ESTUDIO DE TÉCNICAS DE CLUSTERING

Raúl Giráldez y Roberto Ruiz

Departamento de Lenguajes y Sistemas Informáticos
Facultad de Informática y Estadística
Universidad de Sevilla

Informe Técnico LSI-2000-14
Diciembre, 2000

Índice de Contenidos

Índice de Ilustraciones

1. INTRODUCCIÓN

Las técnicas de clustering son técnicas de clasificación no supervisada de patrones en conjuntos denominados clusters. El problema del clustering ha sido abordado por gran cantidad de disciplinas y es aplicable a una gran cantidad de contextos, lo cual refleja su utilidad como uno de los pasos en el análisis experimental de datos. Sin embargo, el clustering es un problema complejo, y diferencias en las hipótesis y contextos en los distintos colectivos de han hecho que su desarrollo sea más lento de lo esperado. Este trabajo presenta una visión global de los distintos métodos de clustering así como las distintas aplicaciones de conceptos relacionados con este entorno, proporcionando información y referencias de conceptos de gran utilidad para la su aplicación en cualquier campo.

El *CLUSTERING* o agrupamiento en clases puede definirse de forma general como el proceso de clasificación no supervisada de objetos. Si al proceso se le indican el número de clases o conjuntos diferentes en que se pueden agrupar los objetos, se le facilita el trabajo. Sin embargo, hay ciertos casos en los que esta información no es necesaria, con lo que tiene un valor meramente informativo.

Al principio del problema no se conoce la distribución de los objetos en las distintas clases. Se dispone de un conjunto de vectores $\{x_1, ..., x_p\}$ denominados *patrones* que representan a los objetos y a partir de este sistema de vectores se desea obtener el conjunto de clases que los engloban a estos patrones de manera que todos los individuos pertenecientes a una misma clase, presentan cualidades similares. De esta forma, los objetos se clasifican de tal forma que cada uno de ellos queda identificado por la clase a la que pertenece. El problema consiste en que no se sabe a priori la distribución de los individuos en las clases, ni siquiera es conocido el número de clases que se formará al final del proceso. Sin embargo, se dispone de información perfectamente definida de las características de los objetos, así como de la estructura de los patrones que contendrán estas características (vectores de características). Con esta información, podemos decir que el problema consiste en conseguir agrupar el conjunto de vectores de características dado en clases en función de las similitudes encontradas entre ellos.

Como veremos, el concepto de similitud o proximidad de vectores tiene diferentes interpretaciones. El agrupamiento se efectúa en función de esta similitud o diferencia de patrones, basadas en las distancias calculadas a partir de los valores de las características para cada patrón. Normalmente se utiliza la distancia euclídea o alguna variante de ésta, aunque existen otras.

Loa algoritmos de clustering los podemos encuadrar dentro de los algoritmos heurísticos y se suelen basar en la minimización de algún índice, por ejemplo, índices cuadráticos basados en distancias, que veremos con detalle.

Por otro lado, la aplicación de las técnicas de clustering para la clasificación de un conjunto de individuos u objetos es apropiada para áreas de trabajo en las que exista poca información sobre la distribución de estos individuos en clases diferentes. De

hecho, en aplicaciones donde el conocimiento es prácticamente nulo son imprescindibles. Otra aplicación interesante de las técnicas de clustering la encontramos en aquellos campos en los que existe conocimiento completo de la clasificación de los objetos, donde la utilidad del clustering radica en probar la calidad de las características escogidas.

En este trabajo se describen algunas de las técnicas de clustering más importantes, constituyendo también una guía de referencia de esta técnicas.

Componentes de una tarea Clustering

Los pasos de una tarea de clustering típica se pueden resumir en cinco pasos siguientes [Jain y Dubes 1988], de los cuales, los tres primeros son los que realizan el agrupamiento de los datos en clusters, mientras que los dos últimos se refieren a la utilización de la salida:

(1) Representación del patrón (opcionalmente incluyendo características de la extracción y/o selección).
(2) Definición de una medida de la proximidad de patrones apropiada para el dominio de los datos.
(3) Clustering propiamente dicho (agrupamiento de los patrones).
(4) Abstracción de los datos (si es necesario).
(5) Evaluación de la salida (si es necesario).

El diagrama de la *ilustración 1* representa el secuenciamiento de los tres primeros pasos, incluyendo una realimentación que refleja la posible influencia de los agrupamientos de salida sobre el proceso de extracción/selección y/o a la semejanza entre patrones.

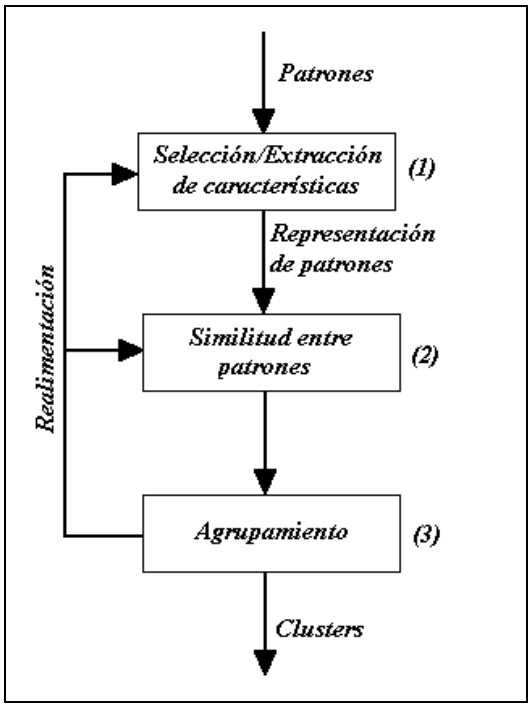


Ilustración 1. Fases del clustering

La *representación del patrón* se refiere al número de clases, el número de patrones disponible, y el número, tipo, y escala de las características disponibles para algoritmo de clustering. Es posible que parte de esta información no sea controlada. La *selección de características* es el proceso de identificar el subconjunto más apropiado de características dentro del conjunto original para utilizarlo en el proceso de agrupamiento. La *extracción de la característica* es el uso de una o más transformaciones de las características de la entrada para producir nuevas características de salida. Cualquiera de estas dos técnicas puede ser utilizada obtener un sistema apropiado de características para utilizarlas en el proceso de clustering.

La *proximidad de patrones* se mide generalmente según una función de distancia definida para pares de patrones. Existen una gran variedad de funciones de distancias que son utilizadas por diferentes autores [Anderberg 1973; Jain y Dubes 1988; Diday y Simon 1976]. Una medida simple de la distancia es la *distancia euclídea*, que es utilizada a menudo para reflejar desemejanza entre dos patrones, aunque es posible utilizar medidas [Michalski y Stepp 1983]. Las diferentes medidas de la distancia y sus características vienen reflejadas en el apartado ***Medidas de semejanza***.

El paso de agrupamiento o clustering propiamente dicho, puede ser realizado de diversas formas. El clustering de salida puede ser *hard* (duro) o *fuzzy* (borroso o difuso). El primero de ellos realiza una partición de los datos en grupos y en el segundo cada patrón tiene un grado variable de la calidad en cada uno de los clusters de la salida. Los algoritmos de clustering *Jerárquicos* una serie jerarquizada de particiones basadas en un criterio de combinación o división de clusters según su semejanza. Los algoritmos de clustering *Particionales* identifican la partición que optimiza (generalmente de manera local) un criterio de agrupamiento. Otras técnicas adicionales incluyen métodos probabilísticos [Brailovski 1991] y gráfico-teórico [Zahn 1971]. Toda esta variedad de técnicas de clustering se detalla en el apartado ***Técnicas de Clustering.***

La *abstracción de los datos* es el proceso de extraer una representación simple y compacta de un conjunto de datos. Aquí, simplemente se trata de seleccionar una forma de representar los datos adecuada para el procesamiento automatizado de la información (de modo que una máquina pueda realizar la transformación posterior eficientemente), o bien, adecuada para la interpretación de la información por parte un ser humano (de modo que la representación que se obtiene sea fácil de comprender y intuitivamente interpretable). En el contexto de clustering, una abstracción típica de los datos es a descripción compacta de cada cluster, generalmente en términos de los patrones representativos tales como el *centro de gravedad* [Diday y Simon 1976].

Es difícil evaluar si la salida de un algoritmo de clustering ha sido "buena" o "mala", es decir, si el algoritmo ha obtenido clusters válidos o útiles para el contexto concreto en el que se aplica. Además, como ocurro normalmente en todo lo relacionado con la computación, aunque está demostrado que ciertos tipos de algoritmos de clustering obtienen mejores resultados que otros, hay que tener en cuenta la cantidad y calidad de recursos de que se dispone, así como las restricciones de tiempo y espacio establecidas. Debido a estas razones, entre otras, es posible que haya que realizar un análisis previo de la información que se desea procesar. Este punto no será tratado en este trabajo.

El *análisis de la validez* de clusters, consiste en la evaluación de la salida obtenida por el algoritmo de clustering. Este análisis utiliza a menudo un criterio específico; sin embargo, estos criterios llegan a ser generalmente subjetivos. Así, existen pocos estándares en clustering excepto en subdominios bien predefinidos. Los análisis de validez deben ser objetivos [Dubes 1993] y se realizan para determinar si la salida es significativa. Cuando se utiliza aproximaciones de tipo estadístico en clustering, validación se logra aplicando cuidadosamente métodos estadísticos e hipótesis de prueba. Hay tres tipos de estudios de la validación:

1. La evaluación *externa* de la validez compara la estructura obtenida con una estructura a priori.

2. La evaluación *interna* intenta determinar si una estructura es intrínsecamente apropiada para los datos.

3. La evaluación *relativa* compara dos estructuras y mide la calidad relativa de ambas.

Estas técnicas de evaluación son discutidas detalladamente en [Jain y Dubes 1988] y [Dubes 1993], y no serán tratadas con más detalle más adelante.

2. DEFINICIONES Y NOTACIÓN

La notación y nomenclatura que se va a utilizar en este trabajo es la siguiente:

- Se denomina clustering al proceso particular de agrupar los datos, así como a todo el proceso completo. Dependiendo del contexto, se puede saber fácilmente a cual de los dos significados se refiere.

- Se denotará por cluster a un conjunto de datos agrupados tras el proceso de clustering.

- Un *patrón* $\boldsymbol{x}$ (vector de características o dato) no es más que un ítem de datos utilizado por el algoritmo de clustering y que normalmente consiste en un vector d componentes: $\boldsymbol{x} = (x_1, ..., x_d)$.

- Los componentes escalares individuales x_i de un patrón $\boldsymbol{x}$ se denominan *características* o *atributos.*

- d es la dimensión del patrón o del sistema de patrones, es decir, el número de características de los patrones.

- Se denota un *sistema de patrones* como: $X=\{x_1, \ldots, x_n\}$, donde cada patrón x_i es denotado por $x_i=(x_{i1}, \ldots, x_{id})$. En ocasiones, un sistema de patrones es representado como una matriz de patrones $n \times d$.

- Una *clase* se refiere al estado el proceso de generación de un patrón. Más concretamente, una clase puede verse como una fuente de patrones cuya distribución en el espacio de características viene determinada por una densidad de probabilidad de la clase. La técnicas de clustering intentan agrupar los patrones de forma que las clases reflejen el proceso de generación de patrones diferentes representado en el conjunto de patrones.

- Las denominadas *técnicas de clustering duras* (*hard clustering*) asignan una etiqueta de clase l a cada patrón x_i, identificando su clase. El conjunto de todas las etiquetas de un sistema de patrones X es $L=\{l_1, \ldots, l_n\}$ con $l_i \in \{1, \ldots, k\}$, donde k es el número de clusters.

- Los procedimientos *Fuzzy clustering* (clustering borrosos) asignan a cada patrón x_i de entrada un grado fraccionario f_{ij} de calidad en cada cluster j de salida.

- Una *medida de la distancia* es una métrica para cuantificar la *similitud* de los patrones. La similitud ente patrones también toma el nombre de *proximidad* entre patrones debido a que dos patrones son más similares cuanto más próximos se encuentran dentro del espacio de características.

La *ilustración 2* muestra gráficamente algunos de estos conceptos así como la representación que se hará de ellos a lo largo de este trabajo. Esta figura representa un ejemplo de clustering donde los patrones tienen dos características o atributos (*altura y temperatura*) que definen un espacio de características donde se representan los patrones por puntos dependiendo del valor que tengan esas características en cada patrón concreto. En este ejemplo, los patrones se han agrupado en dos clusters etiquetados como *A* y *B* (que son las etiquetas de clase).

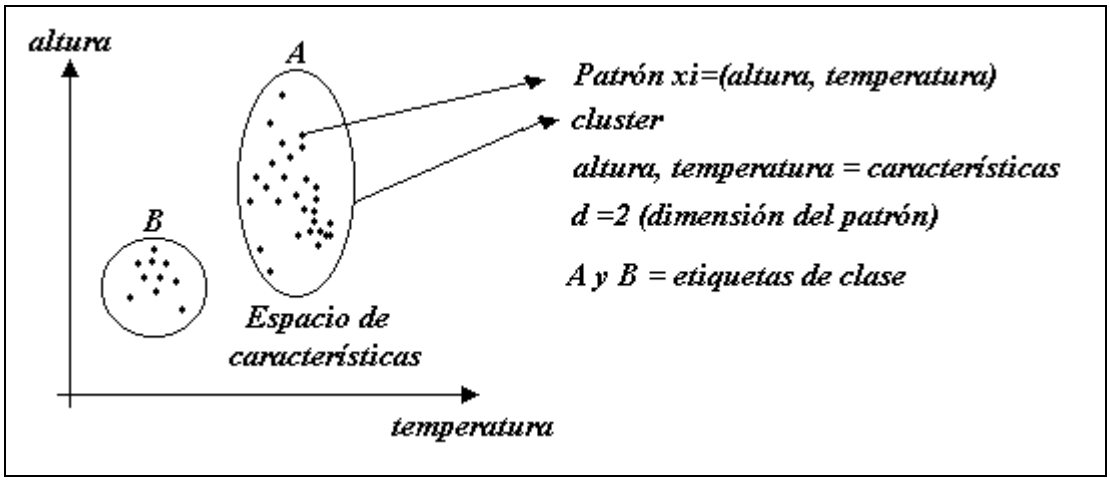


Ilustración 2. Ejemplo de representación gráfica.

Aunque en este caso el espacio de características es bidimensional, en general, se tendrá un hiperespacio de tantas dimensiones con número de características tengan los patrones (*d*).

A parte de estos conceptos y definiciones generales en el contexto del clustering, existen muchas otras nociones más específicas que se irán definiendo en el apartado correspondiente.

3. REPRESENTACIÓN PATRONES. SELECCIÓN Y EXTRACCIÓN DE CARACTERÍSTICAS.

No existe ningún tipo de guía teórica que indique qué patrones y atributos son los más apropiados para utilizar en una situación específica. De hecho, el proceso de generación del patrón a menudo no se puede controlar directamente. El papel del usuario en el proceso de representación del patrón es recolectar hechos y conjeturas sobre los datos, y opcionalmente, la *selección* y *extracción* de características, así como proyectar los elementos posteriores del sistema de clustering. Debido a las dificultades que presenta la representación del patrón, se asume que la representación del patrón está disponible antes de comenzar el proceso de clustering. No obstante, un estudio detallado de las características disponibles y algunas transformaciones pueden dar resultados sensiblemente mejorados del clustering. Un buen patrón de representación puede producir a menudo un clustering simple y fácil de entender; una representación pobre del patrón puede producir un clustering demasiado complejo, difícil o imposible de diferenciar. La *Ilustración 3* muestra un ejemplo sencillo.

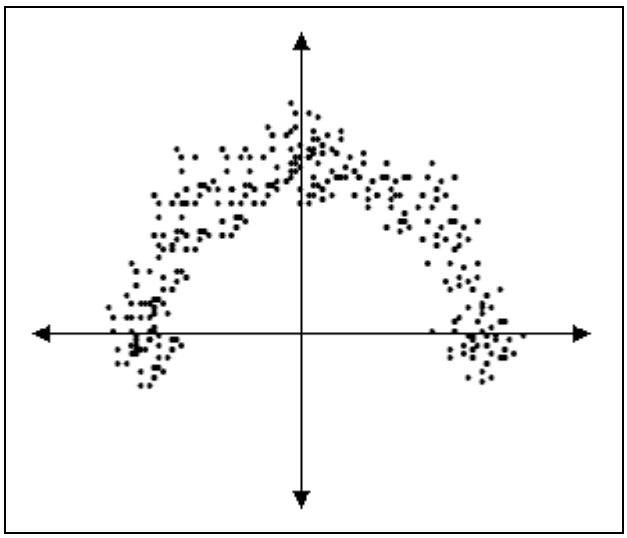


Ilustración 3. Puntos de cluster curvilíneo aproximadamente equidistante del origen.

Los puntos representados en un espacio bidimensional se distribuyen en un cluster curvilíneo de distancia aproximadamente constante del origen. Si se eligen coordenadas cartesianas para representar los patrones, es probable que muchos algoritmos de clustering fraccionaran el cluster varias partes, obteniendo así más clusters, ya que no es compacto. Si embargo, si se utiliza una representación en coordenadas polares para los clusters, la representación del cluster es mucho más sencilla, ya que el radio define claramente el cluster, lo cual facilita encontrar la un único cluster como solución.

Un patrón puede medir tanto objetos físicos como nociones abstractas. Como anteriormente de ha determinado, los patrones se pueden representar como vectores multidimensionales, donde cada dimensión se refiere a un atributo o característica

simple [Duda y Hart 1973]. Los atributo pueden ser cualitativos o cuantitativos. Por ejemplo, si *peso* y *color* son dos atributos usados, (20, *negro*) es la representación de a objeto negro con 20 unidades de peso. Los atributos o características los podemos clasificar en los siguientes tipos: [Gowda y Diday 1992]:

1) Atributos cuantitativos:
   a) Valores continuos (ej. peso)
   b) Valores discretos (ej. número de equipos)
   c) Valores del intervalo (ej. la duración de un acontecimiento)
2) Atributos cualitativos:
   a) Nominal (ej. color)
   b) Ordinal (ej. evaluaciones cualitativas de la temperatura (frío o calor))

Se pueden usar también características estructuradas como árboles [Michalski y Stepp 1983], donde cada nodo padre representa una generalización sus nodos hijos. Un ejemplo de esto lo muestra la *Ilustración 4*.

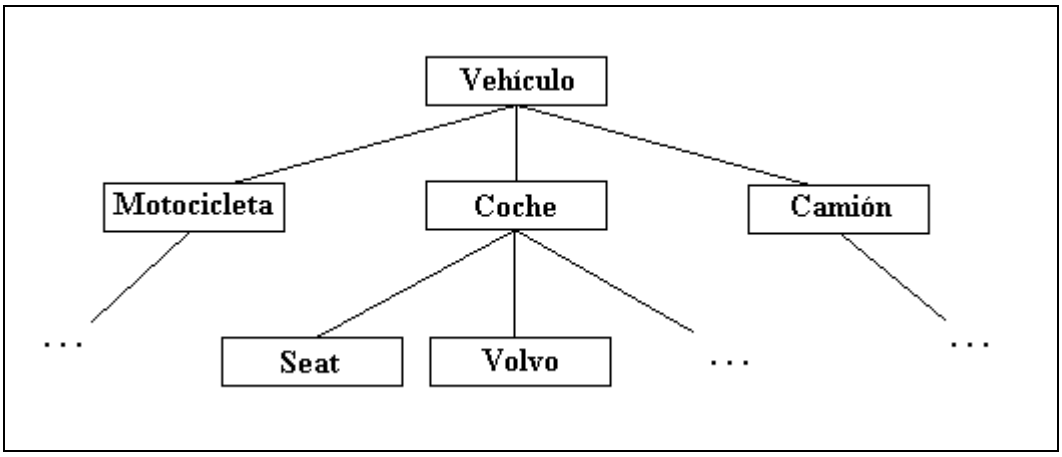


Ilustración 4. Ejemplo de atributo estructurado.

Como podemos observar, el nodo raíz *Vehículo* (que define el atributo), es una generalización de los nodos *Motocicleta, Coche* y *Camión*. A su vez, el nodo coche es una generalización de los nodos *Seat* y *Volvo*, ya que representa los coches de cualquier marca.

En [Diday 1988] se propone una representación generalizada de patrones denominada *Symbolic Objects* (Objetos Simbólicos), donde dichos Objetos Simbólicos vienen representados por una conjunción lógica de eventos. Estos eventos relacionan valores y características en los cuales los atributos pueden tomar uno o varios valores y donde todos los objetos no necesitan estar definidos para el mismo conjunto de atributos.

Es común utilizar solamente los atributos más discriminatorios y más descriptivos en el sistema de entrada, y utilizar exclusivamente estas características en el análisis. Las técnicas de **selección** de características seleccionan atributos de los que ya existen, mientras que las técnicas de **extracción** calculan nuevas características a partir del conjunto original de atributos. En cualquier caso, el objetivo es mejorar el funcionamiento de la clasificación y/o eficacia del cálculo. Las técnicas de selección de características es un campo bien estudiado para el reconocimiento de patrones

estadísticos [Duda y Hart 1973]. Sin embargo, en el contexto del clustering, el proceso de selección de características debe reconocer patrones exactos, lo que implica la posibilidad de tener que realizar un proceso de prueba y error, donde se seleccionan varios subconjuntos de características, se agrupan los patrones resultantes y se evalúa la salida según un índice de validez. Por otro lado, los procesos de extracción de características no necesitan datos etiquetados, por lo que se pueden usar directamente. La reducción del numero de características tiene la ventaja adicional, que es que se pueden producir resultados que se pueden estudiar visualmente.

4. MEDIDAS DE LA SEMEJANZA

Puesto que la semejanza entre patrones es fundamental a la hora de definir un cluster, es necesario establecer una forma de medir esta semejanza. La gran variedad de tipos de atributos hace que la medida (o medidas) de la semejanza deba ser elegida cuidadosamente. Lo más común es calcular el concepto contrario, es decir, la diferencia o *desemejanza* entre dos patrones usando la medida de la distancia en un espacio de características. Nos centraremos en la medida de la distancia entre patrones cuyas características son todas continuas. Para ello, utilizaremos la distancia *Euclídea* que es un caso particular de la de *Minkowski* (p=2).

$$d_2(x_i, x_j) = \left(\sum_{k=1}^{d} (x_{i,k} - x_{j,k})^2 \right)^{1I2} = \| x_i - x_j \|_2$$

(Ecuación de la distancia euclídea)

La ecuación general de la métrica de *Minkowski*, es decir, para cualquier valor de *p*, es la siguiente:

$$d_p(x_i, x_j) = \left(\sum_{k=1}^{d} (x_{i,k} - x_{j,k})^p \right)^{1Ip} = \| x_i - x_j \|_p$$

(Ecuación de la distancia de Minkowski)

La distancia *Euclídea* nos da una medida intuitiva de la distancia entre dos puntos en un espacio de dos o tres dimensiones. Esto puede ser útil cuando los clusters son compactos [Mao y Jain 1996]. El principal inconveniente que presenta la distancia de *Minkowski* es la tendencia de los atributos de mayor magnitud a dominar al resto. Para solucionar esta desventaja podemos normalizar los valores de los atributos continuos de forma que todos tomen valores dentro del mismo rango. Por otro lado, la correlación entre los distintos atributos puede influir negativamente en el cálculo de la distancia. Para dar solución a este problema, usamos la distancia cuadrática de *Mahalanobis*:

$$d_M(x_i, x_j) = (x_i, x_j) \sum^{-1} (x_j, x_i)^T$$

(Ecuación de la distancia de Mahalanibis)

donde x_i y x_j son vectores fila y Σ es la matriz de covarianza de los patrones. La distancia d(_,_) asigna diferentes pesos a cada característica basándose en la varianza y en la correlación lineal de los pares. La distancia de Mahalanobis se usa en [Mao y Jain 1996] para extraer clusters hiperelipsoidales.

Algunos algoritmos de clustering trabajan sobre los valores de una matriz de proximidad en vez de hacerlo directamente con el conjunto de datos original. Esto es útil situaciones donde haya que calcular previamente todas las n(n-1)/2 distancias entre pares de patrones (para n patrones) y almacenarlas en una matriz simétrica.

El cálculo de distancias entre los patrones con algunas o todas las características discretas presenta algunos problemas, desde los tipos diferentes de características no computables hasta (como un ejemplo extremo) el concepto de proximidad, que para atributos discretos en escala nominal es un valor binario. No obstante, en entornos donde es común trabajar con conjuntos de datos mixtos (por ejemplo, en aprendizaje automático) se trabaja con medidas de proximidad heterogéneas.

Pueden representarse también los patrones usando cadenas (*strings*) o estructuras de árbol [Knuth 1973]. Las cadenas son usadas en métodos de agrupamiento sintáctico (*syntactic clustering*)[Fu y Lu 1977]. Varios las medidas de similitud entre cadenas se describe en [Baeza-Yates 1992], mientras que [Zhang 1995] hace un resumen muy completo de medidas de similitud entre los árboles. Haciendo un estudio comparativo entre clusterings sintácticos y estadísticos [Tanaka 1995], se llega a la conclusión de que los métodos sintácticos son inferiores en todos los aspectos. Por consiguiente, no se considerarán los métodos sintácticos en este trabajo.

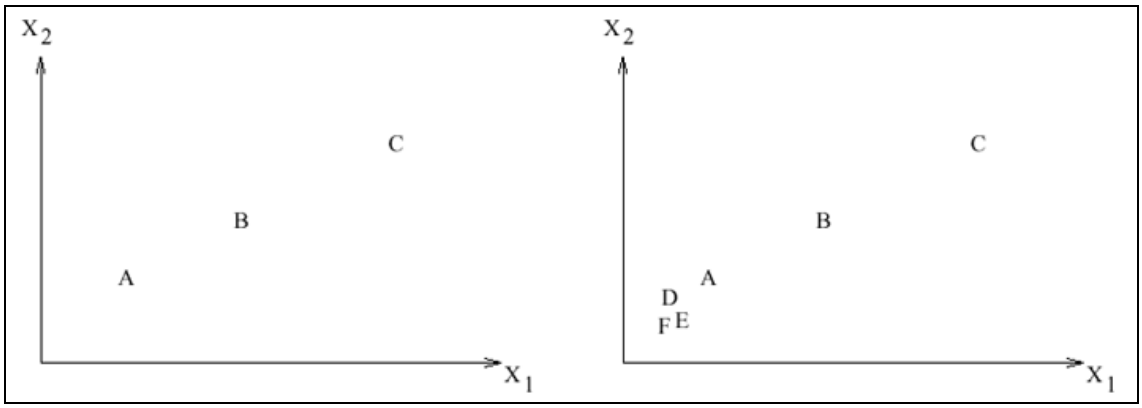
Hay algunas medidas de distancia en las que se considera el efecto del entorno o de los puntos vecinos[Gowda y Krishna 1977; Jarvis y Patrick 1973]. A estos puntos circundantes se les denomina *contexto*. Así, la similitud entre dos puntos x_i y x_j, teniendo en cuenta el contexto, viene dada por la expresión de la forma

$$s\ (x_i\ ,\ x_j) = f\ (\ x_i\ ,\ x_j\ ,\ E\ )$$

donde E es el contexto (conjunto de puntos del entorno). Una métrica definida con el concepto de contexto es la *distancia del vecino común* (MND: Mutual Neighbor Distance), propuesto en [Gowda y Krishna 1977], la cual viene dad por la siguiente expresión:

$$MND(x_i\ ,\ x_j) = NN(x_i\ ,\ x_j) + NN(x_j\ ,\ x_i)$$

donde $NN(x_i\ ,\ x_j)$ es el número del vecino de x_j con respecto al *xi*. Las *ilustraciones 5* y *6* muestran un ejemplo es esto.



Ilustraciones 5 y 6. Ejemplo de distancia MND teniendo en cuenta el contexto de los puntos

En la *ilustración 5*, la distancia entre los puntos *A* y *B* es menor que la distancia entre *A* y *C*, mientras que en la *ilustración 6*, donde el contexto es diferente, *B* y *C* están

más próximos que *B* y *A*. La MND entre *A* y *B* ha aumentado introduciendo un contexto de puntos adicionales, aunque *A* y *B* no se hayan movido.

El teorema del *patito feo* (*ugly duckling*) de Watanabe [Watanabe 1985] implica que es posible construir dos modelos arbitrarios cualesquiera similares codificándolos con un número suficientemente grande de características. Como consecuencia, dos modelo arbitrarios cualesquiera son similares a menos que se use alguna información del dominio adicional, Por ejemplo, en el caso del *clustering conceptual* [Michalski y Stepp 1983], la similitud o proximidad entre x_i y el x_j se define como:

$$s\ (x_i\ ,\ x_j) = f\ (\ x_i\ ,\ x_j\ ,\ C,\ E\ )$$

donde *C* es un conjuntos de *conceptos* predefinidos. Este aspecto se muestra en la *ilustración 7*.

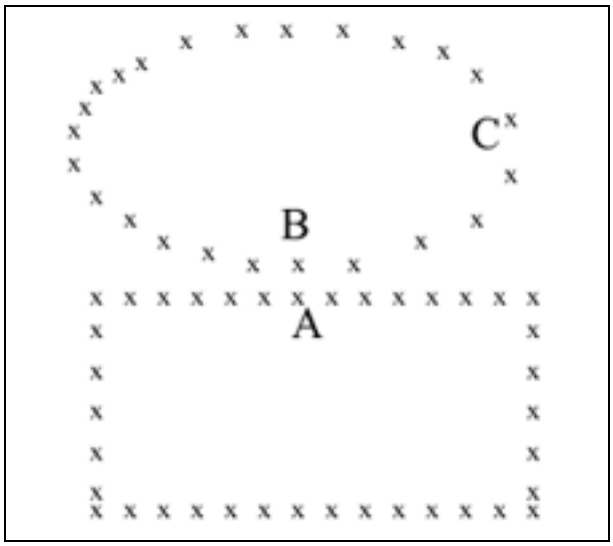


Ilustración 7. Similitud o proximidad conceptual entre puntos.

Aquí, la distancia de Euclídea entre los punto *A* y *B* es menor que la que existe entre *B* y *C*. Sin embargo, *B* y *C* pueden verse como más similares o más próximos que *A* y *B*, ya que *B* y *C* pertenecen al mismo *concepto* (la elipse) y *A* pertenece a un *concepto* diferente (el rectángulo).

5. TÉCNICAS DE CLUSTERING

La *ilustración 8* muestra en una representación de las diferentes metodologías sobre el clustering mediante un esquema jerárquico, aunque existen otras posibles clasificaciones. En este caso, se ha optado por la representación de [Jain y Dubes 1998].

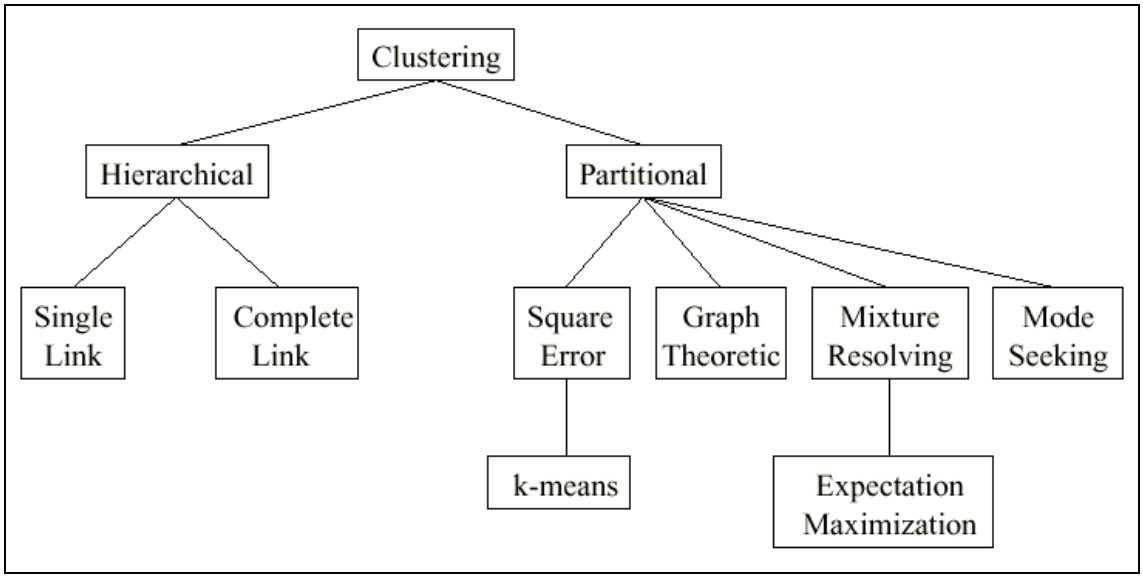


Ilustración 8. Un esquema jerárquico de las metodologías de Clustering

Como podemos observar, en el nivel superior se realiza la separación de las técnicas en dos grupos diferenciados: *Jerárquico* y *Particional*. Los métodos jerárquicos producen series jerarquizadas de particiones, mientras que los métodos particionales producen solamente una. Sin embargo, es necesario tener en cuenta diferentes aspectos o consideraciones a la hora de clasificar las diferentes metodologías de clustering. Estos aspectos se detallan a continuación, haciendo una comparación:

- Aglomerativo *vs.* disgregativo: Este aspecto está relacionado con la estructura algorítmica seguida para la creación de los clusters y el método de aproximación o acercamiento. Un acercamiento aglomerativo comienza con cada patrón en un cluster distinto (singleton), y combina sucesivamente clusters próximos hasta un que se satisface un criterio preestablecido. Por el contrario, el método disgregativo comienza con todos los patrones en un único cluster y se realizan particiones de éste, creando así nuevos clusters hasta satisfacer un criterio predeterminado.

- Monotético *vs.* politético: Este aspecto se refiere al uso secuencial o simultáneo de las características en el proceso de clustering. La mayoría de los algoritmos son *politético*, es decir, todas las características intervienen en el cálculo de las distancias entre los patrones, tomando decisiones en base a esas distancias. Un algoritmo *monotético* simple lo se proporciona en [Anderberg 1973], donde considera características secuencialmente para dividir el conjunto dado de patrones. La *Ilustración 9* muestra este aspecto, donde la colección se divide en dos clusters

usando la característica *x1*; la línea vertical discontinua *V* es la línea de separación. Cada uno de estos clusters son divididos posteriormente de forma independiente usando la característica *x2*, lo cual está representado por las líneas discontinuas *H1* y *H2*. El problema principal de este algoritmo es que genera 2^d clusters, donde *d* es la dimensión de los patrones. Para los valores grandes *de d (d*>100 es un valor típico para michas aplicaciones [Salton 1991]), el número de clusters generados es tan grande que el conjunto de datos es dividido en cluster demasiado pequeños para nuestros intereses.

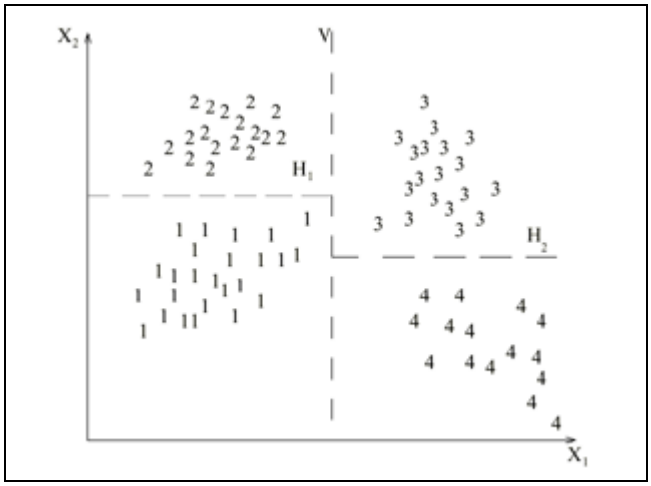


Ilustración 9. Clustering Particional Monotético.

- Hard *vs.* fuzzy: Un algoritmo *hard clustering* asigna cada patrón a un único cluster durante su operación y en su salida. Un algoritmo *fuzzy clustering* asigna grados de calidad en cada cluster para cada patrón de entrada. El *fuzzy clustering* puede ser convertido en *hard clustering* asignando cada patrón al cluster con el grado de calidad más alto.

- Determinista *vs.* estocástico: Esta cuestión es la más relevante en aproximaciones particionales, y está diseñada para optimizar un la función de error cuadrático. Esta optimización puede lograrse usando técnicas tradicionales o a través de una la búsqueda aleatoria del espacio de estados componiendo todas las posibles asignaciones de etiquetas.

- Incremental *vs.* No incremental: Esta cuestión se presenta cuando el sistema del patrones son grandes, y las restricciones de tiempo de ejecución y/o capacidad memoria afecta directamente al algoritmo. En los estudios más antiguos sobre este tema no se contemplaban muchos casos de algoritmos de clustering que trabajaran con gran cantidad de datos; sin embargo, la popularidad adquirida por el *data mining* en los últimos tiempos, ha fomentado el desarrollo de algoritmos de clustering que reducen al mínimo el número de exploraciones a través conjunto de patrones, reducen el número de patrones examinados durante la ejecución, o reducen el tamaño de las estructuras de datos utilizadas en los algoritmos.

A pesar de todo, normalmente la especificación de un algoritmo de clustering deja una relativa flexibilidad en la práctica [Jain y Dubes 1988].

Algoritmos de Clustering Jerárquico

El funcionamiento de los algoritmos de clustering jerárquico se puede observar en la *Ilustración 10*, en la cual se han usado conjuntos de datos de 2 dimensiones. Esta figura representa siete patrones (*A*, *B*, *C*, *D*, *E*, *F*, y *G*) agrupados en tres clusters. Un algoritmo jerárquico produce un árbol jerarquizado que representa grupos anidados de patrones por niveles. Para el caso concreto del ejemplo mostrado por la *ilustración 10*, y tras la aplicación de un algoritmo jerárquico concreto como es el *single-link* (o enlace simple) [Jain y Dubes 1988], el árbol correspondiente a esos siete puntos es el mostrado por la *ilustración 11*. El árbol puede estar dividido en diferentes niveles para proporcionar diferentes clusters.

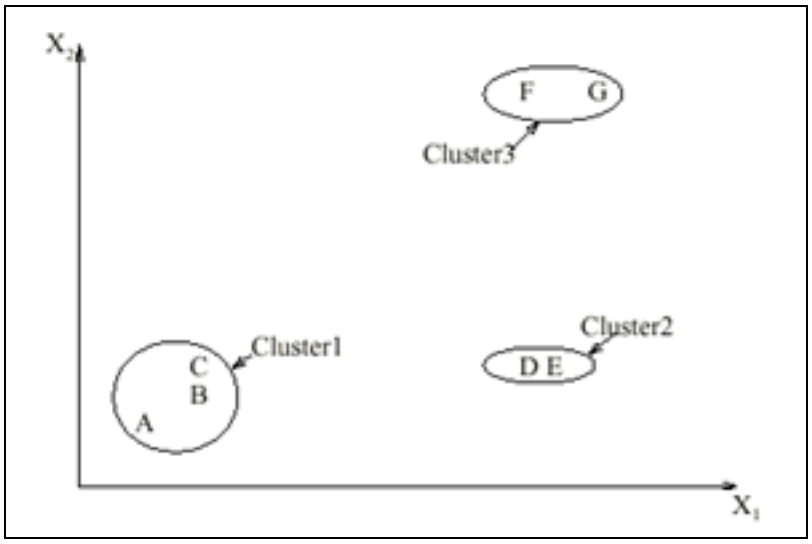


Ilustración 10. Puntos agrupados en tres clusters.

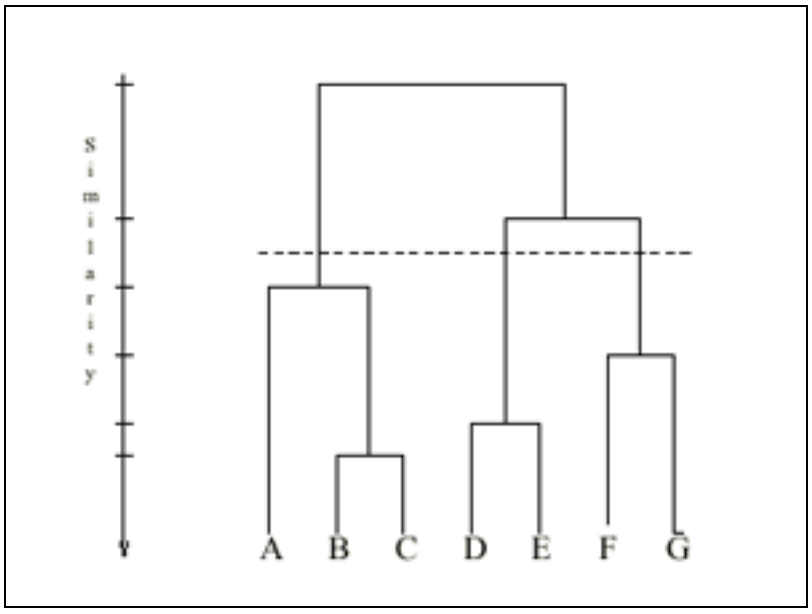


Ilustración 11. Árbol obtenido usando single-link

La mayoría de los algoritmos jerárquicos de clustering son variantes los algoritmos single-link [Sneath and Sokal 1973], complete-link [King 1967], y minimum-variance [Ward 1963; Murtagh 1984], de los cuales, el single-link y del complete-link son los más populares. La diferencia entre estos dos algoritmos radica en la forma de considerar la similitud entre dos clusters. En el método single-link, la distancia entre dos clusters es el mínimo de las distancias entre todos los pares de patrones de ambos clusters (un patrón de cada cluster). Por el contrario, en el método complete-link, la distancia entre dos clusters es el máximo de las distancias entre patrones de los dos clusters. En cualquier caso, dos clusters se combinan para formar uno más grande basado en criterios de distancia. De esta forma, el algoritmo complete-link produce los clusters compactos [Baeza-Yates1992]. Por el contrario, el algoritmo single-link sufre de un efecto de encadenamiento [Nagy 1968] tendiendo a producir clusters alargados o dispersos. En las *ilustraciones 12* y *13*, hay dos clusters separados por un "puente" de patrones ruidosos. El algoritmo del single-link produce los clusters mostrados en la *ilustración 12*, mientras que el algoritmo del complete-link obtiene los clusters mostrados en la *ilustración 13*. Como se pude observar, los clusters obtenidos tras la aplicación del método complete-link son más compactos que los obtenidos por el algoritmo single-link.

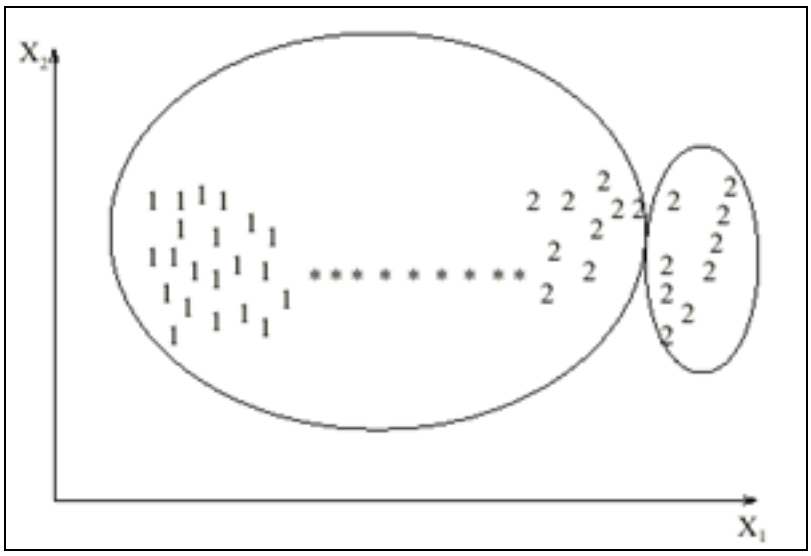


Ilustración 12. Clustering single-link .

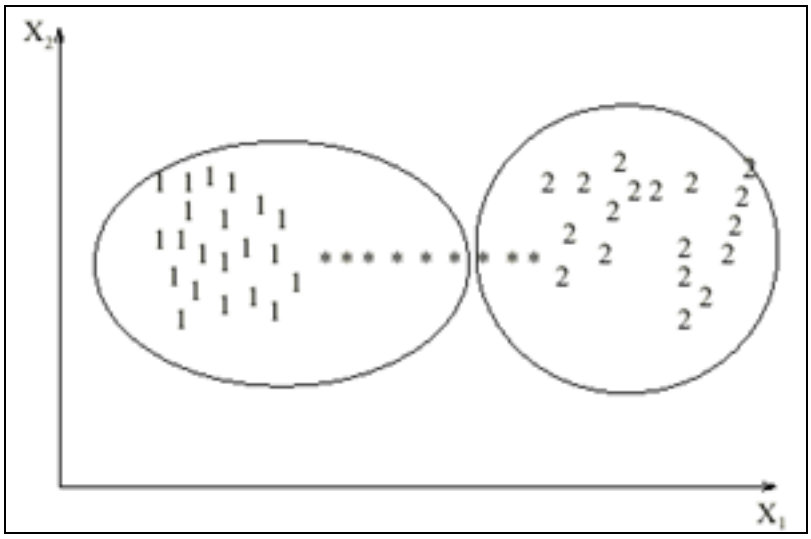


Ilustración 13. Clustering complete-link.

El Cluster etiquetado como 1 obtenido usando el algoritmo single-link es alargado debido a los patrones ruidosos etiquetados como "*". El algoritmo single-link es más versátil que el algoritmo complete-link. Por ejemplo, el algoritmo del single-link puede extraer los clusters concéntricos mostrados en la *ilustración 14*, mientras que el complete-link no pude. Sin embargo, desde un punto de vista pragmático, se ha observado que el algoritmo complete-link produce en muchos casos jerarquías más útiles que el single-link [Jain y Dubes1988].

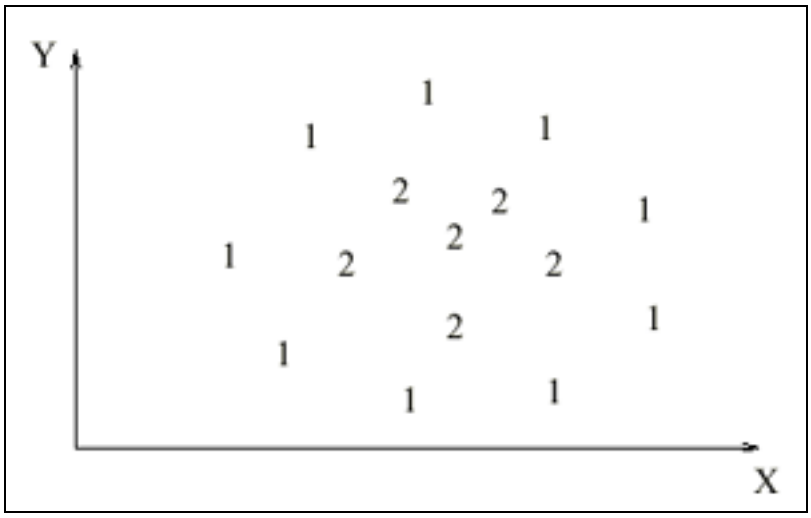


Ilustración 14. Dos clusters concéntricos.

Clustering Single-Link Aglomerativo

(1) Asigna a cada patrón un cluster propio y construye una lista ordenada ascendentemente con las distancias entre pares de patrones distintos.

(2) Recorre la lista de distancias y construye un grafo para cada valor distinto de d_k donde los pares de patrones con distancia menor que d_k son conectados. Si todos los patrones son miembros de un grafo conectado, se para. Si no, se repite este paso.

(3) La salida del algoritmo es una jerarquía anidada de grafos, la cual puede ser cortada en el nivel deseado para la formación de los clusters, que serán los conjuntos de patrones que queden conectados en el grafo correspondiente.

Clustering Complete-Link Aglomerativo

(1) Asigna a cada patrón un cluster propio y construye una lista ordenada ascendentemente con las distancias entre pares de patrones distintos.

(2) Recorre la lista de distancias y construye un grafo para cada valor distinto de d_k donde los pares de patrones con distancia menor que d_k son conectados. Si todos los patrones son miembros de un grafo completamente conectado, se para. Si no, se repite este paso.

(3) La salida del algoritmo es una jerarquía anidada de grafos, la cual puede ser cortada en el nivel deseado para la formación de los clusters, que serán los conjuntos de patrones que queden conectados en el grafo correspondiente

Los algoritmos jerárquicos son más versátiles que los algoritmos particionales. Por ejemplo, el algoritmo single-link funciona bien para conjuntos de datos no isotrópicos, incluso con clusters muy separados, alargados y concéntricos, mientras que un algoritmo particional típico, como el k-medias, funciona bien solamente con conjuntos de datos isotrópicos [Nagy 1968]. Por otra parte, normalmente, la complejidad en tiempo y espacio de los algoritmos particionales [Day 1992] es menor que la de los jerárquicos.

Vistas las características de ambos tipos de algoritmos jerárquicos (single-link y complete-link), es posible desarrollar algoritmos híbridos [Murty y Krishna 1980] que exploten las mejores características de ambos tipos de clustering.

Clustering Jerárquico Aglomerativo

(1) Construir la matriz de proximidad, que contendrá la distancia entre cada uno de los patrones y tratar cada patrón como un cluster independiente.

(2) Buscar el par más similar de clusters usando la matriz de proximidad y combina estos dos clusters en uno sólo. Tras esto, actualiza la matriz la proximidad para reflejar la operación de fusión.

(3) Si todos los patrones están en un mismo cluster, se para. Si no, pasar al paso 2.

Dependiendo de la forma en que se actualiza la matriz de proximidad en el paso 2, se pueden diseñar diferentes tipos de algoritmos aglomerativos Los algoritmos jerárquicos de decisión comienzan con un único cluster que engloba a todos los elementos del conjunto de datos y dividen el cluster siguiendo determinados criterios.

Algoritmos particionales

Un algoritmo de clustering particional obtiene una única partición de los datos en lugar una estructura de clustering, como por ejemplo el árbol que produce un método jerárquico. Los métodos particionales presentan ventajas cuando se trabaja con un conjuntos de datos extenso, para los cuales la construcción de un árbol jerárquico es no es factible. Un problema que presenta los algoritmos particionales es la opción del numero de clusters deseados a la salida, lo cual es un aspecto fundamental. Los métodos particionales producen clusters generalmente optimizando un criterio definido cualquiera, ya sea localmente (en un subconjunto de patrones) o globalmente (definido sobre todos los patrones). La búsqueda combinatoria del conjunto de etiquetas para la optimización del criterio es claramente un problema no factible. Por tanto, en la práctica, el algoritmo es ejecutado varias veces con diferentes estados iniciales, y aquella configuración que sea la mejor de todas la ejecuciones, se utilizará clustering de salida.

Algoritmos de error cuadrático

El criterio más intuitivo y a la vez más utilizado en las técnicas de clustering particional es el de error cuadrático, el cuál tiende a funcionar bien con clusters aislados y compactos. El error cuadrático de un clustering L de un conjunto de patrones X (con K clusters) es

$$e^2(X,L) = \sum_{j=1}^{K} \sum_{i=1}^{n_j} \| x_i^{(j)} - c_j \|^2$$

donde $x_i^{(j)}$ es el patrón i-ésimo que pertenece al j-ésimo cluster y el c_j es el centro de gravedad del j-ésimo cluster.

K-medias es el algoritmo más simple y más usado que emplea un criterio de error cuadrático [MQUEEN 1967]. Comienza con una partición inicial aleatoria, reasignando los patrones a clusters basándose en la similitud de los patrones y el centro de los clusters, hasta que se alcanza el criterio de convergencia (por ejemplo que ningún patrón cambie de cluster o que el error cuadrático deja de disminuir tras un número determinado de iteraciones). El algoritmo k-medias es popular porque es sencillo de implementar y además, la complejidad en tiempo es de *O(n)*, donde *n* es el número de patrones. Un inconveniente principal que presenta este algoritmo es que es sensible a la selección partición inicial, y puede converger a un mínimo local de la función de criterio si la partición inicial no es seleccionada correctamente. La *ilustración 15*

muestra siete patrones bidimensionales. Si comenzamos con los patrones A, B y C como las medias iniciales alrededor de las cuales se forman los tres clusters, entonces terminamos con la partición {{A}, {B, C}, {D, E, F, G}} mostradas como elipses.

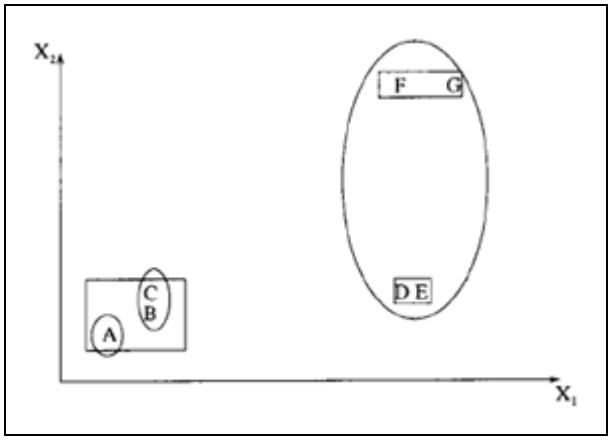


Ilustración 15. El algoritmo k-medias es sensible a la partición inicial

Como de puede observar, el valor del criterio de error cuadrático es mucho mayor para esta partición que para la mejor partición {{A, B, C}, {D, E},{F, G}} mostrada por rectángulos, que se toma el mínimo absoluto de la función de criterio de error cuadrático para un agrupamiento de tres clusters. La solución correcta de tres clusters se obtiene eligiendo, por ejemplo, A, D, y F como las medias iniciales de los clusters.

Método de Clustering de Error cuadrático

(1) Selecciona una partición inicial de los patrones con un número fijo del clusters y los centros de cada uno de ellos.

(2) Asigna cada patrón al más cercano de los clusters según su centro y calcula el nuevo centro de gravedad de los clusters. Repite este paso hasta que converja, es decir, hasta que la calidad del cluster es estable.

(3) Fusiona y divide los clusters según una cierta información heurística, repitiendo opcionalmente el paso 2.

Algoritmo de Clustering K-medias

(1) Elige *k* centros de clusters para coincidir con *k* patrones seleccionados al azar al o *k* puntos aleatorios definidos dentro de hipercubo que contiene el sistema de patrones.

(2) Asigna cada patrón al centro del cluster más cercano.

(3) Calcula de nuevo los centros de gravedad de los clusters.

(4) Si no se cumple el criterio de convergencia, ir al paso 2. Los criterios de convergencia típicos son: no reasignación o reasignación mínima de patrones a los nuevos centros de cluster, o mínima disminución del error cuadrático.

Existen diversas variantes [Anderberg 1973] del algoritmo k-medias que se ha propuesto en la literatura. Algunos de ellos intentan seleccionar una buena partición inicial de modo que la probabilidad de encontrar el mínimo absoluto sea mayor.

Otra variante es permitir combinar y dividir los clusters resultantes. Normalmente, un racimo se divide cuando su varianza supera un umbral preestablecido, y dos clusters se fusionan cuando la distancia entre los centros de gravedad de éstos no supera otro umbral establecido previamente. Usando esta variante, es posible obtener la partición óptima empezando con cualquier partición inicial arbitraria, siempre que se proporcionen valores de umbral apropiados. El algoritmo ISODATA [Ball y Hall 1965] algoritmo emplea esta técnica de combinación y división de clusters. Si se proporciona al ISODATA la partición representada con elipses en la *ilustración 15* como partición inicial, producirá una partición óptima de tres clusters. ISODATA unirá primero los clusters {A} y {B,C} en uno sólo, ya que la distancia entre sus centros de gravedad es pequeño, y después partirá el cluster {D,E,F,G}, que tiene una varianza grande, en dos clusters {D,E} y {F,G}.

Otra variante del algoritmo k-medias implica el seleccionar una función de criterio diferente en conjunto. *El algoritmo clustering dinámico* (que permite representaciones distintas de la del el centro de gravedad para cada cluster) fue propuesto en [Diday 1973], y [Symon 1977]. Este algoritmo describe un agrupamiento dinámico obtenido planteando el problema en el marco de estimación de probabilidad máxima. [Mao y Jain 1996] usa la distancia de *Mahalanobis* para obtener clusters hiperelipsoidales.

Clustering Graph-Theoretic

El algoritmo de clustering particional Graph-Theoretic más conocido se basa en la construcción del *árbol de expansión mínimo* (MST: Minimal Spannig Tree) [Zahn 1971], y eliminar los enlaces del MST con mayor longitud para generar los clusters. La *ilustración 16* representa el MST obtenido a partir de nueve puntos en dos dimensiones

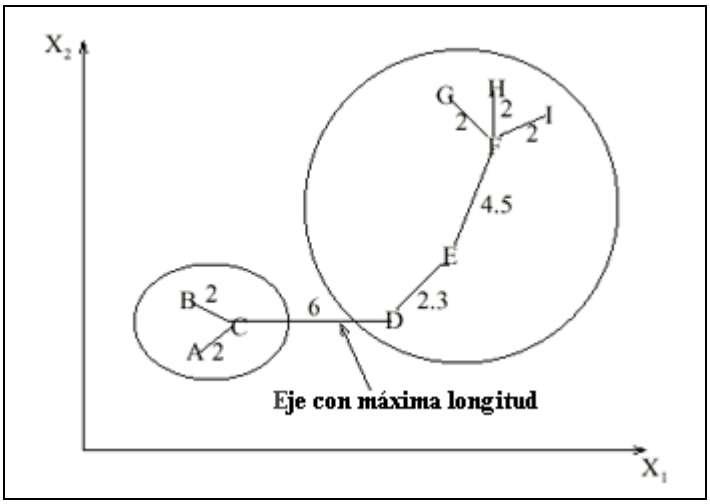


Ilustración 16. Formación de clusters usando MST (1)

Como se puede apreciar, se rompe el enlace *CD* con longitud de 6 unidades (el eje con la máxima longitud euclidiana), generando dos clusters ({A, B, C} y {D, E, F, G, H, I}). El segundo cluster puede ser dividido de nuevo en dos racimos rompiendo el enlace EF, que tiene una longitud de 4,5 unidades. Obteniendo así tres clusters ({A, B, C}, {D, E} y {F, G, H, I}), como muestra la *ilustración 17*.

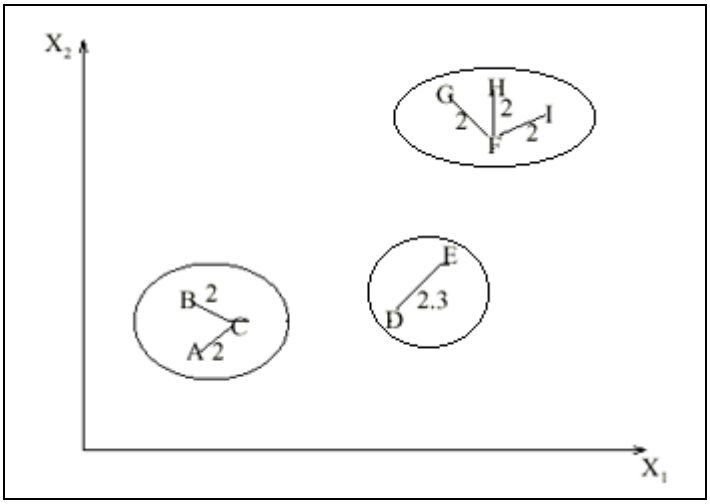


Ilustración 17. Formación de clusters usando MST (2)

Los métodos jerárquicos de clustering están también relacionados con el Graph-Theoretic. Los clusters del single-link son subgrafos del árbol de expansión mínima[Gower y Ross 1969]. Los clusters del complete-link son los subgrafos completos máximos, y se relacionan con coloreado de los nonos de un árbol [Backer y Hubert 1976]. El subgrafo completo máximo es considerado la mejor definición de un cluster en [Augustson y Minker 1970] y [Raghavan y Yu 1981]. Un grafo orientado enfocado a estructuras no jerárquicas y clusters solapados se presenta en [Ozawa 1985]. El *grafo de Delaunay* (DG: Delaunay Graph) es obtenido conectando todos los pares de los puntos que son *Vecinos de Voronoi*. El DG contiene toda la información sobre la vecindad de los nodos del MST y el grafo de vecindad relativa (RNG: Relative Reighbor-hood Graph) [Toussaint 1980].

Algoritmos Mixture-Resolving y Mode-Seeking

Los algoritmos de clustering Mixture-resolving han sido tratados de diversas formas. La suposición subyacente es que los patrones a agrupar son representados a partir de una distribución, y el objetivo es identificar los parámetros de cada uno y, en ocasiones, su número. La mayoría de los estudios referidos a este área asumen una distribución Gaussiana, cuyos parámetros son estimados por el procedimiento (normalmente realizan una estimación de probabilidad máxima).

El algoritmo de Expectation Maximization (EM) (algoritmo de probabilidad máxima de uso general [Dempster et al. 1977] para problemas con falta de datos) aplica al problema de valoración de parámetros. [Mitchell 1997] proporciona una descripción sencilla de esta técnica. En el contexto del EM, la densidad componentes es desconocida, y ésta se estima a partir de los patrones. El procedimiento EM comienza con una estimación inicial del parámetro vector y marca de nuevo iterativamente los patrones respecto a la densidad de la mezcla producida por el parámetro vector. Los patrones marcados se utilizan para actualizar las estimaciones de los parámetros. En el contexto del clustering, las marcas de los patrones se puede interpretar como "pistas" de la clase del patrón. Esos patrones, localizados (por sus marcas) en un componente particular, serían vistos como pertenecientes al mismo cluster.

También han sido desarrolladas técnicas no paramétricas para clustering basados en densidad en [Jain y Dubes 1988]. Otros autores incluyen también algoritmos de clustering particionales o jerárquicos algoritmo usando una medida de la distancia de acuerdo con un estimación de la densidad no paramétrica.

Clustering del Vecino más Cercano

Debido a que el concepto de proximidad desempeña un papel dominante representación intuitiva de un cluster, las distancia entre los vecinos más cercanos pueden servir como la base de procedimientos de clustering. Un procedimiento iterativo es propuesto en [Lu y Fu 1978], en el cual asigna cada patrón sin etiquetar al cluster de su patrón etiquetado más cercano, siempre que la distancia a ese vecino más cercano se encuentre por debajo de un cierto umbral. El proceso continúa hasta que se etiquetan todos los patrones o bien no se etiqueta ninguno nuevo. El valor de la vecindad (descrito anteriormente en el contexto del cálculo de la distancia) puede también utilizarse para unir clusters con sus vecinos cercanos.

Fuzzy Clustering

Los Procedimientos de clustering tradicionales generan particiones. En una partición, cada patrón pertenece a un único cluster. Por lo tanto, los clusters en generados por un *hard clustering* son disjuntos. La técnica de *Fuzzy Clustering* o (clustering borroso o difuso) extiende este concepto para asociar cada patrón a todos los

clusters usando una función de la calidad del miembro [Zadeh 1965]. La salida de tales algoritmos es un agrupamiento, pero no una partición. A continuación de detalla el funcionamiento de uno de estos algoritmos de *fuzzy clustering*.

Algoritmo Fuzzy clustering

(1) Selecciona una partición difusa inicial de N objetos en K clusters para construir la matriz de calidad de miembro U, la cual tendrá una dimensión NxK. Un elemento u_{ij} de esta matriz representa el grado de la calidad de miembro del objeto x_i en el cluster c_j. Normalmente, $u_{ij} \in [0,1]$.

(2) Con la matriz U, encuentra el valor de una función de criterio difuso (por ejemplo, función del error cuadrático ponderada) asociada con la partición correspondiente. Una posible función de criterio difuso es

$$E^2(X,U) = \sum_{i=1}^{N}\sum_{k=1}^{K} u_{ij} \parallel x_i - c_k \parallel^2$$

$$c_k = \sum_{i=1}^{N} u_{ik} x_i$$

donde c_k es el k-ésimo centro de cluster difuso. Reasigna patrones a los clusters para reducir el valor de la función criterio y recalcula U.

(3) Repite el paso 2 hasta que las entradas de la matriz U no cambien sensiblemente.

En Fuzzy clustering, cada cluster es un sistema difuso de todos los patrones. La *ilustración 18* muestra esta idea.

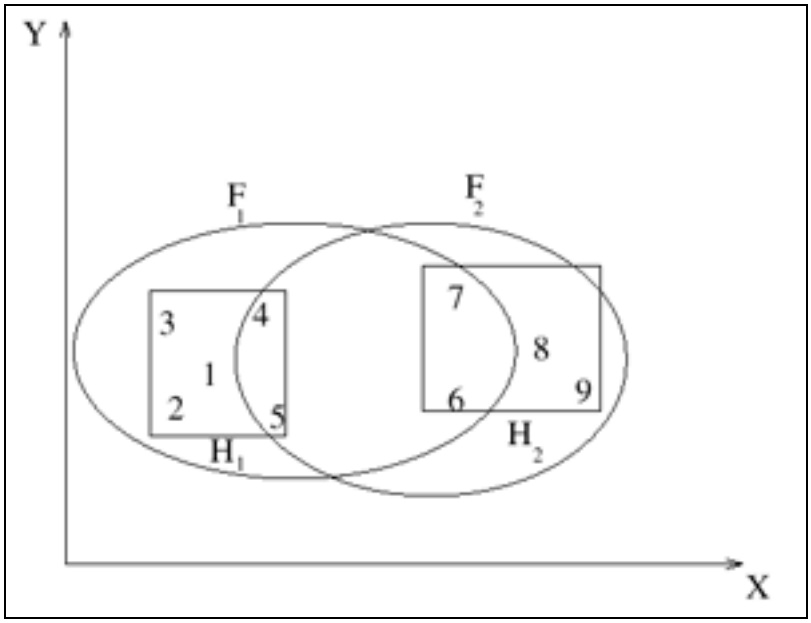


Ilustración 18. Fuzzy clusters

Los rectángulos encierran dos *hard clusters*: $H_1\{1,2,3,4,5\}$ y $H_2\{6,7,8,9\}$. Un *fuzzy* clustering produce los dos clusters difusos F_1 y F_2 representados por elipses. Los patrones tendrán calidad de miembro dentro del intervalo [0, 1] para cada cluster. Por ejemplo, cluster difuso F_1 se podría describir de forma compacta como:

{(1,0.9), (2,0.8), (3,0.7), (4,0.6), (5,0.55), (6,0.2), (7,0.2), (8,0.0), (9,0.0)}

y *el* F_2 se podría describir como:

{(1,0.0), (2,0.0), (3,0.0), (4,0.1), (5,0.15), (6,0.4), (7,0.35), (8,1.0), (9,0.9)}

Los pares (i,μ_i) representan el i-ésimo patrón y su calidad de miembro (μ_i) en cada cluster. Valores grandes de la calidad de miembro indican mayor confianza en la asignación del patrón al cluster. Se puede obtener un *hard* clustering a partir de una partición difusa (fuzzy) limitando el valor de la calidad de miembro.

Las teorías denominadas difusas o borrosas (*fuzzy*), fue aplicada inicialmente al clustering en [Ruspini 1969]. La referencia [Bezdek 198] es una buena fuente de consulta de fuzzy clustering. El fuzzy clustering más popular es el c-*medias* (FCM). Aunque es mejor que el hard clustering *k-medias* en evitar los mínimos locales, FCM puede converger a un mínimo local del cuadrado del criterio de error. El problema más importante que presenta el fuzzy clustering es el diseño de las funciones de calidad del miembro.

Representación de los Clusters

En las aplicaciones dónde el objetivo es encontrar un determinado número de clases o clusters dentro de un conjunto de datos, el resultado final debe ser una partición de ese conjunto de datos. Una partición da una idea sobre la separación de los puntos (datos) dentro de determinados clusters y si esta división es significativa para aplicar sobre los datos una clasificación supervisada, ya que éstas asumen un número dado de clases en el conjunto de datos. Sin embargo, en muchas otras aplicaciones que implican la toma de decisiones, los clusters resultantes deben ser representados de forma compacta e interpretable (*abstracción de datos*). Aunque la representación de los clusters es un paso importante en la toma de decisiones, no ha sido estudiada detalladamente. El concepto de representación de clusters fue introducido por Duran y Odell [1974] y se fue estudiado posteriormente por Diday y Simon [1976] y Michalski [1981]. Estos investigadores sugirieron los siguientes esquemas de representación:

- Representación de un cluster por su centro de gravedad o por un conjunto de puntos distantes en el cluster. La *ilustración 19* muestra estas dos ideas.

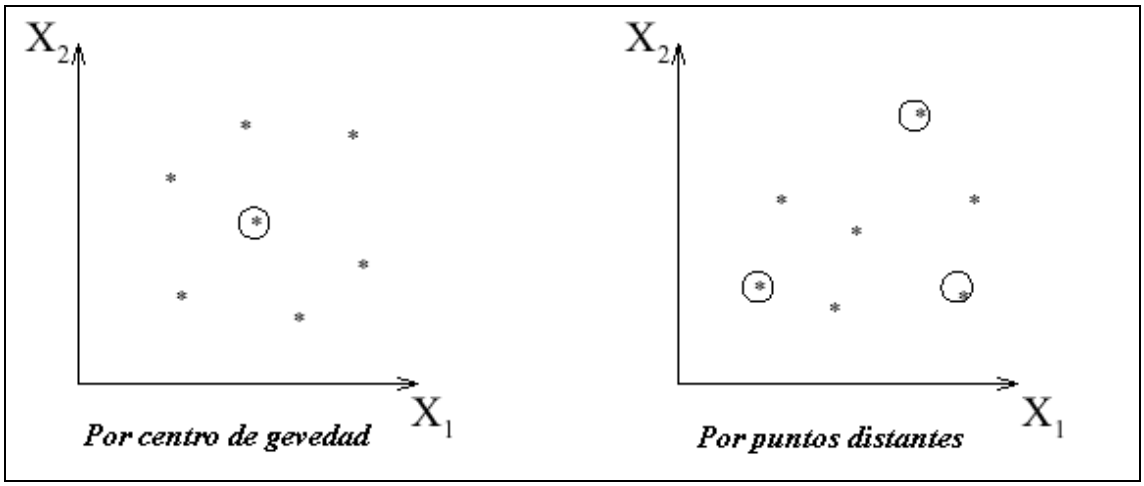


Ilustración 19. Representación de clusters por puntos

- Representación de clusters mediante nodos en un árbol de clasificación, como muestra la *ilustración 20*.

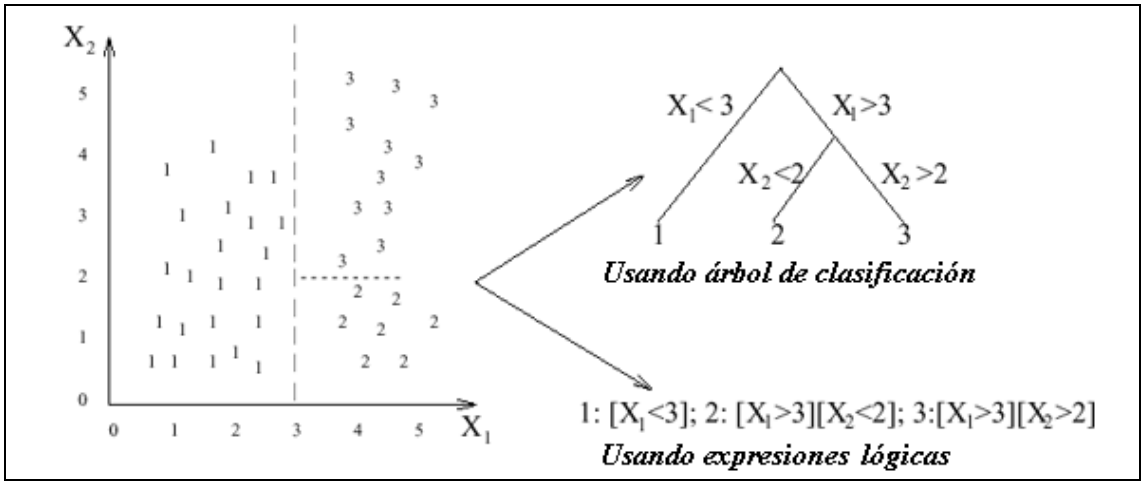


Ilustración 20. Representación de clusters por árbol de clasificación o por expresiones lógicas.

- Representación de los clusters usando conjunciones de expresiones lógicas. Por ejemplo, $[X_1>3][X_2<2]$ en la *ilustración 20*.

De todos estos esquemas de representación, el más popular es el de centros de gravedad. Esta representación es apropiada cuando los clusters son compactos o isotrópicos. En cambio, cuando los clusters son alargados y no isotrópicos, este esquema no resulta una buena representación. En tal un caso, el uso de puntos límite de un cluster es una buena elección. En este caso, el número de puntos que representan al cluster debe aumentar en la medida que aumenta la complejidad de forma del cluster.

Por otro lado, las dos representaciones que muestra la *ilustración 20* son equivalentes. Cada uno de los caminos del árbol de clasificación entre la raíz y cualquier hoja, responde a una de las cláusulas conjuntivas en la representación por expresiones lógicas. Una limitación importante del uso de expresiones lógicas para

representación de clusters es que sólo pueden describir clusters regulares o isotrópicos dentro del espacio de características.

La abstracción del datos es útil la toma de decisiones por las razones siguientes:

(1) Proporciona una descripción simple e intuitiva de los clusters, de forma que ésta sea sencilla para la comprensión humana. Tanto en el clustering conceptual [Michalski y Stepp 1983] como en el clustering simbólico [Gowda y Diday 1992] esta representación se obtiene sin usar ningún paso adicional. Estos algoritmos generan los clusters así como sus descripciones. Se pueden obtener un juego de difusas (*fuzzy rules*) a partir de clusters difusos (*fuzzy* clusters), pudiendo usa estas reglas para la construcción de clasificadores y controladores difusos (*fuzzy classifiers* y *fuzzy controllers*).

(2) Es útil en procesos o aplicaciones de compresión de datos [Murty y Krishna 1980]. La *ilustración 21(a)* muestra los datos agrupados en dos clusters etiquetados como *1* y *2*. Un método de clustering particional como el *k-medias* no puede separa estas dos estructuras de forma apropiada. El algoritmo *single-link* trabaja bien con este tipos de distribución de datos, pero es computacionalmente muy costoso. Un híbrido de estos algoritmos puede usarse para explotar las propiedades deseables de ambos algoritmos. Aplicando el algoritmo *k-medias* al conjunto de datos de la *ilustración 21(a)*, obtenemos ocho subclusters. Cada uno de estos subclusters puede ser representado por su centro de gravedad. Ahora se aplica el algoritmo *single-link* sólo a los centros de gravedad para dividirlos en dos grupos. Los grupos resultantes se muestran en la *ilustración 21(b)*. De esta forma, la compresión de los datos es almacenada mediante la representación de los centros de gravedad de los subclusters.

(3) Aumenta la eficiencia de las tareas de toma de decisiones. En una técnica de recuperación de documentos basada en clusters [Salton 1991], gran cantidad de documentos son agrupados en clusters y cada uno de estos clusters son representados por sus centros de gravedad. Para recuperar documentos relevantes en una consulta, la consulta se empareja o asocia con los centros de gravedad de los clusters en lugar de todos los documentos. Así, se realiza una recuperación de los documentos relevantes de una forma eficiente. También, en aplicaciones que involucran grandes cantidades de datos, el clustering es usado para indexar dichos datos, lo que contribuye a aumentar la eficiencia en la toma de decisiones [Dorai y Jain 1995].

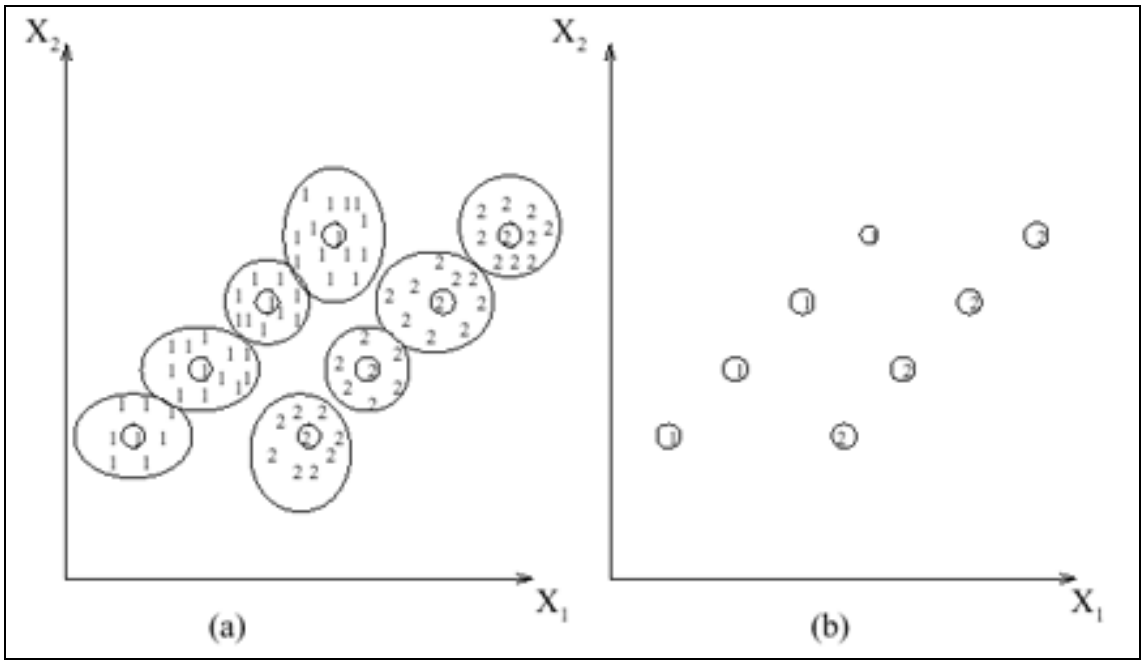


Ilustración 21. Compresión de datos usando clustering

6. REFERENCIAS

ANDERBERG, M. R. 1973. *Cluster Analysis for Application*s. Academic Press, Inc., NewYork, NY.

AUGUSTSON,J.G. Y MINKER, J. 1970. An analysis of some graph theoretical clustering techniques. *J. ACM 1*7, 4 (Oct. 1970), 571–588.

BACKER,F.B. Y HUBERT, L. J. 1976. A graph-theoretic approach to goodness-of-fit in complete-link hierarchical clustering. *J. Am.Stat. Assoc. 7*1, 870–878.

BAEZA-YATES, R. A. 1992. Introduction to data structures and algorithms related to information retrieval. In *Information Retrieval: Data Structures and Algorithm*s,W.B. Frakes and R. Baeza-Yates, Eds. Prentice-Hall, Inc., Upper Saddle River, NJ, 13–27.

BALL,G.H. Y HALL, D. J. 1965. ISODATA, a novel method of data analysis and classification.Tech. Rep.. Stanford University,Stanford, CA.

BEZDEK, J. C. 1981. *Pattern Recognition With Fuzzy Objective Function Algorithm*s. Plenum Press, New York, NY.

BRAILOVSKY, V. L. 1991. A probabilistic approach to clustering. *Pattern Recogn. Lett.1*2, 4 (Apr. 1991), 193–198.

DAY, W. H. E. 1992. Complexity theory: An introduction for practitioners of classification. In *Clustering and Classificatio*n,P. Arabie and L. Hubert, Eds. World Scientific Publishing Co., Inc., River Edge, NJ.

DIDAY, E. 1973. The dynamic cluster method in nonhierarchical clustering. *J. Comput. Inf.Sci.* 2, 61–88.

DIDAY, E. 1988. The symbolic approach in clustering. In *Classification and Related Method*s, H. H. Bock, Ed. North-Holland Publishing Co., Amsterdam, The Netherlands.

DIDAY,E. Y SIMON, J. C. 1976. Clustering analysis. In *Digital Pattern Recognitio*n,K. S. Fu, Ed. Springer-Verlag, Secaucus, NJ, 47–94.

DORAI,C. Y JAIN, A. K. 1995. Shape spectra based view grouping for freeform objects. In *Proceedings of the International Conference on Image Processing* (ICIP-95), 240–243.

DUBES, R. C. 1993. Cluster analysis and related issues. In *Handbook of Pattern Recognition & Computer Visio*n, C. H. Chen, L. F. Pau, and P. S. P. Wang, Eds. World Scientific Publishing Co., Inc., River Edge, NJ, 3–32.

DUDA,R.O. Y HART, P. E. 1973. *Pattern Classification and Scene Analysi*s. John

DURAN,B.S. Y ODELL, P. L. 1974. *Cluster Analysis: A Survey*. Springer-Verlag, New York, NY.

FU,K.S. Y LU, S. Y. 1977. A clustering procedure for syntactic patterns. *IEEE Trans. Syst. Man Cybern.* 7, 734–742.

GARCÍA, J. Junio1999. *Aplicación De Técnicas De Clasificación De Información Y Redes Neuronales Para La Definición De Arquitecturas Para La Mejora Del Proceso Software.*http://www.ie.inf.uc3m.es/Tesis/Documentos/ProcesosSoftware/RedesNeuronales/MejoraRedesNeuronales.html

GOWDA,K.C. Y DIDAY, E. 1992. Symbolic clustering using a new dissimilarity measure. *IEEE Trans. Syst. Man Cybern. 22*, 368–378.

GOWDA,K.C. Y KRISHNA, G. 1977. Agglomerative clustering using the concept of mutual nearest neighborhood. *Pattern Recogn. 1*0, 105–112.

GOWER,J.C. Y ROSS, G. J. S. 1969. Minimum spanning rees and single-linkage cluster analysis. *Appl. Stat. 1*8, 54–64.

JAIN,A.K. Y DUBES, R. C. 1988. *Algorithms for Clustering Data*. Prentice-Hall advanced reference series. Prentice-Hall, Inc., Upper Saddle River, NJ.

JARVIS,R.A. Y PATRICK, E. A. 1973. Clustering using a similarity method based on shared near neighbors. *IEEE Trans. Comput. C-22*, 8 (Aug.), 1025–1034.

KING, B. 1967. Step-wise clustering procedures. *J. Am. Stat. Assoc. 6*9, 86–101.

KNUTH, D. 1973. *The Art of Computer Programming*. Addison-Wesley, Reading, MA.

LU,S.Y. Y FU, K. S. 1978. A sentence-to-sentence clustering procedure for patternanalysis. *IEEE Trans. Syst. Man Cybern.* 8, 381–389.

MAO,J. Y JAIN, A. K. 1996. A self-organizing network for hyperellipsoidal clustering (HEC). *IEEE Trans. Neural Netw.* 7, 16–29.

MCQUEEN, J. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 281–297.

MICHALSKI, R., STEPP,R.E., Y DIDAY, E. 1983. Automated construction of classifications: conceptual clustering versus numerical taxonomy. *IEEE Trans. Pattern Anal. Mach. Intell. PAMI*-5, 5 (Sept.), 396–409.

MICHALSKI, R., STEPP,R.E., Y DIDAY,E. 1981. A recent advance in data analysis: Clustering objects into classes characterized by conjunctive concepts. In *Progress in Pattern Recognition, Vol.* 1, L. Kanal and A. Rosenfeld, Eds. North-Holland Publishing Co., Amsterdam, The Netherlands.

MITCHELL, T. 1997. *MachineLearnin*g. McGraw-Hill, Inc., New York, NY.

MURTAGH, F. 1984. A survey of recent advances in hierarchical clustering algorithms which use cluster centers. *Comput. J. 2*6, 354–359.

MURTY,M.N. Y KRISHNA, G. 1980. A computationally efficient technique for data clustering. *Pattern Recogn. 1*2, 153–158.

NAGY, G. 1968. State of the art in pattern recognition. *Proc. IEEE 5*6, 836–862.
OZAWA, K. 1985. A stratificational overlapping cluster scheme. *Pattern Recogn. 1*8, 279–286.

RAGHAVAN,V.V. Y YU, C. T. 1981. A comparison of the stability characteristics of some graph theoretic clustering methods. *IEEE Trans. Pattern Anal. Mach. Intell.* 3, 393–402.

RUSPINI, E. H. 1969. A new approach to clustering. *Inf. Control 1*5, 22–32.

SALTON, G. 1991. Developments in automatic text retrieval. *Science 25*3, 974–980.

SNEATH,P.H.A. Y SOKAL, R. R. 1973. *Numerical Taxonom*y. Freeman, London,

TANAKA, E. 1995. Theoretical aspects of syntactic pattern recognition. *Pattern Recogn. 2*8, 1053–1061.

TOUSSAINT, G. T. 1980. The relative neighborhood graph of a finite planar set. *Pattern Recogn. 1*2, 261–268.
UK.

WARD,J.H.JR. 1963. Hierarchical grouping to optimize an objective function. *J. Am. Stat. Assoc. 5*8, 236–244.

WATANABE, S. 1985. *Pattern Recognition: Human and Mechanica*l. John Wiley and Sons, Inc., New York, NY.
Wiley and Sons, Inc., New York, NY.

ZADEH, L. A. 1965. Fuzzy sets. *Inf. Control* 8, 338–353.

ZAHN, C. T. 1971. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Trans. Comput. C-20* (Apr.), 68–86.

ZHANG, K. 1995. Algorithms for the constrained editing distance between ordered labeled trees and related problems. *Pattern Recogn. 2*8, 463–474.