Clase N°12. Modelo de Clustering no jerárquico.

Es una técnica de agrupamiento donde la cantidad de particiones es determinada a priori. Entre los métodos más utilizados se encuentran: k-medoids (mediode), k-means (centroide).

K-medoid

Es un método de agrupamiento que genera particiones entre las observaciones minimizando la distancia entre los puntos, tomando como centro de cada agrupamiento puntos al azar que pertenecen al conjunto de datos. Se calcula mediante la función de costos, definida por:

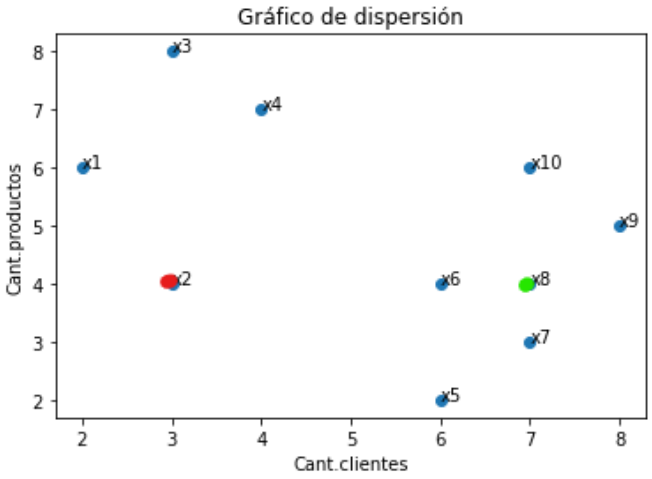
$$costo(x, c) = \sum_{k=1}^{d} |x_i - c_i|$$

Siendo x_i, el punto u observación, c_i el medoid o punto seleccionado al azar perteneciente al conjunto y *d* la cantidad de clústers.

Ejemplo 1: sea x: la cantidad de productos vendidos e y: la cantidad de clientes, con el fin de generar dos grupos (k=2) se seleccionan dos puntos al azar a partir de la sig. tabla:

Observaciones	X: cant. productos	Y: cant. clientes
X1	2	6
X2	3	4
X3	3	8
X4	4	7
X5	6	2
X6	6	4
X7	7	3
X8	7	4
X9	8	5
X10	7	6

Caso 1: Medoids: x2, x8



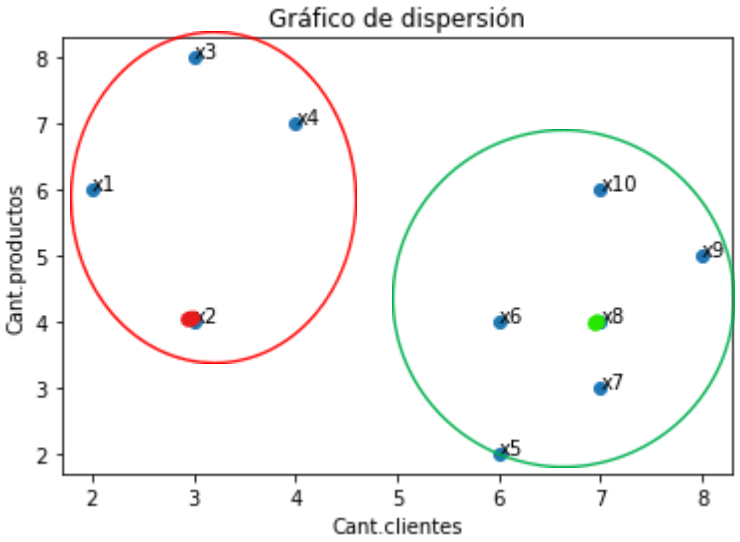
Calcular las distancias entre el medoid x2 y los demás puntos:

X2 (medoid)	**xi**	**Distancia** = | xi-x2 |
(3,4)	**(2,6)**	| (2-3)+(6-4) | =3
(3,4)	**(3,8)**	| (3-3)+(8-4) | =4
(3,4)	**(4,7)**	| (4-3)+(7-4) | =4
(3,4)	(6,2)	| (6-3)+(2-4) | =5
(3,4)	(6,4)	| (6-3)+(4-4) | =3
(3,4)	(7,3)	| (7-3)+(3-4) | =5
(3,4)	(8,5)	| (8-3)+(5-4) | =6
(3,4)	(7,6)	| (7-3)+(6-4) | =6

Calcular las distancias entre el medoid x8 y los demás puntos:

X8 (medoid)	**xi**	**Distancia** = | xi-x8 |
(7,4)	(2,6)	| (2-7)+(6-4) | =7
(7,4)	(3,8)	| (3-7)+(8-4) | =8
(7,4)	(4,7)	| (4-7)+(7-4) | =6
(7,4)	**(6,2)**	| (6-7)+(2-4) | =3
(7,4)	**(6,4)**	| (6-7)+(4-4) | =1
(7,4)	**(7,3)**	| (7-7)+(3-4) | =1
(7,4)	**(8,5)**	| (8-7)+(5-4) | =2
(7,4)	**(7,6)**	| (7-7)+(6-4) | =2

Seleccionar los puntos más cercanos a cada medoid y representamos la primera opción de agrupamiento:



Calcular el costo total de la partición con medoids: x2 y x8

Costo total= (3+4+4) + (3+1+1+2+2) =**20**

Luego se deben seleccionar al azar otro par de puntos, calcular las distancias hacia cada medoid y el costo total.

Caso 2: Medoids: x4, x6

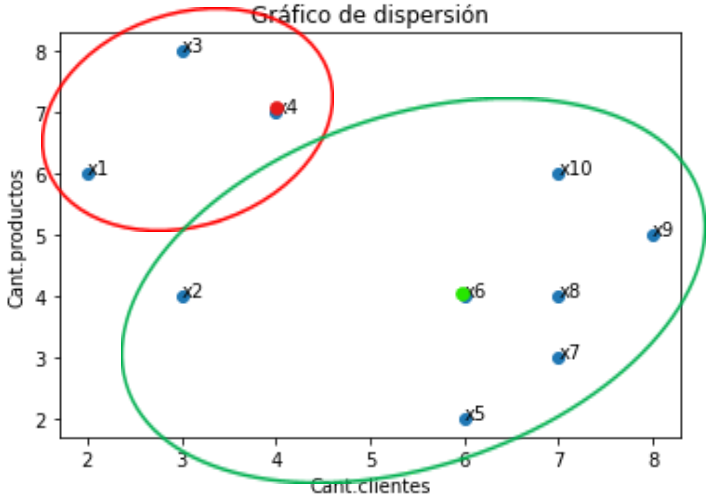
Observaciones	X: cant. productos	Y: cant. clientes
X1	2	6
X2	3	4
X3	3	8
X4	4	7
X5	6	2
X6	6	4
X7	7	3
X8	7	4
X9	8	5
X10	7	6

Se calculan las distancias de los puntos a los nuevos medoids:

X4 (Medoid)	xi	Distancia = | xi-x4 |
(4,7)	**(2,6)**	| (2-4)+(6-7) | =3
(4,7)	(3,4)	| (3-4)+(4-7) | =4
(4,7)	**(3,8)**	| (3-4)+(8-7) | =2
(4,7)	(6,2)	| (6-4)+(2-7) | =7
(4,7)	(7,3)	| (7-4)+(3-7) | =7
(4,7)	(7,4)	| (7-4)+(4-7) | =6
(4,7)	(8,5)	| (8-4)+(5-7) | =6
(4,7)	(7,6)	| (7-4)+(6-7) | =4

X6 (Medoid)	xi	Distancia = | xi-x6 |
(6,4)	(2,6)	| (2-6)+(6-4) | =6
(6,4)	**(3,4)**	| (3-6)+(4-4) | =3
(6,4)	(3,8)	| (3-6)+(8-4) | =7
(6,4)	**(6,2)**	| (6-6)+(2-4) | =3
(6,4)	**(7,3)**	| (7-6)+(3-4) | =2
(6,4)	**(7,4)**	| (7-6)+(4-4) | =1
(6,4)	**(8,5)**	| (8-6)+(5-4) | =3
(6,4)	**(7,6)**	| (7-6)+(6-4) | =3

Se representa la nueva distribución de los clúster:



Calcular el costo total de la partición con medoids: x4 y x6

Costo total= (3+2) + (3+3+2+1+3+3) =**20**

Caso 3: Medoids: x5, x10

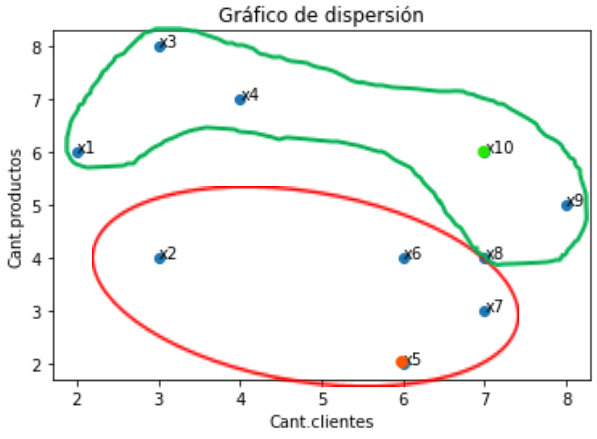
Observaciones	X: cant. productos	Y: cant. clientes
X1	2	6
X2	3	4
X3	3	8
X4	4	7
X5	6	2
X6	6	4
X7	7	3
X8	7	4
X9	8	5
X10	7	6

Se calculan las distancias de los puntos a los nuevos medoids:

X5 (Medoid)	xi	**Distancia =** | xi-x5 |
(6,2)	(2,6)	| (2-6)+(6-2) | =8
(6,2)	**(3,4)**	| (3-6)+(4-2) | =5
(6,2)	(3,8)	| (3-6)+(8-2) | =9
(6,2)	(4,7)	| (4-6)+(7-2) | =7
(6,2)	**(6,4)**	| (6-6)+(4-2) | =2
(6,2)	**(7,3)**	| (7-6)+(3-2) | =2
(6,2)	(7,4)	| (7-6)+(4-2) | =3
(6,2)	(8,5)	| (8-6)+(5-2) | =5

X10 (Medoid)	xi	**Distancia =** | xi-x10 |
(7,6)	**(2,6)**	| (2-7)+(6-6) | =5
(7,6)	(3,4)	| (3-7)+(4-6) | =6
(7,6)	**(3,8)**	| (3-7)+(8-6) | =6
(7,6)	**(4,7)**	| (4-7)+(7-6) | =4
(7,6)	(6,4)	| (6-7)+(4-6) | =3
(7,6)	(7,3)	| (7-7)+(3-6) | =3
(7,6)	**(7,4)**	| (7-7)+(4-6) | =2
(7,6)	**(8,5)**	| (8-7)+(5-6) | =2

Se representa la nueva distribución de los clúster:



Calcular el costo total de la partición con medoids: x5 y x10

Costo total = (5+2+2) + (5+6+4+2+2)=**28**

Por último, el algoritmo selecciona el modelo que logre minimizar la función de costo total. En este caso podríamos seleccionar el caso1 o el caso 2.

K-means

Es otro método de clusterización utilizado para agrupar k elementos a partir de un conjunto de n observaciones en base al valor medio más cercano y se aplica para resolver problemas de optimización.

A partir de un conjunto de observaciones $x_1, x_2, x_3, ..., x_n$ se genera una partición de k subconjuntos con el objetivo de minimizar la suma de los cuadrados dentro de cada grupo

S= $\{s_1, s_2, s_3, ..., s_n\}$ con respecto a su centroide μ_i.

$$\min_{\mathbf{S}} E(\boldsymbol{\mu_i}) = \min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x}_j \in S_i} \|\mathbf{x}_j - \boldsymbol{\mu_i}\|^2$$

Para utilizar el algoritmo k-means, en primer lugar se debe determinar la cantidad de k clúster a agrupar y seleccionar de manera aleatoria los k centroides perteneciente a cada subconjunto. En segundo lugar, se asignan las observaciones a su centroide más próximo (expectation). Luego, se calcula el promedio de los elementos de cada grupo, se actualiza de esta forma el valor del centroide y se vuelven a agrupar los datos más próximos (maximization).

Finalmente, se repite la actualización del centroide y la asignación de objetos hasta que los centroides seleccionados no cambien (condición de stop).

Retomando la tabla del ejemplo 1, se seleccionan dos subconjuntos (k=2) y se determinan aleatoriamente los centroides dentro de cada grupo por ejemplo: X5 y X10.

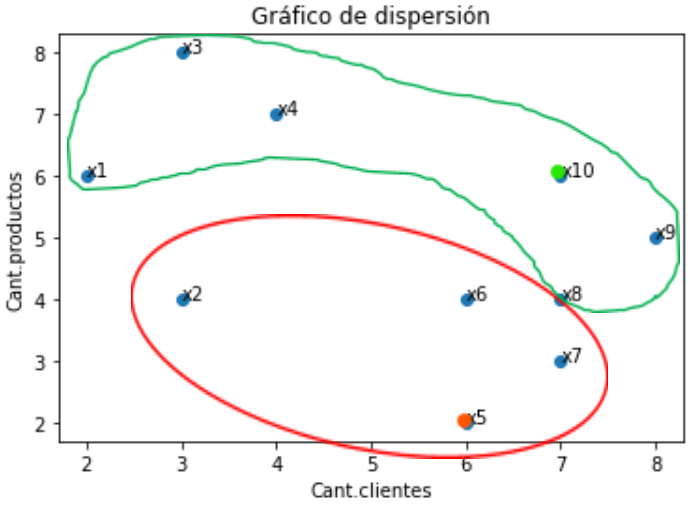
Observaciones	**X: cant. productos**	**Y: cant. clientes**
X1	2	6
X2	3	4
X3	3	8
X4	4	7
X5 (centroide 1)	6	2
X6	6	4
X7	7	3
X8	7	4
X9	8	5
X10 (centroide 2)	7	6

Luego se asignar las observaciones más próximas a cada centroide calculando las distancias euclideas entre cada punto.

X5 (Centroide)	xi	d: $\sqrt{[\Delta x^2 + \Delta y^2]}$
(6,2)	(2,6)	√[(2-6) ² +(6-2) ²] =5.66
(6,2)	**(3,4)**	√[(3-6) ²+(4-2) ²] =3.60
(6,2)	(3,8)	√[(3-6) ²+(8-2) ²] =6.70
(6,2)	(4,7)	√[(4-6) ²+(7-2) ²] =5.38
(6,2)	**(6,4)**	√[(6-6) ²+(4-2) ²] =2
(6,2)	**(7,3)**	√[(7-6) ²+(3-2) ²] =1.41
(6,2)	(7,4)	√[(7-6) ²+(4-2) ²] =2.24
(6,2)	(8,5)	√[(8-6) ²+(5-2) ²] =3.60

X10 (Centroide)	xi	d: $\sqrt{[\Delta x^2 + \Delta y^2]}$
(7,6)	**(2,6)**	√[(2-7) ²+(6-6) ²] =5
(7,6)	(3,4)	√[(3-7) ²+(4-6) ²] =4.47
(7,6)	**(3,8)**	√[(3-7) ²+(8-6) ²] =4.47
(7,6)	**(4,7)**	√[(4-7) ²+(7-6) ²] =3.16
(7,6)	(6,4)	√[(6-7) ²+(4-6) ²] =2.23
(7,6)	(7,3)	√[(7-7) ²+(3-6) ²] =3
(7,6)	**(7,4)**	√[(7-7) ²+(4-6) ²] =2
(7,6)	**(8,5)**	√[(8-7) ²+(5-6) ²] =1.41

- **Distorción:** promedio de las distancias entre el centroide y los puntos. La distorsión en cada cluster es: 3.059 y 2.574.
- **Inercia**: Suma de las distancias entre el centroide y las observaciones conocido como: Sum of Squares Errrors (SSE). ***El algoritmo kmeans tiene como objetivo su minimizar la inercia en cada iteración.*** La inercia para el 1° grupo es: 30.59 y para el 2° cluster es 25.74.



Se determina el promedio de los elementos de cada grupo,

X5 (Centroide)	xi	$\mu_i : (\mu_x, \mu_y)$
(6,2)	**(3,4)**	**(5.33 , 9)**
(6,2)	**(6,4)**	
(6,2)	**(7,3)**	

X10 (Centroide)	xi	$\mu_i : (\mu_x, \mu_y)$
(7,6)	**(2,6)**	**(16.8 , 6)**
(7,6)	**(3,8)**	
(7,6)	**(4,7)**	
(7,6)	**(7,4)**	
(7,6)	**(8,5)**	

Se obtienen los nuevos centroides C1= (5.33, 9) y C2= (16.8, 6) y se vuelven a agrupar los datos más próximos. Se itera hasta que el centroide quede fijo y de esta forma se determinan los elementos pertenecientes a cada clúster.