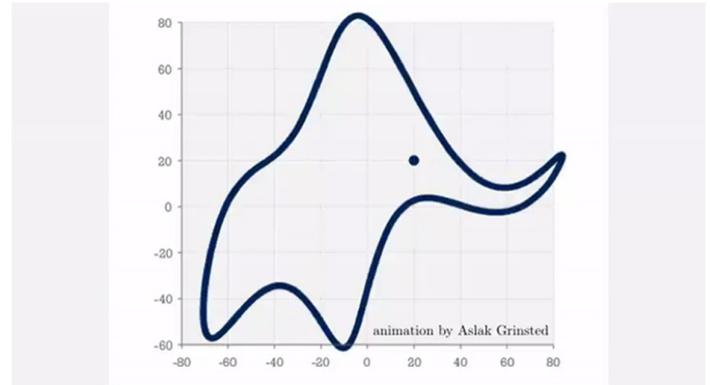
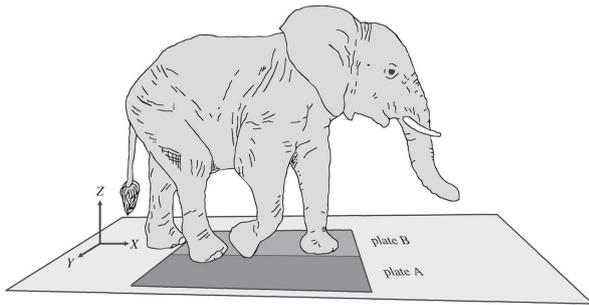
Aprendizaje supervisado



El problema del sobreaprendizaje

“Con cuatro parámetros puedo ajustar un elefante, con cinco puedo hacer que menee su trompa.”

-- John von Neumann



Jürgen Mayer, Khaled Khairy, y Jonathan Howard.
Drawing an elephant with four complex parameters.
American Journal of Physics, 78(6):648–649, 2010.
DOI 10.1119/1.3254017

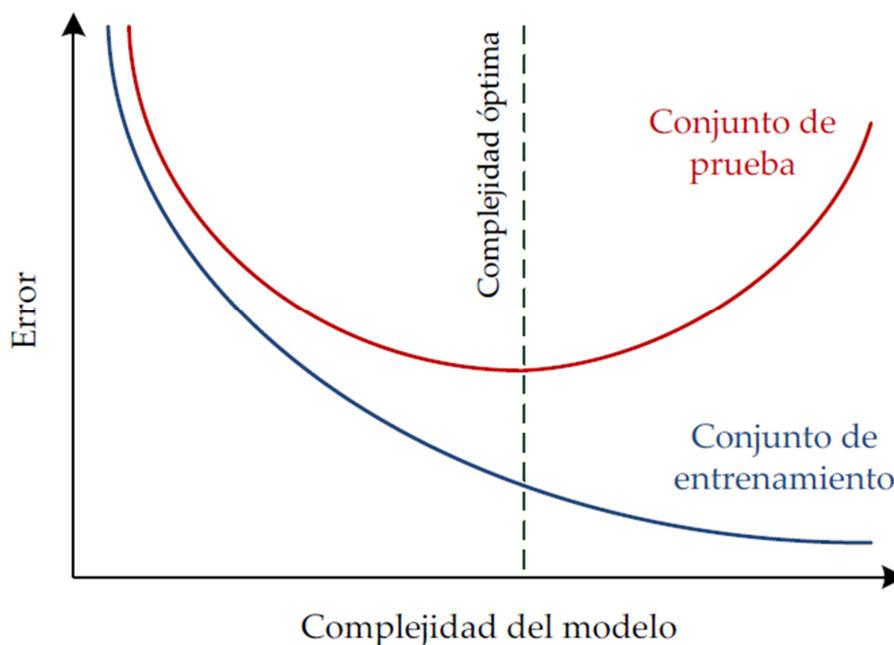


Aprendizaje supervisado



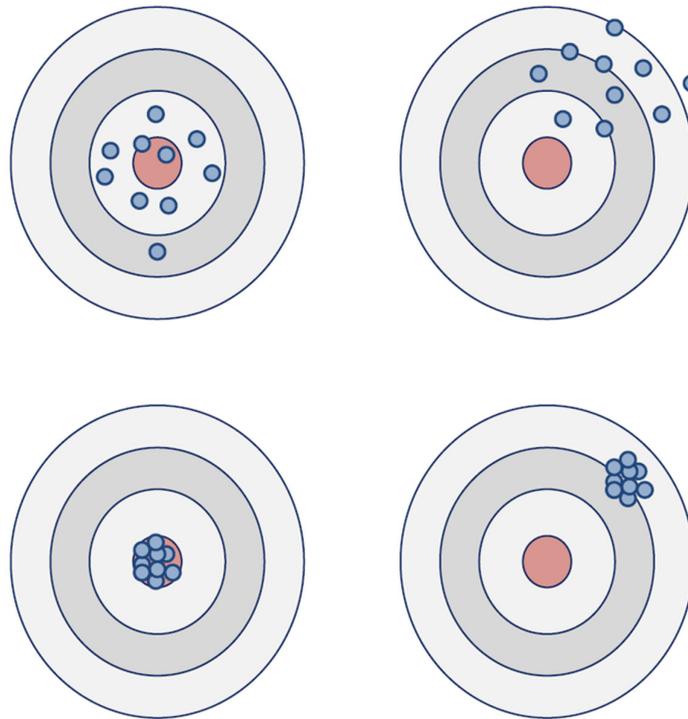
El problema del sobreaprendizaje

debido a la complejidad del modelo

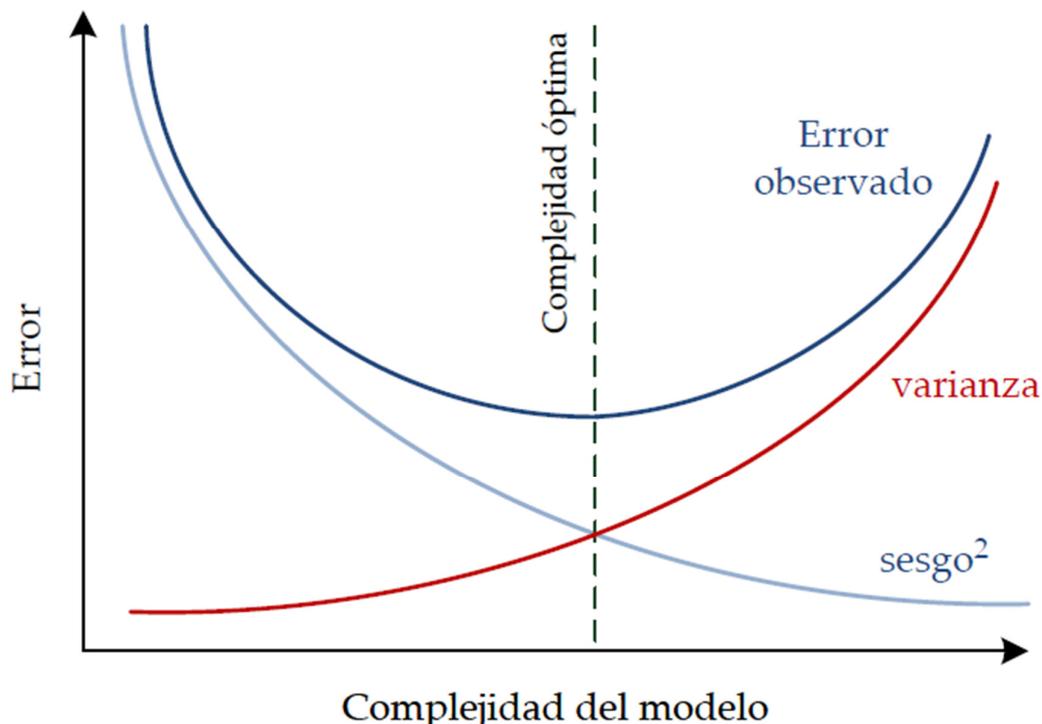




Descomposición del error en sesgo y varianza



Descomposición del error en sesgo y varianza





Criterios de evaluación

- **Precisión**
(porcentaje de casos clasificados correctamente).
- **Eficiencia**
(tiempo necesario para construir/usar el clasificador).
- **Robustez**
(frente a ruido y valores nulos)
- **Escalabilidad**
(utilidad en grandes bases de datos)
- **Interpretabilidad**
(el clasificador, ¿es sólo una caja negra?)
- **Complejidad**
(del modelo de clasificación) → Navaja de Occam.



Limitaciones de la precisión [accuracy]

Supongamos un problema con 2 clases no equilibradas:

- 99900 personas sanas
- 100 personas que padecen una enfermedad

Precisión engañosa: Si el modelo de clasificación siempre dice que los ejemplos son de la clase 1, su precisión es $99900/100000 = 99.9\%$

Paradoja de los falsos positivos: Una prueba diagnóstica con el 99% de precisión identificará 99 de los 100 casos existentes, pero también 999 falsos positivos. Sólo un **9%** de los positivos lo son realmente!!!





Medidas sensibles a costes [cost-sensitive]

		Predicción	
		P	N
Real	P	TP	FN
	N	FP	TN

Accuracy

		Predicción	
		P	N
Real	P	TP	FN
	N	FP	TN

Recall

		Predicción	
		P	N
Real	P	TP	FN
	N	FP	TN

Precision

		Predicción	
		P	N
Real	P	TP	FN
	N	FP	TN

F-measure



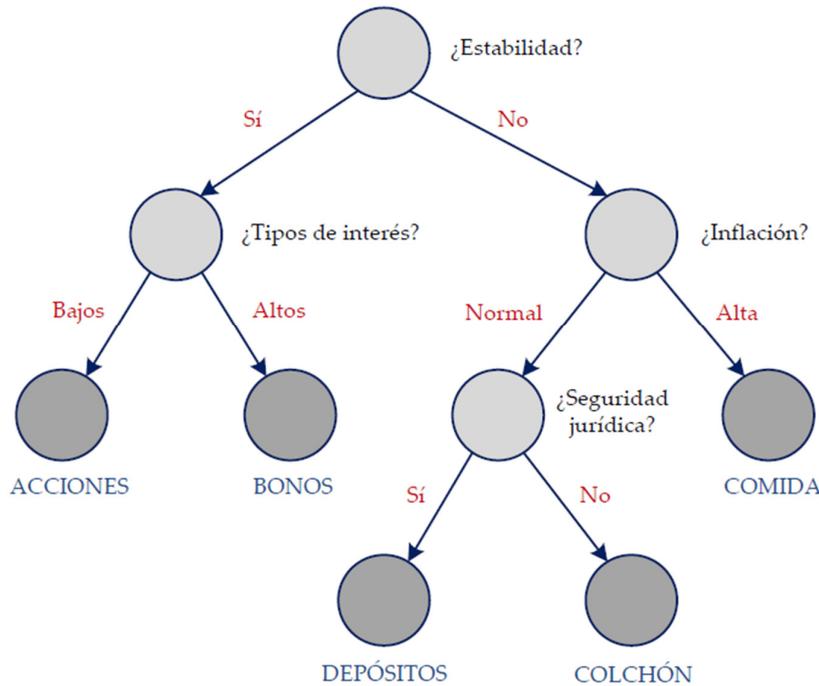
Algunos tipos de modelos de clasificación

- Modelos simbólicos
 - Árboles de decisión
 - Inducción de reglas (p.ej. listas de decisión)
- Modelos "estadísticos"
 - Clasificadores paramétricos
 - Modelos bayesianos, p.ej. redes bayesianas
- Modelos analógicos
 - Clasificadores basados en casos
 - SVMs (Support Vector Machines)
- Modelos conexionistas: Redes neuronales

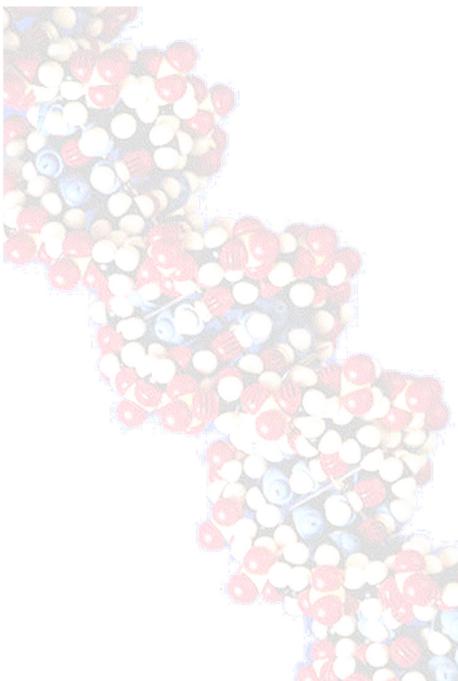




Modelos simbólicos: Árboles de decisión



Clasificadores asociativos (reglas de asociación) ART [Association Rule Trees]

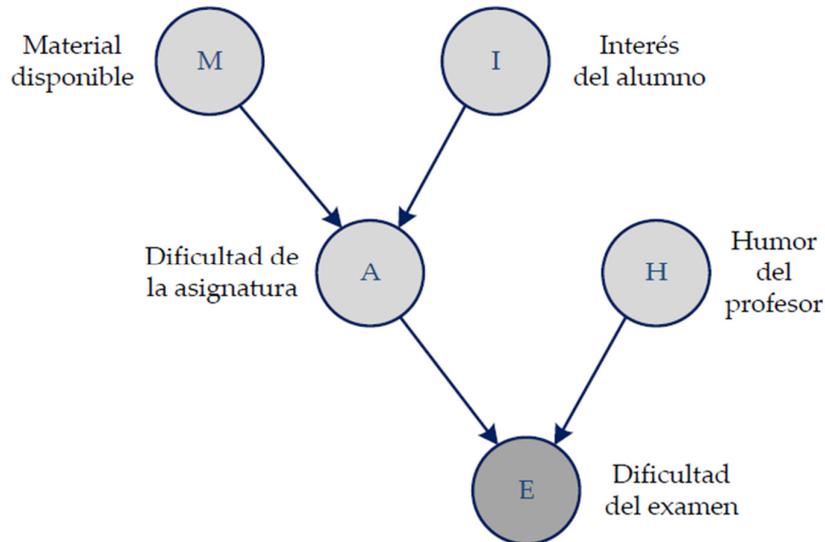


```
P30 = A : TYPE = N (473|62)
P30 = C : TYPE = N (441|24)
P30 = T : TYPE = N (447|57)
else
  P28 = A and P32 = T : TYPE = EI (235|33)
  P28 = G and P32 = T : TYPE = EI (130|20)
  P28 = C and P32 = A : TYPE = IE (160|31)
  P28 = C and P32 = C : TYPE = IE (167|35)
  P28 = C and P32 = G : TYPE = IE (179|36)
else
  P28 = A : TYPE = N (106|14)
  P28 = G : TYPE = N (94|4)
else
  P29 = C and P31 = G : TYPE = EI (40|5)
  P29 = A and P31 = A : TYPE = IE (86|4)
  P29 = A and P31 = C : TYPE = IE (61|4)
  P29 = A and P31 = T : TYPE = IE (39|1)
else
  P25 = A and P35 = G : TYPE = EI (54|5)
  P25 = G and P35 = G : TYPE = EI (63|7)
else
  P23 = G and P35 = G : TYPE = EI (40|8)
  P23 = T and P35 = C : TYPE = IE (37|7)
else
  P21 = G and P34 = A : TYPE = EI (41|5)
else
  P28 = T and P29 = A : TYPE = IE (66|8)
else
  P31 = G and P33 = A : TYPE = EI (62|9)
else
  P28 = T : TYPE = N (49|6)
else
  P24 = C and P29 = A : TYPE = IE (39|8)
else
  TYPE = IE (66|39)
```



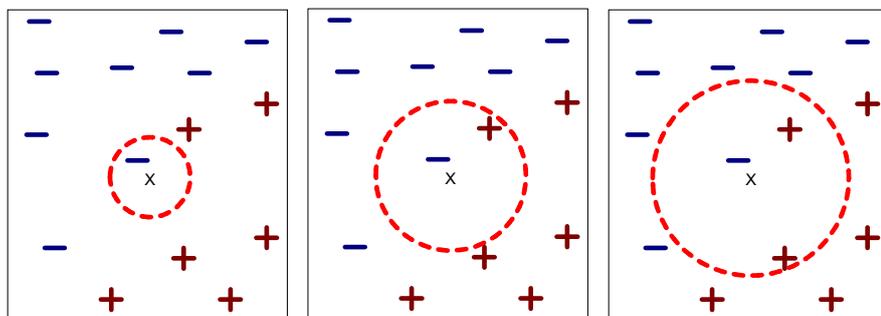


Redes bayesianas



Clasificadores basados en casos [lazy learners]

Almacenan el conjunto de entrenamiento (o parte de él) y lo utilizan directamente para clasificar nuevos datos.



Ejemplos

(a) 1-nearest neighbor

(b) 2-nearest neighbors

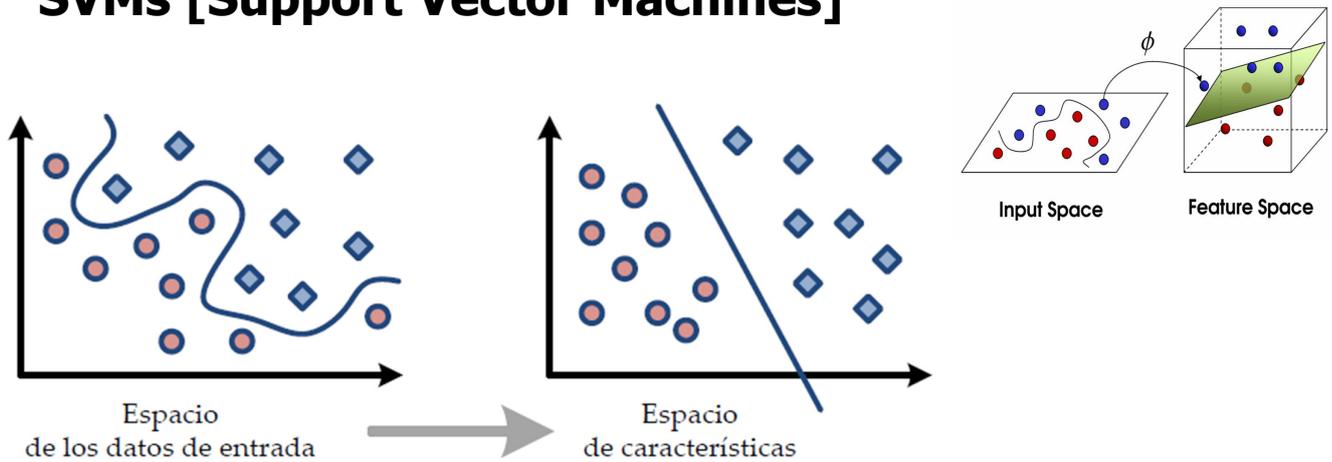
(c) 3-nearest neighbors

- k-NN (k Nearest Neighbors)
- Razonamiento basado en casos (CBR)





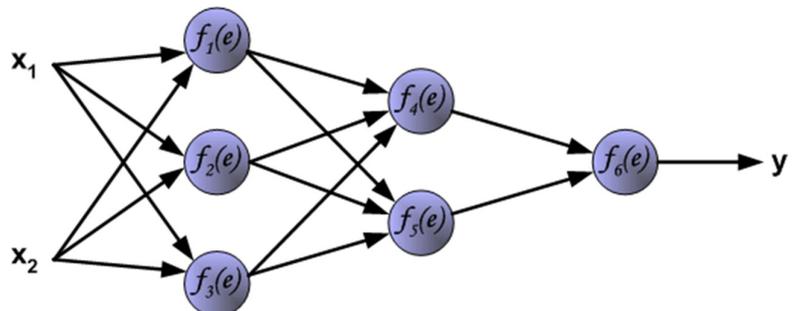
SVMs [Support Vector Machines]



De moda hasta la llegada del Deep Learning...



Redes neuronales p.ej. Backpropagation



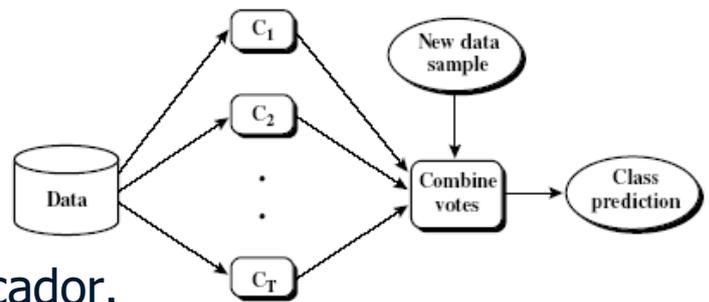
- Como "aproximadores universales", pueden aplicarse para predecir el valor de un atributo (tanto nominal como numérico).
- Ejemplo de modelo predictivo pero no descriptivo (podemos verlo como una caja negra).





“Ensembles”

Combinan varios modelos con el objetivo de mejorar la precisión final del clasificador.



- **Bagging:** Votación por mayoría. Varios clasificadores diferentes votan para decidir la clase de un caso de prueba (usa bootstrapping).
- **Boosting:** Votación ponderada. Los clasificadores tienen distintos pesos en la votación (en función de su precisión), vg: AdaBoost.

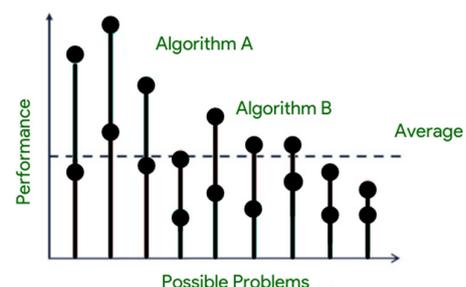


Aprendizaje automático



Teorema de Wolpert

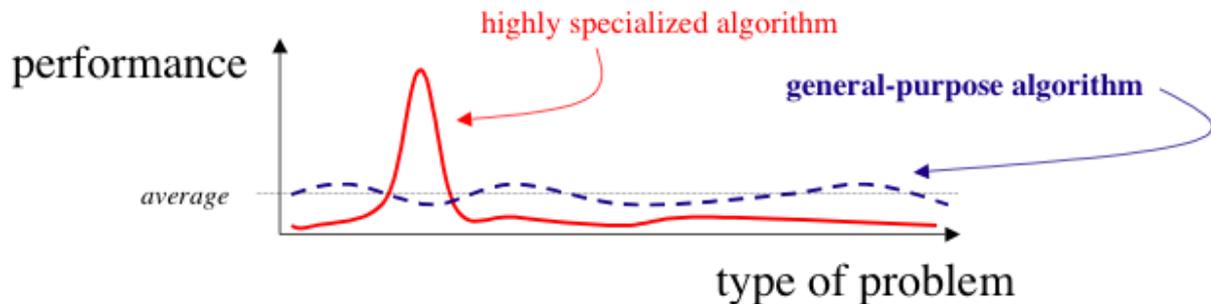
a.k.a. “No free lunch” theorem





Teorema de Wolpert

a.k.a. “No free lunch” theorem



No existe un algoritmo que sea siempre mejor que otro...



Ajuste de hiperparámetros



Hiperparámetros (o meta-parámetros)

Una de las principales dificultades prácticas del uso de técnicas de aprendizaje automático es la destreza que requiere establecer todos sus parámetros:

- **Parámetros** del modelo
(los parámetros que ajusta el algoritmo de aprendizaje)
- **Hiperparámetros** del modelo
(los parámetros que ajusta el que decide cómo ejecutar el algoritmo de aprendizaje).



Ajuste de hiperparámetros



Hiperparámetros (o meta-parámetros)

¿Cómo elegir los hiperparámetros de un modelo?

Método incorrecto: Se prueban montones de alternativas para ver cuál funciona mejor en el conjunto de test.

- Fácil de hacer, pero nos da una impresión engañosa de lo bien que funcionará la red en la práctica:
- La configuración que funcione mejor sobre el conjunto de prueba puede que no sea la que funcione mejor en otros conjuntos de prueba (o los nuevos casos sobre los que queremos aplicar la red neuronal).



Ajuste de hiperparámetros



Hiperparámetros (o meta-parámetros)

¿Cómo elegir los hiperparámetros de un modelo?

Un método mejor: Conjunto de validación.

Se divide el conjunto de datos disponible en tres partes:

- Conjunto de **entrenamiento** (para aprender los parámetros del modelo, e.g. pesos de una red neuronal).
- Conjunto de **validación** (no se utiliza en el entrenamiento, sino para decidir qué hiperparámetros resultan más adecuados)
- Conjunto de **prueba** (para obtener una estimación no sesgada de lo bien que funciona el modelo).



Ajuste de hiperparámetros



Hiperparámetros (o meta-parámetros)

¿Cómo elegir los hiperparámetros de un modelo?

Validación cruzada

- Dividimos el conjunto de datos en N subconjuntos.
- Utilizamos $N-1$ subconjuntos de conjunto de entrenamiento y el subconjunto restante de conjunto de prueba para obtener N estimaciones del error.



Ajuste de hiperparámetros



Hiperparámetros (o meta-parámetros)

¿Cómo elegir los hiperparámetros de un modelo?

AutoML

Aprendizaje automático [Machine Learning]

En vez de probar todas las combinaciones posibles de parámetros, podemos muestrear el espacio de posibles combinaciones.

- p.ej. Metaheurísticas (algoritmos genéticos)
Optimización bayesiana (procesos gaussianos)



Ajuste de hiperparámetros



Hiperparámetros (o meta-parámetros)

¿Cómo elegir los hiperparámetros de un modelo?

AutoML

Aprendizaje automático [Machine Learning]

- Mucho mejor que ir haciendo pruebas manualmente (no es el tipo de tarea que los humanos hacemos bien).
- Evita sesgos psicológicos no deseados: método menos propenso a funcionar mejor con el método que nos gusta y peor con el que no (las personas no podemos evitarlo ;-)

