

# Aprendizaje No Supervisado

Maestría en Ciencia de Datos

---

Lucas Fernández Piana

Primavera 2022

Universidad de San Andrés

# Métodos Jerárquicos

---

## Partición

Sea  $D$  un conjunto recordemos que una partición  $\mathcal{C}$  de  $D$  es una colección de subconjuntos  $\{C_1, \dots, C_k\}$  de  $D$  tales que

- $C_i \cap C_j = \emptyset$ .
- $\bigcup_{i=1}^K C_i = D$ .

## Partición

Sea  $D$  un conjunto recordemos que una partición  $\mathcal{C}$  de  $D$  es una colección de subconjuntos  $\{C_1, \dots, C_k\}$  de  $D$  tales que

- $C_i \cap C_j = \emptyset$ .
- $\bigcup_{i=1}^k C_i = D$ .

## Partición Anidada

Sea  $\mathcal{C}$  y  $\mathcal{B}$  dos particiones de  $D$ . Decimos que  $\mathcal{B}$  está anidada en  $\mathcal{C}$  si se cumple que para cada  $B \in \mathcal{B}$  existe  $C \in \mathcal{C}$  tal que  $B \subset C$ .

# Métodos Jerárquicos

## Partición Anidada

Sea  $\mathcal{C}$  y  $\mathcal{B}$  dos particiones de  $D$ . Decimos que  $\mathcal{B}$  está anidada en  $\mathcal{C}$  si se cumple que para cada  $B \in \mathcal{B}$  existe  $C \in \mathcal{C}$  tal que  $B \subset C$ .

**Ejemplo:**  $D = \{x_1, x_2, x_3, x_4, x_5\}$

$$\mathcal{C} = \{\{x_1, x_2, x_4\}, \{x_3, x_5\}\}.$$

$$\mathcal{B} = \{\{x_1, x_2\}, \{x_4\}, \{x_3, x_5\}\}.$$

$\mathcal{A} = \{\{x_1, x_3\}, \{x_2\}, \{x_4, x_5\}\}$  no está anidada en  $\mathcal{C}$ .

# Métodos Jerárquicos

Un **método de cluster jerárquico** sobre un conjunto de datos  $D = \{x_1, \dots, x_n\}$  consiste de una sucesiones de particiones  $\{\mathcal{C}_0, \dots, \mathcal{C}_{n-1}\}$  de  $D$  tales que

- $\mathcal{C}_0 = D$
- $\mathcal{C}_{i+1}$  está anidada en  $\mathcal{C}_i$ .
- $\mathcal{C}_{n-1} = \{\{x_1\}, \dots, \{x_n\}\}$ .

A grandes rasgos hay dos tipos de algoritmos para construir un método jerárquico,

# Métodos Jerárquicos

## Aglomerativos

Comienzan con la última partición de la sucesión donde cada elemento del conjunto es un cluster y en cada paso, van juntando elementos hasta que  $D$  es considerado un cluster.

## Divisivos

Lo contrario a los aglomerativos, arracan con la primer partición donde  $D$  es un cluster, en cada paso dividen cada cluster de la partición en otros más pequeños. Terminan cuando cada elemento del conjunto forma un cluster.

# Algoritmos Aglomerativos

---



# Single Linkage

Sea  $D = \{x_1, \dots, x_n\}$  un conjunto de datos.

Para construir estos algoritmos es necesario contar con una disimilaridad. Llamemos  $d_{ij} = d(x_i, x_j)$  y para dos clusters  $G$  y  $H$  en  $D$  llamemos **disimilaridad intra grupos SL** a

$$D_{SL}(G, H) = \min_{\substack{i \in G \\ j \in H}} d_{ij}.$$

Veamos cómo funciona el algoritmo con un ejemplo

## Single Linkage - Ejemplo

Consideremos el conjunto  $\{x_1, x_2, x_3, x_4, x_5\}$  con la matriz de disimilaridad,

	1	2	3	4	5
1	0	2,3	3,4	1,2	3,7
2	0	0	2,6	1,8	4,6
3	0	0	0	4,2	0,7
4	0	0	0	0	4,4
5	0	0	0	0	0

## Single Linkage - Ejemplo: paso 0

**Paso 0:** la primera partición  $\mathcal{C}_0 = \{\{x_1\}, \{x_2\}, \{x_4\}, \{x_3\}, \{x_5\}\}$ .

	1	2	3	4	5
1	0	2,3	3,4	1,2	3,7
2	0	0	2,6	1,8	4,6
3	0	0	0	4,2	0,7
4	0	0	0	0	4,4
5	0	0	0	0	0

Por comodidad anotemos  $\mathcal{C}_0 = \{(1), (2), (3), (4), (5)\}$

Además fijemos el valor  $L(0) = 0$ , ya vamos a ver para qué nos sirve.

## Single Linkage - Ejemplo: paso 1

**Paso 1:** Buscar los dos clusters en la partición  $\mathcal{C}_0$  que estén más cercanos según la matriz de disimilaridad y juntarlos.

	1	2	3	4	5
1	0	2,3	3,4	1,2	3,7
2	0	0	2,6	1,8	4,6
3	0	0	0	4,2	0,7
4	0	0	0	0	4,4
5	0	0	0	0	0

## Single Linkage - Ejemplo: paso 1

Entonces según me dice la matriz de disimilaridad debo juntar el cluster (3) con el cluster (5). Por lo tanto, mi siguiente partición será

$$\mathcal{C}_1 = \{(1), (2), (3, 5), (4)\}.$$

Permitanme tomar  $L(1) = d_{35} = 0,7$  prometo que ya vamos a ver por qué lo hago.

## Single Linkage - Ejemplo: paso 2

Para seguir agrupando debo recalcular la matriz de disimilaridad, dado que ahora tengo un nuevo cluster. Utilizaremos la fórmula de  $D_{SL}$  para los clusters en  $\mathcal{C}_1$

	1	2	(3,5)	4
1	0	2,3	nuevo	1,2
2	0	0	nuevo	1,8
(3,5)	0	0	0	nuevo
4	0	0	0	0

## Single Linkage - Ejemplo: paso 2

Rellenando donde dice “nuevo” por los valores que arroja la fórmula de  $D_{SL}$  obtengo esta nueva matriz de disimilaridad.

	1	2	(3, 5)	4
1	0	2,3	3,4	1,2
2	0	0	2,6	1,8
(3, 5)	0	0	0	4,2
4	0	0	0	0

## Single Linkage - Ejemplo: paso 2

Busco los clusters que estén más cerca según la matriz de disimilaridad y agrupo.

	1	2	(3, 5)	4
1	0	2,3	3,4	1,2
2	0	0	2,6	1,8
(3, 5)	0	0	0	4,2
4	0	0	0	0

Mi nueva partición  $\mathcal{C}_2 = \{(1, 4), (2), (3, 5)\}$ .

Me permito tomar  $L(2) = 1,2$ .



## Single Linkage - Ejemplo: paso 3

Recalculo la matriz de disimilaridad para  $\mathcal{C}_2$

	(1, 4)	2	(3, 5)
(1, 4)	0	1,8	3,4
2	0	0	2,6
(3, 5)	0	0	0

## Single Linkage - Ejemplo: paso 3

Busco los clusters que estén más cerca según la matriz de disimilaridad y agrupo.

	(1, 4)	2	(3, 5)
(1, 4)	0	1,8	3,4
2	0	0	2,6
(3, 5)	0	0	0

Mi nueva partición  $\mathcal{C}_3 = \{(1, 4, 2), (3, 5)\}$ .

Me permito tomar  $L(3) = 1,8$ .

## Single Linkage - Ejemplo: paso 3

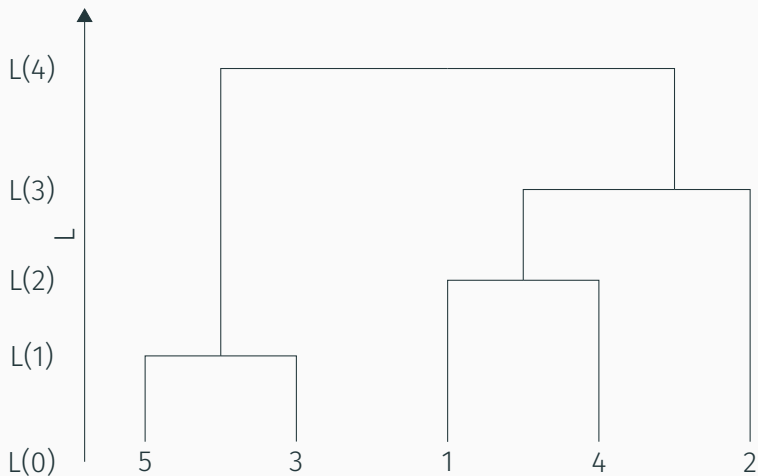
Recalculo la matriz de disimilaridad para  $\mathcal{C}_3$

	(1, 4, 2)	(3, 5)
(1, 4, 2)	0	2,6
(3, 5)	0	0

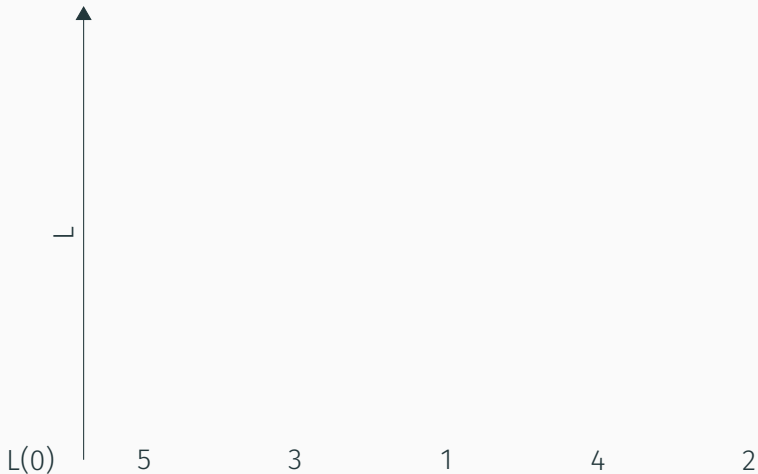
Mi nueva partición  $\mathcal{C}_4 = \{(1, 4, 2, 3, 5)\} = \{D\}$ .

Me permito tomar  $L(4) = 2,6$ .

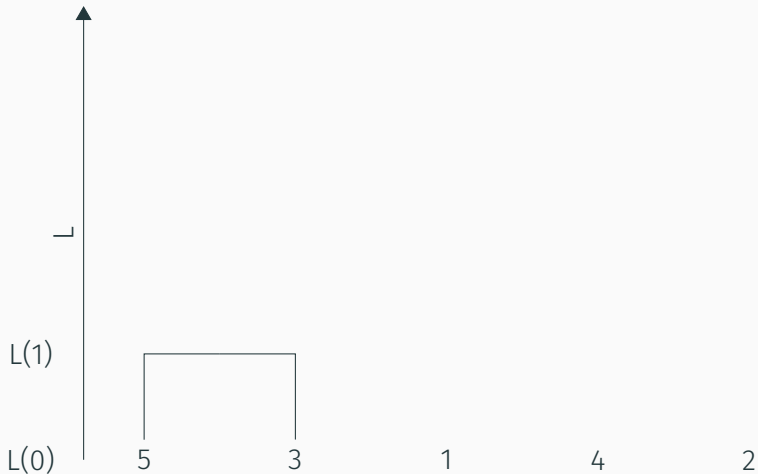
# Single Linkage - Dendrograma



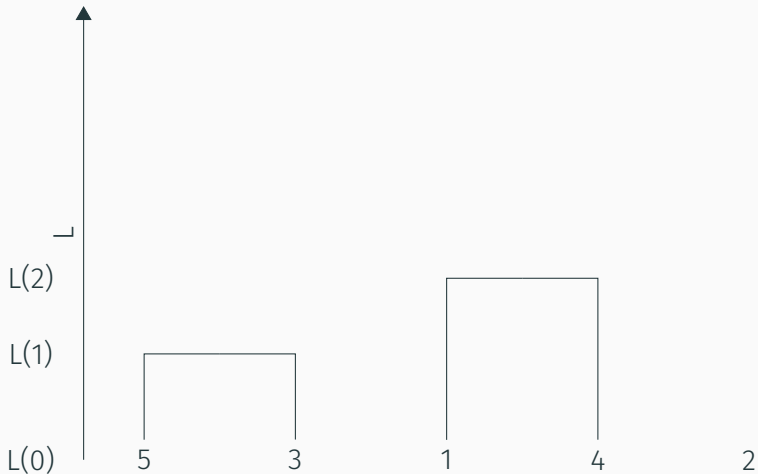
## Single Linkage - Diagrama de Árbol



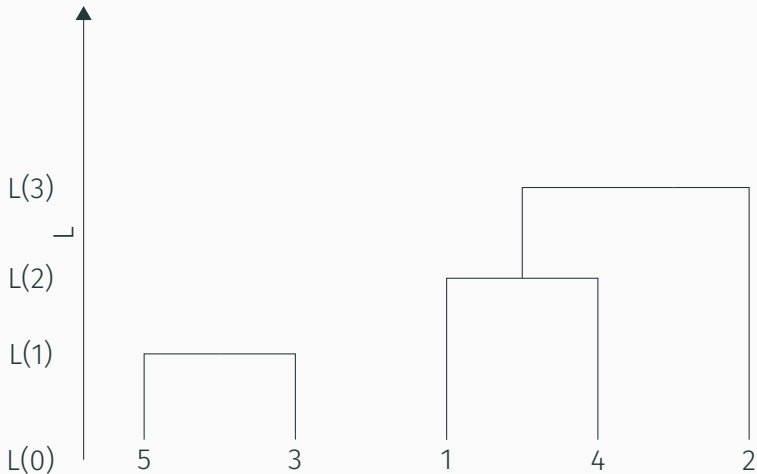
## Single Linkage - Diagrama de Árbol



## Single Linkage - Diagrama de Árbol

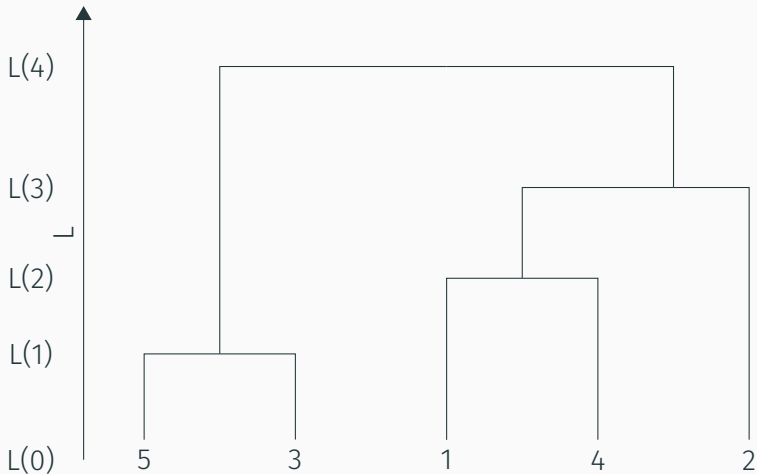


## Single Linkage - Diagrama de Árbol

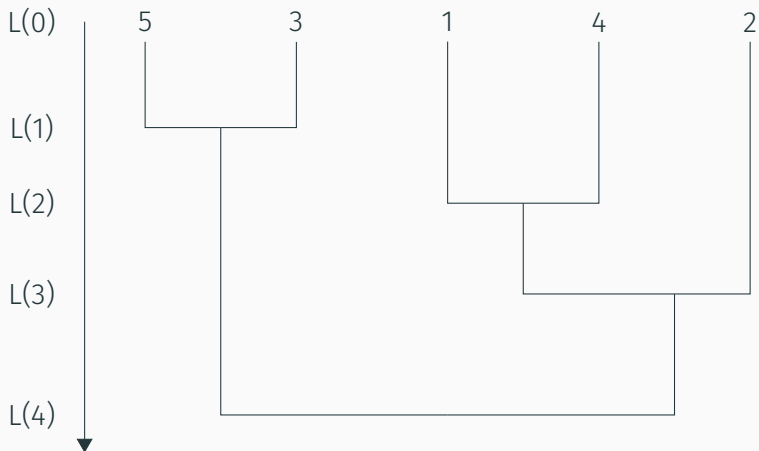




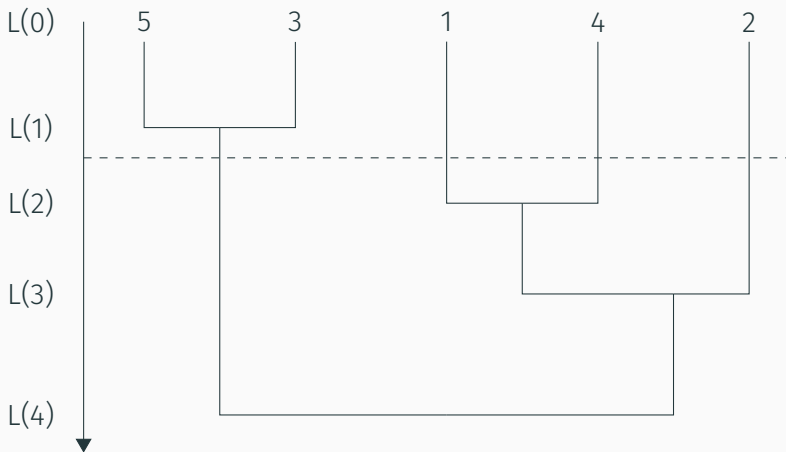
## Single Linkage - Diagrama de Árbol



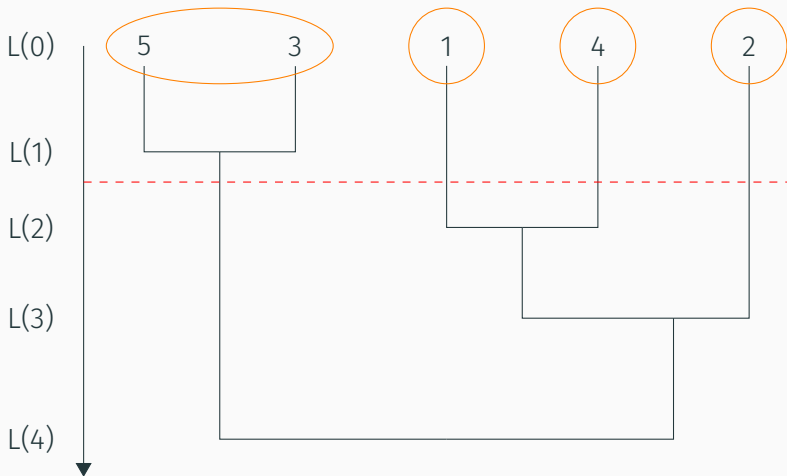
# Single Linkage - Dendrograma



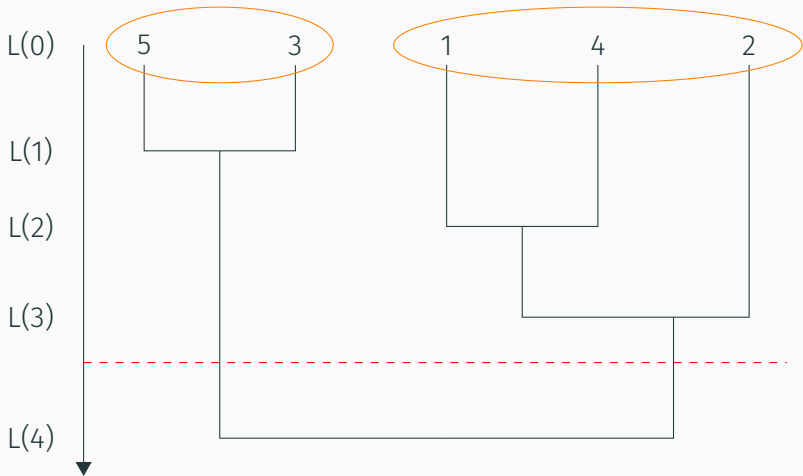
# Single Linkage - Dendrograma y Cortes



## Single Linkage - Dendrograma y Cortes



## Single Linkage - Dendrograma y Cortes



# Complete Linkage

Sea  $D = \{x_1, \dots, x_n\}$  un conjunto de datos.

El algoritmo complete-linkage funciona igual que single-linkage, pero la disimilaridad para dos clusters  $G$  y  $H$  en  $D$  cambia por **disimilaridad intra grupos CL** a

$$D_{CL}(G, H) = \max_{\substack{i \in G \\ j \in H}} d_{ij}.$$

Veamos cómo funciona el algoritmo con un ejemplo

## Complete Linkage - Ejemplo

Consideremos el mismo conjunto de antes  $\{x_1, x_2, x_3, x_4, x_5\}$  con la matriz de disimilaridad,

	1	2	3	4	5
1	0	2,3	3,4	1,2	3,7
2	0	0	2,6	1,8	4,6
3	0	0	0	4,2	0,7
4	0	0	0	0	4,4
5	0	0	0	0	0

## Complete Linkage - Ejemplo: paso 0

**Paso 0:** la primera partición  $\mathcal{C}_0 = \{\{x_1\}, \{x_2\}, \{x_4\}, \{x_3\}, \{x_5\}\}$ .

	1	2	3	4	5
1	0	2,3	3,4	1,2	3,7
2	0	0	2,6	1,8	4,6
3	0	0	0	4,2	0,7
4	0	0	0	0	4,4
5	0	0	0	0	0

Por comodidad anotemos  $\mathcal{C}_0 = \{(1), (2), (3), (4), (5)\}$

Además fijemos el valor  $L(0) = 0$



## Complete Linkage - Ejemplo: paso 1

**Paso 1:** Buscar los dos clusters en la partición  $\mathcal{C}_0$  que estén más cercanos según la matriz de disimilaridad y juntarlos.

	1	2	3	4	5
1	0	2,3	3,4	1,2	3,7
2	0	0	2,6	1,8	4,6
3	0	0	0	4,2	0,7
4	0	0	0	0	4,4
5	0	0	0	0	0

## Complete Linkage - Ejemplo: paso 1

Entonces según me dice la matriz de disimilaridad debo juntar el cluster (3) con el cluster (5). Por lo tanto, mi siguiente partición será

$$\mathcal{C}_1 = \{(1), (2), (3, 5), (4)\}.$$

Tomo  $L(1) = d_{35} = 0,7$ .

## Complete Linkage - Ejemplo: paso 2

Para seguir agrupando debo recalcular la matriz de disimilaridad, dado que ahora tengo un nuevo cluster. Utilizaremos la fórmula de  $D_{CL}$  para los clusters en  $\mathcal{C}_1$

	1	2	(3, 5)	4
1	0	2,3	<b>3,7</b>	1,2
2	0	0	2,6	1,8
(3, 5)	0	0	0	4,2
4	0	0	0	0

## Complete Linkage - Ejemplo: paso 2

Busco los clusters que estén más cerca según la matriz de disimilaridad y agrupo.

	1	2	(3, 5)	4
1	0	2,3	3,4	1,2
2	0	0	2,6	1,8
(3, 5)	0	0	0	4,2
4	0	0	0	0

Mi nueva partición  $\mathcal{C}_2 = \{(1, 4), (2), (3, 5)\}$ .

Tomo  $L(2) = 1,2$ .

## Complete Linkage - Ejemplo: paso 3

Recalculo la matriz de disimilaridad para  $\mathcal{C}_2$

	(1, 4)	2	(3, 5)
(1, 4)	0	2,3	3,4
2	0	0	2,6
(3, 5)	0	0	0

## Complete Linkage - Ejemplo: paso 3

Busco los clusters que estén más cerca según la matriz de disimilaridad y agrupo.

	(1, 4)	2	(3, 5)
(1, 4)	0	2,3	3,4
2	0	0	2,6
(3, 5)	0	0	0

Mi nueva partición  $\mathcal{C}_3 = \{(1, 4, 2), (3, 5)\}$ .

Tomo  $L(3) = 2,3$ .

## Complete Linkage - Ejemplo: paso 3

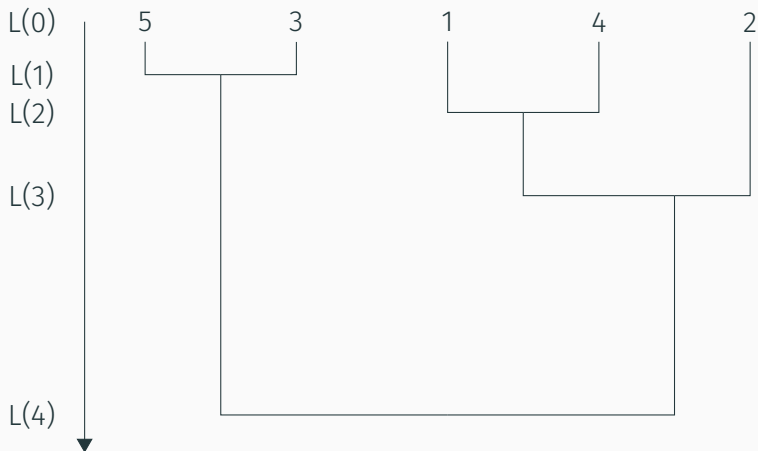
Recalculo la matriz de disimilaridad para  $\mathcal{C}_3$

	(1, 4, 2)	(3, 5)
(1, 4, 2)	0	4,6
(3, 5)	0	0

Mi nueva partición  $\mathcal{C}_4 = \{(1, 4, 2, 3, 5)\} = \{D\}$ .

Me permito tomar  $L(4) = 4,6$ .

## Complete Linkage - Dendrograma y Cortes

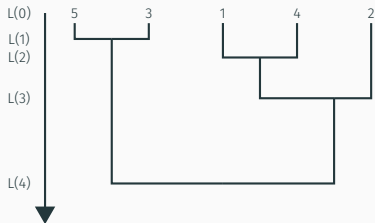




# Complete vs Single - Dendrogramas

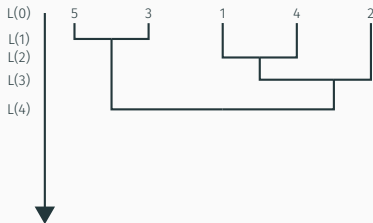
**Usa el maximo de las dist**

Complete Linkage



**Usa el minimo de las dist**

Single Linkage



# Average Linkage

Sea  $D = \{x_1, \dots, x_n\}$  un conjunto de datos.

El algoritmo average-linkage funciona igual que single-linkage o complete-linkage, pero la disimilaridad para dos clusters  $G$  y  $H$  en  $D$  cambia por **disimilaridad intra grupos AL** a

$$D_{AL}(G, H) = \frac{1}{N_G} \frac{1}{N_H} \sum_{i \in G} \sum_{j \in H} d_{ij}.$$

**Observación:** modificar la disimilaridad intra grupos nos genera un algoritmo jerárquico nuevo. Tendremos tantos como disimilaridades podamos inventar. Es importante que esa disimilaridad nos permita construir un dendrograma.

# Updating Matrix Algorithm

- P1 Comenzar con la partición que tiene a cada dato como un cluster, anotar el subíndice de la sucesión de particiones anidadas como  $m = 0$ . Luego, fijar el nivel  $L(0) = 0$
- P2 Encontrar el par de clusters  $\{(r), (s)\}$  con menor disimilaridad de acuerdo a  $d[(r), (s)] = \min d[(i), (j)]$ .
- P3 Juntar los clusters  $(r)$  y  $(s)$  en un nuevo cluster. Actualizar el subíndice de la sucesión de particiones en  $m = m + 1$  y fijar el nivel de este paso  $L(m) = d[(r), (s)]$ .

# Updating Matrix Algorithm

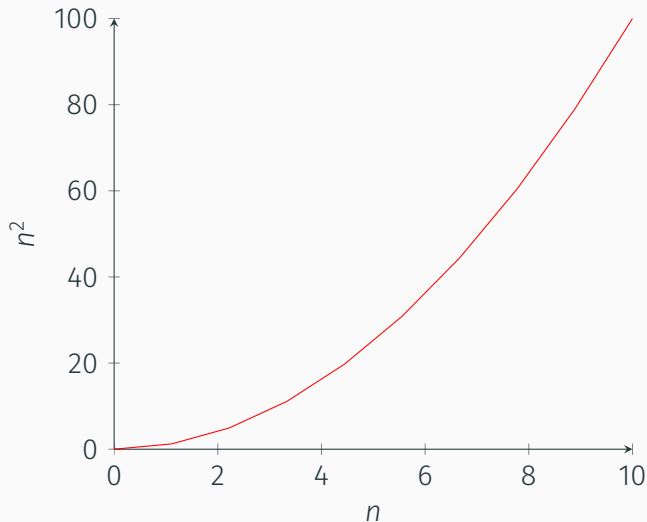
- P4 Actualizar la matriz de disimilaridad borrando las filas y columnas que correspondan a los clusters  $(r)$  y  $(s)$ .  
Agregar una fila y columna que corresponda al nuevo cluster  $(r, s)$  según dicte la fórmula de disimilaridad intra cluster  $d[(k), (r, s)]$  donde  $(k)$  son los clusters que se corresponden con las filas y columnas que no hemos borrado.
- P5 Si tengo un solo cluster terminé, caso contrario vuelvo a P2.

# Updating Matrix Algorithm

Existe una fórmula general para el paso P4 que incluye a muchos algoritmos aglomerativos conocidos,

$$d[(k), (r, s)] = \alpha_s d[(k), (s)] + \alpha_r d[(k), (r)] + \beta d[(r), (s)] + \gamma |d[(k), (r)] - d[(k), (s)]|.$$

**Observación:** notar que si tengo  $n$  datos, con este procedimiento general, tengo que hacer  $n$  pasos para terminar.



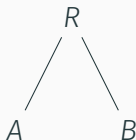
**Metodos pesados computacionalmente**  
**Crece con el cuadrado del numero de datos**

# Algoritmos Divisivos

---

Recordemos que contrariamente a los algoritmos aglomerativos un algoritmo divisivo comienza con la partición trivial:  $\mathbb{C}_0 = D$ .

El algoritmo **DIANA** (Dlvisive ANALysis Clustering) funciona partiendo en cada paso un cluster  $R$  en dos nuevos subconjuntos  $A$  y  $B$ .





¿Cómo funciona esta bisección para el cluster  $R$ ?

Comenzamos tomando  $A = R$  y  $B = \emptyset$ . Para cada elemento  $a$  en  $A$  calculamos su disimilaridad media con respecto al resto de los elementos de  $A$

$$d(a, A - \{a\}) = \frac{1}{|A| - 1} \sum_{s \in A, s \neq a} d(s, a), \quad (1)$$

donde  $|A|$  es el número de elementos en  $A$ .

El elemento  $a^*$  que maximiza (1) será movido a  $B$  siguiendo la regla de actualización,

$$A_{nuevo} = A_{viejo} - \{a^*\} \quad (2)$$

$$B_{nuevo} = B_{viejo} \cup \{a^*\} \quad (3)$$

Ahora tenemos que ver cómo pasar otro elemento de  $A$  a  $B$  si es pertinente.

Mientras  $A$  no me quede vacío, para cada elemento de  $a$  calculo,

$$\text{diss}(a) = d(a, A - \{a\}) - d(a, B) \quad (4)$$

$$= \frac{1}{|A| - 1} \sum_{s \in A, s \neq a} d(s, a) - \frac{1}{|B|} \sum_{h \in B} d(a, h). \quad (5)$$

Consideremos  $a^{**}$  el elemento de  $A$  que maximiza (4). Si  $\text{diss}(a^{**}) > 0$  actualizo según (2) y vuelvo a calcular. En caso que  $\text{diss}(a^{**}) \leq 0$  me detengo y doy por finalizada la bisección.

Finalmente nos queda determinar cómo vamos a elegir el cluster que debemos biseccionar. Para eso tomo los clusters que tengo disponibles y calculo su diametro.

$$diam(R) = \max_{a \in R, b \in R} d(a, b) \quad (6)$$

El cluster que debemos considerar para biseccionar es el que tenga diametro más grande y este diametro se asigna como nivel del paso.