Aprendizaje No-Supervisado

Inteligencia Artificial

Profesor: Alejandro Figueroa

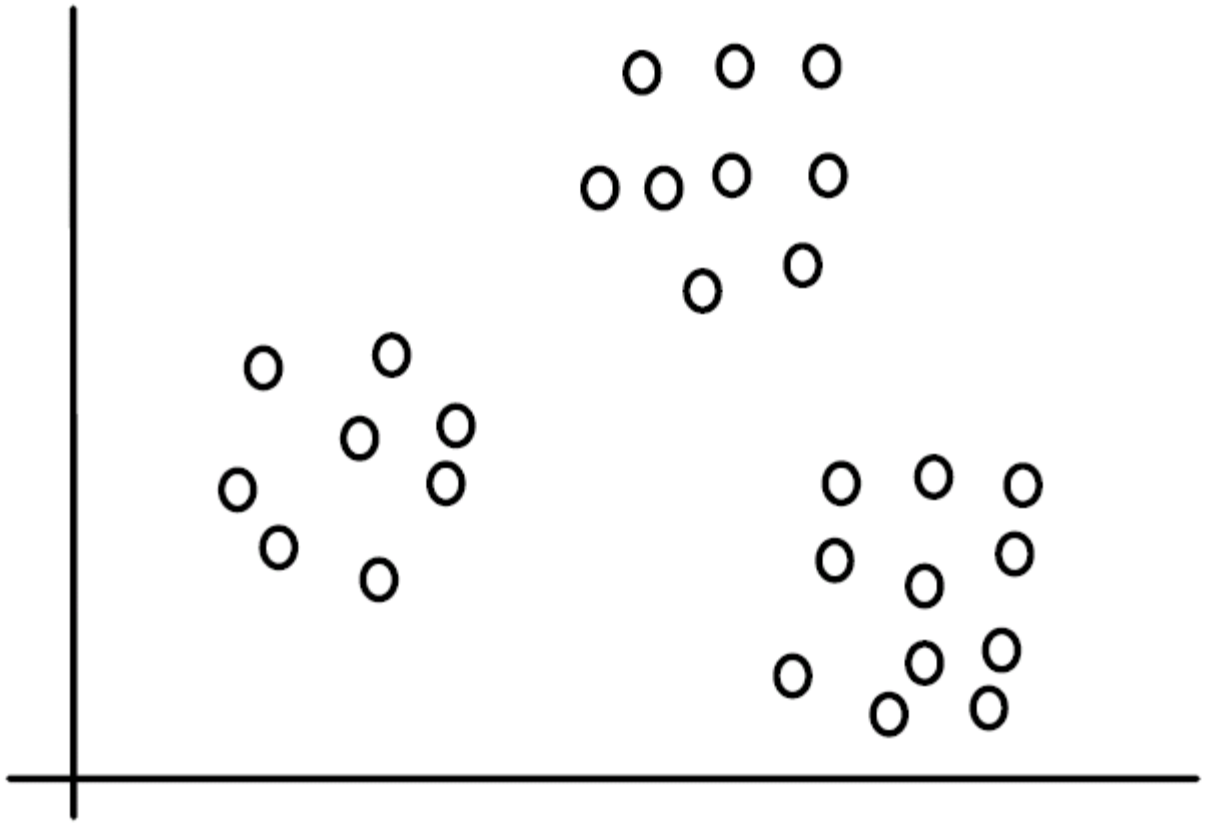
Conceptos Básicos

- En algunas ocasiones no tenemos las clases asociadas en los datos de entrenamiento.
- Pero aún así queremos encontrar estructuras intrínsecas en los datos.
- Clustering es un método para lograr ésto.
- Clustering es llamado aprendizaje no-supervizado, sin embargo, no es lo único que es llamado así.

Conceptos Básicos

- Clustering consiste en organizar los datos en grupos cuyos miembros son parecidos de alguna forma.
- Es decir, un clúster es un conjunto de datos que son similares de alguna forma, y a su vez son “disimiles” a los ejemplos contenidos en otros clústeres.
- En la jerga de clustering, cada instancia en los datos es llamada objeto o data point.

Conceptos Básicos



Conceptos Básicos

- Hay dos tipos de clustering: particional y jerárquico.
- La idea es agrupaciones intrínsecas de los datos de entradas mediante el uso de un algoritmo de clustering y una medida de distancia.

Clustering Particional: K-means

- Es simple y eficiente.
- Dado un número de k de clústeres y un conjunto de datos, este algoritmo particiona iterativamente los datos en los k clústeres de acuerdo a una medida de distancia.
- Supongamos que tenemos un conjunto de datos $D=\{x_1,x_2,x_3,\ldots,x_n\}$, donde $x_i=(x_{i1},x_{i2},\ldots,x_{ir})$ es un vector de valores reales $X\subseteq\Re^r$. Donde r es el número de atributos.

Clustering Particional: K-means

- Cada clúster tiene un centro, llamado centroid, que representa al clúster.
- El centroid es la media de todos los puntos dentro del clúster.
- Al comienzo el algoritmo escoge k datos como los centroides semilla.
- Después, calcula la distancia de centroide semilla con cada dato. Cada dato es asignado al centroide que está más cerca.

Clustering Particional: K-means

- Después que todos los datos son asignados, el centroide de cada clúster es re-calculado utilizando los datos en cada clúster.
- El proceso se repite hasta una condición de término:
  - No hay más, o muy pocas, re-asignaciones de datos a otros clústeres. Es decir, no ha mucho movimiento.
  - No hay más, o muy pequeño, cambio en los centroides.
  - Muy poca disminución en la suma del error cuadrático (SSE).

Clustering Particional: K-means

- La suma del error cuadrático se define como:

$$SSE = \sum_{j=1}^{k} \sum_{x \in C_j} dist(x, m_j)^2$$

- Donde k es el número de clústeres requeridos, C_j es el j-ésimo clúster, m_j es el centroide de C_j y dist(x,m_j) es la distancia entre x y m_j.
- El centroide es el vector de medias de todos los datos en C_j.

Clustering Particional: K-means

- En el espacio euclidiano, la media de es calculada como:

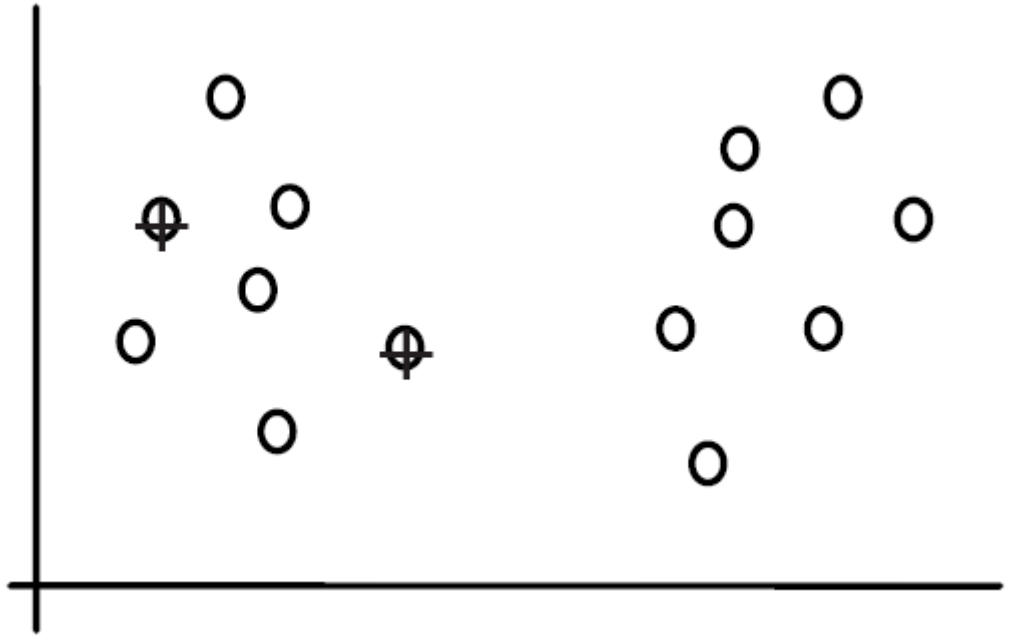
$$m_j = \frac{1}{|C_j|} \sum_{x_j \in C_j} x_i$$

- Donde $|C_j|$ es la cantidad de puntos en C_j. La distancia desde un punto x_j al centroide m_j se calcula:

$$dist(x_i, m_j) = \sqrt{(x_{i1} - m_{j1})^2 + (x_{i2} - m_{j2})^2 + ... + (x_{ir} - m_{jr})^2}$$

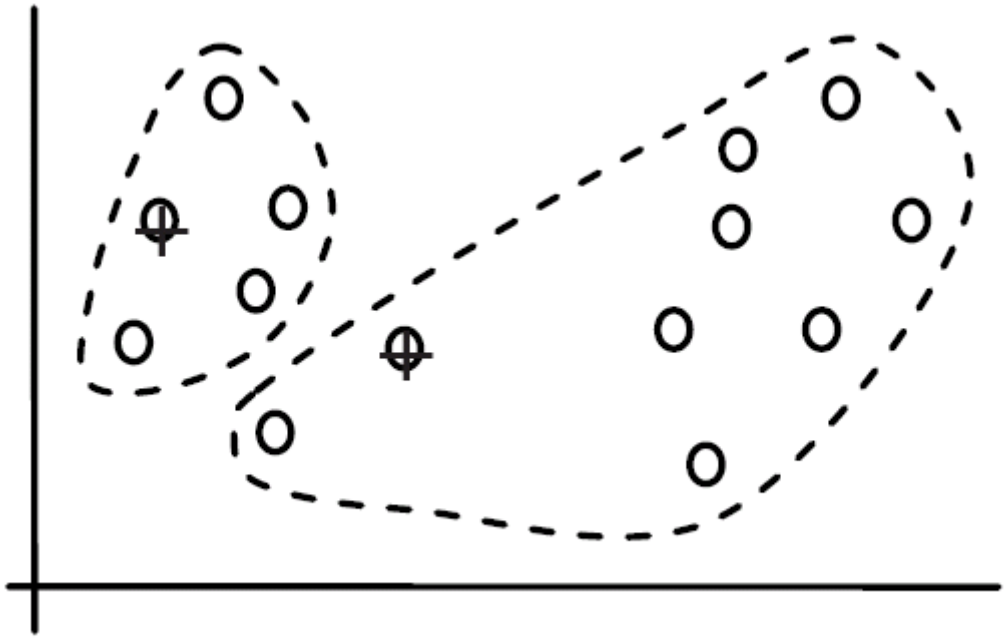
Clustering Particional: K-means

- Al comienzo se escogen k centroides aleatoriamente:



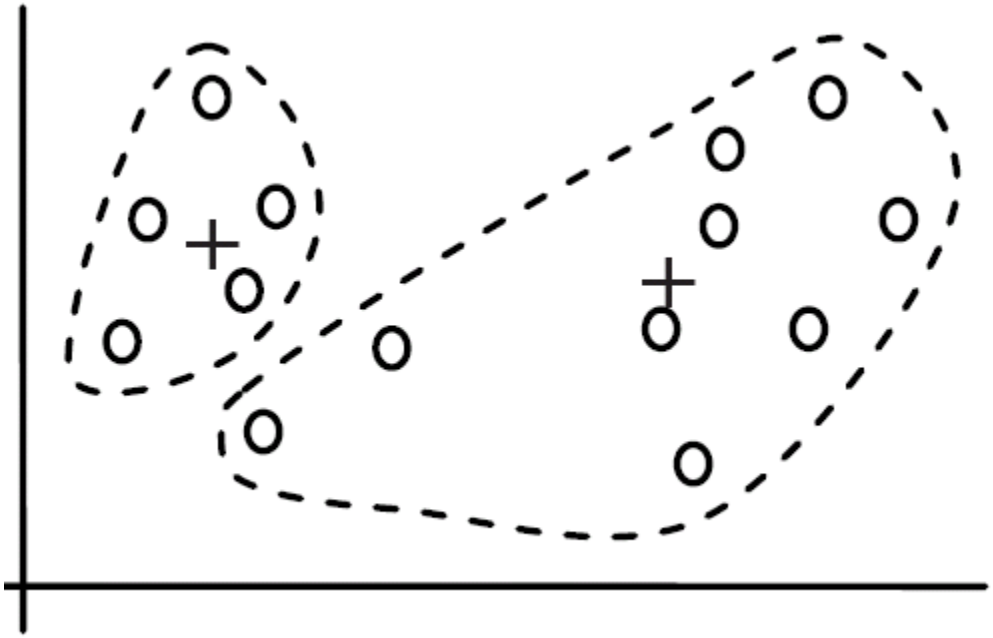
Clustering Particional: K-means

- Se asigna cada punto a uno de los dos clústeres:



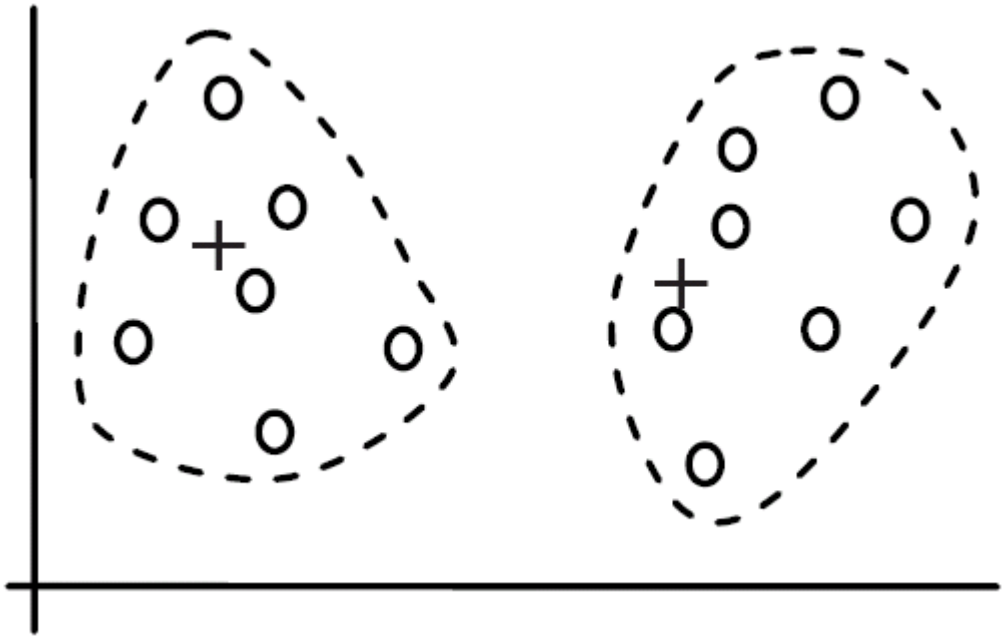
Clustering Particional: K-means

- Los centroides se re-calculan:



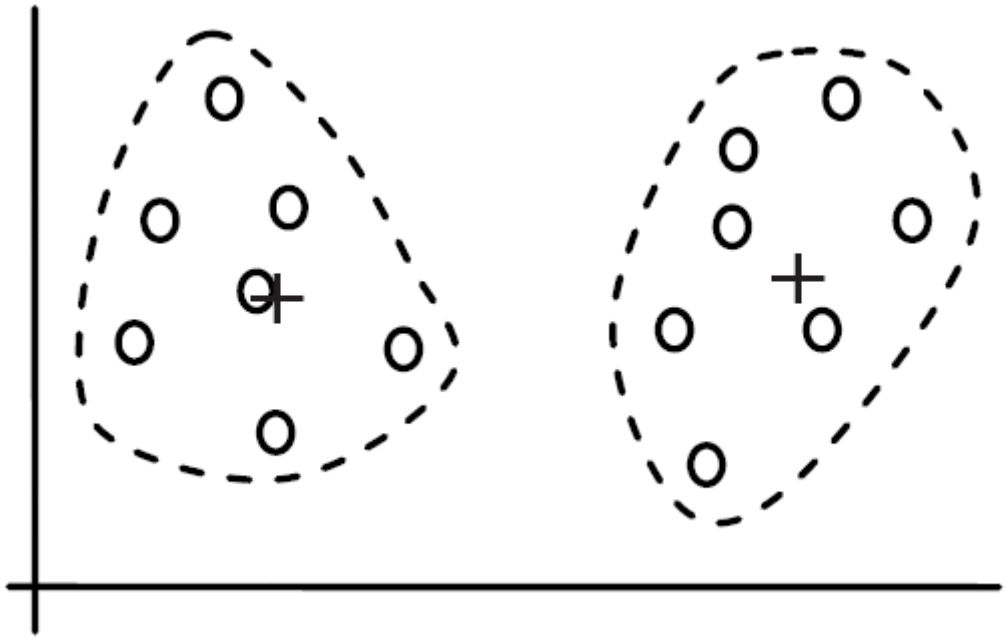
Clustering Particional: K-means

- Los puntos se re-asignan:



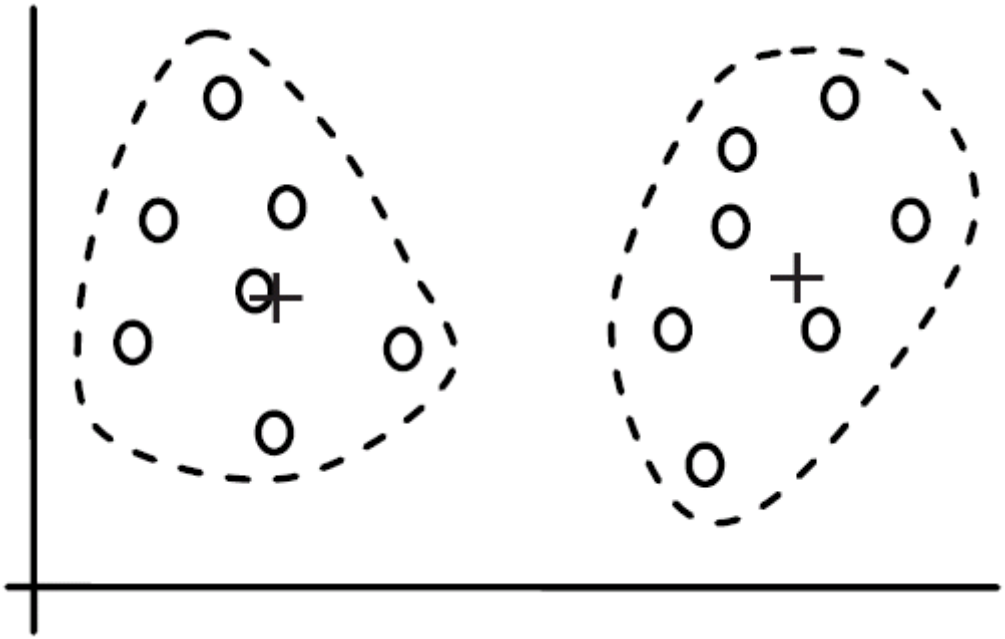
Clustering Particional: K-means

- Se recalculan los centroides:



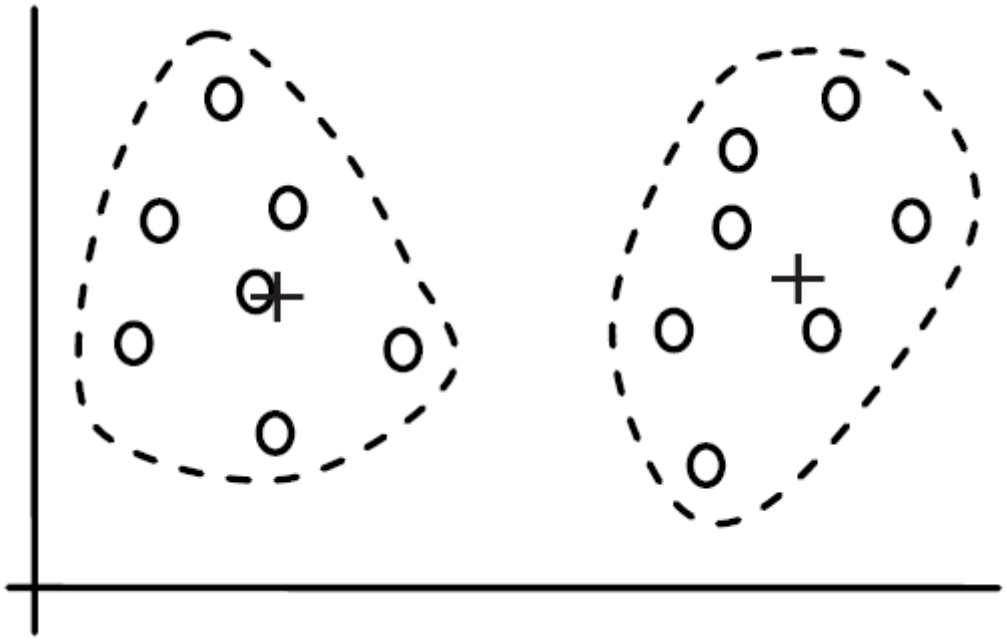
Clustering Particional: K-means

- Se re-asignan los puntos:



Clustering Particional: K-means

- Se recalculan los centroides:



Clustering Particional: K-means

- Clústeres vacios:
  - Durante el proceso algún clúster puede quedar vacio.
  - Se utiliza un data point como el centroid de reemplazo, e.g., el que este más lejos del otro centroid.
  - Si se utiliza SSE, se puede escoger el punto que minimiza el error.
- Es un problema de K-means.

Clustering Particional: K-means

- Las principales ventajas de K-means:
  - Es simple y eficiente.
  - Es fácil de entender y de implementar.
  - Su complejidad es O(tkn), donde n es el número de datos, k el de clústeres y t el número de iteraciones.
  - Dado que k y t son pequeños, normalmente, se considera un algoritmo lineal con respecto al tamaño del conjunto de datos.

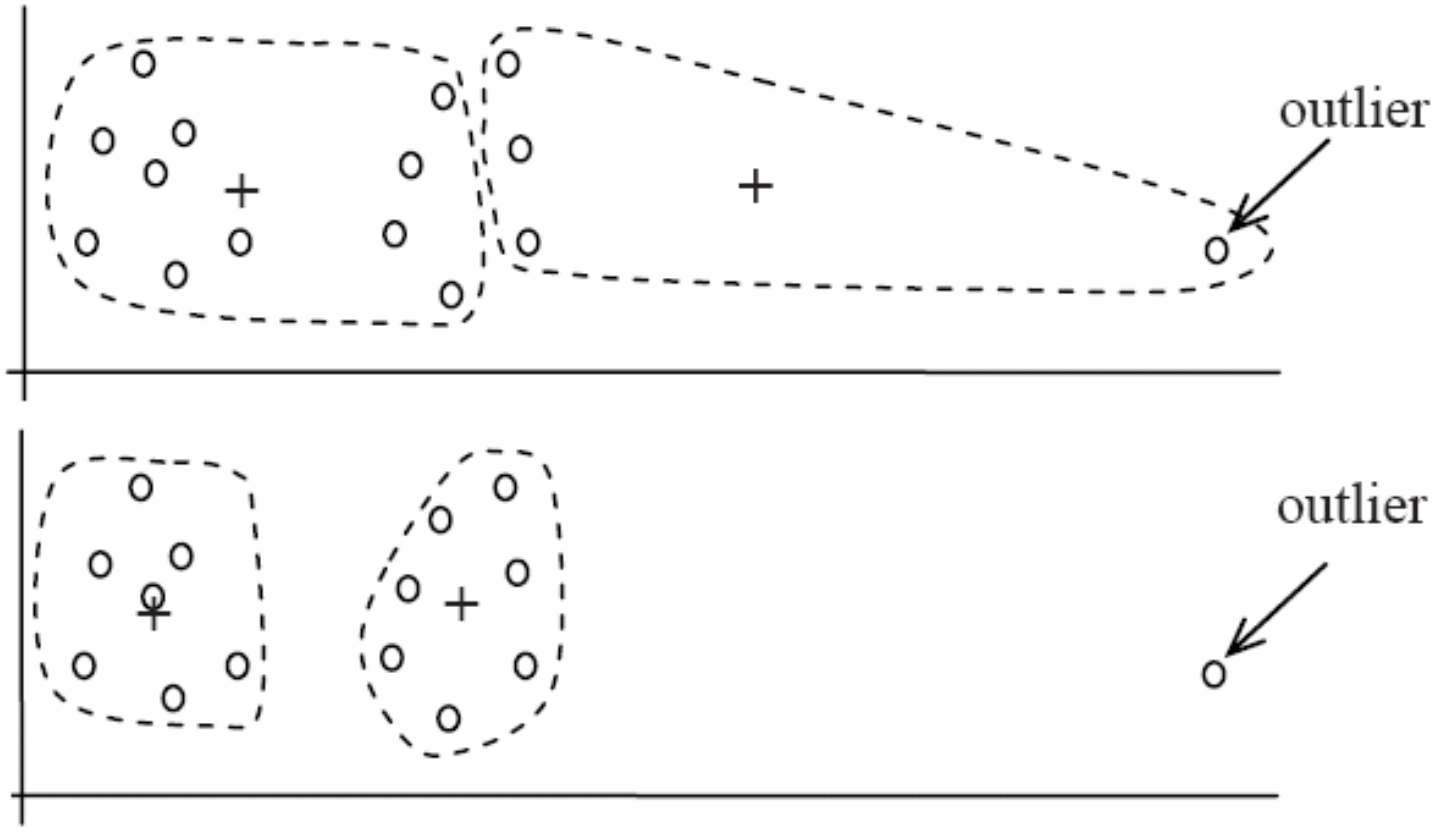
Clustering Particional: K-means

- Otras debilidades:
  - Es necesario tener definido el concepto de media. Por ende, hay un problema con atributos basados en categorías. Hay una variación llamada k-modes, que utiliza la moda en vez de la media. Entonces, cada componente es el valor más frecuente de cada atributo, y la distancia es el número de atributo-valor que coinciden.
  - Muchas veces no se sabe el valor de k de ante-mano
  - El algoritmo es sensible a outliers.

Clustering Particional: K-means

- Una forma de lidiar con los outliers es remover los puntos que están más lejos de los centroides que otros puntos.
- Es primordial esperar varias iteraciones antes de remover un punto, porque podría ser un clúster pequeño de puntos.
- Se puede utilizar una cota.

Clustering Particional: K-means



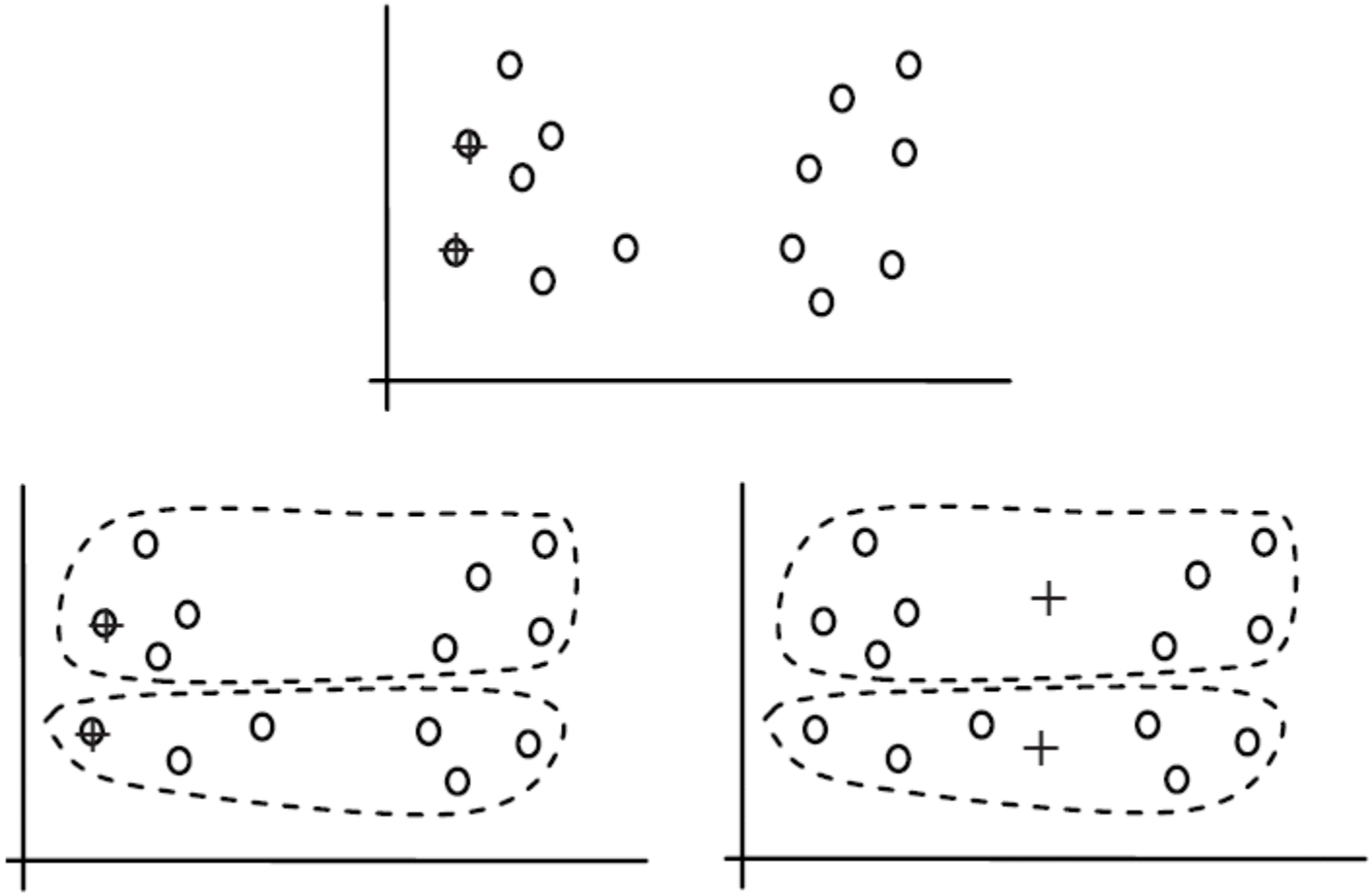
Clustering Particional: K-means

- Otra forma de combatir los outliers es random sampling:
  - Se escoge un pequeño subconjunto de los datos.
  - Se hace aleatoriamente.
  - Se utiliza este subconjunto para calcular los centroides.
  - Después se añaden el resto de los puntos:
    - Al centroide más cercano, o
    - Utilizar los clústeres para hacer aprendizaje supervisado y etiquetar el resto de los puntos.
    - Usar los clústeres como semillas para aprendizaje semi-supervisado.

Clustering Particional: K-means

- Una rápida reseña a lo que es aprendizaje semi-supervisado.
  - Es un paradigma que aprende de un conjunto reducido de ejemplos etiquetados y de un conjunto grande de datos no etiquetados.
- Otro problema de K-means son las semillas iniciales:
  - Diferentes semillas pueden terminar en diferentes centroides alcanzando un óptimo local.

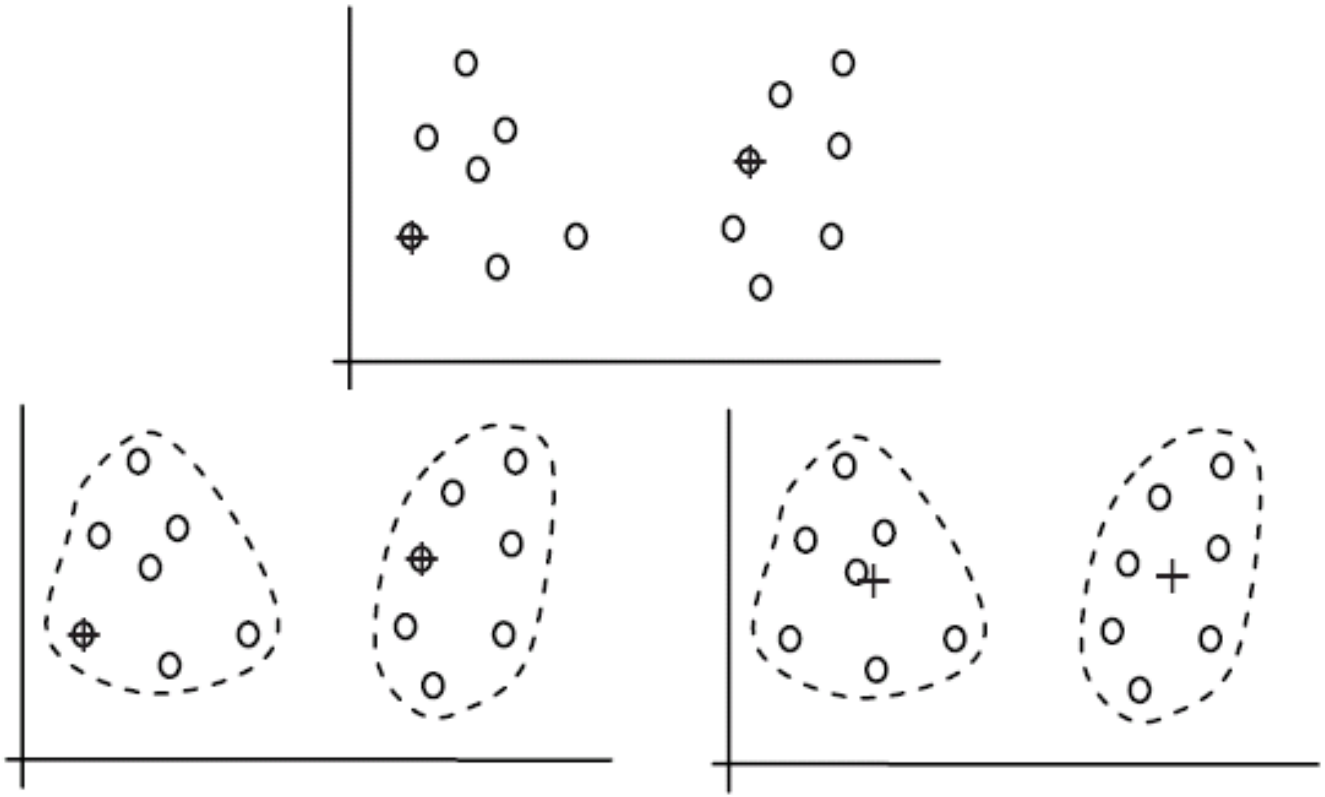
Clustering Particional: K-means



Clustering Particional: K-means

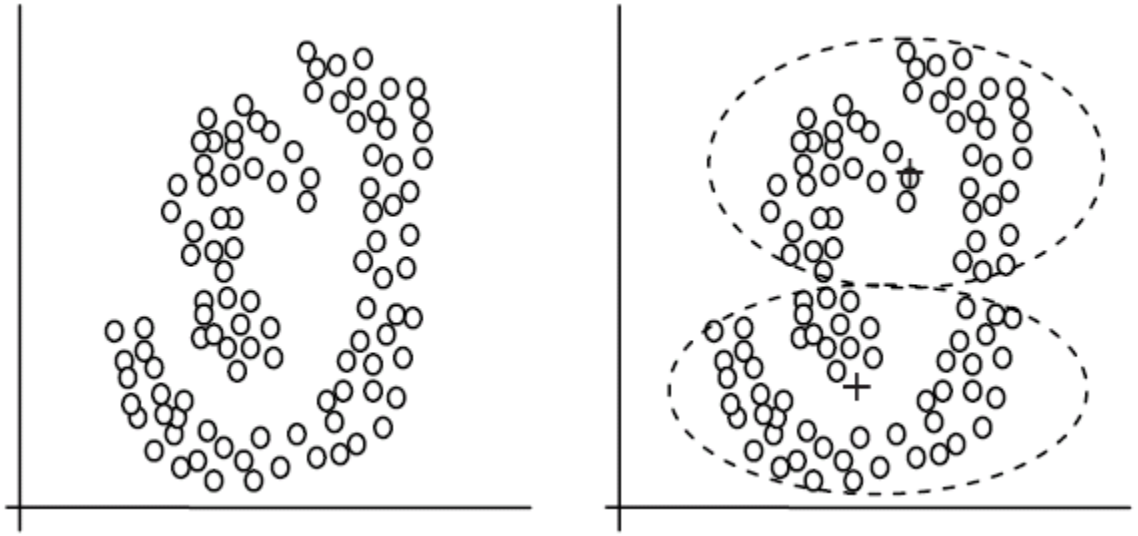
- Hay varios métodos para escoger semillas iniciales:
  - Calcular el centroide del conjunto completo de datos. Se escoge el punto más lejano al centroide, y después se escoge el punto más lejano al previamente escogido, así sucesivamente. Este método no funciona bien con outliers.

Clustering Particional: K-means



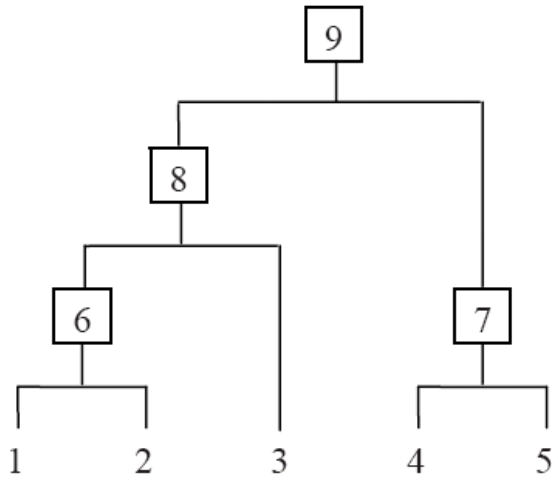
Clustering Particional: K-means

- En la práctica, frecuentemente resulta fácil para los humanos escoger las semillas. Por ejemplo, data points que son muy diferentes.
- K-means no funciona bien cuando hay que descubrir datos que son hiper-esferas.



Clustering Jerárquico

- Produce un árbol, también llamado dendrogram (a), que consiste en un conjunto de clústeres anidados.
- Los datapoint están en las hojas del árbol, y el árbol parte con un único nodo llamado raíz.
- Cada nodo del árbol tiene hijos, y cada hermano particiona los datos que tiene el padre



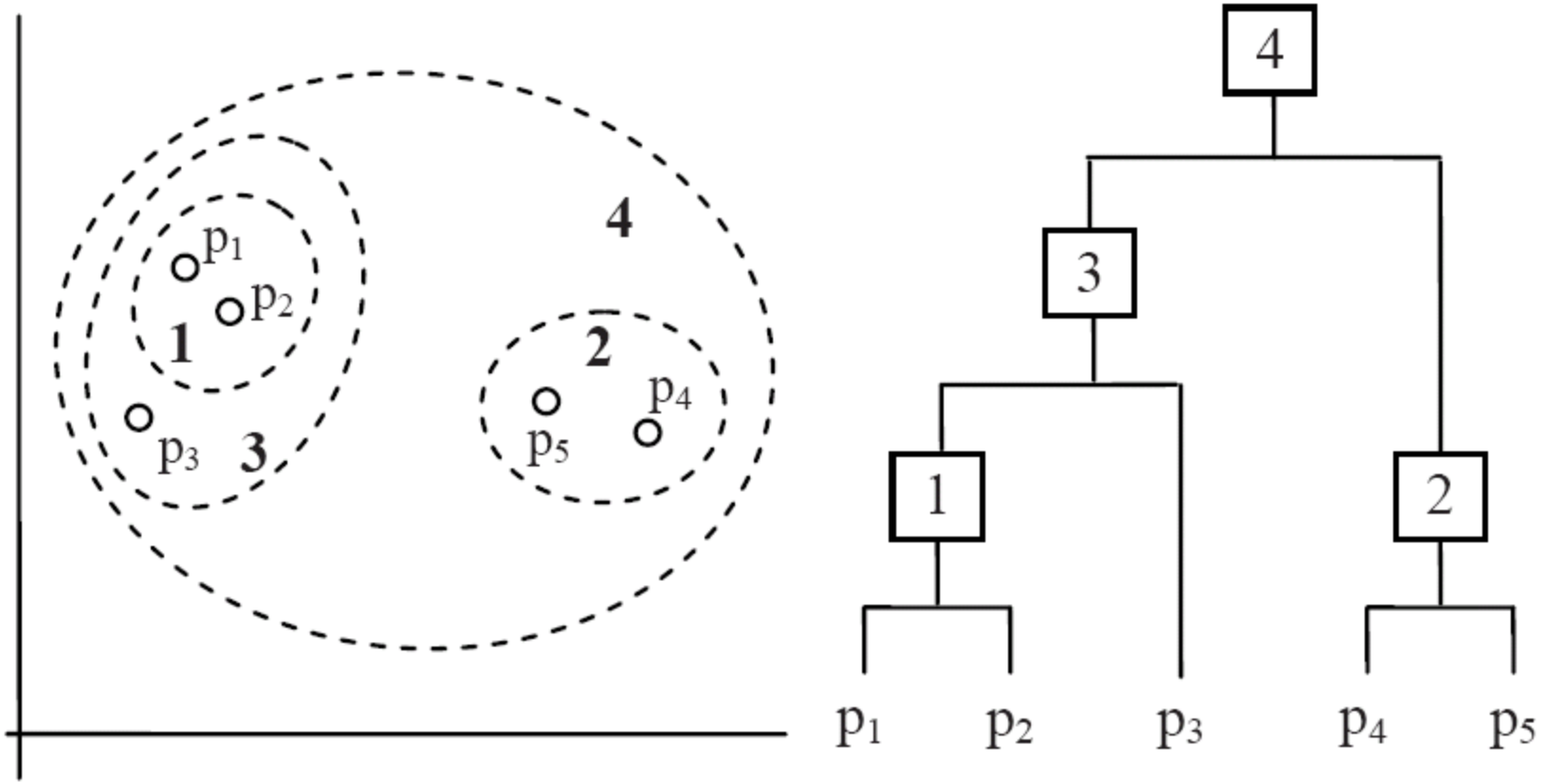
Clustering Jerárquico

- Hay dos tipos:
  - **Agglomerativos (bottom-up):** Construye el árbol mediante la amalgama de los clústeres más cercanos. Así sube niveles en el árbol. Es el método más usado
  - **Divise (top-down):** Parte con todos los datos en un clúster, la raíz. Divide los nodos en hijos sucesivamente, hasta alcanzar clústeres de puntos individuales.

Clustering Jerárquico

- ***Agglomerative*** *(D)*
  1. Hacer de cada data point un clúster.
  2. Calcular todas las distancias entre los puntos en D
  3. **Repetir**
     1. Encontrar los dos clústeres que están más cerca.
     2. Unirlos para formar un nuevo clúster C.
     3. Calcular la distancia de C a todos los clústeres.
  4. **Hasta que** haya un solo clúster.

Clustering Jerárquico



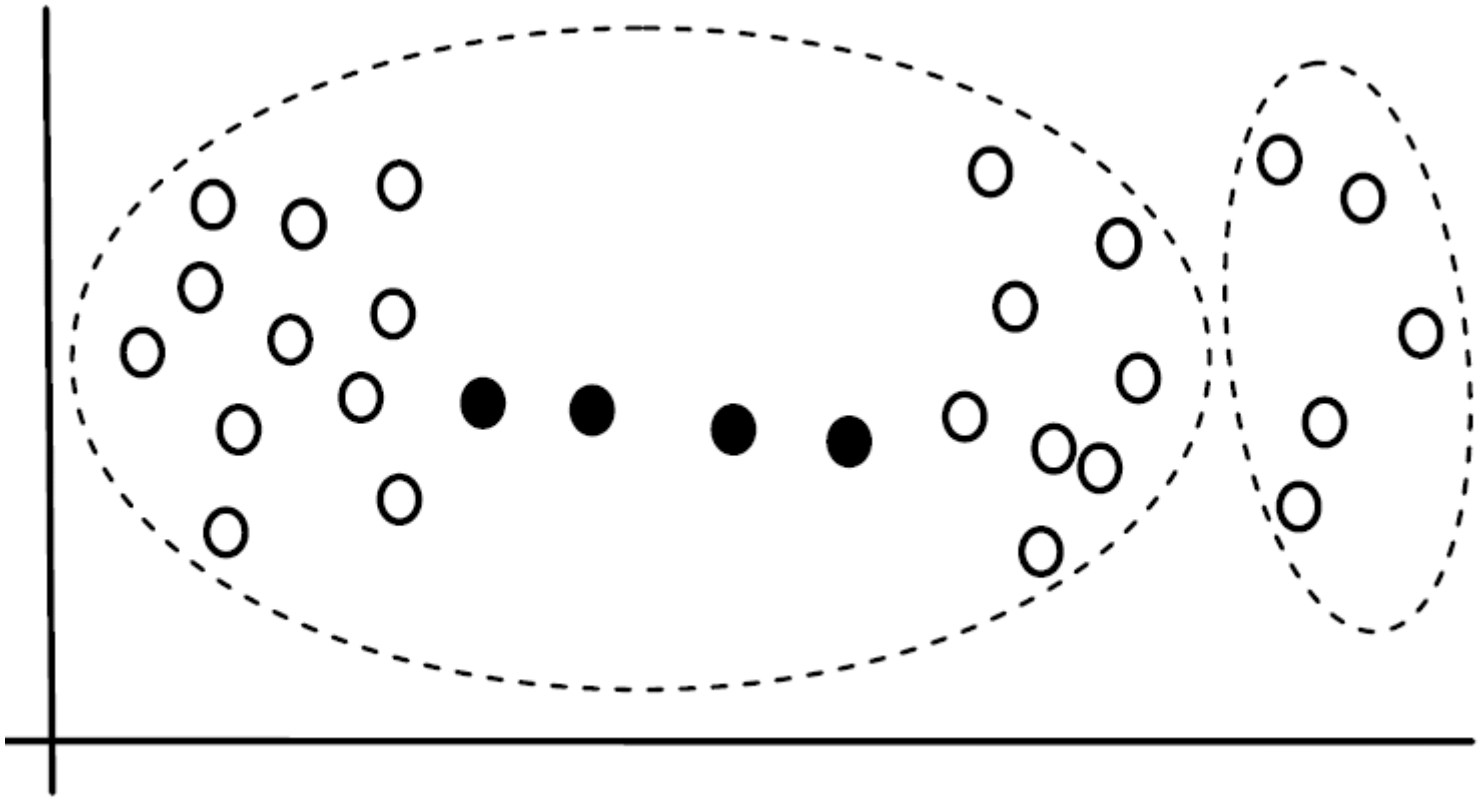
Clustering Jerárquico

- Para calcular la distancia entre dos clústeres:
  - Single-Link
  - Complete-Link
  - Average-Link

Clustering Jerárquico

- Single-Link:
  - La distancia entre dos clústeres es la distancia de los dos puntos más cercanos.
  - Bueno para encontrar clústeres de forma no-eliptica.
  - Es sensible a datos ruidoso.
  - La complejidad es $O(n^2)$, con n el número de datos.

Clustering Jerárquico



Clustering Jerárquico

- Complete-Link:
  - Une dos clústeres tal que su distancia máxima es la mínima entre todos los clústeres.
  - No tiene el problema de los datos ruidosos, pero tiene el problema de los outliers.
  - En general, produce mejores resultados que el método de Single-Link.
  - La complejidad es $O(n^2 \log n)$, con n como el número de datos.

Clustering Jerárquico

- Average-Link:
  - La distancia es el promedio entre todos los pares de puntos de un clúster.
  - La complejidad es $O(n^2 \log n)$, con n como el número de datos.
- Otros métodos:
  - **Centroid**: la distancia está dada por el centroide de los dos clústeres.
  - **Ward**: Se amalgama los clústeres que incrementan el error de menor forma. El error (SSE) se calcula como la diferencia entre el clúster nuevo y los que se une.

Clustering Jerárquico

- Clustering jerárquico tiene las ventajas:
  - Puede tomar cualquier métrica de distancia.
  - No tiene un k definido, es decir puede explorar cualquier nivel de granularidad.
  - Clúster jerárquico agglomerativo produce generalmente los mejores resultados.
- Las desventajas:
  - Son más complejos computacionalmente.
  - Ineficiente y poco práctico para conjuntos de datos muy grandes.

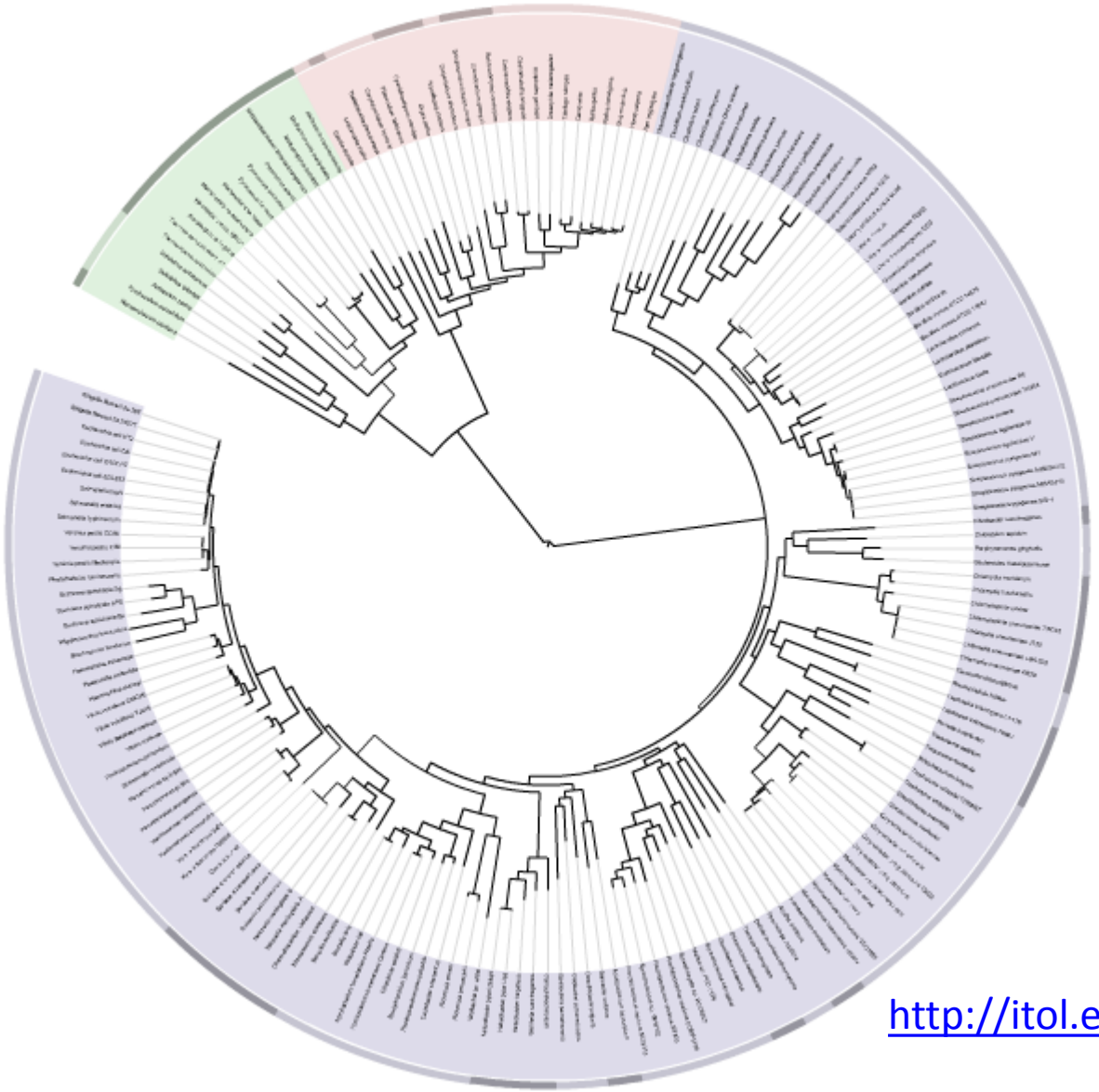
Clustering Jerárquico

- Alguna veces se puede aplicar métodos de escalamiento (scale-up methods) para combatir el problema con los conjuntos de datos muy grandes.
- La idea es encontrar en primera instancia muchos clústeres pequeños utilizando un algoritmo eficiente.
- Después se utiliza el centroide de esos clústeres para representar los clústeres, y de ahí ejecutar el clustering jerárquico final.

Clustering Jerárquico

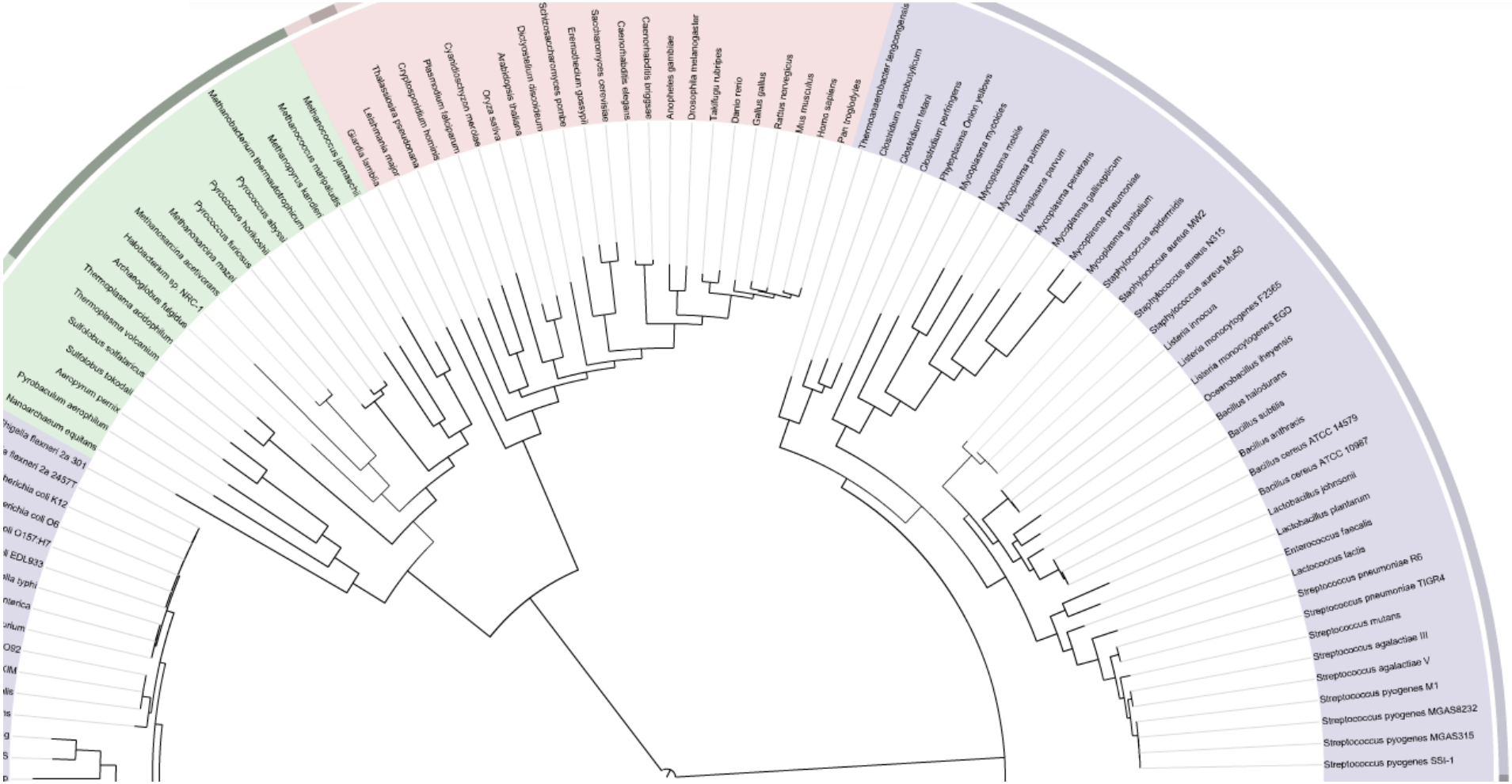
- Permite ver las agrupaciones a diferentes niveles, es decir podemos cortar a diferentes niveles el árbol y ver cómo se van formando los clústeres.
  - Cada nivel utiliza un número diferente de clústeres.
- Agrupa instancias de acuerdo a cómo sus similitudes los conectan entre ellos.
- Los outliers tienden a unirse muy tarde cuando hacemos clustering agglomerativo, entonces se verán aislados hasta niveles muy altos del árbol.

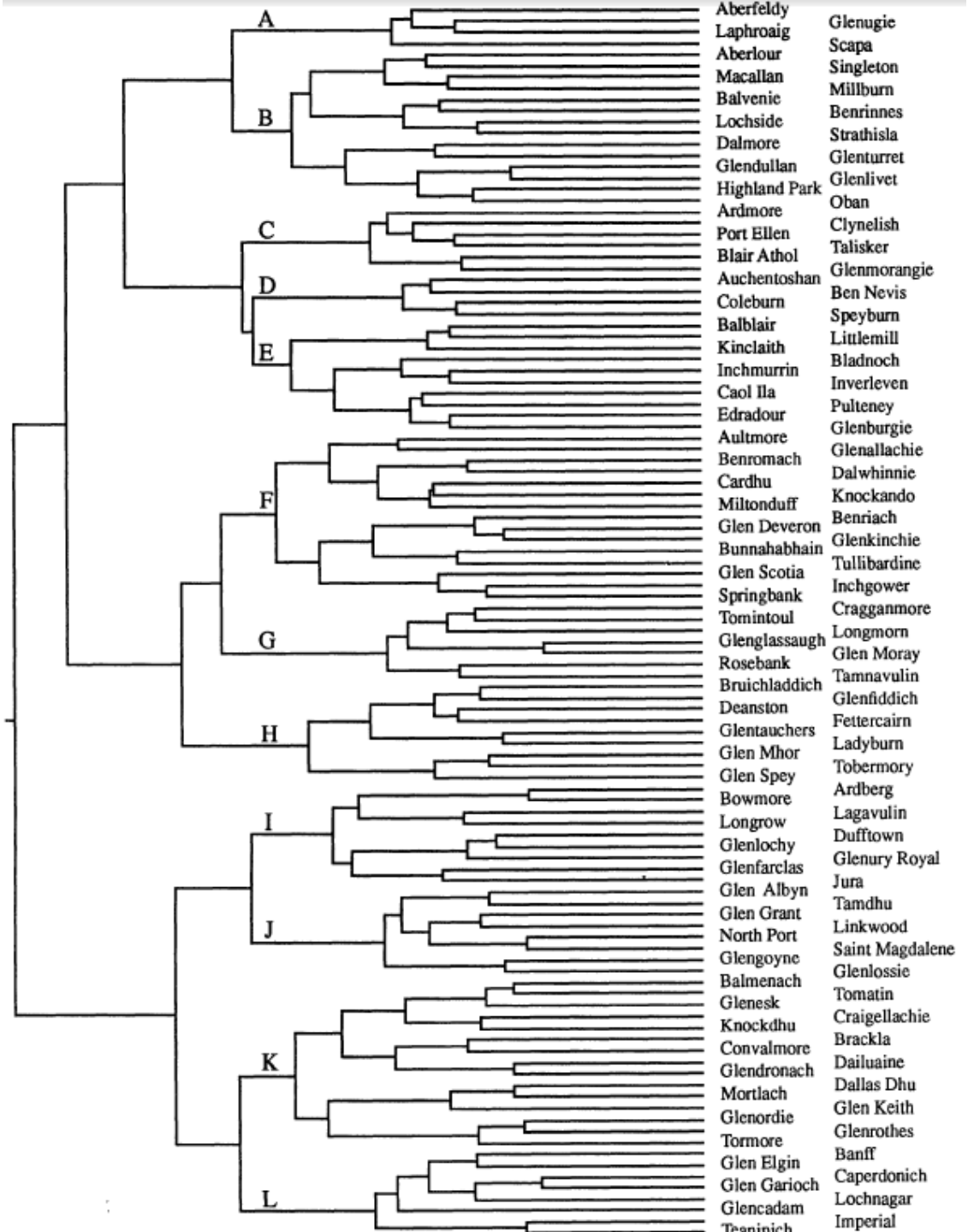
Clustering Jerárquico



http://itol.embl.de/itol.cgi

Clustering Jerárquico





Clustering jerárquico para un grupo de whiskies escoses de malta.

Se puede ver que si se corta el dendograma de manera vertical, se ven los distintos grupos que se forman.

El orden de los nombres es arbitrario.

Datos en:

http://adn.biol.umontreal.ca/~numericalecology/data/scotch.html

Funciones de Distancia

- Para atributos numéricos:
  - Minkowski
  - Euclidean
  - Manhattan
  - Weighted Euclidean
  - Squared Euclidean
  - Chebychev

Funciones de Distancia

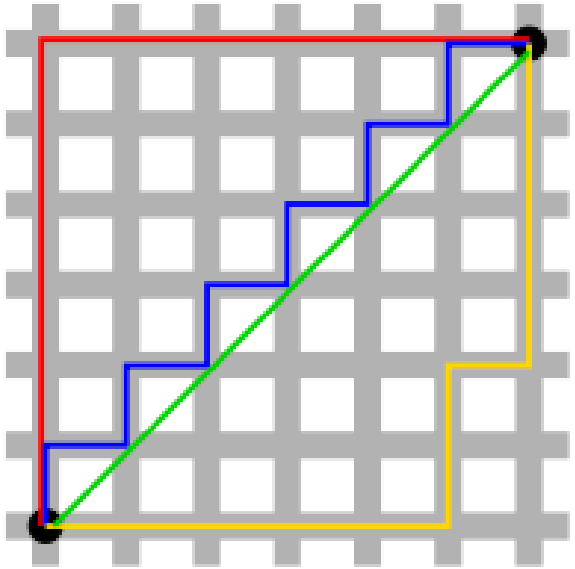
- La distancia de **Minkowski** de dos puntos x_i y x_j (h es un entero positivo) de r-dimensiones:

$$dist(x_i, x_j) = \sqrt[h]{\left|x_{i1} - x_{j1}\right|^h + \ldots + \left|x_{ir} - x_{jr}\right|^h}$$

- Si h=1, tenemos la distancia de **Manhattan**.
- Si h=2, tenemos la distancia **euclidiana**.

Funciones de Distancia

- La distancia Manhattan toma su nombre porque es la distancia que se necesita (en términos de número de cuadras) para viajar desde un punto a otro en una ciudad que tiene forma de grilla como Manhattan o el plano de Viña del Mar.
- La distancia total es una combinación de las cuadras este-oeste y norte-sur caminadas.



Fuente: Wikipedia

Funciones de Distancia

- La distancia **euclidiana con pesos** está dada por :

$$dist(x_i, x_j) = \sqrt[2]{w_1(x_{i1} - x_{j1})^2 + \ldots + w_r(x_{ir} - x_{jr})^2}$$

- Esta distancia le asocia un peso/importancia a cada atributo.
- La distancia **euclidiana al cuadrado**:

$$dist(x_i, x_j) = (x_{i1} - x_{j1})^2 + \ldots + (x_{ir} - x_{jr})^2$$

- La idea es ubicar pesos más grande de manera progresiva en puntos que están más separados.

Funciones de Distancia

- La distancia de **Chebychev**:

$$dist(x_i, x_j) = \max(|x_{i1} - x_{j1}|, \ldots, |x_{ir} - x_{jr}|)$$

- Esta distancia intenta acrecentar el contraste de puntos que tienen cualquier atributo diferente.

Ejemplo

- Si X_1 es (1,3) y X_2 es (5,6), entonces:
  - **Minkowski:**

$$dist(x_1, x_2) = \sqrt[h]{2^h + 1^h}$$

$$h = 1 \Rightarrow \sqrt[1]{4^1 + 3^1} = \sqrt[1]{7} = 7$$

$$h = 2 \Rightarrow \sqrt[2]{4^2 + 3^2} = \sqrt[2]{25} = 5$$

$$h = 3 \Rightarrow \sqrt[3]{4^3 + 3^3} = \sqrt[3]{91} = 9{,}53939201$$

  - Si consideramos w_1=0.1 y w_2=0.2 como los pesos de los features X_1 y X_2 respectivamente, tendríamos como distancia euclidiana con pesos:

$$dist(x_1, x_2) = \sqrt[2]{0.1 * 4^2 + 0.2 * 3^2} = 1{,}84390889$$

Ejemplo

- Si X_1 es (1,3) y X_2 es (5,6), entonces:
  - **Euclidiana al cuadrado:**

$$dist(x_1, x_2) = 4^2 + 3^2 = 25$$

  - La distancia de **Chebychev** sería el máximo entre 4 y 3, es decir 4.
    - Basta con que haya un atributo muy diferente para que la distancia sea grande, aún cuando el resto tengan el mismo valor. Ergo, debe ser usada con cuidado.

Otras Funciones de Distancia

- Otras funciones menos conocidas:
  - Canberra

$$dist(x_i, x_j) = \frac{|x_{i1} - x_{j1}|}{x_{i1} + x_{j1}} + \ldots + \frac{|x_{ir} - x_{jr}|}{x_{ir} + x_{jr}}$$

  - Squared Cord

$$dist(x_i, x_j) = (\sqrt{x_{i1}} - \sqrt{x_{j1}})^2 + \ldots + (\sqrt{x_{ir}} - \sqrt{x_{jr}})^2$$

  - Squared Chi-squared

$$dist(x_i, x_j) = \frac{(x_{i1} - x_{j1})^2}{x_{i1} + x_{j1}} + \ldots + \frac{(x_{ir} - x_{jr})^2}{x_{ir} + x_{jr}}$$

Ejemplo

- Si X_1 es (1,3) y X_2 es (5,6), entonces:
  - **Canberra**:

$$\frac{4}{6}+\frac{3}{9}=\frac{36+18}{54}=\frac{54}{54}=1$$

  - **Squared Cord**:

$$(\sqrt{1}-\sqrt{5})^2+(\sqrt{3}-\sqrt{6})^2=(-1{,}23606798)^2+(-0{,}71743894)^2=0{,}58144723=2{,}04258267$$

  - **Squared Xi-squared**:

$$\frac{4^2}{6}+\frac{3^2}{9}=5,\overline{6}$$

Atributos Heterogéneos

- Si los atributos son numéricos, las fórmulas de distancias se aplican de manera directa.
- Pero cuando los vectores tienen datos de tipos heterogéneos el asunto se puede complicar:

Feature/Atributo	Persona A	Persona B
Género	Masculino	Femenino
Edad	23	40
Años en la residencia actual	2	10
Estado residencial (1-propietario; 2-Inquilino; 3-Otro	2	1
Ingreso	50,000	90,000

Atributos Heterogéneos

- Este ejemplo nos muestra varios problemas con el cálculo de la distancia:
  - El atributo género es simbólico (binario o categórico). Para un atributo binario tal vez sea suficiente usar 0's y 1's, pero no si tiene múltiples valores.
  - La edad puede fluctuar entre 18 y 100, pero el ingreso no. El problema que una diferencia de 10 dólares sería tan significante como 10 años de edad.

Funciones de Distancia

- Otros tipos de atributos:
  - **Binarios**: Por ejemplo, el genero (masculino y femenino). Normalmente no hay una relación de orden. Las métricas se basan en la proporción de coincidencias en los valores: ambos falsos o verdaderos.

	1	0	
1	a	b	$a+b$
0	c	d	$c+d$
	$a+c$	$b+d$	$a+b+c+d$

Funciones de Distancia

- Otros tipos de atributos (binario):
  - **Simétricos**: si ambos estados tienen la misma importancia, por ende tienen el mismo peso. La función de distancia más usada es la **distancia de simple coincidencias**:

$$dist(x_i, x_j) = \frac{b + c}{a + b + c + d}$$

Funciones de Distancia

- Ejemplo:

x1	1	1	1	0	1	0	0
x2	0	1	1	0	0	1	1

$$dist(x_i, x_j) = \frac{2+2}{2+2+1+2} = \frac{4}{7} = 0.57$$

Funciones de Distancia

- Otros tipos de atributos (binario):
  - **Asimétricos**: Si un valor es más importante que el otro. Por convención, se toma como el positivo (1) como el estado más importante o raro. Una distancia que se puede utilizar es la **Jaccard**:

$$dist(x_i, x_j) = \frac{w_1(b+c)}{w_2 a + w_1(b+c)}$$

w_1 y w_2 son pesos que se le pueden dar a las no-coincidencias. En general, ambos son 1.

Funciones de Distancia

- Nótese que también existe el **coeficiente Jaccard** que mide la similitud y no la distancia: a/(a+b+c).
- Para atributos nominales con más de dos estados o valores, la medida de distancia más usada también basada en la distancia de coincidencias simples.
  - Dado dos puntos x_i y x_j, el número de atributos r, y q el número de valores que coinciden entre x_i y x_j:

$$dist(x_i, x_j) = \frac{r - q}{r}$$

Funciones de Distancia

- Supongamos dos vectores de r=4 atributos X_1={0,”naranja”, true, 1} y X_2={1,”naranja”, falso, 0}
  - El valor de “q” entre estos dos vectores es 1, porque sólo el atributo “naranja” coincide.
  - Por consiguiente, el valor de la fracción es ¾.
  - De esta misma forma, la distancia es ¾.

Estandarización de Datos

- La idea de estandarización es que todos los atributos puedan tener igual impacto en el cómputo de la distancia.
- En especial, cuando se usa la distancia euclidiana.
- El objetivo es prevenir que hayan clústeres dominados por atributos con una alta variación.

Estandarización de Datos

- Ejemplo: En un conjunto de dos dimensiones, el rango de un atributo es de cero a uno, mientras el rango del otro atributo es de 0 a 1000. Considere los siguientes data points:
  - X_i: (0.1,20)
  - X_j:(0.9,720)
- La distancia euclidiana entre dos puntos es:

$$dist(x_i, x_j) = \sqrt{(0.9-0.1)^2 + (720-20)^2} = 700.000457$$

Estandarización de Datos

- En el ejemplo vemos que la distancia está casi completamente dominada por (720-20)=700.
- Lo que se hace es estandarizar los atributos, es decir que todos estén en el rango 0 a 1.
- Entonces los valores 20 y 720 se transforman en 0.02 y 0.72.
- La distancia en la primera dimensión se transforma en 0.8 y en la segunda 0.7.
- Por ende, la dist(x_i,x_j)=1.063.

Estandarización de Datos

- Este tipo de fenómeno también afecta a una representación como la de bolsa de palabras.
  - Una representación donde cada palabra representa un atributo, y el valor de ese atributo es la frecuencia en el documento.
- Supongamos que tenemos los siguientes tres documentos:
  - Doc 1={a=100, the=100, ball=5, nadal=10, net=2, win=2}
  - Doc 2={a=95, the=102, ball=7, federer=8, net=3, win=3}
  - Doc 3={a=98, the=102, phone=8, call=5, cost=3}
- En términos de distancia euclidiana, ¿Cuál par está más cerca?

Estandarización de Datos

1		Doc 1	Doc 2	Doc 3	Doc 1 y 2	Doc 1 y 3	Doc 2 y 3
2	a	100	95	98	25	4	9
3	the	100	102	102	4	4	0
4	ball	5	7		4	25	49
5	nadal	10			100	100	0
6	net	2	3		1	4	9
7	win	2	3		1	4	9
8	federer		8		64	0	64
9	phone			8	0	64	64
10	call			5	0	25	25
11	cost			3	0	9	9
12				Suma	199	239	238
13				Raíz Cuadrada	14,10673598	15,4596248	15,4272486

1		Doc 1	Doc 2	Doc 3	Doc 1 y 2	Doc 1 y 3	Doc 2 y 3
2	ball	5	7		4	25	49
3	nadal	10			100	100	0
4	net	2	3		1	4	9
5	win	2	3		1	4	9
6	federer		8		64	0	64
7	phone			8	0	64	64
8	call			5	0	25	25
9	cost			3	0	9	9
10				Suma	170	231	229
11				Raíz Cuadrada	13,03840481	15,1986842	15,132746

Vemos que al eliminar los stopwords, quedamos con vectores que tienen atributos que se mueven en intervalos de valores similares.

Estandarización de Datos

- Atributos que representan escalas en intervalos:
  - Continuos y numéricos.
  - Números reales que siguen una escala.
  - E.g., edad, alto, peso, costo, etc.
- La diferencia en edad oscila 10 y 20 es la misma que entre 40 y 50.
- Hay dos técnicas: **range** y **z-score**.

Estandarización de Datos

- **Range** divide cada valor por el rango de valores válidos de los atributos talque los rangos de los valores transformados estén entre 0 y 1.
- Dado el valor x_{if} del f-ésimo atributo del i-ésimo data point, el nuevo valor $rg(x_{if})$ es,

$$rg(x_{if}) = \frac{x_{if} - \min(f)}{\max(f) - \min(f)}$$

Estandarización de Datos

- Donde min(f) y max(f) son el mínimo y máximo valor posible del atributo.
- Ergo, max(f)-min(f) es el rango de valores válidos del atributo f.
  - ¿Calcular el rango de acuerdo a los valores posibles o los máximos y mínimos vistos en los datos?
- El método de **z-score** transforma el valor de un atributo basado en la media y en la desviación estándar del atributo.
- El **z-score** indica cual lejos y en qué dirección el valor se desvía desde la media del atributo, expresado en unidades de la desviación estándar del atributo.

Estandarización de Datos

- La desviación estándar de f, denotada por σ_f, se calcula mediante:

$$\sigma_f = \sqrt{\frac{\sum_{i=1}^{n}(x_{if} - \mu_f)^2}{n-1}}$$

- Donde n es el número de data points en el conjunto de datos, y u_f es la media del atributo f:

$$\mu_f = \frac{1}{n}\sum_{i=1}^{n} x_{if}$$

Estandarización de Datos

- Entonces, el nuevo valor después de la transformación:

$$z(x_{if}) = \frac{x_{if} - \mu_f}{\sigma_f}$$

- Si consideramos la bolsas de palabras como atributos a intervalos (en estricto rigor no los son, pero sólo con efectos ilustrativos):

1		Doc 1	Doc 2	Doc 3	Min	Max	Range Doc 1	Range Doc 2	Range Doc 3	Media	viación Stanc	Z-score Doc 1	Z-score Doc 2	Z-score Doc 3
2	a	100	95	98	80	105	0,8	0,6	0,72	97,6666667	2,51661148	0,92717265	-1,05962589	0,13245324
3	the	100	102	102	85	110	0,6	0,68	0,68	101,333333	1,15470054	-1,15470054	0,57735027	0,57735027
4	ball	5	7	0	0	15	0,333333333	0,466666667	0	4	3,60555128	0,2773501	0,83205029	-1,10940039
5	nadal	10	0	0	0	15	0,666666667	0	0	3,33333333	5,77350269	1,15470054	-0,57735027	-0,57735027
6	net	2	3	0	0	14	0,142857143	0,214285714	0	1,66666667	1,52752523	0,21821789	0,87287156	-1,09108945
7	win	2	3	0	0	12	0,166666667	0,25	0	1,66666667	1,52752523	0,21821789	0,87287156	-1,09108945
8	federer	0	8	0	0	15	0	0,533333333	0	2,66666667	4,61880215	-0,57735027	1,15470054	-0,57735027
9	phone	0	0	8	0	11	0	0	0,727272727	2,66666667	4,61880215	-0,57735027	-0,57735027	1,15470054
10	call	0	0	5	0	14	0	0	0,357142857	1,66666667	2,88675135	-0,57735027	-0,57735027	1,15470054
11	cost	0	0	3	0	11	0	0	0,272727273	1	1,73205081	-0,57735027	-0,57735027	1,15470054

Estandarización de Datos

- Atributos que representan escalas en radios:
  - Numéricos que toman valores reales.
  - Al contrario de los atributos anteriores, las escalas no son lineales.
  - Por ejemplo los atributos que crecen exponencialmente: Ae^{Bt} (A y B son constantes positivas),

Estandarización de Datos

- Para este tipo de atributo, tenemos dos opciones:
  - Aunque no es recomendable, tratarlo como atributos que representan escalas en intervalos.
  - Desarrollar una transformación logarítmica para cada valor x_{if}, i.e., $\log(x_{if})$. Después de la transformación, el atributo puede ser tratado con un atributo que representa escalas en intervalos.

Estandarización de Datos

- Atributos nominales: el valor puede tomar cualquier estado dentro de un conjunto. Normalmente, no tienen lógica o orden numérico.
  - Por ejemplo el atributo fruta puede tener los siguientes valores posibles: manzana, naranja, y pera. Estos atributos no tienen orden.
- Un atributo binario es un caso especial de atributo nominal con dos estados.

Estandarización de Datos

- Se pueden convertir a un conjunto de atributos binarios.
- Por ejemplo, para el atributo nominal fruta, podemos crear tres atributos binarios: manzana, naranja y pera en los datos nuevos. Si en particular un dato original tiene manzana como fruta, entonces en los datos transformados, se asigna al set al valor manzana como uno, y el valor de los otros atributos un cero.

Estandarización de Datos

- Atributos ordinales son como los atributos nominales.
- Por ejemplo, el atributo edad puede tener valores: joven, mediana edad y viejo.
- Lo común es tratarlos atributos que representan escalas en intervalos.

Evaluación

- Como no tenemos etiquetas, no sabemos los clústeres correctos.
- **Inspección del usuario**: un panel de expertos examina los resultados. Como los expertos difieren frecuentemente, se toma un promedio.
  - Es una tarea que consume mucho tiempo.
  - Consume muchos recursos humanos.
  - Es fácil para algunos tipos de datos, pero no para otros.

Evaluación

- Utilizar un set de datos anotados y que se pueden utilizar para clasificación.
- Entropía: para cada clúster podemos medir su entropía.

$$entropy(D_i) = -\sum_{j=1}^{k} \Pr_i(c_j) \log_2 \Pr_i(c_j)$$

- Donde $Pr_i(c_j)$ es la proporción de data points de la clase c_j en el clúster i o D_i.

$$entropy_{total}(D) = \sum_{i=1}^{k} \frac{|D_i|}{|D|} \times entropy(D_i)$$

Evaluación

- Purity: mide el grado en que un clúster contiene sólo una clase de datos:

$$purity(D_i) = \max_j (\Pr_i(c_j))$$

- La purity total está dada por:

$$purity_{total}(D) = \sum_{i=1}^{k} \frac{|D_i|}{|D|} \times purity(D_i)$$

- También se puede utilizar recall, precisión y F-Score basado en la clase más frecuente de un clúster.

Evaluación

- Ejemplo: Una colección de 900 documentos de tres tipos: ciencias, deportes, y política. Cada clase tiene 300 documentos, y cada uno es etiquetado con uno de los tópicos.
  - Las etiquetas no se usan para el clustering.

Clúster	Ciencia	Deportes	Política	Entropía	Purity
1	250	20	10	0.589	0.893
2	20	180	80	1.198	0.643
3	30	100	210	1.257	0.617
TOTAL	300	300	300	1.031	0.711

Evaluación

2	Clúster	Ciencia	Deporte	Política			$Pr_i(c_j)$			$Pr_i(c_j)*\log_2(Pri(cj))$		Entropía	Purity
3	1	250	20	10	280	0,89285714	0,07142857	0,03571429	-0,14598101	-0,27195392	-0,17169125	0,58962618	0,89285714
4	2	20	180	80	280	0,07142857	0,64285714	0,28571429	-0,27195392	-0,40977638	-0,51638712	1,19811742	0,64285714
5	3	30	100	210	340	0,08823529	0,29411765	0,61764706	-0,30904415	-0,51927493	-0,42935452	1,2576736	0,61764706

Evaluación

- La precisión de los documentos de ciencia en el clúster 1 es 0.89. El recall es 0.83, y la F_1-score es 0.86.

$$\text{Precision}_{\text{ciencia}} = \frac{250}{280} = 0.89285714$$

$$\text{Recall}_{\text{ciencia}} = \frac{250}{300} = 0.8\overline{3}$$

$$F_1\text{-Score}_{\text{ciencia}} = \frac{(1+1^2) * \text{Precision} * \text{Recall}}{1^2 * \text{Precision} + \text{Recall}} = 0.86206897$$

Métricas

- Rand Index: determina el grado de similitud entre las etiquetas dadas U y la solución V generada por el algoritmo de clustering:

$$R = \frac{a+d}{a+b+c+d}$$

- Donde a, b,c, d son calculadas para todos los puntos de datos i y j y sus asignaciones respectivas $C_u(i)$, $C_u(j)$, $C_v(i)$ y $C_v(j)$:
  - $a=|\{i,j|C_u(i)=C_u(j) \wedge C_v(i)=C_v(j)\}|$
  - $b=|\{i,j|C_u(i)=C_u(j) \wedge C_v(i)\neq C_v(j)\}|$
  - $c=|\{i,j|C_u(i) \neq C_u(j) \wedge C_v(i)=C_v(j)\}|$
  - $d=|\{i,j|C_u(i) \neq C_u(j) \wedge C_v(i) \neq C_v(j)\}|$
- R está entre [0,1] y debe ser maximizada.

Métricas

- La varianza interna calcula la suma de las desviaciones cuadradas entre todos los ítems de datos y su centro asociado:

$$I = \sum_{c \in C} \sum_{i \in c} dist(i, \mu_c)^2$$

- Donde C es el conjunto de clústeres, μ_c es el centroide del clúster c, dist(i,μ_c) es la distancia utilizada para calcular la desviación entre cada ítem i y su centro asociado. Hay que minimizarlo.

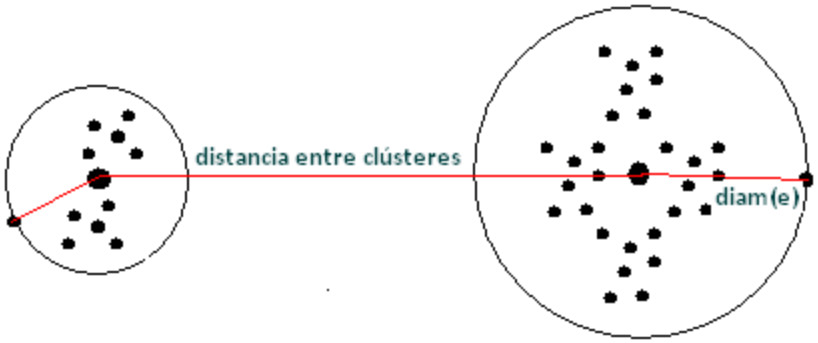
Métricas

- El índice de Dunn determina el ratio mínimo entre el diámetro del clúster y la distancia inter clúster para un particionamiento dado.
  - La idea es que elementos dentro de un clúster deben estar más cerca que aquellos en clústeres diferentes.

$$D = \min_{c,d \in C} \left[\frac{dist(\mu_c, \mu_d)}{\max_{e \in C} diam(e)} \right]$$

- Donde el diámetro diam(c) de un clúster c es calculado como la distancia máxima dentro (interna) de un clúster, dist(μ_d, μ_c) es la distancia entre los centroides de los clústeres c y d. Debe ser maximizada.

Metricas



Hay que buscar el máximo de dim(e) y después el mínimo entre los clústeres.

Fuzzy C-Means

- Es un método que permite a un data-point pertenecer a dos o más clústeres.
- Se basa en la minimización de la siguiente función objetivo:

$$J_m = \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij}^{m} \left\| x_i - c_j \right\|^2, \qquad 1 < m < \infty$$

m es cualquier número real mayor que uno (coeficiente de difusión),u_{ij} es el grado de pertenencia de x_i al clúster j, x_i es un i-ésimo vector d-dimensional, c_j es un centro que también tiene d-dimensiones, y ||*|| es una norma que mide la similitud de un dato a su centro.

Fuzzy C-Means

- El particionamiento fuzzy o difuso es un proceso iterativo que busca optimizar J_m mediante la actualización de los grados de membrecía u_{ij} y los centros c_j:

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}}$$

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^m x_i}{\sum_{i=1}^{N} u_{ij}^m}$$

Fuzzy C-Means

- El proceso iterativo termina cuando:

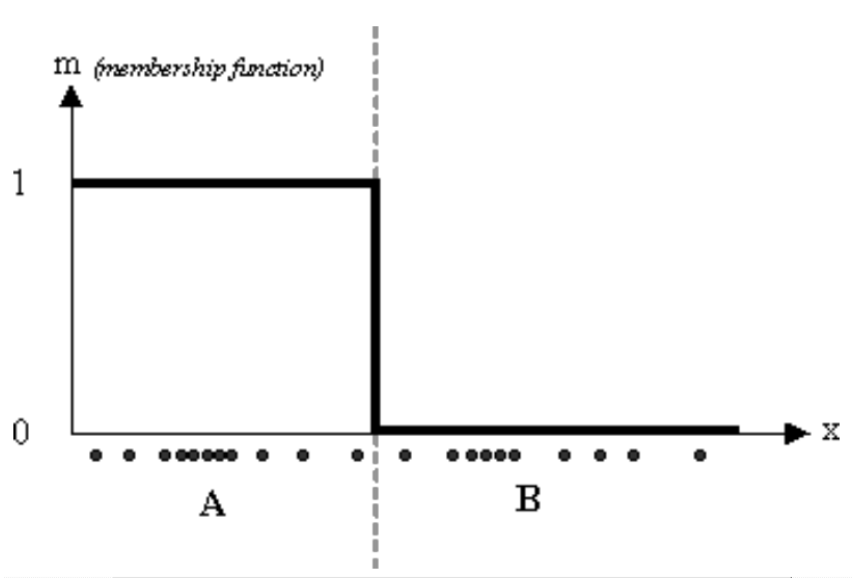
$$\max_{ij}\left\{\left|u_{ij}^{(k+1)} - u_{ij}^{(k)}\right|\right\} < \varepsilon$$

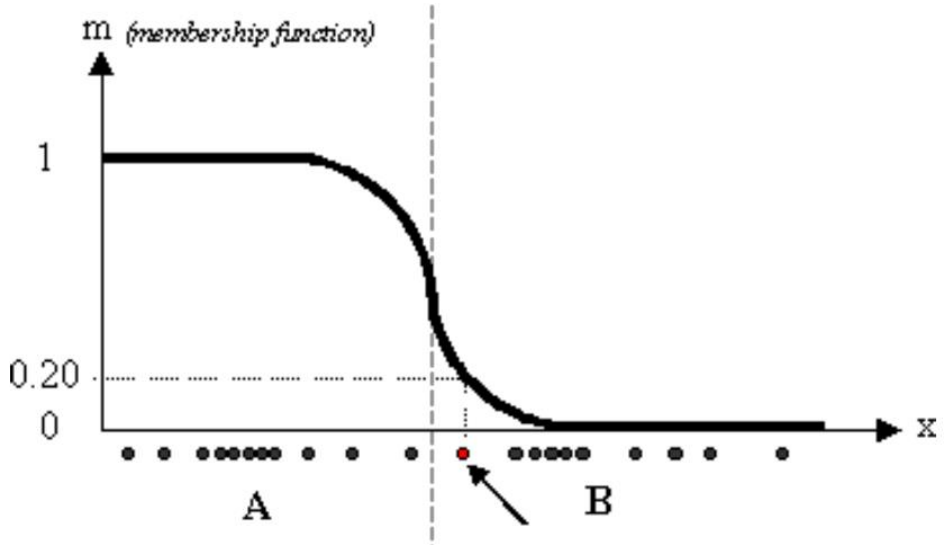
- Donde k es el número de iteraciones.
- El proceso puede converger a un óptimo local.
- La matriz u_{ij} va a tener los valores de membrecía de cada punto x_i a cada clúster c_j.

Fuzzy C-Means

- El algoritmo:
  1. Inicializar $U=[u_{ij}] = U^{(0)}$
  2. Calcular los centros $C^{(k)}=[c_j]$ con $U^{(k)}$
  3. Actualizar $U^{(k)}=U^{(k+1)}$
  4. Si se cumple cualquiera de los dos criterios de termino, terminar, sino volver a 2.

Fuzzy C-Means





K-means vs. FCM: En las funciones de membrecía, K-means tiene 0's y 1's, y FCM grados de membrecía [0,1].

Fuzzy C-Means

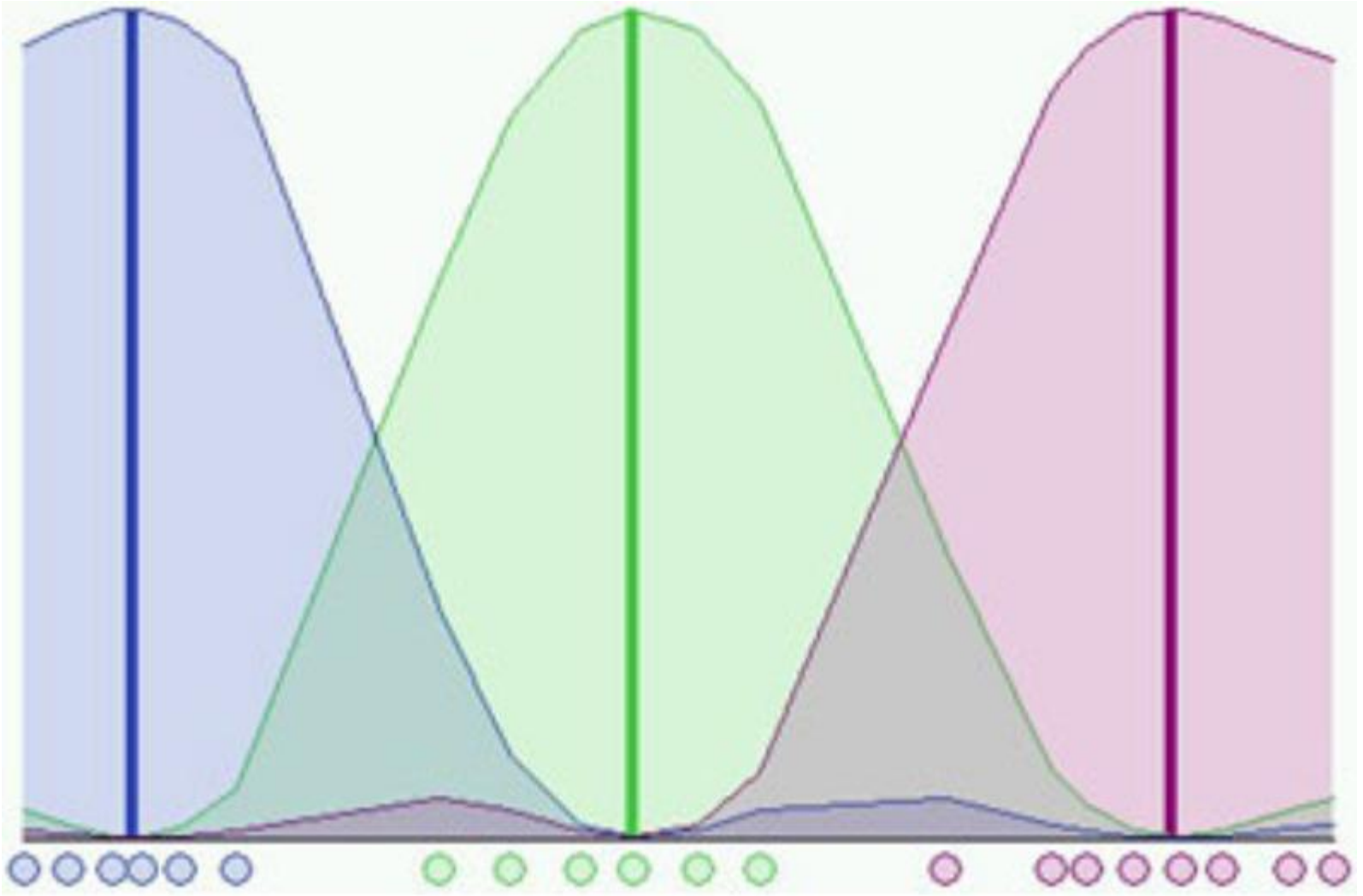
- En FCM, tenemos las siguientes propiedades:

$$u_{ij} \in [0,1] \quad \forall i, j$$

$$\sum_{j=1}^{C} u_{ij} = 1 \quad \forall i$$

$$0 < \sum_{i=1}^{N} u_{ij} < N \quad \forall j$$

Fuzzy C-Means



Clustering con Algoritmo Genéticos

- Uno de los problemas de los algoritmos de clustering tipo K-means es que el valor de “k” es ingresado por el usuario. O bien, varios “k” deben probarse.
- Por lo general, el valor de “k” es desconocido.
- La idea es utilizar algoritmos genéticos para solucionar este problema.
- La solución de K-means también depende del punto de partida.
  - K-means puede converger a un óptimo local.

Clustering con Algoritmo Genéticos

- Hay varias versiones de algoritmos genéticos que apuntan a resolver algunos de estos problemas de clustering.
- Si uno considera el k fijo, podría diseñar un GA donde cada individuo de la población representa los centros de cada uno de los “k” clústers.
- Entonces, uno podría utilizar como función objetivo la suma de las distancias de cada punto al centro del clúster respectivo.
- El objetivo, entonces, sería minimizar esta distancia.

Clustering con Algoritmo Genéticos

- Para este tipo de GA, un individuo sería la concatenación de los valores reales de cada centro.
- Por ejemplo si cada datapoint tiene r=2 features, y tenemos K=3 clústers, el cromosoma sería algo así como la secuencia: 51.6 72.3 18.3 15.7 29.1 32.2, que representa (51.6, 72.3), (18.3, 15.7) y (29.1, 32.2).
- La población inicial puede ser escogida aleatoriamente.

Clustering con Algoritmo Genéticos

- Cada punto entonces se le asigna al clúster más cercano.
- Después se recalculan los centros de los clústers, y estos nuevos valores reemplazan los antiguos.

$$z_i = \frac{1}{n_i} \sum_{x_j \in C_i} x_j$$

Z_i es el centro para el clúster C_i con n_i datos.

Clustering con Algoritmo Genéticos

- En el ejemplo el centro del primer clúster es (51.6, 72.3), consideremos que este clúster contiene dos puntos más: (50.0, 70.0) y (52.0, 74.0).
  - El nuevo centro del clúster sería ((50.0+52.0+51.6)/3, (70.0+74.0+72.3)/3) = (51.2, 72.1)
- El nuevo valor reemplaza al antiguo:
  - Nótese que es necesario reemplazar los centros, ya que los asignados aleatoriamente no necesariamente son los centros exactos. Y necesitamos los centros exactos para calcular el error.

Clustering con Algoritmo Genéticos

- La función objetivo se calcula por clúster.
- Por cada clúster se suma la distancia al centro.
- Nótese que es necesario utilizar el inverso de la suma para que una minimización implica una maximización de la función objetivo.
- Como mecanismo de selección se puede utilizar la selección proporcional, donde a cada individuo se le asigna un número de copias en relación a su fitness.

Clustering con Algoritmo Genéticos

- Para lograr este muestreo se puede utilizar la ruleta. Los individuos seleccionado pasan al “mating pool”.
- Como cruzamiento se puede utilizar el de 1-punto, escogido aleatoriamente.
- Como operador de mutación: se escoge un σ aleatorio entre [0,1] con distribución uniforme. Si el valor del gen “v” es $v \pm 2 * \sigma * v$, si $v \neq 0$, $v \pm 2 * \sigma$, si $v=0$.

Clustering con Algoritmo Genéticos

- Se puede enriquecer el GA con elitismo.
- Y como criterio de término un cierto número de iteraciones.
- En general, se ha demostrado experimentalmente que GA tienen mejor performance que K-means.
- Pero aún no se resuelve el problema principal: ¿Cómo independizarse de “k”?

Clustering con Algoritmo Genéticos

- El problema se puede replantear de diversas formas. Por ejemplo,
  - Sabemos que podemos utilizar meta-heurísticas para re-asignar los centroides.
- Podemos ver que un cromosoma puede codificar no sólo la posición del espacio vectorial sino que además la id del centroide más cercano (clúster).
  - Se podría pensar en agregar otro gen que indique el número de clúster utilizados, y plantear el problema como optimización multi-objetivos.

Clustering con Algoritmo Genéticos

- Una forma estándar es ejecutar el genético con diversos valores de “k”.
  - ¿Dista mucho de ejecutar K-means con diferentes valores de “k”?
- Codificar los centroides y/o los identificadores de clúster aumenta la dificultad del operador de cruzamiento, ya que el orden de los centroides puede ser diferentes, con ids diferentes.
  - Mantener la consistencia es costoso computacionalmente

Clustering con Algoritmo Genéticos

- Otra solución es manejar individuos con diversos números de centroides.
  - En este caso, incluso podrían haber operadores que insertan y eliminar centroides.
  - Se podrían importar y exportar centroides entre soluciones.
- Todas estas soluciones hacen que los GAs sean una buena herramientas para clustering. Pero requiere diseño.
  - ¿Podría un GA también buscar la mejor métrica de distancia?

Clustering con Algoritmo Genéticos

- En general, la selección y la mutación no presentan problemas.
- La función objetivo (minimizar el SSE) puede optimizarse recalculando las partes “nuevas”.
  - Si los centroides no cambian, entonces, no es necesario calcular la función objetivo completa de nuevo.
  - Sin embargo, si cambian los centroides …. Habría que ver si el actual sigue siendo el más cercano.

Tipo	Algoritmo Típico	Formas de clúster	Eliminación de ruido	Datos de alta dimensión	Estándar	Marco algoritmico
Particionamiento	K-means	Convex	Weak	Weak	Distance	Optimization
	PAM	Convex	Weak	Weak	Distance	Optimization
	CLARA	Convex	Strong	Weak	Distance	Optimization
Jerárquico	BIRCH	Arbitrary	Strong	Weak	Distance	Divise
	CURE	Arbitrary	Strong	Moderate	Distance	Aggl.
	ROCK	Arbitrary	Moderate	Strong	Linkage	Aggl.
	Chamaleon	Arbitrary	Moderate	Strong	Linkage	Aggl. / Divise
Densidad	DBSCAN	Arbitrary	Strong	Moderate	Density	Search
	OPTICS	Arbitrary	Strong	Moderate	Density	Search
	DENCLUE	Arbitrary	Very Strong	Very strong	Density	Optimization
Grilla	STING	Arbitrary	Very Strong	Weak	Density	Aggl.
	Wavecluster	Arbitrary	Very Strong	Strong	Density	Search
	CLIQUE	Convex	Moderate	Very strong	Density	Search
Enjambre	Ant Colony	Arbitrary	Strong	Strong	Density	Optimization

Referencias

- "*Web Data Mining: Exploring Hyperlinks, contents, and Usage Data*" by Bing Liu, Second Edition, chapter 4, 2011.
- "*Fast Distance Metric Based Data Mining Techniques Using P-trees: k-Nearest-Neighbor Classification and k-Clustering*", by Abdul Maleq Khan, 2001.
- "A Genetic Approach to the Automatic Clustering Problem", Lin Yu Tseng, Shiueng Bien Yang, 2000.
- "Genetic algorithm-based clustering technique", Ujjwal Maulik and Sanghamitra Bandyopadhyay, 2000.
- "*Genetic K-Means Algorithm*", K. Krishna and M. Narasimha Murty, 1999.

Referencias

- "**Fuzzy C-Means Clustering"**
  - http://home.deib.polimi.it/matteucc/Clustering/tutorial_html/cmeans.html
- "*On the Performance of Ant-based Clustering*", by J. Handl, J.D. Knowles and M. Dorigo, in HIS 2003, pp. 204-213, 2003.
- "*An Analysis of Ant Colony Clustering Methods: Models, Algorithms and Applications*", by G. Zhe, L. Dan, A. Baoyu, O. Yangxi, C. Wei, N. Xinxin, and X. Yang, in International Journal of Advancements in Computing Technology(IJACT), Vol. 3,N. 11, 2011.
- http://en.wikipedia.org/wiki/Taxicab_geometry (Visitado 9 de septiembre 2014)

Referencias

- “Data Science for Business”, Foster Provost and Tom Fawcett, ISBN: 978-1-449-36132-7, 2013.
- “A Classification of Pure Malt Scotch Whiskies”, Francois-Joseph Lapointe and Pierre Legendre, Applied Statistics, Vol. 43, No. 1, pp. 237-257, 1994.