Introducción a IA

Aprendizaje No Supervisado

Definición

Clustering

Clustering: es la organización de objetos en grupos o clusters. Más precisamente, es el particionamiento de un conjunto de ejemplos en subconjuntos (clusters) de manera que los datos en un mismo cluster tengan características en común.

Un buen agrupamiento es aquel que produce clusters de calidad, con:

→ alta similitud intra-cluster
→ baja similitud inter cluster

Definición

Clustering

Cómo se mide la similitud/distancia:

- La similitud es subjetiva en muchos casos
- La medida depende de los datos y sus características
- Medidas comunes son la distancia Euclídea, Manhattan, correlación y coseno
- Características heterogéneas (por ejemplo, ingresos, edad, hábitos de consumo, nivel de educación, etc.) pueden requerir una definición ad-hoc de similitud

Definición

Clustering

Euclídea

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{n} \left(x_i^k - x_j^k\right)^2}$$

Manhattan

$$d(x_i, x_j) = \sum_{k=1}^{n} \left| x_i^k - x_j^k \right|$$

Coseno

$$cos(x_i, x_j) = \frac{\sum_{k=1}^{n} x_i^k \cdot x_j^k}{\sqrt{\sum_{k=1}^{n} \left(x_i^k\right)^2} \sqrt{\sum_{k=1}^{n} \left(x_j^k\right)^2}}$$

Definición

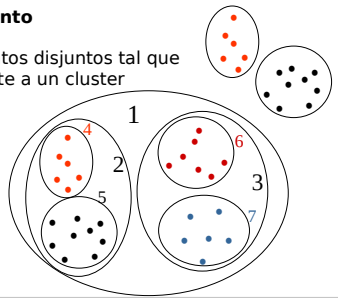
Clustering

Clustering basado en **particionamiento**

- Dividen los ejemplos en subconjuntos disjuntos tal que cada objeto pertenece exactamente a un cluster

Clustering **jerárquico**

- Generan un conjunto de clusters anidados en un árbol o jerarquía



Definición

Clustering

Basado en particionamiento:

- El objetivo es alcanzar una única partición de la colección de ejemplos en clusters
- Usualmente se basan en optimizar iterativamente un criterio o función objetivo que refleja el consenso entre los ejemplos y las particiones

→ ***k*-Means**: ejemplo de clustering basado en particionamiento, ampliamente usado en minería de datos

Algoritmos

k-Means

k-Means agrupa *n* ejemplos en *k* particiones en base a sus atributos, donde *k*<*n*

- Cada cluster está representado por su **centroide** o centro de gravedad:

$$\vec{\mu}(c) = \frac{1}{|c|} \sum_{\vec{x} \in c} \vec{x}$$

- Se optimiza la función:

$$\arg\min_{C} = \sum_{i=1}^{k} \sum_{x_j \in c_i} |x_j - \mu_i|^2$$

Algoritmos

k-Means

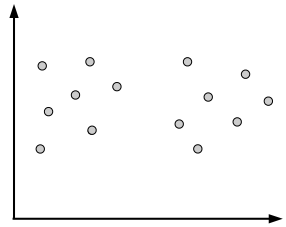
Dado un número de clusters *k*, el algoritmo *k*-Means ejecuta tres pasos después de la inicialización:

Inicialización: seleccionar aleatoriamente *k* ejemplos (semillas) para ser centroides de los clusters

1. Asignar cada ejemplo al centroide con el que tenga mayor similitud
2. Calcular nuevos centroides de los clusters de la participación
3. Si no se satisface el criterio de terminación (no hay cambios en los clusters), volver al paso 1

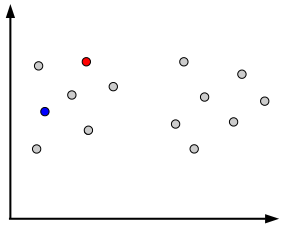
Algoritmos

k-Means



Algoritmos

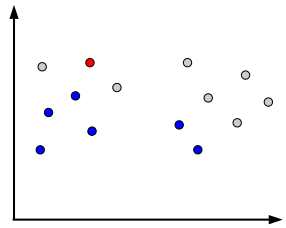
k-Means



Para k=2, elegir aleatoriamente dos semillas entre los ejemplos

Algoritmos

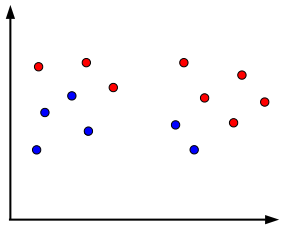
k-Means



Asignar los ejemplos al centroide más cercano

Algoritmos

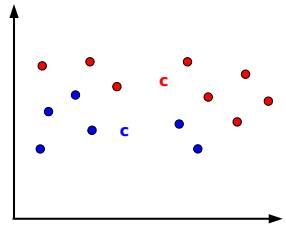
k-Means



Asignar los ejemplos al centroide más cercano

Algoritmos

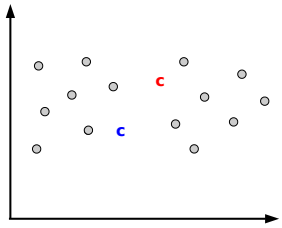
k-Means



Calcular los nuevos centroides para el actual particionamiento

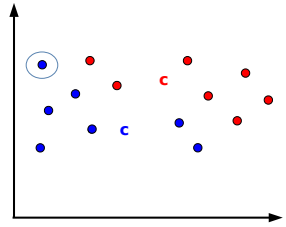
Algoritmos

k-Means



Algoritmos

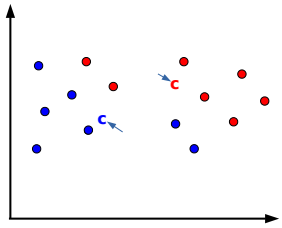
k-Means



Asignar los ejemplos al centroide más cercano

Algoritmos

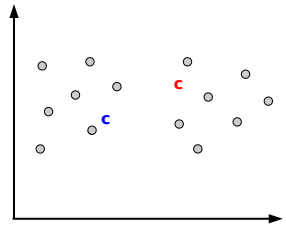
k-Means



Calcular los nuevos centroides para el actual particionamiento

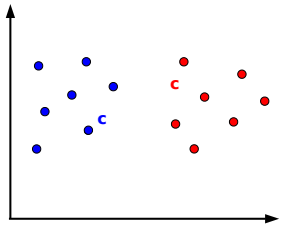
Algoritmos

k-Means



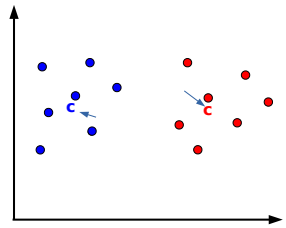
Algoritmos

k-Means



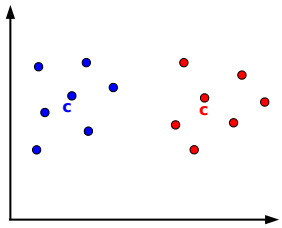
Algoritmos

k-Means



Algoritmos

k-Means



Los ejemplos no cambian de cluster, se satisface el criterio de terminación

Algoritmos

k-Means

Pros

- Simple y eficiente dentro de los algoritmos de particionamiento

Contras

- Necesita establecer *k* de antemano
- Sensible a ruido y outliers, puede caer en mínimos locales (*k*-Medoids)
- Sensitivo a la elección de las semillas iniciales (*k*-Means++)
  → según las semillas puede tener mejores tazas de convergencia
  → la selección de semillas puede basarse en heurísticas o resultados obtenidos con otros métodos
- Es aplicable cuando es posible calcular el centroide (*k*-Modes para atributos categóricos)

Algoritmos

Clustering Jerárquico

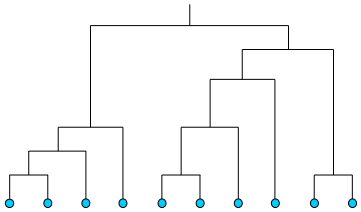
Métodos de clustering jerárquico:

- **Aglomerativo (bottom-up)**: métodos que comienzan con cada ejemplo en un cluster diferente y combinan iterativamente los clusters en clusters de mayor tamaño
- **Divisivo (top-down)**: métodos que comienzan con todos los ejemplos en un mismo cluster y los separan sucesivamente en clusters de menor tamaño

Algoritmos

Jerárquico Aglomerativo

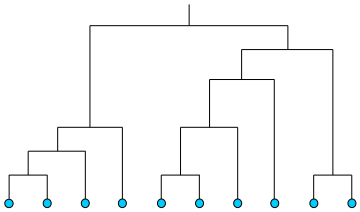
Los algoritmos jerárquicos aglomerativos construyen un árbol binario o **dendograma** a partir de un conjunto de ejemplos



Algoritmos

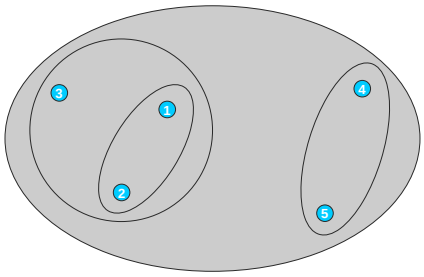
Jerárquico Aglomerativo

Un dendograma muestra como se combinan los clusters
→ La raíz contiene todos los ejemplos y las hojas los ejemplos individuales
→ Cortando en diferentes niveles se obtiene distinto número de clusters



Algoritmos

Jerárquico Aglomerativo



Algoritmos

Jerárquico Aglomerativo

El clustering aglomerativo jerárquico se desarrolla en los siguientes pasos:

1. Asignar cada ejemplo a un cluster diferente (*n* ejemplos, *n* clusters)
2. Encontrar el par de **clusters más similares** y combinarlos en uno único
3. Calcular las similitudes o distancias entre el nuevo cluster y los clusters restantes
4. Hasta que solo quede un cluster de tamaño *n*, volver a 2

Algoritmos

Jerárquico Aglomerativo

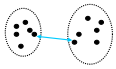
Medidas de similitud:

- Función de similitud entre ejemplos: determina la similitud/distancia entre dos ejemplos individuales
- Función de similitud entre clusters: determina la similitud de dos clusters conteniendo múltiples ejemplos
  - → Single link
  - → Complete link
  - → Group average

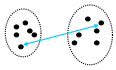
Algoritmos

Jerárquico Aglomerativo

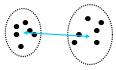
Single link: la similitud de los clusters está dada por los dos ejemplos más similares entre ambos



Complete link: la similitud de los clusters está dada por los dos ejemplos menos similares entre ambos



Group average: la similitud es el promedio las similitudes entre los ejemplos de ambos clusters



Algoritmos

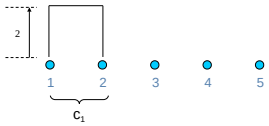
Jerárquico Aglomerativo

	1	2	3	4	5
1	0				
2	2	0			
3	6	3	0		
4	10	9	7	0	
5	9	8	5	4	0

Algoritmos

Jerárquico Aglomerativo

	1	2	3	4	5
1	0				
2	2	0			
3	6	3	0		
4	10	9	7	0	
5	9	8	5	4	0



Algoritmos

Jerárquico Aglomerativo

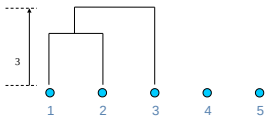
	1,2	3	4	5
1,2	0			
3	3	0		
4	9	7	0	
5	8	5	4	0

$d_{(1,2),3} = \min\{d_{1,3}, d_{2,3}\} = \min\{6,3\} = 3$
$d_{(1,2),4} = \min\{d_{1,4}, d_{2,4}\} = \min\{10,9\} = 9$
$d_{(1,2),5} = \min\{d_{1,5}, d_{2,5}\} = \min\{9,8\} = 8$

Algoritmos

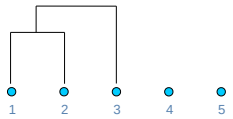
Jerárquico Aglomerativo

	1,2	3	4	5
1,2	0			
3	3	0		
4	9	7	0	
5	8	5	4	0



Algoritmos

Jerárquico Aglomerativo

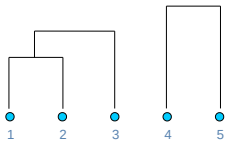


	1,2,3	4	5
1,2,3	0		
4	7	0	
5	5	4	0

$d_{(1,2,3),4} = \min\{d_{(1,2),4}, d_{3,4}\} = \min\{9,7\} = 7$
$d_{(1,2,3),5} = \min\{d_{(1,2),5}, d_{3,5}\} = \min\{8,5\} = 5$

Algoritmos

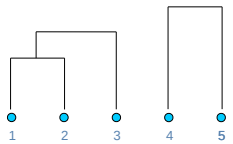
Jerárquico Aglomerativo



	1,2,3	4	5
1,2,3	0		
4	7	0	
5	5	4	0

Algoritmos

Jerárquico Aglomerativo

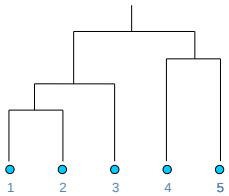


	1,2,3	4,5
1,2,3	0	
4,5	5	0

$d_{(1,2,3),(4,5)} = \min\{d_{(1,2,3),4}, d_{(1,2,3),5}\} = 5$

Algoritmos

Jerárquico Aglomerativo



	1,2,3	4,5
1,2,3	0	
4,5	5	0

Algoritmos

Jerárquico Aglomerativo

Pros

- No se necesita especificar el número de clusters de antemano
- La estructura jerárquica ofrece una forma natural de navegar los datos, más rico para análisis que el particionamiento

Contras

- No escala bien en el número de ejemplos, costoso computacionalmente
- No se recupera de decisiones incorrectas
- La interpretación de los datos es subjetiva

Próxima clase

Algoritmos de Clustering

- *k*-Means
- Jerárquico Aglomerativo

Evaluación del Aprendizaje

- Métricas