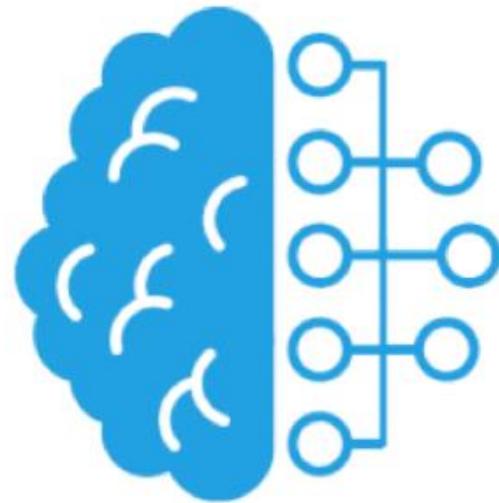

Aprendizaje de Máquina

Clase 5

Claudio Delrieux – DIEC - UNS
cad@uns.edu.ar



Clustering: reconocimiento de patrones

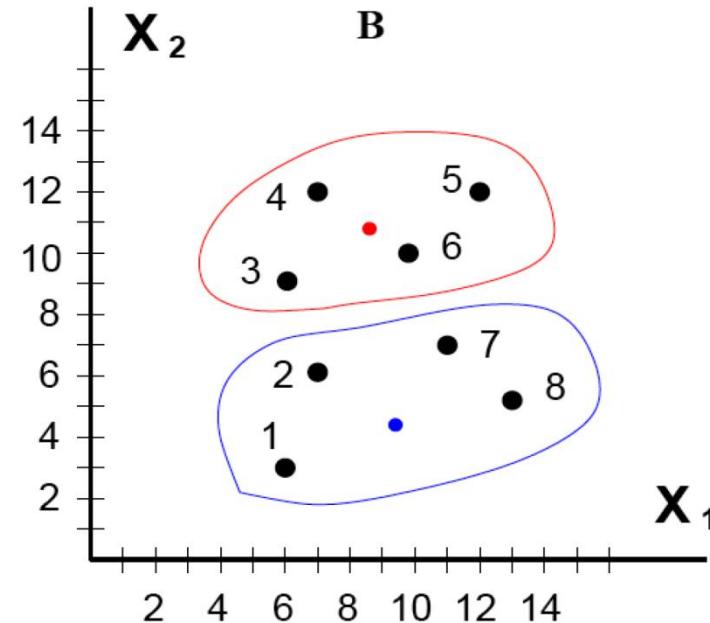
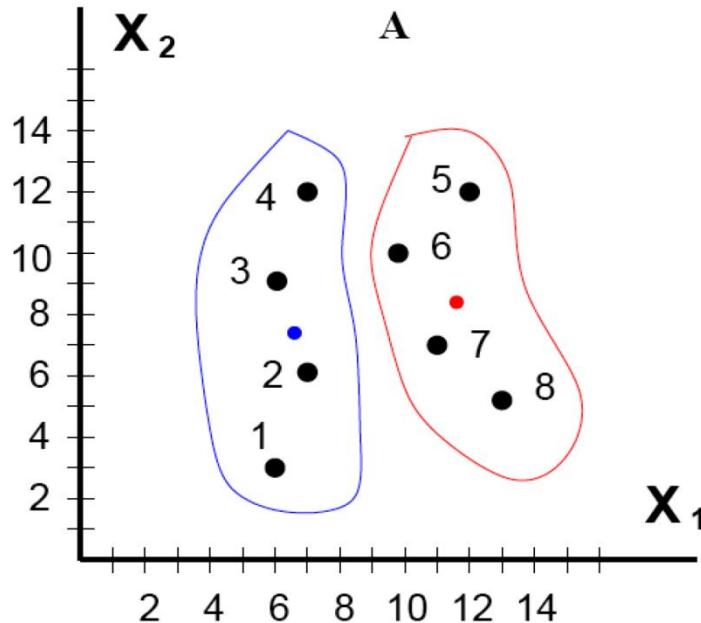
Se conoce como «aprendizaje no supervisado», porque las categorías o clases subyacentes es desconocida (o inexistente).

El objetivo consiste en agrupar el conjunto de registros (tradicionalmente denominados *patrones*) en agrupamientos (o clases), por algún criterio particular (similitud, proximidad, etc.). La similitud o proximidad se determinan por distancia Euclídea, Mahalanobis, etc.

Atributos nominales requieren tratamiento especial.

El espacio de búsqueda es enorme: $m^k/m!$ para k patrones en m clases.

Clustering: reconocimiento de patrones



Clustering: reconocimiento de patrones

Si Z_i es el centro del agrupamiento S_i , calculado como $Z_i = \frac{1}{N_i} \sum_{X \in S_i} X$

la **suma de los errores al cuadrado, SSE** $J_e = \sum_{i=1}^K \sum_{X \in S_i} \|X - Z_i\|^2$

$$S_1 = \{X_1, X_2, X_3, X_4\} \quad S_2 = \{X_5, X_6, X_7, X_8\}$$

$$Z_1 = (6.5, 7.5) \quad Z_2 = (11.5, 8.5)$$

$$J_e = \sum_{X \in S_1} \|X - Z_1\|^2 + \sum_{X \in S_2} \|X - Z_2\|^2 = 23.27$$

$$S_1 = \{X_1, X_2, X_7, X_8\} \quad S_2 = \{X_3, X_4, X_5, X_6\}$$

$$Z_1 = (9.25, 5.25) \quad Z_2 = (8.75, 10.75)$$

$$J_e = \sum_{X \in S_1} \|X - Z_1\|^2 + \sum_{X \in S_2} \|X - Z_2\|^2 = 22.9$$

Clustering: k-medias

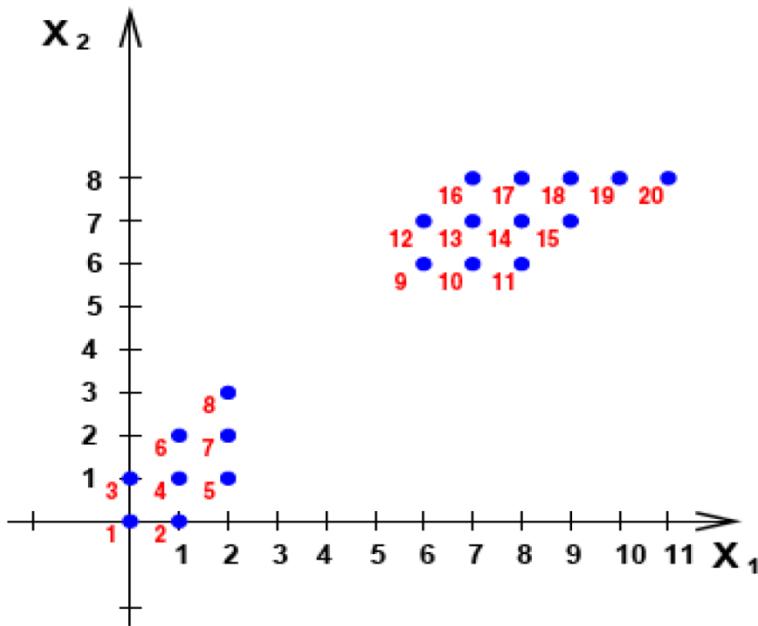
Es el algoritmo más difundido. Se basa en iterar asignaciones de centroides a las clases, seguidas de una etapa de ajuste. Es un algoritmo *aglomerativo*, con k conocido o asignado arbitrariamente.

Mientras algún patrón cambie de grupo repetir:

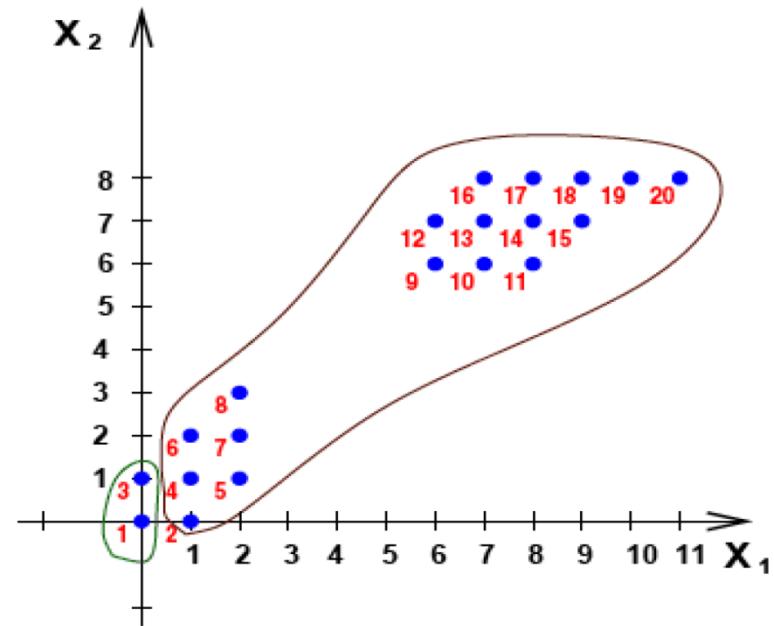
1. Asignar cada patrón al grupo cuyo centroide sea más cercano
2. Recalcular los centroides de cada grupo

Existen teoremas que muestran que este algoritmo converge siempre, aunque depende de las asignaciones iniciales.

Clustering: k-medias



Situación inicial ($k=2$, asignamos centroides a 1 y 2)



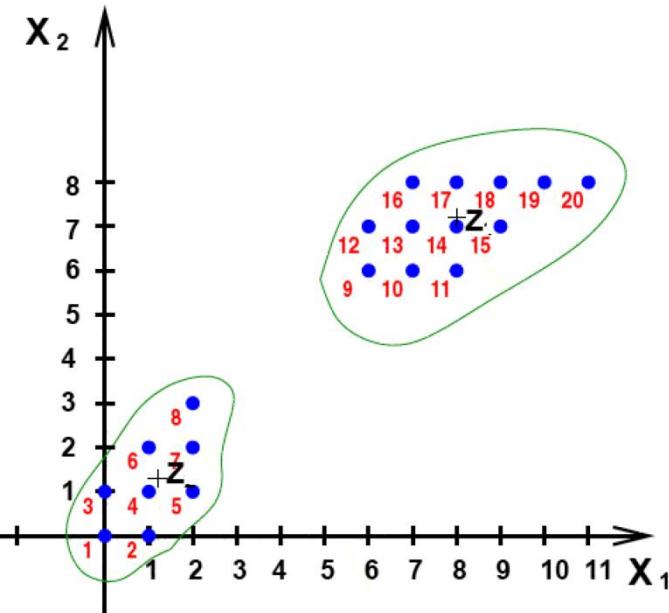
Situación luego de paso 1.

Clustering: k-medias

Luego de la primera asignación, los datos 1 y 3 van al primer centroide, los demás al segundo (el dato 2).

Al recalcular, los nuevos centroides se ubican (resp.) en $(0, 0.5)$ y $(5.8, 5.3)$. Eso hace que algunos datos cambien de grupo (2, 4..8 se van al primer grupo).

Los nuevos grupos tienen diferentes centroides, pero no se generan nuevos cambios: la solución es estable.



Clustering: k-medias

Algunos criterios de evaluación de la calidad del clustering:

<code>metrics.adjusted_mutual_info_score(...[, ...])</code>	Adjusted Mutual Information between two clusterings.
<code>metrics.adjusted_rand_score(labels_true, ...)</code>	Rand index adjusted for chance.
<code>metrics.calinski_harabasz_score(X, labels)</code>	Compute the Calinski and Harabasz score.
<code>metrics.davies_bouldin_score(X, labels)</code>	Computes the Davies-Bouldin score.
<code>metrics.completeness_score(labels_true, ...)</code>	Completeness metric of a cluster labeling given a ground truth.
<code>metrics.cluster.contingency_matrix(...[, ...])</code>	Build a contingency matrix describing the relationship between labels.
<code>metrics.fowlkes_mallows_score(labels_true, ...)</code>	Measure the similarity of two clusterings of a set of points.
<code>metrics.homogeneity_completeness_v_measure(...)</code>	Compute the homogeneity and completeness and V-Measure scores at once.
<code>metrics.homogeneity_score(labels_true, ...)</code>	Homogeneity metric of a cluster labeling given a ground truth.
<code>metrics.mutual_info_score(labels_true, ...)</code>	Mutual Information between two clusterings.
<code>metrics.normalized_mutual_info_score(...[, ...])</code>	Normalized Mutual Information between two clusterings.
<code>metrics.silhouette_score(X, labels, *[..., ...])</code>	Compute the mean Silhouette Coefficient of all samples.
<code>metrics.silhouette_samples(X, labels, *[..., ...])</code>	Compute the Silhouette Coefficient for each sample.
<code>metrics.v_measure_score(labels_true, ..., [beta])</code>	V-measure cluster labeling given a ground truth.



Clustering: evaluación intrínseca

Coeficiente de silueta: No utiliza etiquetas, evalúa un cociente que tiene en cuenta la distancia media intra-cluster y las distancias de cada patrón del cluster a patrones de otros clusters.

Variancia explicada: mide cuánta varianza de los datos originales es explicada por los clusters.

Inercia: La media de las distancias cuadráticas de cada patron al centroide de su cluster.

Índices Davies-Bouldin, Dunn, Calinski: Evalúan la compactación, dispersión, relaciones de distancia, etc. entre clusters.

Clustering: evaluación extrínseca

Homogeneidad: Si tenemos etiquetas, entonces la homogeneidad evalúa la “pureza” de las etiquetas de los patrones que fueron a cada cluster.

Compleitud: Si tenemos etiquetas, la completitud evalúa si todos los patrones de una misma etiqueta fueron a un mismo cluster.

Valor-V: La media armónica entre homogeneidad y completitud.

Clustering: evaluación extrínseca

Esta forma de evaluar clustering es muy útil cuando tenemos “algunos” datos etiquetados y queremos inducir un etiquetado completo (esta es una posible instancia del aprendizaje semi-supervisado).

La homogeneidad es “análoga” a la especificidad (un cluster más homogéneo es más “escéptico” a la hora de aceptar un registro, y la completitud es “análoga” a la sensibilidad (un cluster más completo es más “crédulo”).

Ambas magnitudes están “en pugna” y el valor V evalúa el punto de equilibrio.

Clustering: evaluación extrínseca

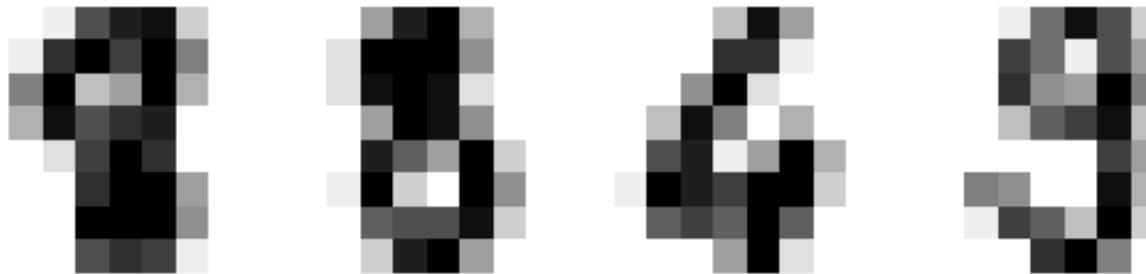
Indice Rand (ajustado): Es una medida de consenso entre dos etiquetados (en este caso entre el dato y el clustering). Evalúa las coincidencias luego de eliminar las que podrían haber ocurrido por azar.

Información mútua: Utiliza la entropía de Shannon para evaluar la similitud entre conjuntos (el dato y el clustering).

Medida Fowlkes-Mallows (FMI): Mide la geometría de las relaciones de similitud entre los puntos de datos y las etiquetas verdaderas. Puede ayudar a evaluar la calidad de los clusters en función de la similitud y la disimilitud.

Clustering: k-medias

En https://mybinder.org/v2/gh/scikit-learn/scikit-learn/0.23.X?urlpath=lab/tree/notebooks/auto_examples/cluster/plot_kmeans_digits.ipynb tenemos un ejemplo ya conocido:



Clustering: Mean Shift

Es un algoritmo para inducir modas locales en un espacio de atributos continuo. Alrededor de cada patrón se evalúa la media de los registros de un determinado entorno, y luego el valor del patrón es asignado a esa media (i.e., tenemos una estructura intermedia donde los valores de los patrones van cambiando).

El proceso se itera hasta llegar a un punto fijo (las modas locales).

Luego a cada patrón original se le asigna como cluster el que corresponde a su moda más cercana.

Clustering: Mean Shift

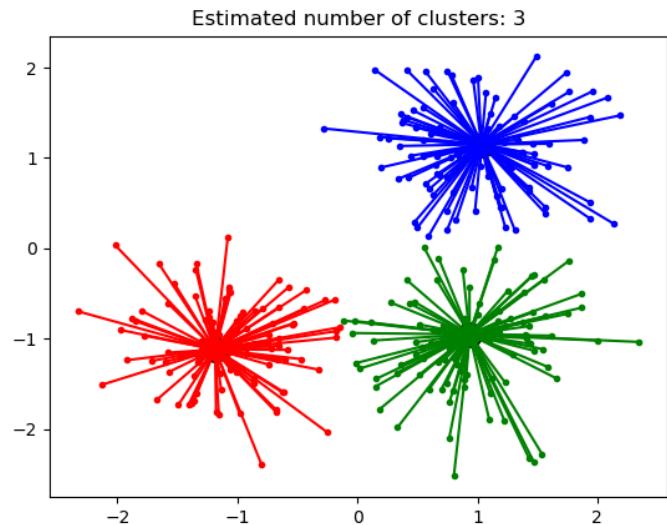
En https://github.com/manlio99/Materia-de-aprendizaje/blob/master/5_DataMining/3_clustering.ipynb podemos ver un ejemplo de clustering del espacio de colores RGB utilizando mean shift.



Clustering: Propagación

El método de *affinity propagation* se basa en enviar mensajes entre pares de patrones que intentan agruparse juntos, hasta llegar a una convergencia. Durante esta iteración se atenúa la posibilidad de interacción para evitar oscilaciones.

Esto permite controlar la cantidad de datos totales que representan al cluster (parámetro de *preferencia*).



Clustering Jerárquico

Es un método *divisivo*, basado en un único parámetro θ (fracción de distancia).

1. Arbitrariamente elegir el primer patrón x_1 como centro del primer agrupamiento Z_1 .
2. Determinar el patron x_i más alejado de Z_1 y denominarlo Z_2 .
3. Computar y guardar los pares de distancias $\delta(x_j, Z_k)$ para todos los restantes patrones a los centros de las clases.
4. Elegir el máximo δ entre las distancias de cada patrón a centro del agrupamiento más cercano a dicho patrón.
5. Si δ es mayor que θ por la mayor de la distancia entre los centros de las clases, se crea una clase nueva con este patrón. Caso contrario, el algoritmo para con las clases cuyos centros ya se computaron.
6. Retornar al paso 3.

Clustering Jerárquico

El primer patrón forma el centro del primer agrupamiento:

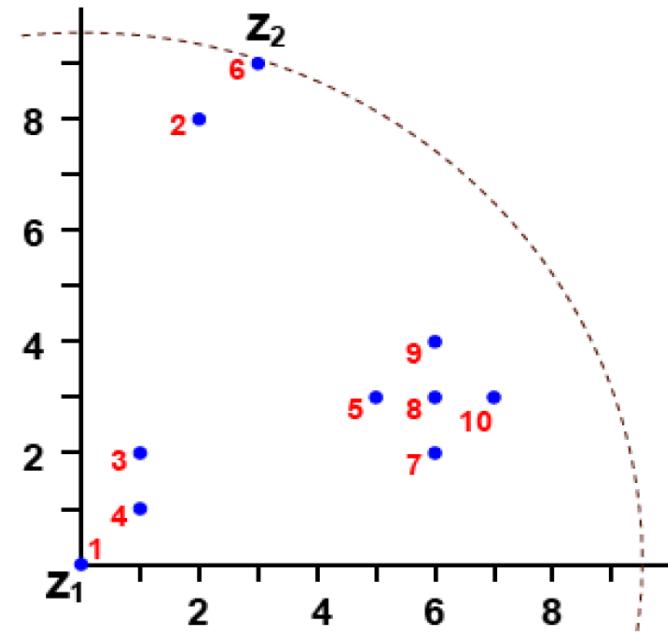
$$Z_1 = X_1 \quad A = 1 \quad S_1 = \{X_1\}$$

y en consecuencia, $L = \{X_2, X_3, \dots, X_{10}\}$. Entre éstos seleccionamos X_6 porque es el más alejado de Z_1

$$\delta(X_6, Z_1) = 9.5 = \max_{X_i \in L} \{\delta(X_i, Z_1)\}$$

Así, X_6 forma el centro del segundo agrupamiento:

$$Z_2 = X_6 \quad A = 2 \quad S_2 = \{X_6\} \quad L = \{X_2, \dots, X_5, X_7, \dots, X_{10}\}$$



Clustering Jerárquico

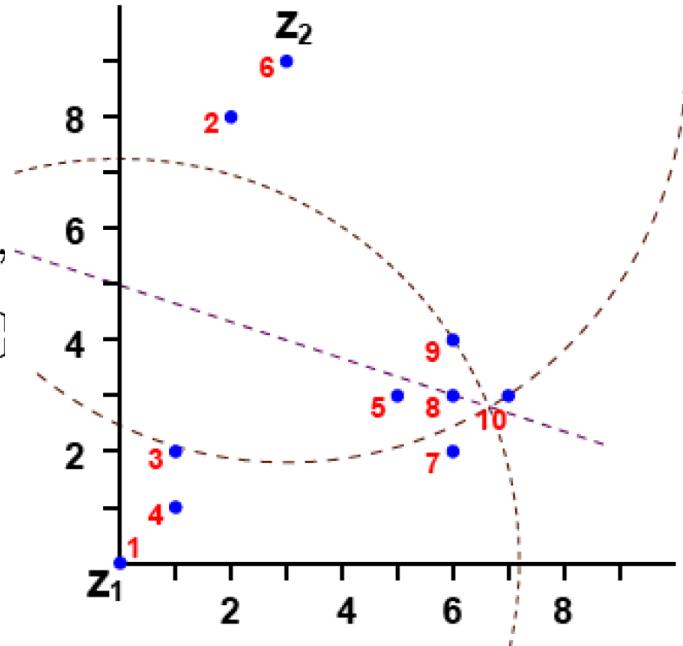
Paso 2. ¿Crear agrupamientos?

En primer lugar construimos el conjunto T :

$$T = \{(X_2, Z_2), (X_3, Z_1), (X_4, Z_1), (X_5, Z_1), \\ (X_7, Z_1), (X_8, Z_1)^*, (X_9, Z_2), (X_{10}, Z_2)\}$$

y sobre éste seleccionamos la pareja (X_{10}, Z_2) porque

$$\delta(X_{10}, Z_2) = 7.2 = \max_{(X, Z) \in T} \{\delta(X, Z)\}$$



Clustering Jerárquico

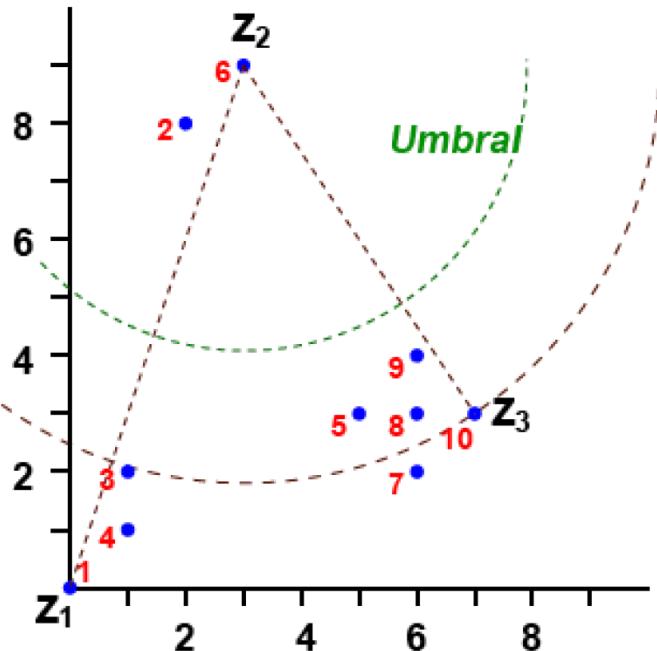
A continuación se calcula el valor umbral considerando Z_1 , Z_2 y θ . En este caso, $umbral = 4.75$ ya que:

$$umbral = 0.5 \frac{\delta(Z_1, Z_2)}{1} = 0.5 \times 9.5 = 4.75$$

Ahora, como $\delta(X_{10}, Z_2) = 7.2 > umbral = 4.75$
se crea un nuevo agrupamiento cuyo centro es X_{10} :

$$Z_3 = X_{10} \quad A = 3 \quad S_3 = \{X_{10}\} \quad L = \{X_2, \dots, X_5, X_7, X_8, X_9\}$$

y se fuerza a otra iteración



Clustering Jerárquico

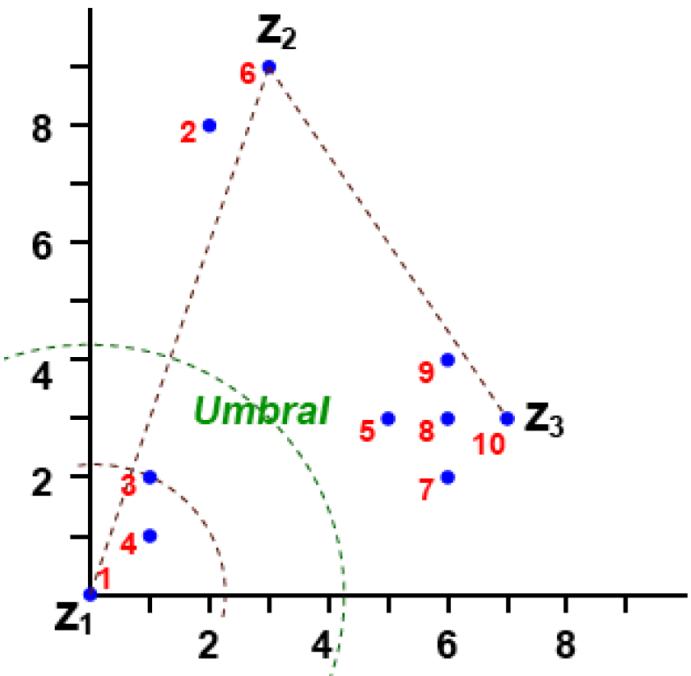
$$\mathbf{T} = \{(X_2, Z_2), (X_3, Z_1), (X_4, Z_1), (X_5, Z_3), \\ (X_7, Z_3), (X_8, Z_3), (X_9, Z_3)\}$$

y sobre éste seleccionamos la pareja (X_3, Z_1) porque

$$\delta(X_3, Z_1) = 2.3 = \max_{(X, Z) \in \mathbf{T}} \{\delta(X, Z)\}$$

$$umbral = 0.5 \frac{\delta(Z_1, Z_2) + \delta(Z_2, Z_3)}{2} = 0.5 \frac{9.5 + 7.2}{2} = 4.2$$

Ahora, como $\delta(X_3, Z_1) = 2.3 \nless umbral = 4.2$
no se puede crear un nuevo agrupamiento, por lo que se termina el algoritmo de creación de clases.

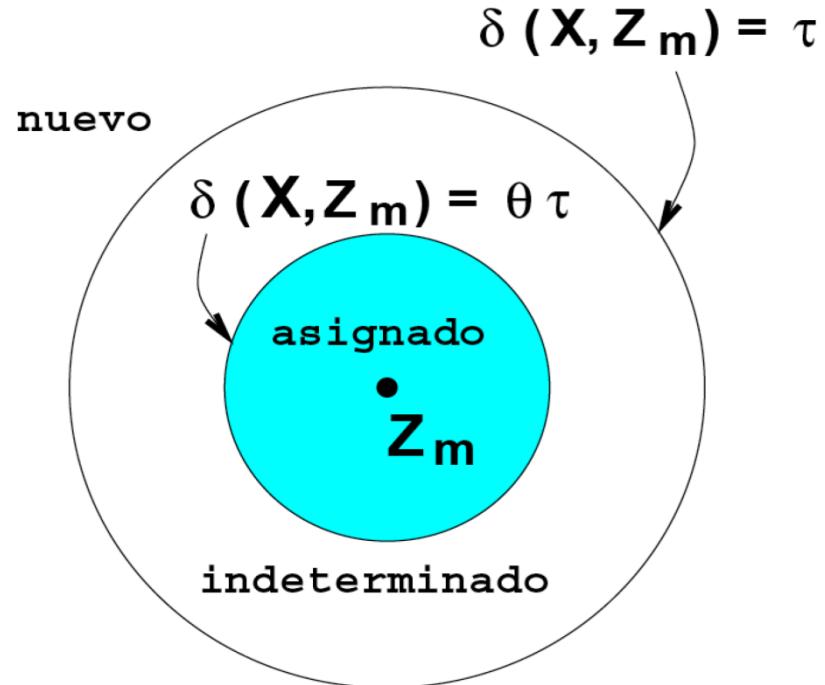


Clustering: Algoritmo de distancia adaptativa

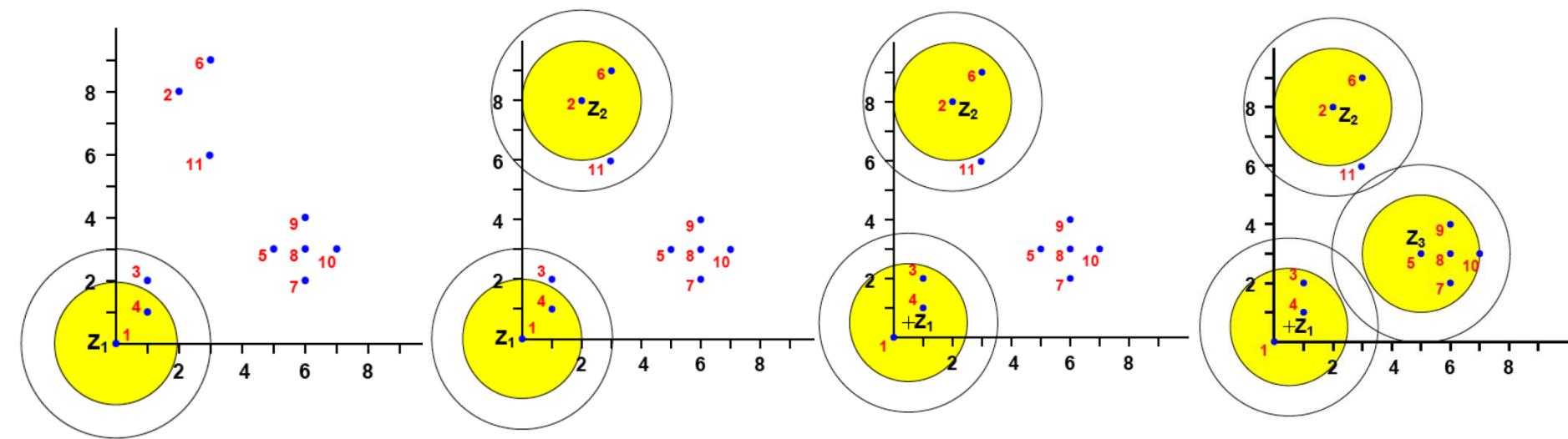
Es un método aglomerativo adaptativo, utiliza dos parámetros τ , θ .

A partir de un centroide arbitrario, los patrones que están a mayor q generan un nuevo grupo, los que están a $\theta\tau$ pertenecen “debilmente”.

Nuevos grupos cambian la asignación, y se recalculan los centroides, hasta llegar a una asignación estable.



Clustering: Algoritmo de distancia adaptativa



Evolución de los grupos y centroides agregando los patrones de a uno.

Clustering: Algoritmo ISODATA

Algoritmo aglomerativo-divisivo. Elimina grupos poco numerosos, separa grupos dispersos, une grupos cercanos. Requiere varios parámetros (cantidad máxima de iteraciones y de clases, distancias umbral para unir o para dividir grupos, etc.):

1. Elegir arbitrariamente k centroides
2. Asignar todos los patrones al centroide más cercano
3. Eliminar grupos poco numerosos (y reasignar sus patrones)
4. Si un grupo tiene SSE por encima del valor umbral, se divide
5. Recalcular los centroides
6. Si dos centroides están a menos de la distancia umbral, se unen los grupos

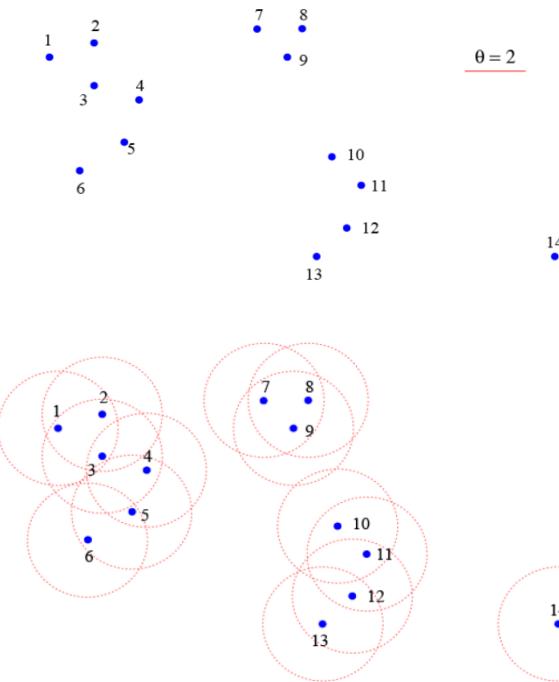
Clustering: Algoritmos basados en grafos

Se genera una matriz binaria de adyacencia de $k \times k$ (k =cantidad de patrones)

Dada una distancia θ , si dos patrones i y j están a menor distancia en el espacio de atributos, entonces los elementos (i,j) y (j,i) en la matriz son 1, caso contrario son 0.

Los grupos se forman encontrando sub-matrices cuadradas no separables dentro de la matriz de adyacencia.

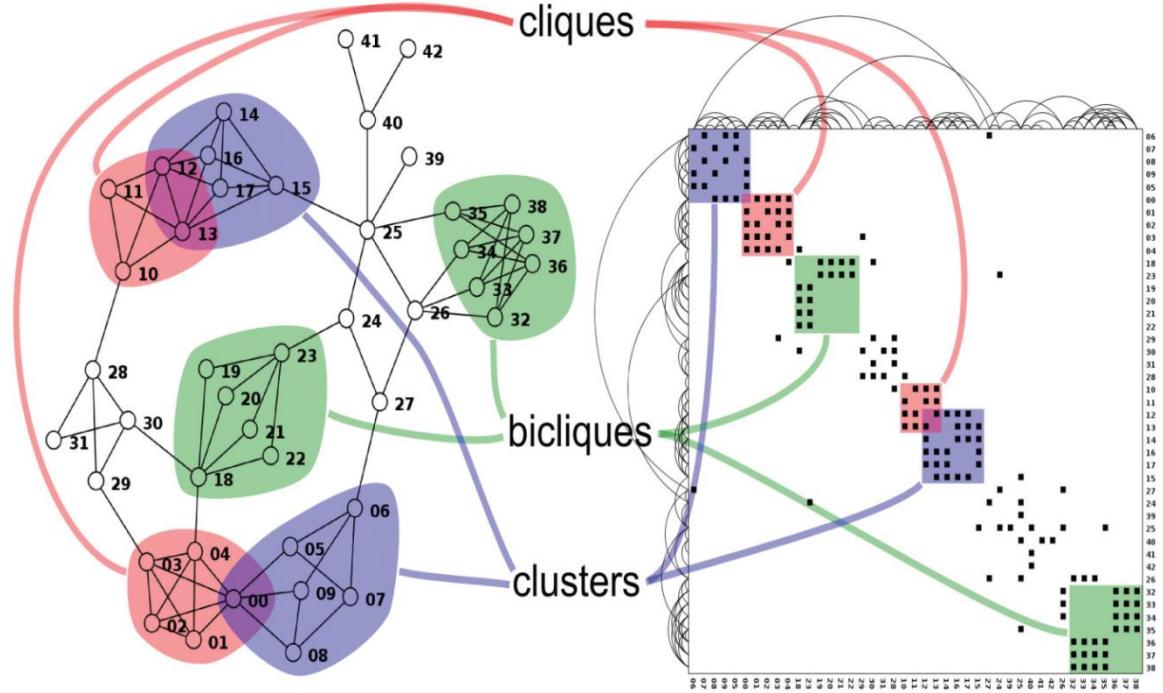
Clustering: Algoritmos basados en grafos



	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
2	1	1	1	0	0	0	0	0	0	0	0	0	0	0
3	1	1	1	1	0	0	0	0	0	0	0	0	0	0
4	0	0	1	1	1	0	0	0	0	0	0	0	0	0
5	0	0	0	1	1	1	0	0	0	0	0	0	0	0
6	0	0	0	0	1	1	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	1	1	1	0	0	0	0	0
8	0	0	0	0	0	0	0	1	1	1	0	0	0	0
9	0	0	0	0	0	0	0	1	1	1	0	0	0	0
10	0	0	0	0	0	0	0	0	0	1	1	0	0	0
11	0	0	0	0	0	0	0	0	0	1	1	1	0	0
12	0	0	0	0	0	0	0	0	0	0	1	1	1	0
13	0	0	0	0	0	0	0	0	0	0	1	1	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Clustering: Algoritmos basados en grafos

En aplicaciones específicas es importante analizar la matriz de adyacencia en búsqueda de grupos con otros tipos de propiedades.



Clustering:más modelos y ejemplos

En <https://scikit-learn.org/stable/modules/clustering.html#> se pueden ver todos los métodos de clustering disponibles en scikit-learn y en https://scikit-learn.org/stable/auto_examples/index.html#clustering una batería de ejemplos.

Práctico 5

Elegir uno de entre los siguientes proyectos **sugeridos** y aplicar clustering para resolverlo. Cada proyecto tiene datasets de diferentes características y dificultad (indicada con un semáforo).



Para los más sencillos trabajar individualmente y aplicar más de un método de clustering, utilizar diferentes parámetros, comparar evaluaciones. Para los más complejos, trabajar en grupos de dos personas y aplicar un único modelo.

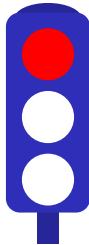
Si encuentran algún otro caso de clústering (que no esté totalmente resuelto) lo pueden proponer para negociarlo con la cátedra.

Ejercicio 5.1

Sommelier de cerveza: El dataset contiene 1.5 M reportes de cervezas scrappeados del sitio [BeerAdvocates](#). El objetivo es agrupar los tipos de cervezas de acuerdo a dichos reportes, y elegir una de ellas como la mejor representante de cada grupo.

Otras posibles preguntas: (i) Si tuviésemos que elegir las N más recomendables, cuáles serían y por qué? (ii) Cuáles son los factores que más influyen en la calidad de una cerveza?

https://github.com/manlio99/Materia-de-aprendizaje/tree/master/3_MidtermProjects/ProjectBEE



Ejercicio 5.2

Segmentación de clientes: El objetivo es encontrar segmentos (grupos) de clientes minoristas en función de su patrón de consumo anual en diversas categorías de productos. https://github.com/manlio99/Materia-de-aprendizaje/tree/master/3_MidtermProjects/ProjectWC



Ejercicio 5.3

Crímenes violentos en los estados de EEUU: El dataset contiene registros de crímenes ocurridos durante 1973 en EEUU en diferentes estados, así como el porcentaje de la población que vive en zonas urbanas en dichos estados. El objetivo es agrupar a estos estados en diferentes categorías de acuerdo a estos factores.



https://github.com/manlio99/Materia-de-aprendizaje/tree/master/3_MidtermProjects/ProjectUSA



Ejercicio 5.4

Chatbot: El objetivo consiste en crear un chatbot utilizando patrones de entrada y respuestas predefinidos.

https://github.com/manlio99/Materia-de-aprendizaje/tree/master/3_MidtermProjects/ProjectPCB

