Aprendizaje de Máquina
No supervisado

REGLAS DE ASOCIACIÓN

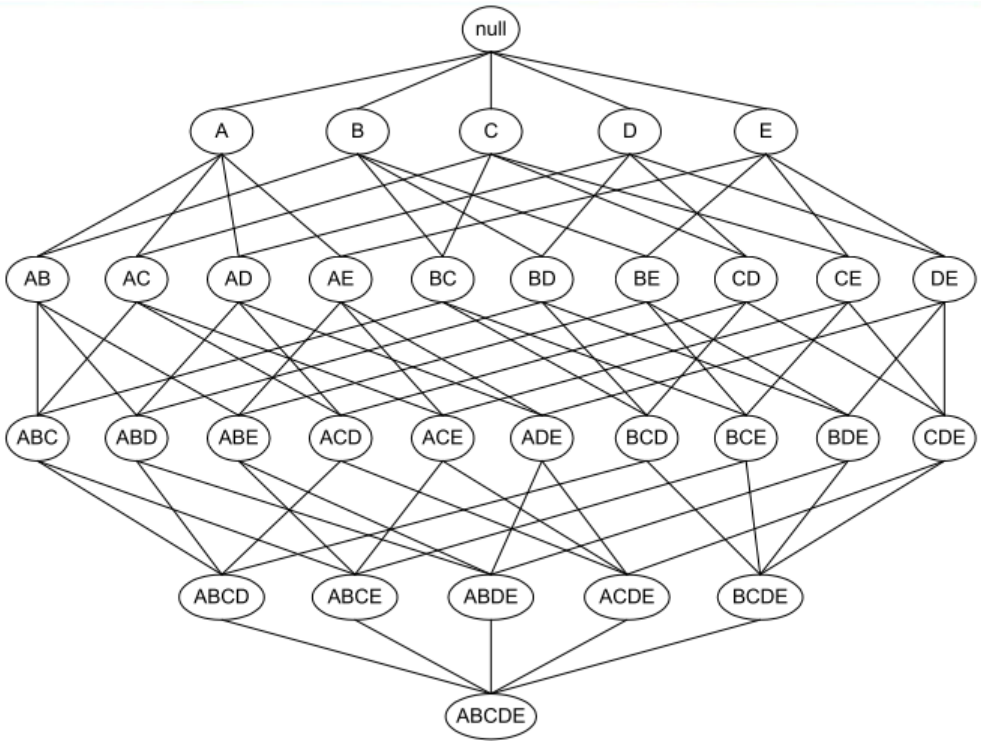
Algoritmo a-priori

- Algoritmo para encontrar Reglas de asociación en un conjunto de datos. Este algoritmo se basa en el conocimiento previo o “a priori” de los conjuntos frecuentes para reducir el espacio de búsqueda y aumentar la eficiencia.

Algoritmo a-priori

- 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.
  - Análisis de afinidad (Market Basket Analysis)

Algoritmo a-priori



Para 5 productos a 2^5 combinaciones

Algoritmo a-priori

- Solución por fuerza bruta
  - Enumerar todas las reglas de asociación posibles
  - Calcular el soporte y la confianza de cada regla
  - Eliminar las reglas que no superen los umbrales de soporte y confianza

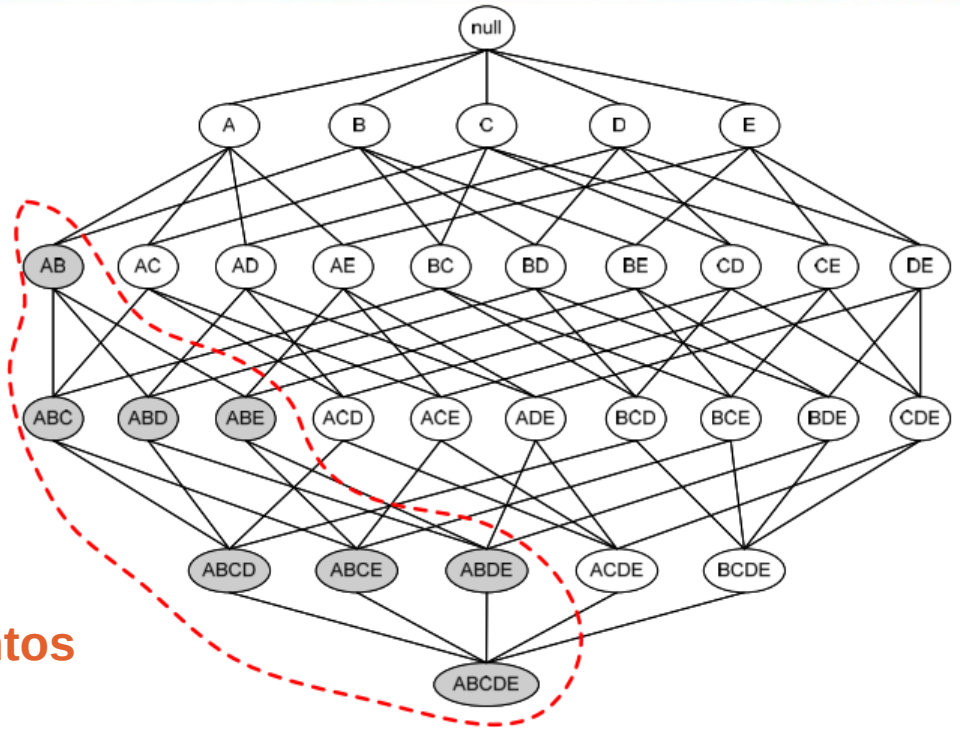
- Cálculo por fuerza bruta es computacionalmente imposible cuando el número de productos es grande. $O(T \cdot 2^d)$

Algoritmo a-priori

- Propiedad a-priori
  - Si un conjunto de artículos es frecuente también lo son todos los superconjuntos
  - Si un artículo no es frecuente podemos eliminar todos los superconjuntos asociados (podamos el árbol de combinaciones).

Algoritmo a-priori

Transacción no frecuente



Superconjuntos podados

Algoritmo a-priori

- Tablas

  L[k] = Conjunto de artículos de tamaño k frecuentes

  C[k] = Conjunto de artículos de tamaño k potencialmente frecuentes
- Algoritmo

  Generar L[1] (artículos frecuentes de tamaño 1)

  Repetir mientras se descubran nuevos conjuntos de artículos frecuentes

    Generar los candidatos C[k+1] a partir de los patrones frecuentes L[k]

    Contabilizar el soporte de cada candidato de C[k+1] recorriendo la base de datos secuencialmente

    Eliminar candidatos no frecuentes, dejando en L[k+1] sólo aquellos que son frecuentes

Algoritmo a-priori

ID	Artículos
1	Pan, Leche, Huevos
2	Pan, Pañales, Cerveza
3	Leche, Pañales, Cerveza
4	Pan, Leche, Pañales, Cerveza
5	Pan, Leche, Huevos, Cerveza

L[1]	Soporte
Pan	4
Leche	4
~~Huevos~~	~~2~~
Pañales	3
Cerveza	4

C[2]
{Pan, Leche}
{Pan, Pañales}
{Pan, Cerveza}
{Leche, Pañales}
{Leche, Cerveza}
{Pañales, Cerveza}

Soporte mínimo 3

Algoritmo a-priori

ID	Artículos
1	Pan, Leche, Huevos
2	Pan, Pañales, Cerveza
3	Leche, Pañales, Cerveza
4	Pan, Leche, Pañales, Cerveza
5	Pan, Leche, Huevos, Cerveza

L[2]	Soporte
{Pan, Leche}	3
~~{Pan, Pañales}~~	~~2~~
{Pan, Cerveza}	3
~~{Leche, Pañales}~~	~~2~~
{Leche, Cerveza}	3
{Pañales, Cerveza}	3

C[2]
{Pan, Leche, Cerveza}
{Pan, Cerveza, Pañales}
{Leche, Cerveza, Pañales}

Algoritmo a-priori

ID	Artículos
1	Pan, Leche, Huevos
2	Pan, Pañales, Cerveza
3	Leche, Pañales, Cerveza
4	Pan, Leche, Pañales, Cerveza
5	Pan, Leche, Huevos, Cerveza

L[2]	Soporte
~~{Pan, Leche, Cerveza}~~	~~2~~
~~{Pan, Cerveza, Pañales}~~	~~2~~
~~{Leche, Cerveza, Pañales}~~	~~2~~

Algoritmo a-priori

- Terminamos con

C
Pan
Leche
Pañales
Cerveza
{Pan, Leche}
{Pan, Cerveza}
{Leche, Cerveza}
{Pañales, Cerveza}

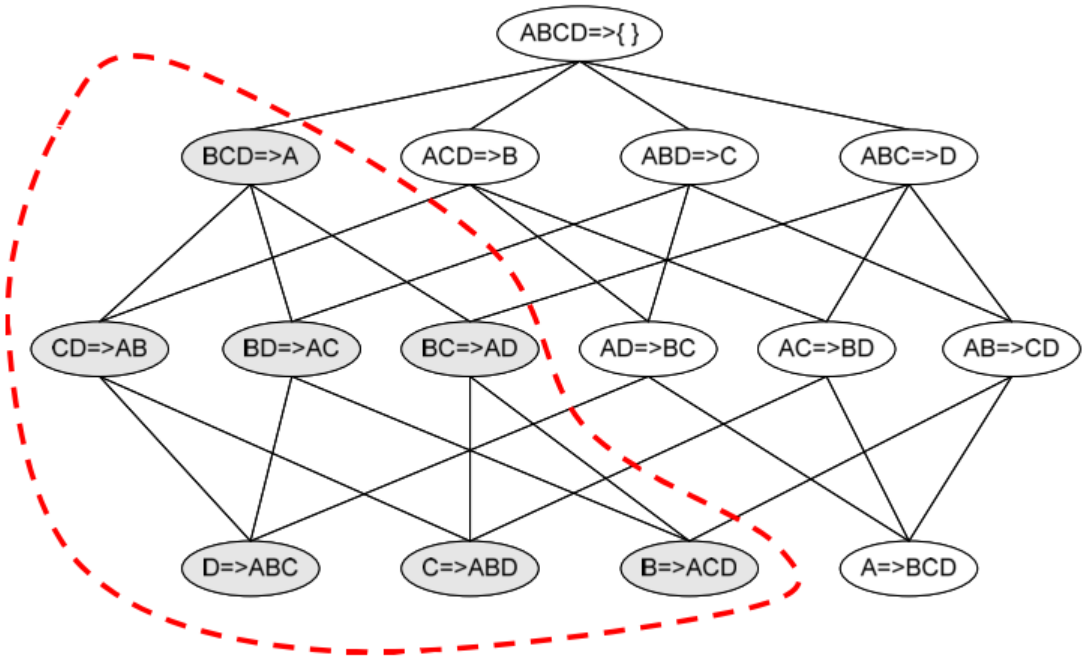
Algoritmo a-priori

- Dado un artículo frecuente L, se encuentran todos los subconjuntos no vacíos $f \subset L$ tales que $f \rightarrow L - f$ satisfaga el umbral mínimo de confianza
- La confianza de las reglas generadas de un mismo conjunto de artículos tienen la siguiente propiedad:

  L={A,B,C,D}

  c(ABC → D) ≥ c(AB → CD) ≥ c(A → BCD)

Algoritmo a-priori



Reglas de baja confianza

Algoritmo a-priori

- Confianza 75%

L	Confianza
Pan → Leche	3/4 - 75%
Leche → Pan	3/4 - 75%
Pan → Cerveza	3/4 - 75%
Cerveza → Pan	3/4 - 75%
Cerveza → Leche	3/4 - 75%
Leche → Cerveza	3/4 - 75%
Pañales → Cerveza	3/3 – 100%
Cerveza → Pañales	3/4 - 75%

No se incluyen las reglas de un solo producto

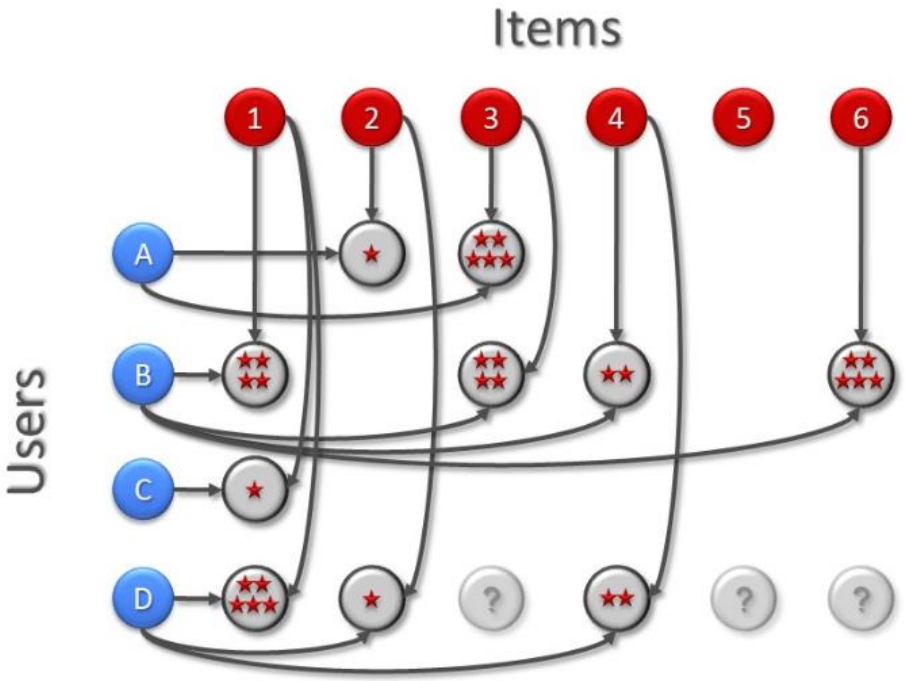
SISTEMAS DE RECOMENDACIÓN

Sistemas de recomendación

- Son útiles para entender mejor la demanda y planear adecuadamente la oferta a fin de optimizar el proceso de producción.
- Muchos sectores económicos (Transporte, retail, entretenimiento, finanzas) dependen de las preferencias de las personas involucradas en las transacciones.

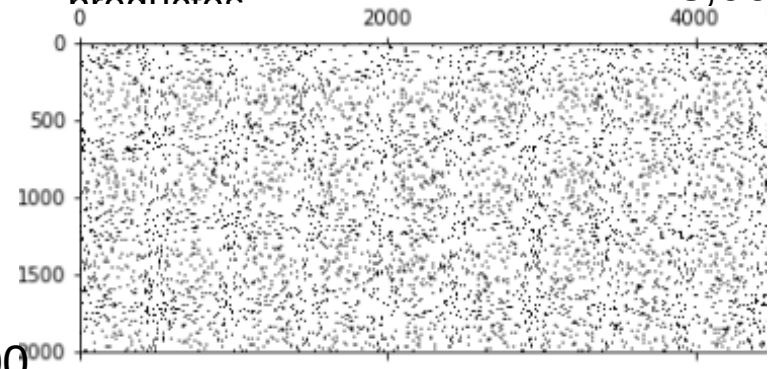
La **teoría de la elección social** es un marco teórico para el análisis de la combinación de opiniones individuales, preferencias, intereses o bienestar para llegar a una decisión colectiva o bienestar social.

Cómo predecir ratings?



- Es necesario construir una matriz con los usuarios como renglones, los productos como columnas y las calificaciones en las entradas.
- Hay un gran porcentaje de datos faltantes porque típicamente los usuarios sólo califican un pequeño subconjuntos de productos

18,000



500,000

Cada película 5,000 calif. en promedio

¿Cómo predecir ratings? Y^T

	Juan	Paco	Pedro	Mar
La sociedad de los poetas muertos	5	5	0	0
Cinema Paradiso	5	?	?	0
La lista de Schindler	?	4	0	?
Star Wars	0	0	5	4
Star Trek	0	0	5	?

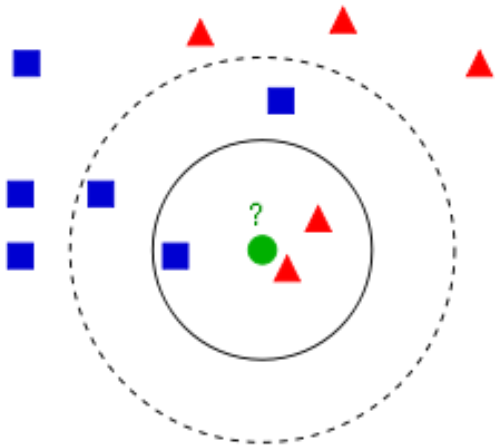
Problema de regresión

- Obtengo las características de las películas (actores, categoría, país de origen, tuvo final feliz, etc.) para predecir las calificaciones con base en las que han sido calificadas.
- ¿Tenemos acceso a todas las características, o a las principales por lo menos, que influyen en la calificación de un usuarios?
- ¿Qué pasa si en vez de películas tengo productos como en Amazon? Cada artículo tiene un conjunto distinto de características que lo describe.
- ¿Tenemos suficientes evaluaciones de cada usuario para predecir los faltantes?

K-vecinos cercanos

- A partir de un nuevo conjunto de valores de entrada, se predice el resultado utilizando los k datos de entrenamiento más cercanos al nuevo dato
  - Regresión: se regresa el promedio del valor a predecir de los k datos más cercanos

K-vecinos cercanos



En este caso, para el nuevo dato representado por el punto verde, sería clasificado como triángulo para k=3 y como cuadrado para k=5

Distancia euclidiana

- Se utiliza la distancia euclidiana entre los datos para determinar su cercanía

$$x^{(i)} = \left(x_1^{(i)}, x_2^{(i)}, \ldots, x_n^{(i)}\right)$$

$$d\left(x^{(i)}, x^{(j)}\right) = \sqrt{\sum_{r=1}^{p}\left(x_r^{(i)} - x_r^{(j)}\right)^2}$$

Similitud coseno

- La similitud existente entre dos vectores en un espacio que posee un producto punto con el que se evalúa el valor del coseno del ángulo comprendido entre ellos
- Valores:
  - 1 ambos vectores apuntan en el mismo sentido
  - 0 ambos vectores son ortogonales
  - -1 ambos vectores apuntan en sentido contrario

$$A \cdot B = \|A\|\|B\| \cos\theta$$

$$similitud = \cos\theta = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

¿Cómo predecir ratings? Y^T

	Juan	Paco	Pedro	Mar
La sociedad de los poetas muertos	5	5	0	0
Cinema Paradiso	5	?	?	0
La lista de Schindler	?	4	0	?
Star Wars	0	0	5	4
Star Trek	0	0	5	?

¿Cómo predecir ratings? Y^T

- Y_{ai} es la calificación que el usuario a le asignó a la película i
- Calcularemos la estimación de las calificaciones que no tenemos para algún usuario en particular usando KNN

$$\hat{Y}_{ai} = \frac{\sum_{b \in KNN(a,i)} Y_{bi}}{K}$$

$$\hat{Y}_{ai} = \frac{\sum_{b \in KNN(a,i)} sim(a,b) Y_{bi}}{\sum_{b \in KNN(a,i)} sim(a,b)}$$

- Donde obtengo los k vecinos más cercanos dónde la película i sí fue calificada

¿Cómo predecir ratings? Y^T

- El éxito del método KNN depende en gran medida de la elección de la medida de similitud.
- Los usuarios miden de forma distinta las películas, hay algunos más exigentes que otros.
- Podemos utilizar en vez de la similitud en calificaciones, la similitud entre las desviaciones (positivas y negativas) con respecto a la media de las calificaciones de cada usuario.
- Este método no permite detectar las estructuras ocultas que hay en los datos: un usuario puede ser similar a un grupo de usuarios en una dimensión, pero similar a algún otro conjunto de usuarios en una dimensión diferente.

Filtrado colaborativo

- El filtrado colaborativo o factorización de matrices me permite encontrar las agrupaciones ocultas tanto para los usuarios como para las productos.
- De esta forma no se tiene que crear una medida de similitud sofisticada específica para distintos escenarios

¿Cómo predecir ratings? Y^T

	Juan	Paco	Pedro	Mar	Drama	Acción
La sociedad de los poetas muertos	5	5	0	0	90%	0%
Cinema Paradiso	5	?	?	0	100%	1%
La lista de Schindler	?	4	0	?	99%	0%
Star Wars	0	0	5	4	10%	100%
Star Trek	0	0	5	?	0%	90%

¿Cómo predecir ratings? Y^T

	Juan	Paco	Pedro	Mar	Drama	Acción
La sociedad de los poetas muertos	5	5	0	0	90%	0%
Cinema Paradiso	5	?	?	0	100%	1%
La lista de Schindler	?	4	0	?	99%	0%
Star Wars	0	0	5	4	10%	100%
Star Trek	0	0	5	?	0%	90%

¿Cómo predecir ratings? Y^T

	Juan	Paco	Pedro	Mar	Drama	Acción
La sociedad de los poetas muertos	5	5	0	0	90%	0%
Cinema Paradiso	5	?	?	0	100%	1%
La lista de Schindler	?	4	0	?	99%	0%
Star Wars	0	0	5	4	10%	100%
Star Trek	0	0	5	?	0%	90%

¿Cómo predecir ratings? Y^T

	Juan	Paco	Pedro	Mar	Drama	Acción
La sociedad de los poetas muertos	5	5	0	0	90%	0%
Cinema Paradiso	5	?	?	0	100%	1%
La lista de Schindler	?	4	0	?	99%	0%
Star Wars	0	0	5	4	10%	100%
Star Trek	0	0	5	?	0%	90%

¿Cómo predecir agrupaciones (traits)? Y^T

	Juan	Paco	Pedro	Mar	Drama	Acción
La sociedad de los poetas muertos	5	5	0	0	?	?
Cinema Paradiso	5	?	?	0	?	?
La lista de Schindler	?	4	0	?	?	?
Star Wars	0	0	5	4	?	?
Star Trek	0	0	5	?	?	?

¿Cómo predecir agrupaciones? Y^T

	Juan	Paco	Pedro	Mar	Drama	Acción
La sociedad de los poetas muertos	5	5	0	0	90%	0%
Cinema Paradiso	5	?	?	0	100%	1%
La lista de Schindler	?	4	0	?	99%	0%
Star Wars	0	0	5	4	10%	100%
Star Trek	0	0	5	?	0%	90%

Función objetivo (método ingenuo)

- Queremos obtener una matriz resultante X completamente calificada a partir de la matriz rala Y que se nos proporciona como entrada.
- La función de costo inicial la definiremos como que tanto se parecen los valores que sí tengo en la matriz Y a los que tengo en mi matriz resultante X

$$D = \{(a,i)|Y_{ai}\ es\ dada\}$$

$$J(X) = \frac{1}{2}\sum_{(a,i)\in D}(Y_{ai} - X_{ai})^2 + \frac{\lambda}{2}\sum_{(a,i)} {X_{ai}}^2$$

Optimización (método ingenuo)

- La función de costo no implica interacción entre usuarios y productos, por lo que podemos calcularla para cada elemento por X_{ai} separado

1. $(a, i) \in D$

$$\frac{\partial J(X_{ai})}{\partial X_{ai}} = -(Y_{ai} - X_{ai}) + \lambda X_{ai} = 0$$

$$X_{ai} = \frac{Y_{ai}}{1 + \lambda}$$

3. $(a, i) \notin D$

$$\frac{\partial J(X_{ai})}{\partial X_{ai}} = \lambda X_{ai} = 0$$

$$X_{ai} = 0$$

¿Qué está mal del método ingenuo?

- Debido a que tomamos como independiente cada elemento X_{ai}, no estamos modelando las interacciones entre los distintos usuarios y películas que es lo que queríamos hacer en primer lugar.
- Teneos un número muy grande de parámetros (n usuarios*m productos) siendo estimado a partir de un conjunto muy pequeño valores conocidos.

Factorización de matrices

- El objetivo es no estimar de forma independiente los parámetros para encontrar las agrupaciones ocultas que desconocemos
- Para ello debemos reducir el número de parámetros a calcular
- Supuesto: X es una matriz de bajo rango
  - Rango de una matriz: el número máximo de columnas (filas respectivamente) que son linealmente independientes.
- Por ejemplo, la siguiente matriz es de rango 1

$$\begin{bmatrix} 1 & 2 & 3 \\ 5 & 10 & 15 \end{bmatrix} = \begin{bmatrix} 1 \\ 5 \end{bmatrix} [1 \quad 2 \quad 3] = \begin{bmatrix} 10 \\ 50 \end{bmatrix} [0.1 \quad 0.2 \quad 0.3]$$

$$U \qquad V$$

$$O(n * m) \qquad O(n + m) \qquad X = UV^T$$

Factorización de matrices

- En el caso de que el rango de la matriz sea 1, U y V son vectores, donde U estaría representando el sentimiento general de cada usuario hacia las películas y V representa como cada una de las películas es percibida por los usuarios.
- Conforme aumentamos el rango de la matriz X, vamos agregando mas agrupaciones que caracterizan a los usuarios y a las películas.
- El rango K constituiría un hiperparámetro a escoger de forma que optimice nuestra función de costo

Minimización alternada

$$D = \{(a,i)|Y_{ai}\ es\ dada\}$$

$$J(X) = \frac{1}{2}\sum_{(a,i)\in D}(Y_{ai} - [UV^T]_{ai})^2 + \frac{\lambda}{2}\sum_{a=1}^{n}\sum_{j=1}^{k}{U_{aj}}^2 + \frac{\lambda}{2}\sum_{i=1}^{m}\sum_{j=1}^{k}{V_{ij}}^2$$

- Seleccionamos V al azar y la dejamos fija y optimizamos con respecto a U
- Una vez actualizada la U, la dejamos fija y optimizamos con respecto a V
- Repetimos hasta que converja (variaciones entre las estimaciones de los vectores es pequeña) (optimo local)

AGRUPAMIENTO

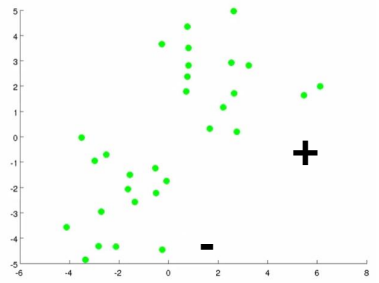
Revisión de conceptos básicos

- Vectores de características, etiquetas $x \in \mathbb{R}^d, y \in \{-1,1\}$
- Conjunto de entrenamiento $\mathcal{S}_n = \{(x^{(i)}, y^{(i)}), i = 1,2,\dots,n\}$
- Clasificador $h: \mathbb{R}^d \to \{-1,1\}, h(x) = 1, \mathcal{X}^+ \{x \in \mathbb{R}^d : h(x) = 1\}$, $\mathcal{X}^- \{x \in \mathbb{R}^d : h(x) = -1\}$
- Error de entrenamiento $\mathcal{E}_n(h) = \frac{1}{n}\sum_{i=1}^{n} [\![h(x^{(i)}) \neq y^{(i)}]\!]$
- Error de prueba $\mathcal{E}(h)$
- Conjunto de clasificadores $h \in \mathcal{H}$

Aprendizaje supervisado vs. no supervisado

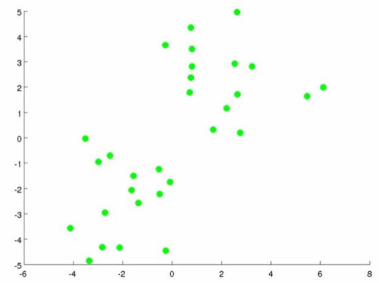
- Supervisado

$$S_n = \{(x^{(i)}, y^{(i)}) | i = 1 \dots n\}$$



- No supervisado

$$S_n = \{(x^{(i)}) | i = 1 \dots n\}$$

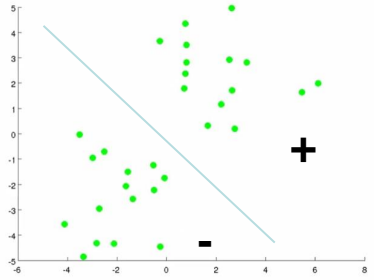


Aprendizaje supervisado vs. no supervisado

- Supervisado

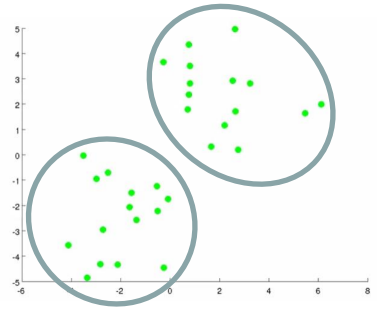
$$S_n = \{(x^{(i)}, y^{(i)}) | i = 1 \dots n\}$$

Clasificación



- No supervisado

$$S_n = \{(x^{(i)}) | i = 1 \dots n\}$$



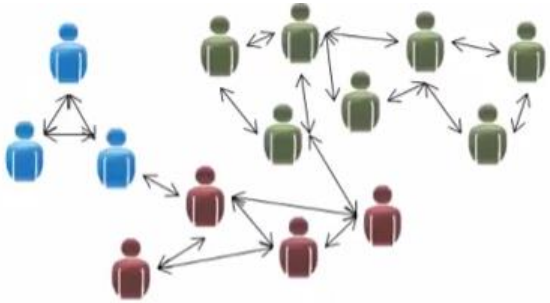
Agrupamiento

Aprendizaje no supervisado

- Infiere una función que describe la estructura de datos no etiquetados
- No hay señal de error ni de recompensa para evaluar una solución potencial
- Busca resumir y explicar las principales características de los datos
- Típicamente trata las entrada como un conjunto de variables aleatorias, siendo construido un modelo de densidad para ese conjunto de datos.

Ejemplos de aplicación

Segmentación de mercado

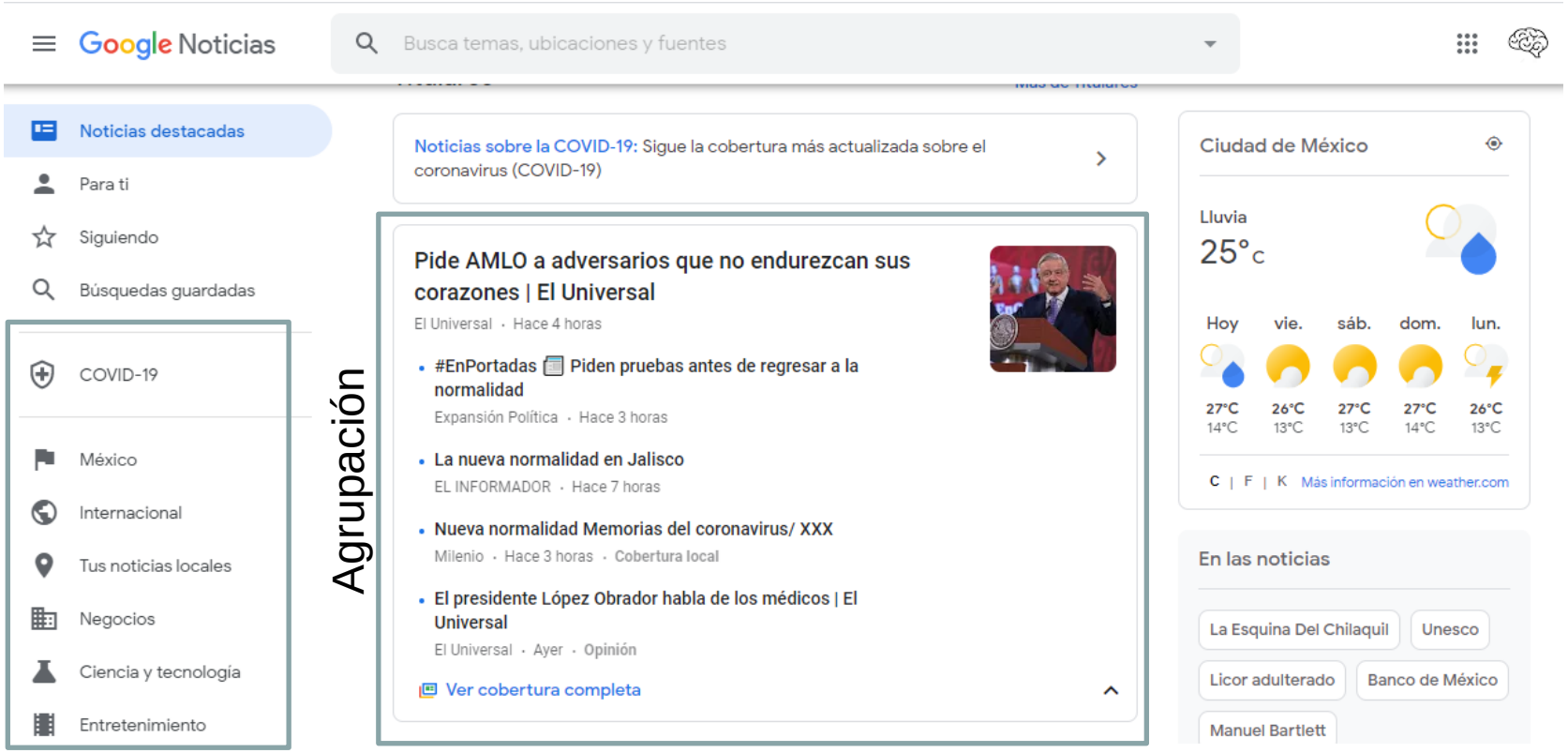


Análisis de redes sociales

Organizar centros de datos

Análisis de datos astronómicos

Ejemplos


Google Noticias
Busca temas, ubicaciones y fuentes
Noticias destacadas
Para ti
Siguiendo
Búsquedas guardadas
COVID-19
México
Internacional
Tus noticias locales
Negocios
Ciencia y tecnología
Entretenimiento
Clasificación
Noticias sobre la COVID-19: Sigue la cobertura más actualizada sobre el coronavirus (COVID-19)
Pide AMLO a adversarios que no endurezcan sus corazones | El Universal
El Universal · Hace 4 horas
#EnPortadas Piden pruebas antes de regresar a la normalidad
Expansión Política · Hace 3 horas
La nueva normalidad en Jalisco
EL INFORMADOR · Hace 7 horas
Nueva normalidad Memorias del coronavirus/ XXX
Milenio · Hace 3 horas · Cobertura local
El presidente López Obrador habla de los médicos | El Universal
El Universal · Ayer · Opinión
Ver cobertura completa
Agrupación
Ciudad de México
Lluvia
25°C
Hoy vie. sáb. dom. lun.
27°C 14°C
26°C 13°C
27°C 13°C
27°C 14°C
26°C 13°C
C | F | K Más información en weather.com
En las noticias
La Esquina Del Chilaquil
Unesco
Licor adulterado
Banco de México
Manuel Bartlett

Cuantificación de una imagen

- Tamaño de una imagen con pixeles de 24 bits

  1024 * 1024 * 24 bits ~ 3MB (rojo: 8 bits, azul: 8 bits, verde: 8 bits)
- No usar todos los bits

  En vez de 24, 16

2^24 colores

2 colores

4 colores

16 colores

Distancia y disimilitud

- $D \in \mathbb{R}^{n \times n}$ es una matriz de distancia si:

$$D_{ii} = 0, D_{ij} \geq 0, D_{ij} = D_{ji}, D_{ij} = D_{ik} + D_{kj} \; para \; toda \; i, j, k$$

  - Por ejemplo: distancia euclidiana, distancia de Manhattan, distancia máxima,…
- $D \in \mathbb{R}^{n \times n}$ es una matriz de disimilitud si:

$$D_{ii} = 0, D_{ij} \geq 0, D_{ij} = D_{ji} \; para \; toda \; i, j$$

  - Más flexible que las distancias, funciona p. ejemplo para rankings

Distancia euclidiana

- Se utiliza la distancia euclidiana entre los datos para determinar su cercanía

$$x^{(i)} = \left(x_1^{(i)}, x_2^{(i)}, \ldots, x_n^{(i)}\right)$$

$$d\left(x^{(i)}, x^{(j)}\right) = \sqrt{\sum_{r=1}^{p} \left(x_r^{(i)} - x_r^{(j)}\right)^2}$$

Variables cualitativas

- Ordinales

$$\frac{i - {}^{1}/_{2}}{M}, i = 1, \dots, M$$

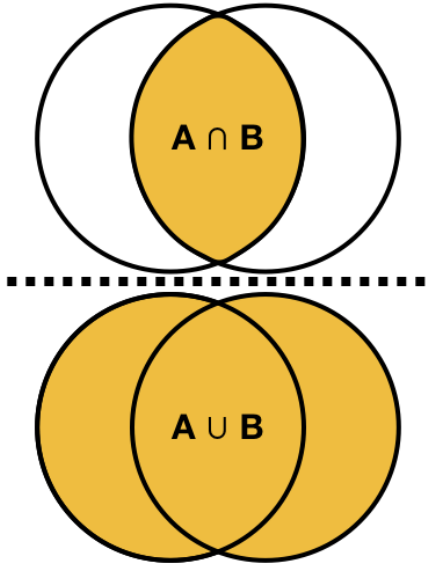
- Nominales
  - Si la variable tiene M valores distintos, se define explícitamente la distancia como una matriz de M x M con elementos

$$L_{rr'} = L_{r'r}$$
$$L_{rr} = 0$$
$$L_{rr'} \geq 0$$

  - Normalmente se escoge $L_{rr'} = 1$ para todas las $r \neq r'$

Similitud de Jaccard

- Mide el grado de similitud entre dos conjuntos, sea cual sea el tipo de elementos.



$$\frac{|A \cap B|}{|A \cup B|}$$

Similitud coseno

- La similitud existente entre dos vectores en un espacio que posee un producto punto con el que se evalúa el valor del coseno del ángulo comprendido entre ellos
- Valores:
  - 1 ambos vectores apuntan en el mismo sentido
  - 0 ambos vectores son ortogonales
  - -1 ambos vectores apuntan en sentido contrario

$$A \cdot B = \|A\|\|B\| \cos\theta$$

$$similitud = \cos\theta = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

Disimilitud de observaciones

- Definimos el procedimiento para combinar p disimilitudes de atributos individuales en un solo valor que indica la disimilitud global $D\left(x^{(i)}, x^{(j)}\right)$ entre dos observaciones $\left(x^{(i)}, x^{(j)}\right)$.

$$D\left(x^{(i)}, x^{(j)}\right) = \sum_{r=1}^{p} w_r \cdot d\left(x_r^{(i)} - x_r^{(j)}\right)^2$$

$$\sum_{r=1}^{p} w_r = 1$$

Agrupamiento

- Una forma de análisis exploratorio de datos (EDA) donde las observaciones se dividen en grupos significativos que comparten características (rasgos) comunes.

Agrupamiento estricto

- No supervisado
  - Entradas

$$S_n = \{(x^{(i)}) | i = 1 \dots n\}$$

$$k$$

  - Salidas

$$C_1, \dots, C_k$$

$$\bigcup C_k = \{i = 1 \dots n\}$$

$$C_i \cap C_j = \emptyset \, \{i \neq j\}$$

    - Representantes

$$z^{(1)}, \dots, z^{(k)}$$

Costo de las particiones

$$costo(C_1, \dots, C_k) = \sum_{j=1}^{k} costo(C_j)$$

$$costo(C, z) = \sum_{i \in C} costo(x^{(i)}, z)$$

Costo de las particiones

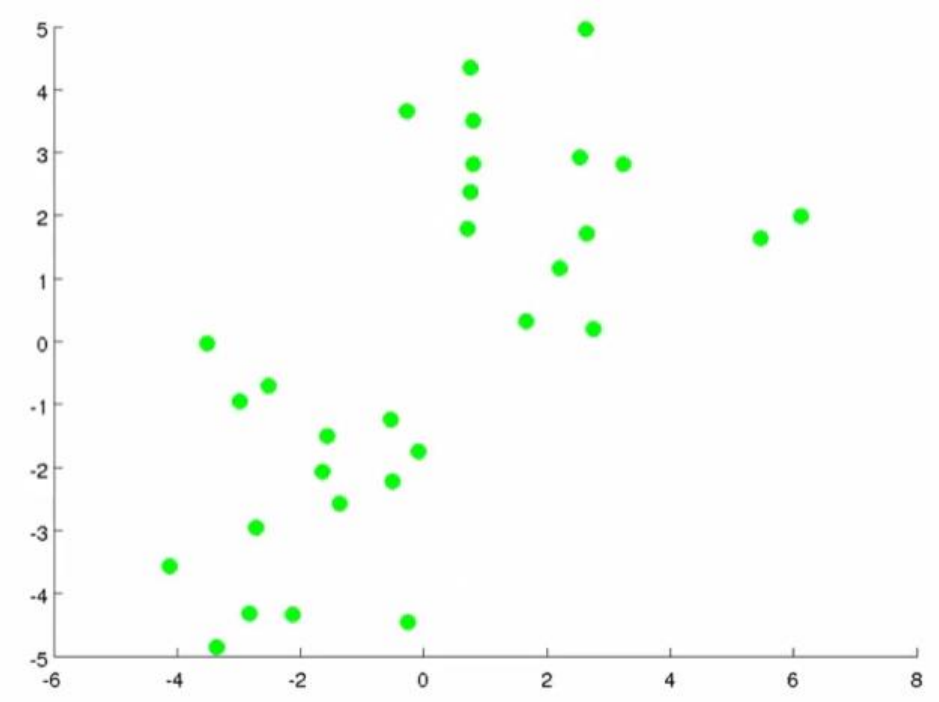
- Distancia euclidiana

$$costo\left(C_1, \dots, C_k, z^{(1)}, \dots, z^{(k)}\right) = \sum_{j=1}^{k} \sum_{i \in C_j} \left\| x^{(i)} - z^{(j)} \right\|^2$$

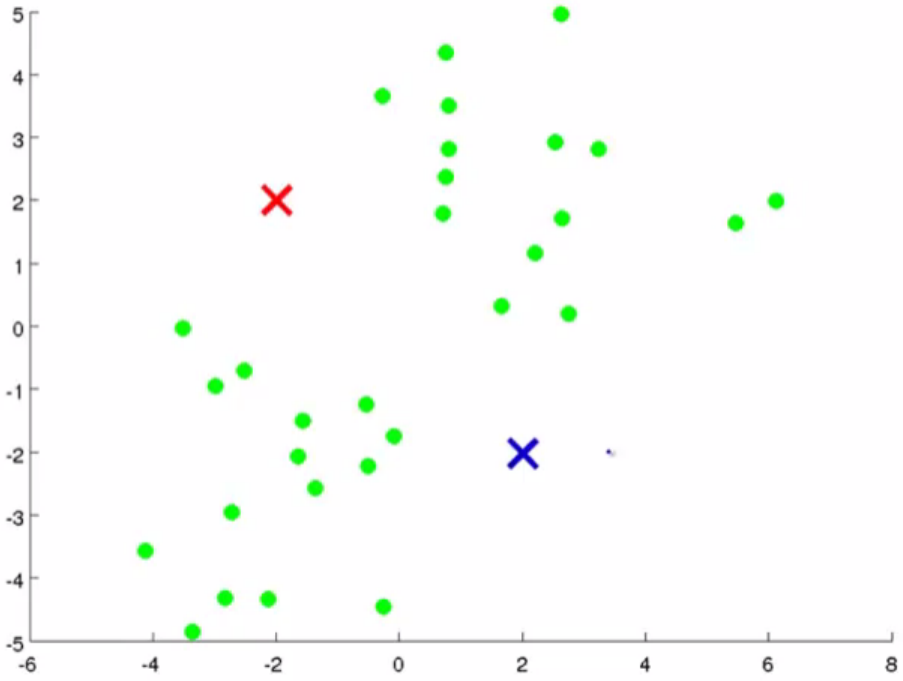
K-medias

- Es un método de agrupamiento, que tiene como objetivo la partición de un conjunto de n observaciones en k grupos en el que cada observación pertenece al grupo más cercano a la media.

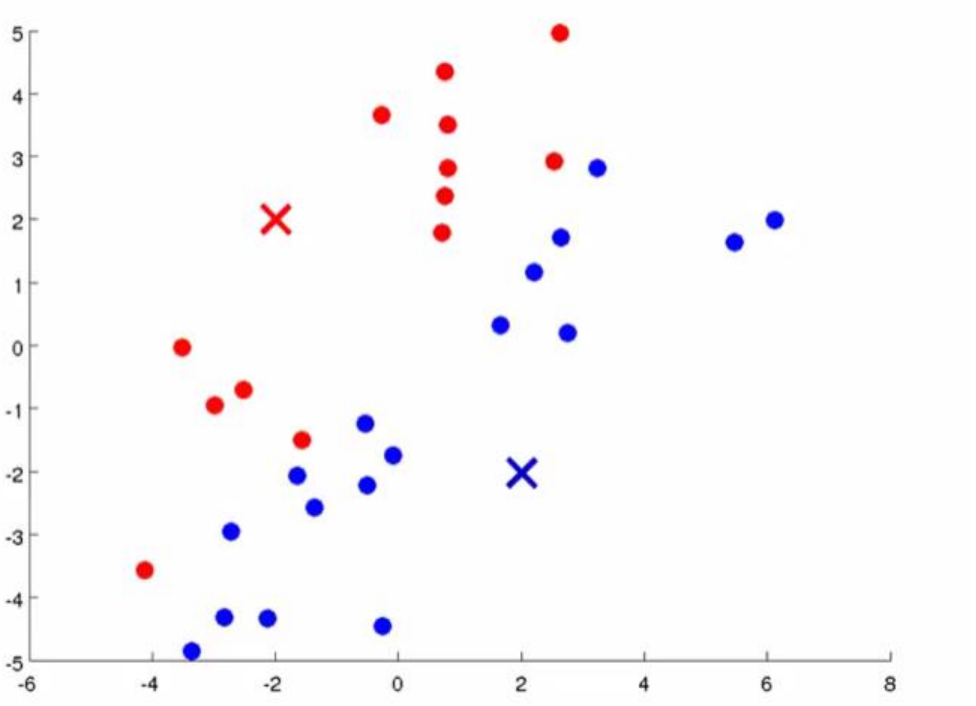
K-medias



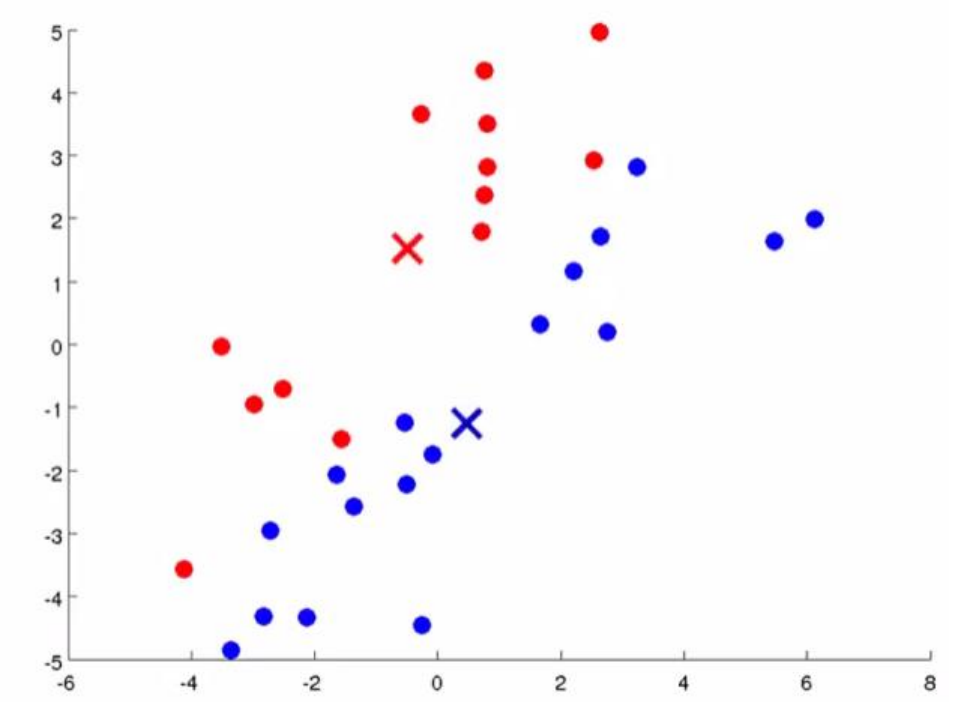
K-medias



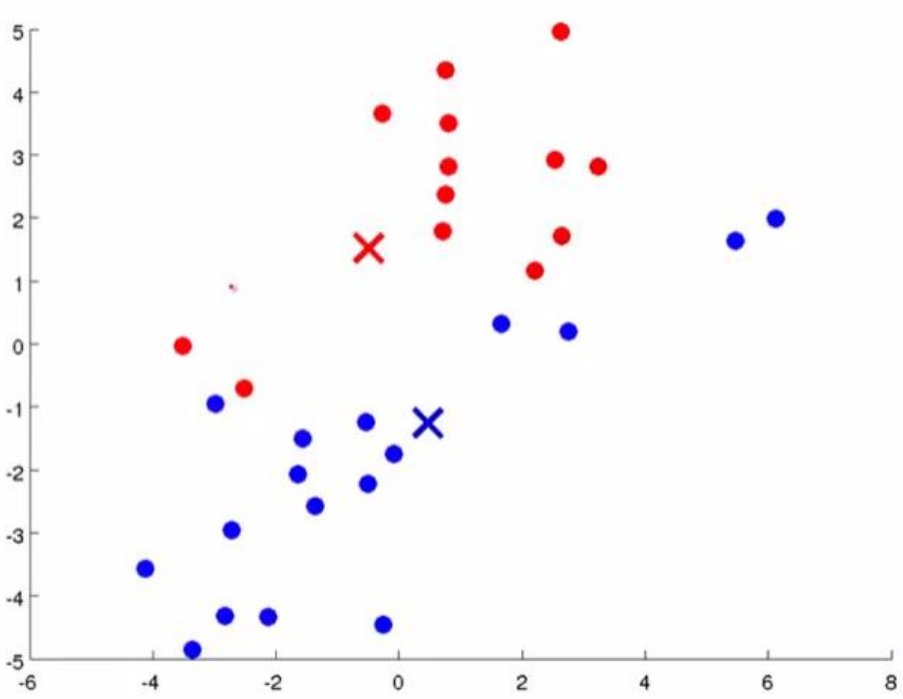
K-medias



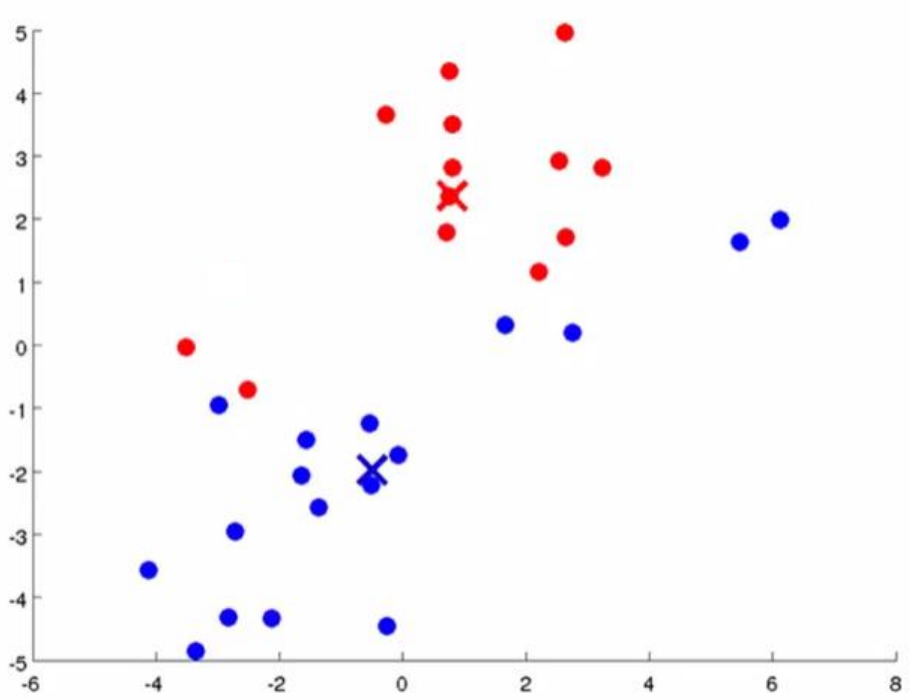
K-medias



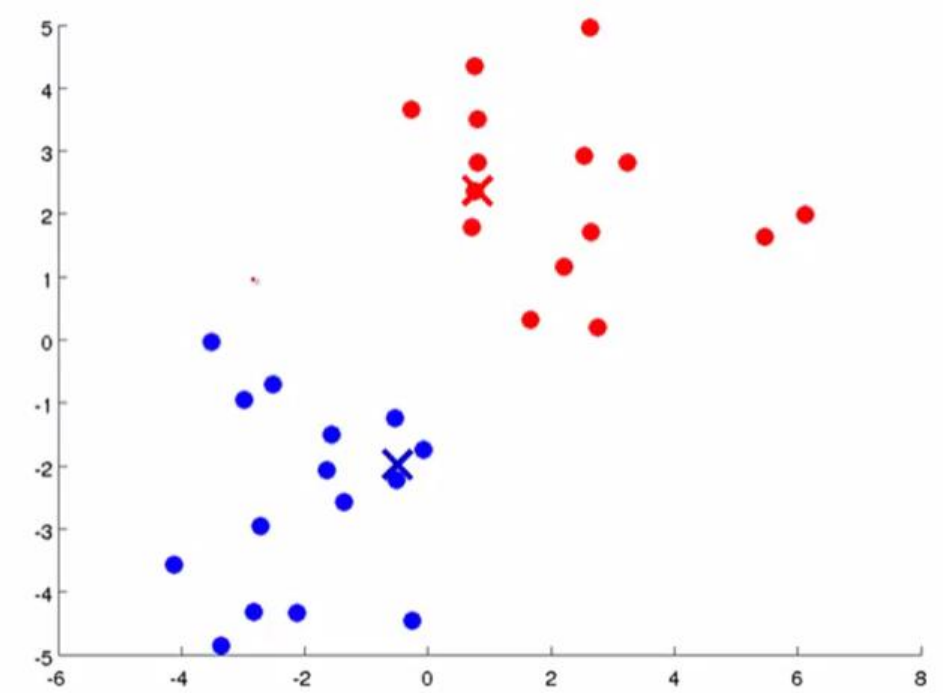
K-medias



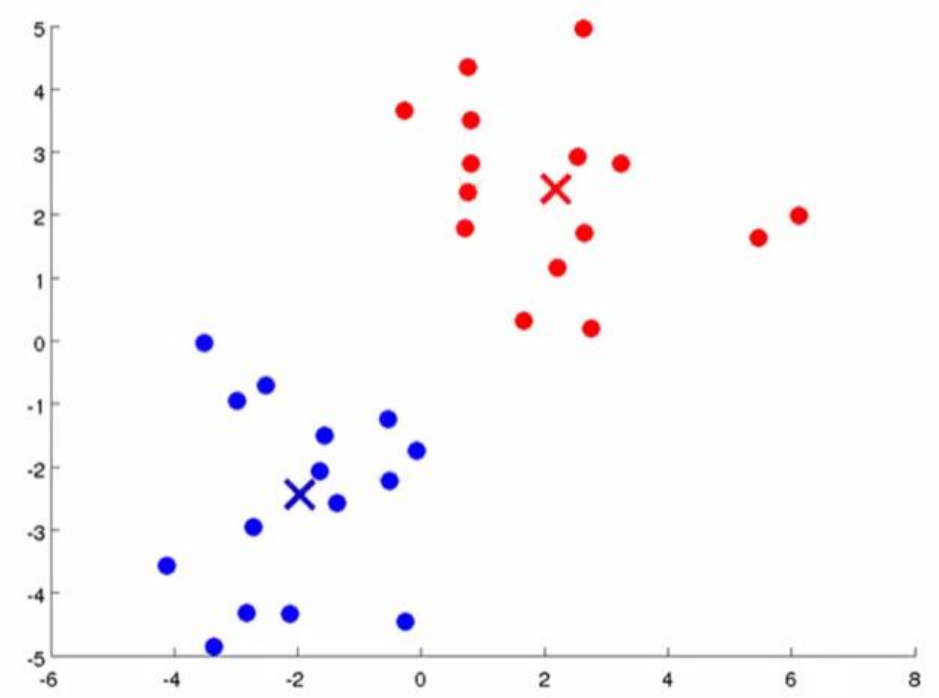
K-medias



K-medias



K-medias



K-medias

- Entradas
  - K (número de grupos)
  - Datos de entrenamiento $\{x^{(1)}, x^{(2),\ldots}, x^{(m)}\}$
  - Cada vector $x^{(i)}$ tiene n atributos

K-medias

- Algoritmo

  Se inicializan aleatoriamente K centroides de grupo $\mu_1, \mu_2, \ldots, \mu_k$ (de dimensión n)

  Repetir {

  for i=1 to m

  c(i):= índice (1 a K) del centroide de grupo más cercano a $x^{(i)}$

  for k=1 to K

  μ_k := promedio de los puntos asignados al grupo k

  }

K-medias

- Objetivo de optimización

$$J\left(c^{(1)},\ldots,c^{(m)},\mu_1,\ldots,\mu_k\right)=\frac{1}{m}\sum_{i=1}^{m}\left\|x^{(i)}-\mu_{c^{(1)}}\right\|^2$$

$$\min_{\substack{c^{(1)},\ldots,c^{(m)},\\ \mu_1,\ldots,\mu_k}} J\left(c^{(1)},\ldots,c^{(m)},\mu_1,\ldots,\mu_k\right)$$

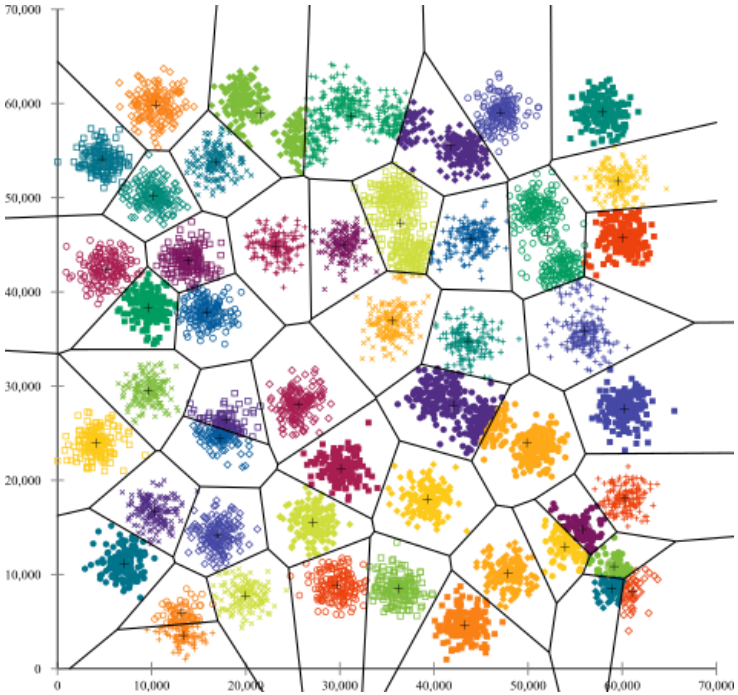
$O(n^{dk+1}\log n)$ Es un problema NP−Duro

Es de orden Exponencial

K-medias

- Inicialización aleatoria:
  - Para j = 1 hasta k repetir
    - Se obtiene un entero aleatorio i del rango [1 a m]
    - Se asignan el centroide del grupo j de la siguiente forma: $\mu_j = x^{(i)}$

K-medias



Óptimos locales

K-medias

- Para superar óptimos locales

  for i=1 to tope (50 – 1,000) {

  Inicializa aleatoriamente K-medias

  Ejecuta K-medias para obtener $c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K$

  Se obtiene lo función de costo (distorsión)

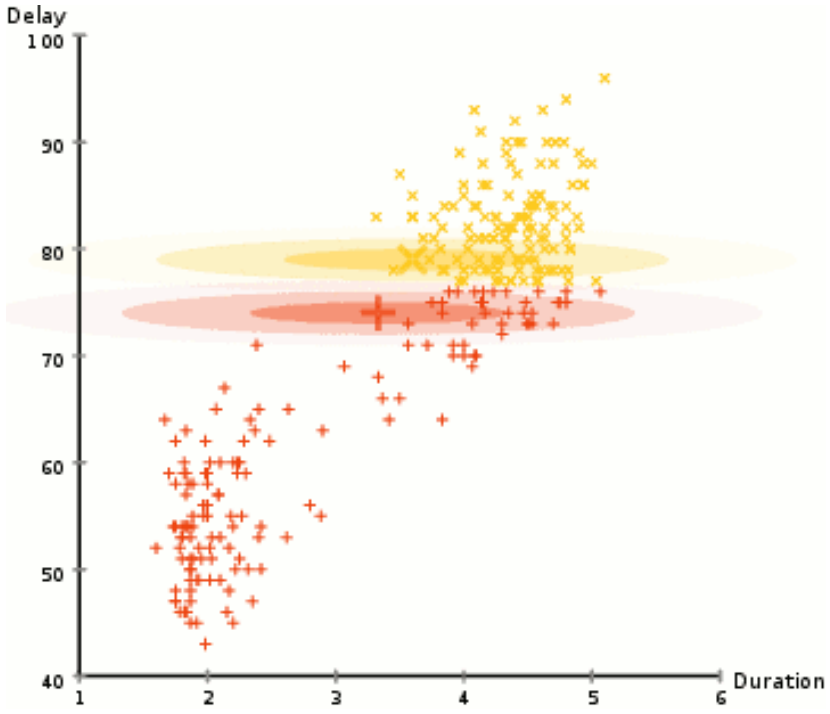
  $J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K)$

  }

- Se usa la corrida que dio la función de costo $J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K)$ mínima

MEZCLA DE MODELOS

Expectation maximization



Expectation maximization

- Algoritmos con el que encontramos los parámetros óptimos (media, varianza, covarianza, etc.) de modelos (distribuciones de probabilidad) de los que pueden provenir los datos a agrupar
  - Distribuciones gaussianas para datos continuos
  - Distribuciones multinomiales para datos discretos
- Similar a k-medias, pero aquí cada dato no se marca como totalmente perteneciente a un grupo, sino que se le asigna una probabilidad de que así sea

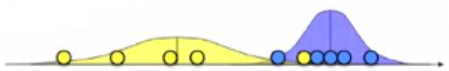
Expectation maximization

- Asumamos que tenemos los siguientes datos etiquetados

- Para calcular las distribuciones gaussianas que los generaron, me bastaría calcular la media y la varianza

$$\mu_b = \frac{x_1 + \ldots + x_{n_b}}{n_b}$$

$$\sigma_b^2 = \frac{(x_1 - \mu_b)^2 + \ldots + (x_{n_b} - \mu_b)^2}{n_b}$$



Expectation maximization

- El problema es cuando no sabemos la etiqueta de cada uno de los datos, ni cuantos modelos los generaron

- Si supiéramos los parámetros de las distribuciones, podríamos fácilmente determinar que modelo generó el dato

$$P(b|x_i) = \frac{P(x_i|b)P(x_i)}{P(x_i|a)P(x_i) + P(x_i|b)P(x_i)}$$

$$P(x_i|b) = \frac{1}{\sqrt{2\pi\sigma_b^2}} e^{\left(-\frac{x_i-\mu_b}{2\sigma_b^2}\right)}$$

Expectation maximization

Se inicializan p(c), μ_c, σ_c^2 aleatoriamente

- Repetir hasta que converja {

Para una dimensión

Paso E (Adivinar la probabilidad de que un dato pertenezca a una gaussiana)

$$P(b|x_i) = \frac{P(x_i|b)P(b)}{P(x_i|a)P(a) + P(x_i|b)P(b)}$$

$$b_i = P(b|x_i) \qquad P(x_i|b) = \frac{1}{\sqrt{2\pi\sigma_b^2}} e^{\left(-\frac{x_i-\mu_b}{2\sigma_b^2}\right)}$$

$$a_i = P(a|x_i) \qquad P(x_i|a) = 1 - P(x_i|b)$$

Paso M (actualizamos la estimación de los parámetros con la nueva información

$$\mu_b = \frac{b_1 x_1 + \ldots + b_{n_b} x_{n_b}}{n_b}$$

$$\sigma_b^2 = \frac{b_1(x_1 - \mu_b)^2 + \ldots + b_{n_b}(x_{n_b} - \mu_b)^2}{b_1 + \ldots + b_{n_b}} \qquad P(b) = \frac{b_1 + \ldots + b_{n_b}}{n}$$

Lo mismo para a

}

Expectation maximization

Se inicializan p(c), μ_c, σ_c^2 aleatoriamente

- Repetir hasta que converja {

Para más de una dimensión

Paso E (Adivinar la probabilidad de que un dato pertenezca a una gaussiana)

$$P(c|x_i) = \frac{P(x_i|c)P(c)}{\sum_{c'=1}^{k} P(x_i|c')P(c')}$$

$$P(x_i|c) = \frac{1}{\sqrt{2\pi|\Sigma_c|}} e^{\left(-\frac{1}{2}(x_i-\mu_c)^T\Sigma^{-1}(x_i-\mu_c)\right)}$$

Paso M (actualizamos la estimación de los parámetros con la nueva información

$$P(c) = \frac{1}{n}\sum_{i=1}^{n} P(c|x_i)$$

$$\mu_{c,j} = \sum_{i=1}^{n} \frac{P(c|x_i)}{nP(c)} x_{i,j}$$

$$(\Sigma_c)_{j.k} = \sum_{i=1}^{n} \frac{P(c|x_i)}{nP(c)} (x_{i,j} - \mu_{c,j})(x_{i,k} - \mu_{c.k})$$

}

Expectation maximization

1. E – step

$$p(j|i) = \frac{p_j N\left(x^{(i)}; \mu^{(j)}, \sigma_j^2 I\right)}{p(x|\theta)}$$

2. M – Step

$$\hat{n}_j = \sum_{i=1}^{n} p(j|i) \qquad \hat{p}_j = \frac{\hat{n}_j}{n}$$

$$\hat{\mu}^{(j)} = \frac{1}{\hat{n}_j} \sum_{i=1}^{n} p(j|i) x^{(i)} \qquad \sigma_j^2 = \frac{1}{\hat{n}_j d} \sum_{i=1}^{n} p(j|i) \left\| x^{(i)} - \mu^{(j)} \right\|^2$$

AGRUPAMIENTO JERÁRQUICO

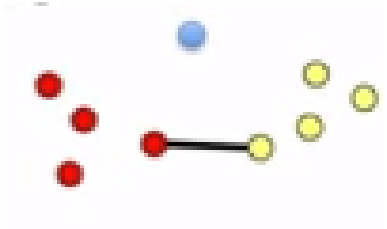
Agrupamiento jerárquico

- Es un método que busca construir una jerarquía de grupos. Estrategias para agrupamiento jerárquico generalmente caen en dos tipos:
  - **Aglomerativas**: Este es un acercamiento ascendente, cada observación comienza en su propio grupo, y los pares de grupos son mezclados mientras uno sube en la jerarquía.
  - **Divisivas**: Este es un acercamiento descendente, todas las observaciones comienzan en un grupo, y se realizan divisiones mientras uno baja en la jerarquía.

Distancia entre grupos

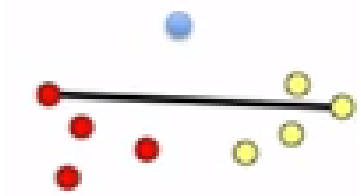
- Liga única: $D(c_1, c_2) =$

$$\min_{x_1 \in c_1, x_2 \in c_2} D(x_1, x_2)$$



- Liga completa $D(c_1, c_2) =$

$$\max_{x_1 \in c_1, x_2 \in c_2} D(x_1, x_2)$$

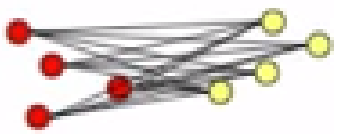


Distancia entre grupos

- Liga promedio: $D(c_1, c_2) =$

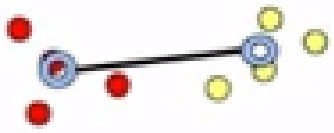
$$\frac{1}{|c_1|}\frac{1}{|c2|}\sum_{x_1 \in c_1}\sum_{x_2 \in c_2} D(x_1, x_2)$$

- Centroide: $D(c_1, c_2) =$



$$D\left(\frac{1}{|c_1|}\sum_{x \in c_1} x\,, \frac{1}{|c_2|}\sum_{x \in c_2} x\right)$$

Sólo datos numéricos

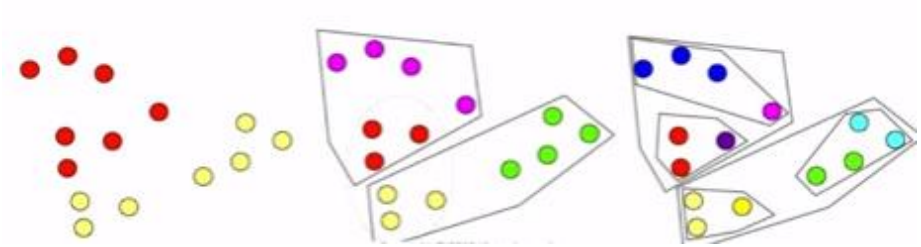


Agrupamiento jerárquico

- En vez de escoger el número de grupos en los que quiero dividir mis datos, construyo una jerarquía
- Cada nivel que vea me da distinto nivel de granularidad

K-medias jerárquico

- Comienzo con un nodo con todos los datos de entrenamiento y le aplico K-medias con una K fija
- Para cada grupo resultante aplico el algoritmo de K-medias recursivamente
  - Es un algoritmo rápido, pero codicioso

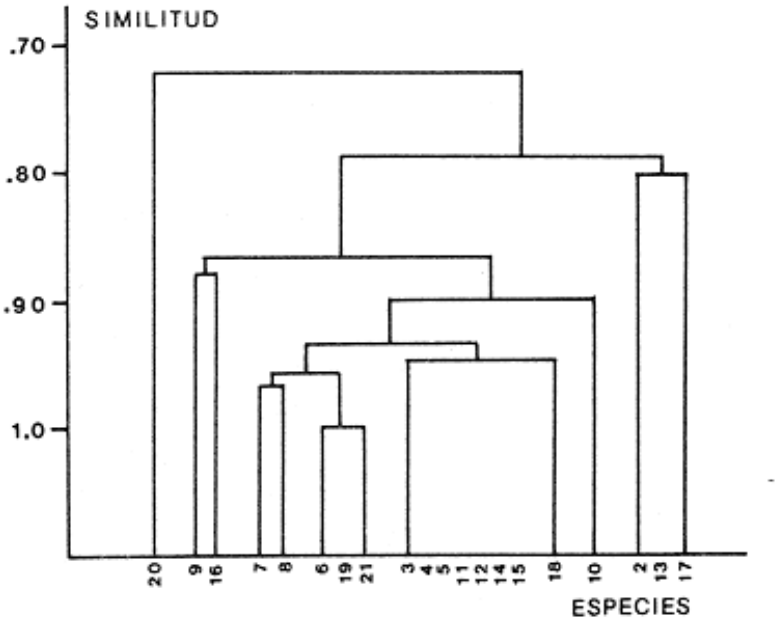


Agrupamiento aglomerativo

- Idea: asegurarse que puntos cercanos terminen en el mismo grupo
- Empezar con una colección C de n grupos con un solo elemento
- Repetir hasta que quede un solo grupo
  - Encontrar un par de grupos cercanos
  - Unir los grupos c_i y c_j en uno nuevo c_{i+j}
  - Eliminar los grupos c_i y c_j de C y agregar c_{i+j}
- El algoritmo es lento
- Produce un dendograma

$$\min_{i,j} D(c_i, c_j)$$

Dendograma


SIMILITUD
.70
.80
.90
1.0
20 9 16 7 8 6 19 21 3 4 5 11 12 14 15 18 10 2 13 17
ESPECIES

AGRUPAMIENTO POR DENSIDAD

Agrupamiento por densidad

- DBSCAN es un algoritmo de agrupamiento basado en densidad (density-based clustering) porque encuentra un número de grupos (clusters) comenzando por una estimación de la distribución de densidad de los nodos correspondientes.
- DBSCAN es uno de los algoritmos de agrupamiento más usados y citados en la literatura científica.

Agrupamiento por densidad

- Grupo se define como el conjunto máximo de puntos conectados por densidad
- Tiene 2 parámetros:
  - ε - máximo radio del vecindario
  - minPts – número mínimo requerido de puntos en el vecindario ε de un punto
- El vecindario ε de un punto q -
  - $N_{\varepsilon}(q)$: { p pertenece a D | dist(p,q)≤ ε }

Agrupamiento por densidad

- p es directamente densamente alcanzable desde q si
  - p pertenece a $N_\varepsilon(q)$
  - $|N_\varepsilon(q)| \geq minPts$
- p es densamente alcanzable desde q si
  - existe una secuencia de puntos $p_1,\ldots p_n$ donde $p_1=p$ y $p_n=q$ tal que cada punto p_i+1 es directamente densamente alcanzable desde p_i

Agrupamiento por densidad

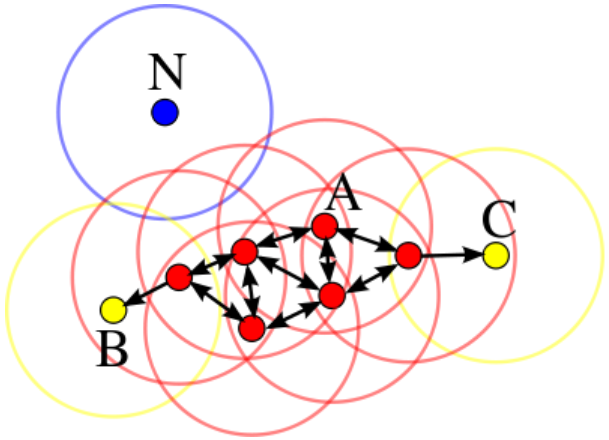
- p es densamente conectado a q (pertenecen al mismo grupo) si
  - Existe un punto o del que p y q puedan ser densamente alcanzables

Agrupamiento por densidad

- Algoritmo
  - Selecciona aleatoriamente un punto p
  - Se obtienen todos los puntos que sean densamente alcanzables desde p usando los parámetros ε y minPts
  - Si p es un punto nuclear (tiene minPts o más en el vecindario ε) un grupo es formado
  - Si p es un punto borde (en el grupo, pero no tiene minPts en el vecindario) y no hay puntos que puedan ser densamente alcanzables por p se marca como ruido
  - Se visita el siguiente punto
  - Repetir hasta que todos los puntos sean visitados

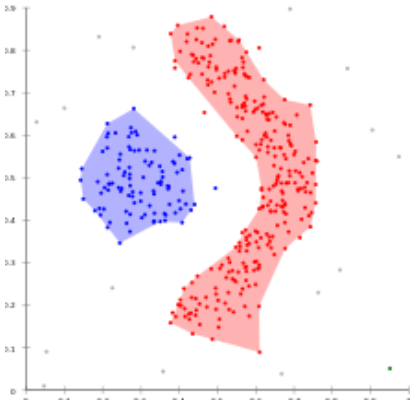
Agrupamiento por densidad

- Los puntos marcados como A son puntos núcleo. Los puntos B y C son *densamente alcanzables* desde A y *densamente conectados* con A, y pertenecen al mismo clúster. El punto N es un punto ruidoso que no es núcleo ni densamente alcanzable. (MinPts=3 o MinPts=4)



Agrupamiento por densidad

- DBSCAN puede encontrar grupos que no son linealmente separables. Este conjunto de datos no puede ser correctamente agrupado con K-medias o con Mezclas Gaussianas EM.



Agrupamiento por densidad

- DBSCAN es muy sensible a cambios en los parámetros

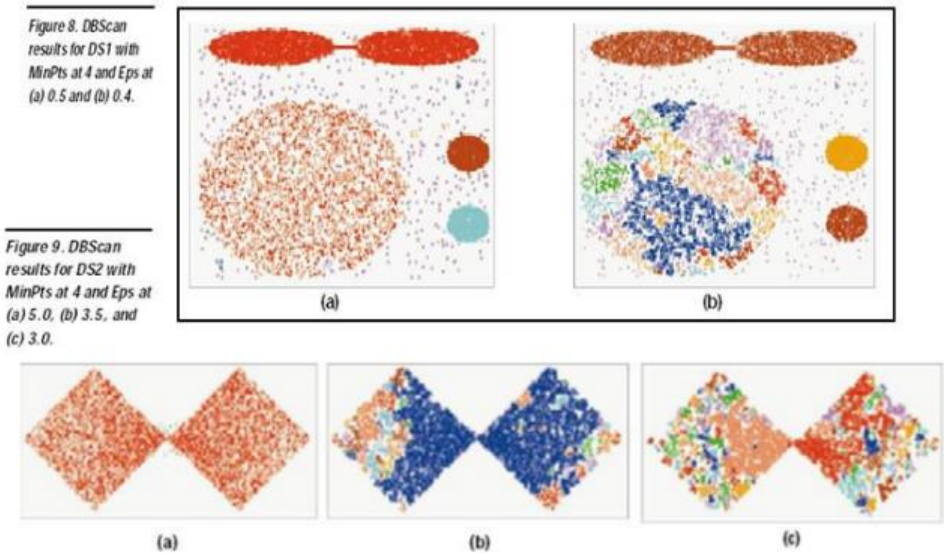


Figure 8. DBScan results for DS1 with MinPts at 4 and Eps at (a) 0.5 and (b) 0.4.

Figure 9. DBScan results for DS2 with MinPts at 4 and Eps at (a) 5.0, (b) 3.5, and (c) 3.0.

Agrupamiento por densidad

- OPTICS puede verse como una generalización de DBSCAN para múltiples rangos, reemplazando el parámetro e por el radio máximo de búsqueda.

REDUCCIÓN DE LA DIMENSIONALIDAD

Diferentes enfoques

- **Análisis de componentes principales**: proyección que dispersa los datos tanto como sea posible
- **Multidimensional scaling** : proyección que conserva las distancias originales tanto como sea posible
- **Stochastic neighbor embedding**: Encaje no lineal que intenta mantener cerca los puntos cercanos

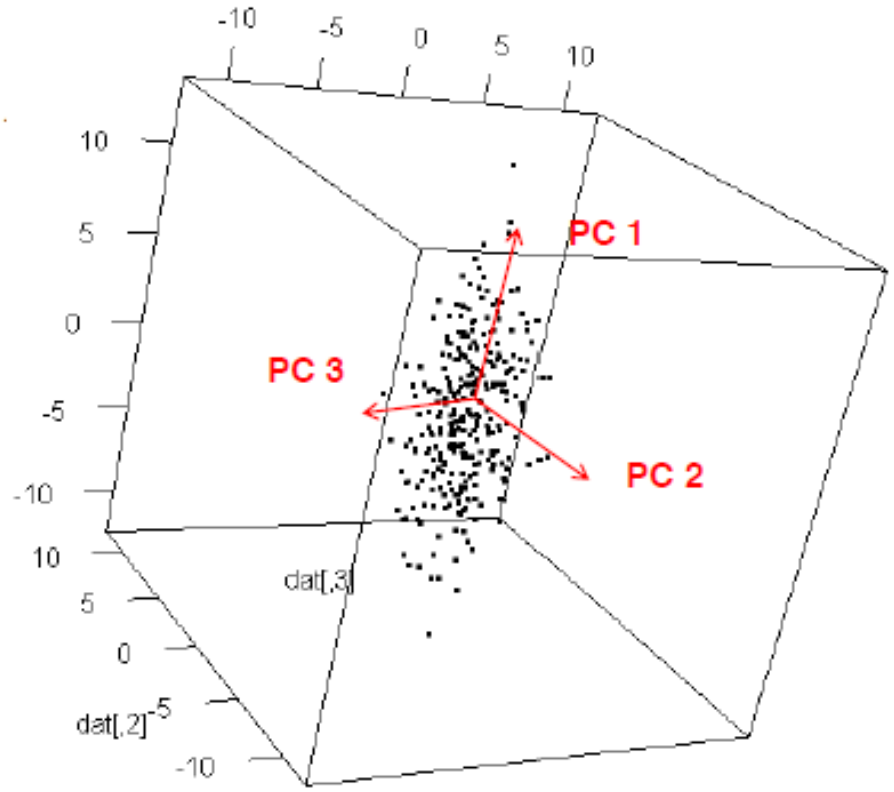
COMPONENTES PRINCIPALES

Análisis de componentes principales (PCA)

- **Objetivo**: reducción de dimensiones a unas cuantas
- **Intuición**: encontrar una proyección de baja dimensión con la mayor dispersión
- Uno de los métodos más utilizados para encontrar patrones en los datos
- Usado frecuentemente cuando cada observación
  - contiene muchas características y no todas ellas son significativas.
  - Existe mucha covarianza entre las características

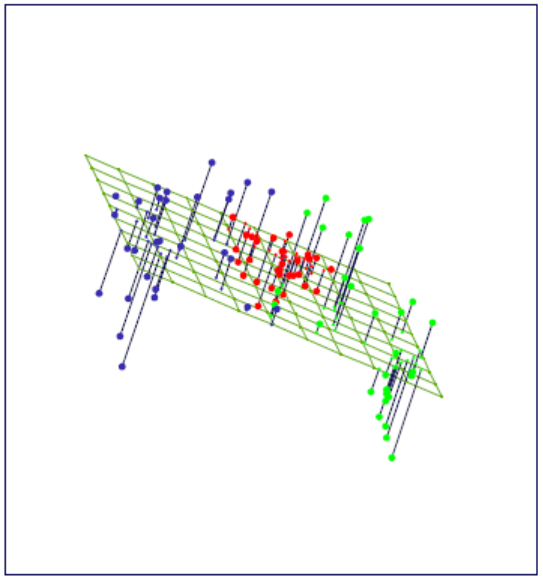
Definición 1: maximizar la variación de la proyección

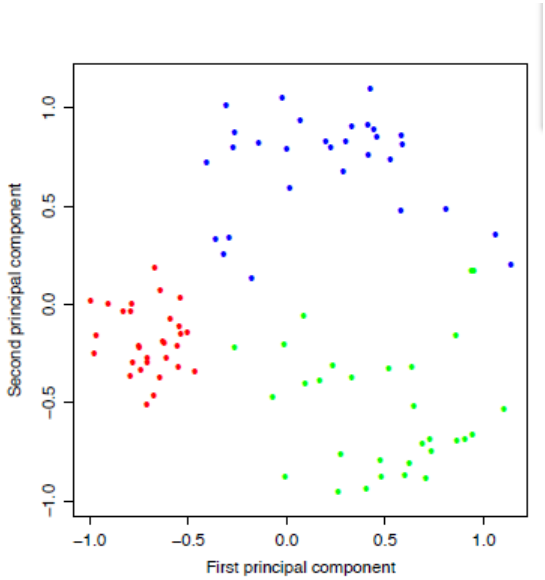
- Comience con datos centrados $X \in \mathbb{R}^{n \times p}$
  - PC 1 es la dirección de mayor varianza
  - PC 2 es perpendicular a PC 1 de nuevo con la varianza más grande
  - PC 3 es perpendicular a PC 1 y PC 2 de nuevo con la varianza más grande
  - etc.



Definición 2: Minimizar los residuos de proyección

- PC 1: línea recta con la menor distancia ortogonal a todos los puntos
- PC 1 y PC 2: plano con la menor distancia ortogonal a todos los puntos
- etc.





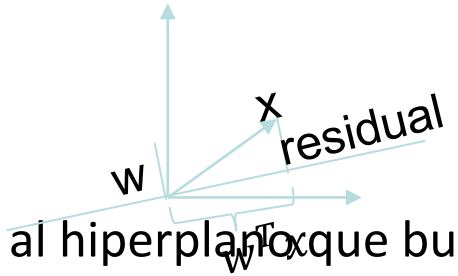
Definición 3: Descomposición espectral

- Matriz de covarianza (o matriz de correlación) $R = \frac{1}{n} X^T X$ es simétrica y semidefinida positiva
- **Teorema de descomposición espectral**: toda matriz simétrica real R se puede descomponer como

$$R = \frac{1}{n} V \Lambda V^T$$

- donde Λ es diagonal y V es ortogonal
- Las columnas de V (= vectores propios de R) son las componentes principales
- Las entradas diagonales de Λ (= valores propios de R) son la varianza a lo largo de las componentes principales

3 definiciones



- w es un vector unitario ortogonal al hiperplano que buscamos
- La longitud de la proyección de x en w es $w^T x$
- Los residuales al cuadrado

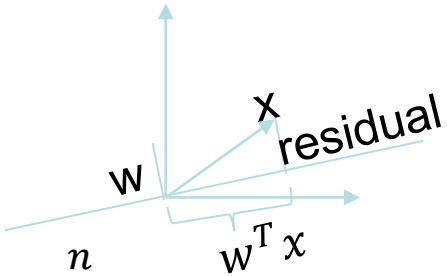
$$\|x - (w^T x)w\|_2^2 = \|x\|_2^2 - 2(w^T x)^2 + (w^T x)^2 w^T w$$
$$= \|x\|_2^2 - (w^T x)^2$$

- Minimizar residuales

$$\min_{w \epsilon \mathbb{R}^p . \|w\|=1} \sum_{i=1}^{n} \|x_i\|_2^2 - (w^T x_i)^2$$

3 definiciones

- Maximizar varianza



$$\max_{w \epsilon \mathbb{R}^p . \|w\|=1} \sum_{i=1}^{n} (w^T x_i)^2$$

$$\max_{w \epsilon \mathbb{R}^p . \|w\|=1} w^T \frac{1}{n} \sum_{i=1}^{n} x_i {x_i}^T w$$

ESCALADO MULTIDIMENSIONAL

Distancia y disimilitud

- $D \in \mathbb{R}^{n \times n}$ es una matriz de distancia si:

$$D_{ii} = 0, D_{ij} \geq 0, D_{ij} = D_{ji}, D_{ij} = D_{ik} + D_{kj} \; para \; toda \; i, j, k$$

  - Por ejemplo: distancia euclidiana, distancia de Manhattan, distancia máxima,…
- $D \in \mathbb{R}^{n \times n}$ es una matriz de disimilitud si:

$$D_{ii} = 0, D_{ij} \geq 0, D_{ij} = D_{ji} \; para \; toda \; i, j$$

  - Más flexible que las distancias, funciona p. ejemplo para rankings

Multidimensional scaling (MDS)

- Dado una matriz $D \in \mathbb{R}^{n \times n}$, determinar puntos $y_1, \ldots, y_n \in \mathbb{R}^q$ tales que minimicemos:

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \left(D_{ij} - \|y_i - y_j\|_2 \right)^2$$

- Asumiendo D como una matriz con distancias Euclidianas

MDS clásico

- Primero transformamos la matriz de distancia D, con $D_{ij} = \left\| x_i - x_j \right\|_2$ en una matriz positiva semidefinida XX^T:

$$XX^T = -\frac{1}{2}\left(I - \frac{1}{n}ee^T\right)D^2\left(I - \frac{1}{n}ee^T\right)$$

- Donde e es un vector de unos

$$D_{ij}^2 = \|x_i\|_2 + \left\|x_j\right\|_2 - 2x_i x_j^T$$

$$= (XX^T)_{ii} - (XX^T)_{jj} - 2(XX^T)_{ij}$$

MDS clásico

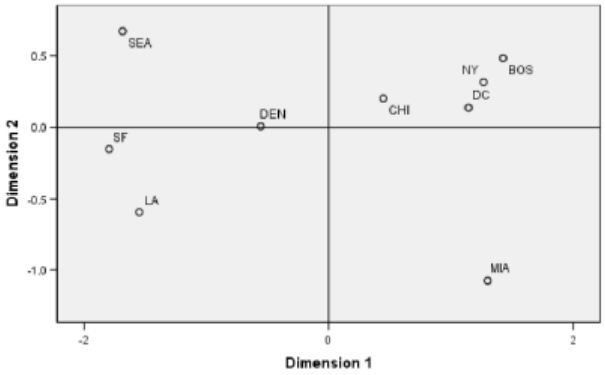
- $\min_{Y} \sum_{i=1}^{n} \sum_{j=1}^{n} \left(D_{ij}^2 - \left\| y_i - y_j \right\|_2^2 \right)^2$ es equivalente a:

$$\min_{Y} trace(XX^T - YY^T)^2$$

- Descomposición de valores propios: $XX^T = V\Lambda V^T$, donde las columnas de V son vectores propios de XX^T, Λ es una matriz diagonal que contiene los valores propios de XX^T
- La mejor aproximación de rango q de XX^T se obtiene eligiendo los q más grandes valores propios y vectores propios correspondientes, es decir, $YY^T = V_1 \Lambda V_1^T$
- MDS clásico es PCA en $\mathrm{B} = XX^T$; PCA clásico opera en $X^T X$

Ejemplo MDS

	BOS	CHI	DC	DEN	LA	MIA	NY	SEA	SF
BOS	0	963	429	1,949	2,979	1,504	206	2,976	3,095
CHI	963	0	671	996	2,054	1,329	802	2,013	2,142
DC	429	671	0	1,616	2,631	1,075	233	2,684	2,799
DEN	1,949	996	1,616	0	1,059	2,037	1,771	1,307	1,235
LA	2,979	2,054	2,631	1,059	0	2,687	2,786	1,131	379
MIA	1,504	1,329	1,075	2,037	2,687	0	1,308	3,273	3,053
NY	206	802	233	1,771	2,786	1,308	0	2,815	2,934
SEA	2,976	2,013	2,684	1,307	1,131	3,273	2,815	0	808
SF	3,095	2,142	2,799	1,235	379	3,053	2,934	808	0



T-SNE

Stochastic neighbor embedding (SNE)

- Enfoque probabilístico para colocar objetos de un espacio de alta dimensión en un espacio de baja dimensión para preservar la identidad de los vecinos
- Centrar una gaussiana en cada objeto en un espacio de alta dimensión
- Encontrar un *embedding* de modo que la distribución resultante de alta dimensión se aproxime bien mediante la distribución resultante de baja dimensión
- Determinar la distribución de baja dimensión minimizando la divergencia de Kullback-Leibler

Stochastic neighbor embedding (SNE)

- dada la matriz de disimilitud D, para cada objeto calculo la probabilidad de elegir j como vecino:

$$p_{ij} = \frac{e^{-D_{ij}^2}}{\sum_{k \neq l} e^{-D_{kl}^2}}$$

- en un espacio de baja dimensión, para cada punto y_i calcule la probabilidad de elegir y_j como vecino:

$$q_{ij} = \frac{e^{-\|y_i - y_j\|_2^2}}{\sum_{k \neq l} e^{-\|y_k - y_l\|_2^2}}$$

- Minimizar la divergencia KL

$$KL(P\|Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

- mantiene los objetos cercanos cerca y los objetos separados relativamente lejos

Créditos

- Parte de este material está basado en cursos de estadística, aprendizaje de máquina y aprendizaje estadístico del MIT, Stanford y Caltech