

---

---

## Capítulo 8

---

---

### Métodos de agrupamiento [Clustering]

---

---

|   |    |
|---|----|
| 1. INTRODUCCIÓN .....                                 | 2  |
| 1.1 Formulación matemática del problema .....         | 3  |
| 2 ALGORITMOS DE AGRUPAMIENTO .....                    | 4  |
| 2.1 Agrupamiento mediante enfriamiento simulado ..... | 6  |
| 2.2 Método adaptativo .....                           | 8  |
| 2.3 Algoritmo de Batchelor y Wilkins .....            | 10 |
| 2.4 Algoritmo de las K medias .....                   | 11 |
| 2.5 Algoritmo GRASP .....                             | 12 |
| 2.6 Algoritmo de agrupamiento secuencial .....        | 17 |
| 2.7 Algoritmo ISODATA .....                           | 18 |
| 2.8 Métodos basados en grafos .....                   | 19 |
| 3. BIBLIOGRAFÍA .....                                 | 21 |

## 1. Introducción

Los métodos de agrupamiento o clustering (arracimamiento en algunas traducciones no demasiado buenas) constituyen un tipo de aprendizaje por descubrimiento muy similar a la inducción. En el aprendizaje inductivo, un programa aprende a clasificar objetos basándose en etiquetados proporcionados por un profesor (aprendizaje supervisado). En los métodos de agrupamiento no se suministran los datos etiquetados: el programa debe descubrir por sí mismo las clases naturales existentes.

Por ejemplo, el programa AUTOCLASS (Cheeseman, Self, Kelly, Taylor, Freeman y Stutz, “*Bayesian Classification*”, Proceedings AAAI88, 1988) usa razonamiento bayesiano para, dado un conjunto de datos de entrenamiento, sugerir un conjunto de clases plausible. Este programa encontró nuevas clases significativas de estrellas a partir de sus datos del espectro infrarrojo, lo que puede considerarse ejemplo de descubrimiento por parte de una máquina (los hechos descubiertos eran desconocidos para los astrónomos).

Las funciones de densidad de probabilidad suelen tener una moda o un máximo en una región; es decir, las observaciones tienden a agruparse en torno a una región del espacio de patrones cercana a la moda. Las técnicas de agrupamiento analizan el conjunto de observaciones disponibles para determinar la tendencia de los patrones a agruparse. Estas técnicas permiten realizar una clasificación asignando cada observación a un agrupamiento [cluster], de forma que cada agrupamiento sea más o menos homogéneo y diferenciable de los demás.

Los agrupamientos naturales obtenidos mediante una técnica de agrupamiento mediante similitud resultan muy útiles a la hora de construir clasificadores cuando no están bien definidas las clases (no existe un conocimiento suficiente de las clases en que se pueden distribuir las observaciones), cuando se desea analizar un gran conjunto de datos (“divide y vencerás”) o, simplemente, cuando existiendo un conocimiento completo de las clases se desea comprobar la validez del entrenamiento realizado y del conjunto de variables escogido.

Los métodos de agrupamiento asocian un patrón a un agrupamiento siguiendo algún *criterio de similitud*. Tales medidas de similitud deben ser aplicables entre pares de patrones, entre un patrón y un agrupamiento y, finalmente, entre pares de agrupamientos. Generalmente, como medidas de similitud se emplean métricas de distancia. Algunas de las más habituales son la distancia euclídea, la distancia euclídea normalizada, la distancia euclídea ponderada y la distancia de Mahalanobis.

## 1.1 Formulación matemática del problema

Consideremos un conjunto  $M$  de ejemplos y  $K$  clusters ( $C_1, C_2, \dots, C_k$ ):

$$\Leftrightarrow C_i \neq \emptyset \ (\#C_i > 0)$$

$$\Leftrightarrow i \neq j \rightarrow C_i \cap C_j = \emptyset$$

$$\Leftrightarrow M = \cup C_i = \{X_1, X_2, \dots, X_m\} \text{ tal que } X_i \in \mathbb{R}^n$$

Matemáticamente, el problema puede formularse como la minimización de

$$f(W, Z; X) = \sum_{i=1}^m \sum_{j=1}^k w_{ij} \|X_i - Z_j\|^2$$

$$\sum_{j=1}^k w_{ij} = 1, \quad 1 \leq i \leq m$$

$$w_{ij} \in \{0,1\}$$

- $X_i$  es el vector patrón correspondiente al ejemplo  $i$ -ésimo ( $X_i \in \mathbb{R}^n$ ).
- $Z_j$  es el centro del  $j$ -ésimo cluster ( $Z_j \in \mathbb{R}^n$ ).
- $W$  es la matriz de pertenencia ( $m \times k$ ) tal que  $w_{ij}$  es 1 si  $X_i \in C_j$  y 0 en caso contrario.
- $k$  es el número de clusters (clases en el problema de clasificación)
- $m$  es el número de ejemplos del conjunto de casos de entrenamiento.

### ANOTACIONES

- ✓ La función  $f$  no es convexa, por lo que pueden existir mínimos locales no óptimos.
- ✓ La minimización de la función  $f$  requiere conocer a priori el número deseado de agrupamientos  $k$  (si no el problema sería trivial: un agrupamiento para cada caso de entrenamiento). No obstante, existen técnicas que permiten ajustar el número de agrupamientos (fusionando y dividiendo agrupamientos), así como tratar elementos discordantes [*outliers* en inglés] debidos, por ejemplo, a ruido en la adquisición de datos.

## 2. Algoritmos de agrupamiento

La agrupamientos detectados dependen del algoritmo empleado, del valor dado a sus parámetros, de los datos utilizados y de la medida de similaridad adoptada.

Se han propuesto cientos de algoritmos de agrupamiento más o menos específicos. Según se use o no una función criterio se distinguen los algoritmos directos o constructivos (basados en aproximaciones heurísticas) de los algoritmos indirectos o por optimización.

La estructura general de un algoritmo iterativo de agrupamiento directo es la siguiente:

```
Seleccionar una partición inicial
del conjunto de ejemplos en K clusters.

Calcular los centros de los clusters

Mientras que no se estabilicen los clusters
    Repetir
        Generar una nueva partición
        (asignando cada patrón al cluster más cercano)

        Calcular los centroides de los nuevos clusters

    Hasta que se obtenga un valor óptimo de la función de evaluación

Ajustar el número de clusters
mezclando y separando clusters existentes
o eliminando clusters pequeños o "periféricos"
```

En las siguientes páginas se muestran algunos algoritmos de los más empleados. Sin embargo, existen otros muchos métodos de agrupamiento. Por ejemplo, los algoritmos DHB y DHF utilizan dos estrategias de búsqueda básicas: el algoritmo DHB realiza una búsqueda en anchura mientras el algoritmo DHF hace una búsqueda en profundidad. El conocido algoritmo de Forgy no es más que una extensión de DHF que permite adaptar el número de agrupamientos [*clusters*] obtenidos.

Los tres algoritmos citados en el párrafo anterior (DHB, DHF y algoritmo de Forgy) realizan una exploración exhaustiva del espacio de búsqueda, lo cual impide su utilización en la muchos problemas reales. En las siguientes referencias se puede obtener más información sobre esos tres algoritmos:

➤ *A.K.Jain & R.C.Dubes: "Algorithms for Clustering Data". Prentice Hall, 1988.*

➤ *M.A.Ismael & M.S.Kamel: "Multidimensional Data Clustering Utilizing Hybrid Search Strategies". Pattern Recognition 22, 1989, pgs. 75-89.*

Técnicas de Algorítmica, como el *Simulated Annealing* [enfriamiento simulado], nos permiten agrupar conjuntos de datos más grandes sin la necesidad de realizar una exploración exhaustiva del espacio de posibles asignaciones. Las siguientes referencias recogen cómo se puede emplear el enfriamiento simulado para resolver problemas de agrupamiento:

➤ *S.Z.Selim & K.Alsultan: "A Simulated Annealing for the Clustering Problem". Pattern Recognition 24, 1991, pgs. 1003-1008.*

➤ *R.W.Klein & R.C.Dubes: "Experiments in Projection and Clustering by Simulated Annealing". Pattern Recognition 22, 1989, pgs. 213-220.*

No obstante, los algoritmos de agrupamiento más utilizados en problemas reales suelen basarse en heurísticas. Existen multitud de algoritmos heurísticos de agrupamiento, de los cuales se expondrán algunos de los más representativos. No existe ninguna medida objetiva que permita su comparación, por lo que los comentarios acerca de la bondad de cada método son, generalmente, sesgados y subjetivos.

La exposición de los algoritmos de clustering concluirá con un método basado en grafos que utiliza la matriz de similaridad para construir los distintos agrupamientos. Éste método es muy fácil de entender aunque, como sucede con todos los métodos basados en grafos, consume demasiados recursos.

---



---

### *Algunas medidas de distancia utilizadas frecuentemente*

---



---

1. *Distancia euclídea:*  $d^2(X_i, X_j) = (X_i - X_j)^T (X_i - X_j) = \|X_i - X_j\|^2$
  2. *Distancia euclídea normalizada:* Igual que la anterior, salvo que los valores de cada variable se normalizan en el intervalo [0,1] para que unas características no influyan más que otras al calcular las distancias.
  3. *Distancia euclídea ponderada:*  $d_w^2(X_i, X_j) = \sum_{k=1}^d \frac{1}{\sigma_k^2} (X_{ki} - X_{kj})^2$
  4. *Distancia de Mahalanobis:*  $r^2(X_i, X_j) = (X_i - X_j)^T \Sigma^{-1} (X_i - X_j)$   
donde  $\Sigma^{-1}$  es la inversa de la matriz de covarianza.
- 
-

## 2.1 Agrupamiento mediante enfriamiento simulado [Simulated Annealing]

El problema del agrupamiento se puede resolver con cualquier técnica general que permita realizar una exploración del espacio de búsqueda (el espacio de las posibles asignaciones). Una posible forma de resolverlo (bastante ineficiente por cierto) se basa en la utilización de un algoritmo de enfriamiento simulado:

| <i>Simulated Annealing</i>                        | <i>Clustering</i>                                  |
|---|--|
| Sólido que se enfría                              | Problema de optimización                           |
| Configuración actual                              | Asignación actual de patrones a clusters           |
| Perturbación de la configuración actual           | Generación de una nueva asignación                 |
| Energía E asociada a la configuración             | Función objetivo J                                 |
| Aceptación de la configuración si disminuye E     | Aceptación de la asignación si mejora J            |
| Aceptación de una configuración con mayor energía | Aceptación de la asignación si se espera una mejor |

Nuestra función objetivo puede ser, por ejemplo, la suma de las distancias al cuadrado de los patrones a los centroides de los agrupamientos (la función que minimiza el K-Means).

Simplemente hemos de definir un método de generación de vecinos (construcción de una nueva asignación a partir de la asignación actual), fijar un esquema de enfriamiento (constante proporcional en este caso), establecer el valor de temperatura inicial, escoger la temperatura final del proceso de enfriamiento y, por último, tener algo de paciencia para obtener buenos resultados...

### Generación de una nueva asignación

```
flag = falso
Mientras flag = falso
  Para i de 1 a n
    Generar un número aleatorio  $u \sim U(0,1)$ 
    Si  $u > P$ 
      flag = true
       $S_i = \{ c \mid 1 \leq c \leq C, c \neq a_i \}$ 
      Seleccionar aleatoriamente un elemento c de  $S_i$ 
      Asignar el patrón i al cluster c [ $a_i := c$ ]
```

## Algoritmo de agrupamiento con enfriamiento simulado

---

Inicialización:

- Fijar  $T_0$  (temperatura inicial),  $\varepsilon$  (temperatura final) y  $\mu$  (factor [ $<1$ ])
- Seleccionar un conjunto arbitrario de  $C$  clusters (vectores  $A_{\text{best}}$  y  $A_{\text{candidate}}$ )
- Calcular la función objetivo:  $J_{\text{best}}$  y  $J_{\text{candidate}}$
- $T = T_0$

Mientras  $T \geq \varepsilon$

    Generar una nueva asignación  $A_{\text{trial}}$  (con valor  $J_{\text{trial}}$ )

    Si  $J_{\text{trial}} > J_{\text{candidate}}$

        Generar un número aleatorio  $y \sim U(0,1)$

        Si  $y \leq \exp(-(J_{\text{trial}} - J_{\text{candidate}})/T)$

$J_{\text{candidate}} = J_{\text{trial}}$

$A_{\text{candidate}} = A_{\text{trial}}$

    si no

$A_{\text{candidate}} = A_{\text{trial}}$

$J_{\text{candidate}} = J_{\text{trial}}$

        Si  $J_{\text{trial}} < J_{\text{best}}$

$A_{\text{best}} = A_{\text{trial}}$

$J_{\text{best}} = J_{\text{trial}}$

            count = 0

        si no

            count++

    Si count  $\geq N$

$T = \mu T$

---

El enfriamiento simulado es menos dependiente de la configuración inicial de los clusters que el algoritmo K-MEANS y, además, no se queda estancado en óptimos locales. El esquema de enfriamiento  $T_i = 0.9T_{i-1}$  obtiene prácticamente los mismos resultados que otros criterios de convergencia más lenta (vg: criterio de Cauchy, criterio de Boltzmann...). Aun a riesgo de realizar una peor exploración del espacio de búsqueda, en muchas ocasiones es preferible obtener una buena solución en un tiempo limitado.

El esquema de generación de vecinos empleado para el algoritmo SA [Simulated Annealing] tiende a mantener siempre una fracción  $1/k$  de las muestras en cada uno de los  $k$  clusters, algo que no sucede con el K-MEANS. Además, no todos los agrupamientos han de tener el mismo número de casos.

En definitiva, el enfriamiento simulado es mucho más lento que el sencillo algoritmo de las  $K$  medias y suele obtener peores resultados.

---

## 2.2 Método adaptativo [*adaptive sample set construction*]

El método adaptativo es un algoritmo heurístico de agrupamiento que se puede utilizar cuando no se conoce de antemano el número de clases del problema. Entre sus ventajas se encuentran su simplicidad y eficiencia. Además, las observaciones se procesan secuencialmente. Por desgracia, su comportamiento está sesgado por el orden de presentación de los patrones y presupone agrupamientos compactos separados claramente de los demás.

El primer agrupamiento se escoge arbitrariamente. Se asigna un patrón a un cluster si la distancia del patrón al centroide del cluster no supera un umbral. En caso contrario, se crea un nuevo agrupamiento.

Este algoritmo incluye una clase de rechazo a la hora de clasificar observaciones. Los patrones se asignan por la regla de mínima distancia. Algunos patrones no son clasificados si el cuadrado de la distancia al agrupamiento más cercano es mayor que el umbral  $\tau$ .

---

---

### Parámetros

|           |                                   |
|-----------|-----------------------------------|
| $\tau$    | Umbral de distancia (al cuadrado) |
| $\theta$  | Fracción (entre 0 y 1)            |
| $\{X_i\}$ | Conjunto de patrones              |

### Variables

|       |                                |
|-------|--------------------------------|
| A     | Número actual de agrupamientos |
| $Z_i$ | Centroide del agrupamiento i   |

### Algoritmo de agrupamiento

Inicialización:  $A=1$ ,  $Z_1=X_1$

Mientras queden patrones por asignar

Obtener el siguiente patrón X y calcular  $d(X, Z_i)$   $i=1..A$ .

Asignar X al agrupamiento más cercano  $Z_i$  si  $d(X, Z_i) \leq \theta\tau$

Formar un nuevo agrupamiento con X si  $d(X, Z_i) > \tau$  :  $A++$ ,  $Z_A=X$

Recalcular el centroide  $Z_i$  y la varianza  $C_i$  del agrupamiento.



### Resultados obtenidos

La tabla siguiente recoge los resultados obtenidos con la aplicación del algoritmo de agrupamiento adaptativo a un conjunto de cuatro imágenes 256x256 de una galaxia espiral (correspondientes a las bandas R, G, B e IR). La columna “bondad estimada” corresponde al porcentaje de clasificación obtenido utilizando los agrupamientos obtenidos para clasificar las muestras no etiquetadas del conjunto de datos. Para este conjunto de datos se establecen tres clases: estrellas, brazos de la galaxia y fondo.

| $\tau$       | $\theta$ | Agrupamientos | Bondad estimada |
|--------------|----------|---------------|-----------------|
| 10           | 8        | 347           | 74.89 %         |
| 25           | 8        | 284           | 81.83 %         |
| 50           | 8        | 202           | 87.34 %         |
| 75           | 8        | 176           | 87.79 %         |
| 100          | 5        | 152           | 90.81 %         |
| 100          | 8        | 151           | 90.81 %         |
| 100          | 10       | 150           | 91.02 %         |
| 150          | 5        | 122           | 92.65 %         |
| 150          | 8        | 120           | 92.85 %         |
| 150          | 10       | 117           | 93.26 %         |
| 200          | 8        | 96            | 94.89 %         |
| 250          | 8        | 85            | 95.10 %         |
| 1000         | 5        | 30            | 98.36 %         |
| 1000         | 8        | 28            | 97.95 %         |
| 1000         | 10       | 27            | 98.16 %         |
| 10000        | 8        | 4             | 96.32 %         |
| <b>25000</b> | <b>8</b> | <b>3</b>      | <b>95.91 %</b>  |
| 100000       | 8        | 2             | 94.89 %         |

☞ En esta tabla se puede apreciar como el parámetro fundamental a la hora de conseguir un “buen” agrupamiento con este algoritmo es el umbral  $\tau$ . Cuanto mayor sea este umbral, menos agrupamientos se formarán.

☞ El parámetro  $\theta$  influye menos en el comportamiento final del método adaptativo.

### 2.3 Algoritmo de Batchelor y Wilkins (algoritmo de máxima distancia)

Como sucedía con el método adaptativo, el algoritmo de Batchelor y Wilkins es un método de agrupamiento con número de clases desconocido.

---

---

#### *Parámetros*

f      Fracción de la distancia media entre agrupamientos

#### *Algoritmo de agrupamiento*

Primer agrupamiento: Patrón escogido al azar

Segundo agrupamiento: Patrón más alejado del primer agrupamiento

Mientras se creen nuevos agrupamientos

    Obtener el patrón más alejado de los agrupamientos existentes  
    (máximo de las distancias mínimas de los patrones a los agrupamientos)

    Si la distancia del patrón escogido al conjunto de agrupamientos  
    es mayor que una fracción de la distancia media entre los  
    agrupamientos, crear un agrupamiento con el patrón seleccionado

Asignar cada patrón a su agrupamiento más cercano

---

---

#### *Ejemplo: Galaxia espiral*

| Fracción $f$ | Agrupamientos | Bondad estimada |
|--------------|---------------|-----------------|
| 0            | 1027          | -               |
| 1            | 10            | 98.16 %         |
| 2            | 3             | 93.67 %         |
| $\geq 0.3$   | 2             | 57.34 %         |

Igual que sucedía con el método adaptativo, los resultados obtenidos dependen en gran medida de los parámetros con los que se llama al algoritmo.

## 2.4 Algoritmo de las K medias (J.B. MacQueen, 1967)

El algoritmo de las K medias (o *K-Means*) es probablemente el algoritmo de agrupamiento más conocido. Es un método de agrupamiento heurístico con número de clases conocido (K). El algoritmo está basado en la minimización de la distancia interna (la suma de las distancias de los patrones asignados a un agrupamiento al centroide de dicho agrupamiento). De hecho, este algoritmo minimiza la suma de las distancias al cuadrado de cada patrón al centroide de su agrupamiento.

El algoritmo es sencillo y eficiente. Además, procesa los patrones secuencialmente (por lo que requiere un almacenamiento mínimo). Sin embargo, está sesgado por el orden de presentación de los patrones (los primeros patrones determinan la configuración inicial de los agrupamientos) y su comportamiento depende enormemente del parámetro K.

---

---

Seleccionar arbitrariamente una configuración inicial de los clusters

Repetir

    Calcular los centros de los clusters  $Z_j$

    Redistribuir los patrones entre los clusters utilizando la mínima distancia euclídea al cuadrado como clasificador:

$$X_i \in C_j \leftrightarrow ||X_i - Z_j||^2 < ||X_i - Z_l||^2 \quad \forall l \neq j$$

Hasta que no cambien los centros de los clusters

---

---

*Ejemplo: Galaxia espiral*

| K | Bondad estimada |
|---|-----------------|
| 3 | 94.89 %         |
| 5 | 92.04 %         |
| 7 | 98.16 %         |

NOTA: La convergencia del algoritmo depende de la configuración inicial de los clusters. El algoritmo es eficiente y encuentra siempre un óptimo local, por lo que podemos utilizarlo como técnica de búsqueda local en algoritmos GRASP [*Greedy Randomized Adaptive Search Procedure*] o GLS [*Genetic Local Search*].

## 2.5 Algoritmo GRASP [Greedy Randomized Adaptive Search Procedure]

GRASP es una técnica de los años 80 que tiene como objetivo resolver problemas difíciles en el campo de la optimización combinatoria. Esta técnica dirige la mayor parte de su esfuerzo a construir soluciones de alta calidad que son posteriormente procesadas para obtener otras aún mejores.

Los algoritmos GRASP son algoritmos de tipo iterativo en los que cada iteración incluye una fase de construcción de una solución y otra de postprocesamiento en la cual se optimiza la solución generada en la primera fase.

Se puede establecer una analogía con el problema de la Programación Lineal, donde primero se construye una solución factible y después se aplica el algoritmo Simplex. Sin embargo, en GRASP se le da bastante importancia a la calidad de la solución generada inicialmente.

La estructura básica de un algoritmo GRASP es la siguiente:

---

---

```
Mientras no se satisfaga el criterio de parada
```

```
    Construir una solución greedy aleatoria
```

```
    Aplicar una técnica de búsqueda local a la solución greedy
    aleatoria obtenida en el paso anterior (para mejorarla)
```

```
    Actualizar la mejor solución encontrada
```

---

---

Se puede extender este algoritmo si se le añade un operador de mutación (mecanismo de generación de vecinos) que introduzca mayor diversidad en la búsqueda realizada por el procedimiento GRASP básico:

---

---

```
Mientras no se satisfaga el criterio de parada
```

```
    Solución = Solución greedy aleatoria
```

```
    Para i de 1 a L
```

```
        Solución = Búsqueda local (Solución)
```

```
        Actualizar la mejor solución (si corresponde)
```

```
        Solución = Mutación(Solución)
```

---

---

### *Algoritmo greedy*

Como algoritmo greedy se puede utilizar una versión simplificada del algoritmo de Batchelor y Wilkins. Como centros de los agrupamientos se escogerán patrones del conjunto de entrenamiento de forma que el patrón escogido en cada momento sea el más alejado a los centroides ya fijados (siendo la distancia a un conjunto de centroides el mínimo de las distancias a cada centroide). Obviamente, el primer centroide ha de escogerse de forma aleatoria.

### *Mecanismo de generación de soluciones greedy aleatorias*

En la construcción de soluciones greedy aleatorias se utiliza una lista de candidatos [*RCL: Restricted Candidate List*] en la que se incluyen los mejores aspirantes a formar parte de la solución. De esa lista se escoge un candidato aleatoriamente.

Un posible algoritmo greedy aleatorio consiste en utilizar como RCL la lista de los patrones más alejados a los centroides ya escogidos. El tamaño de esta lista puede ser, por ejemplo, igual al 5% de las muestras disponibles.

El algoritmo greedy aleatoria es análogo al algoritmo greedy clásico, teniendo en cuenta que, en cada momento, se escoge aleatoriamente uno de los patrones más alejados al conjunto de centroides ya establecido.

### *Técnica de búsqueda local*

La estrategia de búsqueda local se utiliza para mejorar la solución obtenida mediante en el mecanismo de generación de soluciones greedy aleatorias. Mientras la solución no sea un óptimo local, se encuentra una solución mejor entre los vecinos de la solución actual.

Como técnica de búsqueda local se puede emplear, por ejemplo, el conocido algoritmo de las K medias, cuya convergencia depende de la configuración inicial de los clusters.

### *Operador de mutación (algoritmo GRASP extendido)*

En el algoritmo GRASP extendido es necesario considerar un operador de mutación que se aplicará sobre la solución actual cada vez que finalice la ejecución del algoritmo de búsqueda local.

Una solución se representa por un vector en el cual la componente  $i$ -ésima indica el agrupamiento al que pertenece el patrón  $i$ -ésimo. Para mutar una solución se altera cada componente del vector solución con una probabilidad  $P$ . Como es lógico, las componentes del vector se mantendrán con una probabilidad  $1-P$ . Como valor de  $P$  escogeremos 0.3 para aplicar una mutación fuerte que permita una buena exploración del espacio de búsqueda.

---

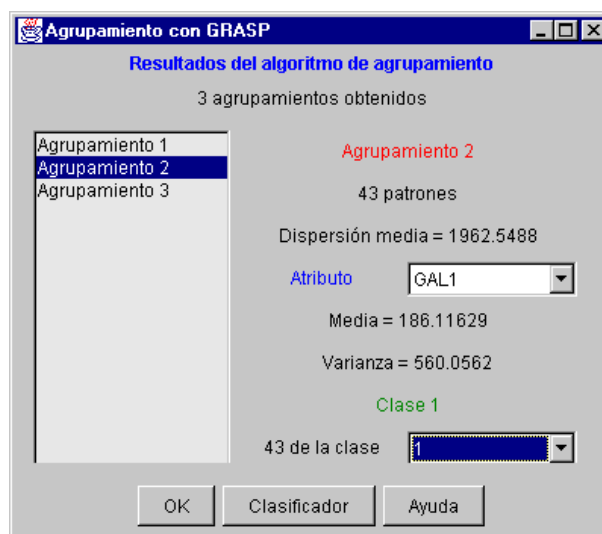
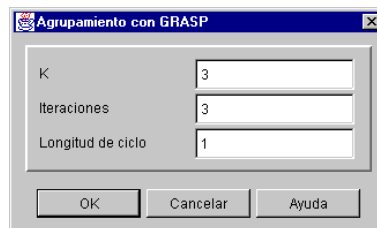
### *Pseudocódigo del operador de mutación*

---

```
flag = falso
Mientras flag = falso
    Para i de 1 a n
        Generar un número aleatorio  $u \sim U(0,1)$ 
        Si  $u > P$ 
            flag = true
             $S_i = \{ c \mid 1 \leq c \leq C, c \neq a_i \}$ 
            Seleccionar aleatoriamente un elemento  $c$  de  $S_i$ 
            Asignar el patrón  $i$  al cluster  $c$  [ $a_i := c$ ]
```

---

Los algoritmos GRASP no requieren estructuras de datos especialmente complejas y conducen siempre a buenas soluciones. De hecho, con la base de datos de la galaxia espiral, con ejecutar este algoritmo unas pocas iteraciones (sin necesidad de utilizar el operador de mutación) se consigue siempre la división del conjunto de datos en tres agrupamientos que corresponden a las tres clases del problema:



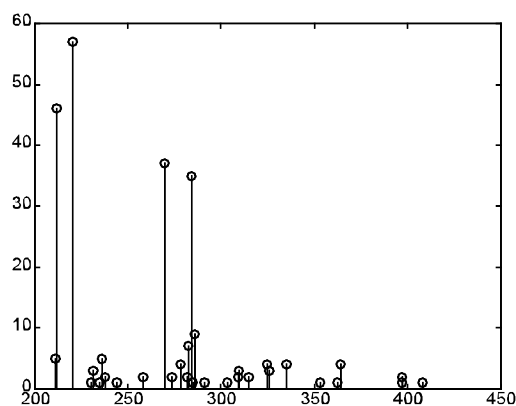
A continuación se mostrarán los valores obtenidos para distintos parámetros a lo largo de las iteraciones de un algoritmos GRASP. Para ilustrar la distribución de estos valores se utilizarán cinco particiones distintas de la **base de datos TITANIC** (por su simplicidad) y el **algoritmo GRASP básico** (sin operador de mutación).

La tabla siguiente recoge el mínimo, el máximo y la moda de las distribuciones de cada uno de los parámetros para las ejecuciones del algoritmo GRASP básico aplicadas sobre la tabla TITANIC cuando el número inicial de clusters es igual a 5 ( $K=5$ ). Recuérdese que el valor del parámetro  $J$  (la suma de las distancias al cuadrado) se está intentando minimizar:

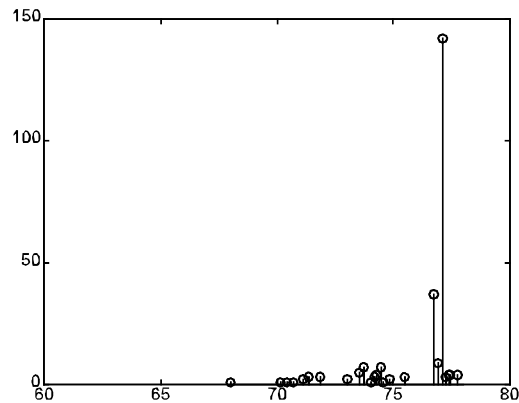
| Datos | J     |       |       | %TRA |      |      | %TST |      |      |
|-------|-------|-------|-------|------|------|------|------|------|------|
|       | min   | mod   | max   | min  | mod  | max  | min  | mod  | max  |
| 1     | 21121 | 22058 | 40842 | 6003 | 7713 | 7778 | 6843 | 7855 | 7961 |
| 2     | 19013 | 22351 | 41061 | 6816 | 7739 | 7895 | 6616 | 7795 | 7931 |
| 3     | 18833 | 22161 | 41907 | 6810 | 7791 | 7953 | 6828 | 7674 | 7795 |
| 4     | 17994 | 22187 | 39960 | 6829 | 7719 | 7862 | 6782 | 7840 | 8006 |
| 5     | 18867 | 22486 | 42428 | 6823 | 7771 | 7869 | 6601 | 7719 | 7825 |

Se puede apreciar a partir de los datos de esta tabla que la moda de las distribuciones de los distintos parámetros suele encontrarse muy cerca del máximo obtenido. Esta moda corresponde con la solución que se obtendría con un algoritmo greedy simple (seguido de la técnica de búsqueda local correspondiente). La inclusión de cierta aleatoriedad en el algoritmo GRASP permite obtener de vez en cuando soluciones mejores a costa de obtener también algunas soluciones peores que las que se obtendrían con el algoritmo simple.

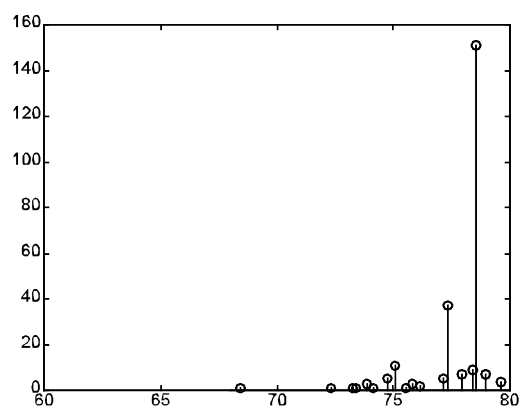
Los diagramas siguientes, construidos utilizando MATLAB, muestran las distribuciones de los distintos valores:



*Distribución de J*



*Porcentaje de clasificación en el conjunto de entrenamiento*



*Porcentaje de clasificación en el conjunto de prueba*

- ✓ Las metaheurísticas como la utilizada en la técnica GRASP permiten mejorar las soluciones obtenidas en problemas complejos. Los algoritmos GRASP consiguen mejorar los resultados obtenidos por el algoritmo de las K medias para el problema del agrupamiento. GRASP consigue soluciones de alta calidad.
- ✓ El uso adecuado de la lista de candidatos RCL (Restricted Candidate List) es esencial para el algoritmo GRASP, ya que de esa forma se consiguen distintos puntos de partida para el algoritmo de búsqueda local concreto, el K-Medias en nuestro caso. Dado que este algoritmo específico del problema es altamente dependiente de la configuración inicial de los clusters, el método de construcción de soluciones greedy aleatorias debe proporcionar buenos puntos de partida en distintas regiones del espacio de búsqueda (lo que consigue con la RCL).
- ✓ En cuanto al algoritmo greedy empleado, se puede apreciar que no obtiene soluciones excesivamente buenas aunque sí proporciona buenos puntos de partida para un algoritmo de búsqueda local como el algoritmo de las K Medias, el cual obtiene resultados netamente superiores a los que obtenía cuando se partía de una configuración inicial aleatoria.



## 2.6 Algoritmo de agrupamiento secuencial

La utilización del método de agrupamiento secuencial es algo más compleja que la de algoritmos como el K-Means o el de Batchelor y Wilkins. Igual que ellos, realiza cálculos sencillos, procesa los patrones secuencialmente y los resultados obtenidos depende del orden de presentación de los patrones (está sesgado por los primeros patrones).

Su funcionamiento puede ajustarse gracias a un amplio conjunto de parámetros. Los valores adecuados para los parámetros son difíciles de establecer a priori, por lo que se suele emplear un mecanismo de prueba y error.

| Parámetro | Descripción  |
|-----------|--|
| <b>K</b>  | Máximo número de agrupamientos   |
| <b>R</b>  | Umbral de distancia para crear agrupamientos                                   |
| <b>C</b>  | Umbral de distancia para mezclar agrupamientos                                 |
| <b>M</b>  | Longitud del lote:<br>Número de patrones considerados entre procesos de mezcla |
| <b>T</b>  | Umbral para la eliminación de agrupamientos (porcentaje respecto a M)          |

El algoritmo selecciona arbitrariamente el centro del primer agrupamiento. Posteriormente procesa secuencialmente los demás patrones. Calcula la distancia del patrón actual al agrupamiento más cercano (a su centroide). Si ésta es menor o igual a R se asigna el patrón a su agrupamiento más cercano. En caso contrario, se crea un nuevo agrupamiento con el patrón actual.

Cada M patrones, se mezclan agrupamientos por cercanía (se mezclan dos agrupamientos si la distancia entre ellos es menor que C). Si, tras la mezcla por cercanía, quedan más agrupamientos que los deseados por el usuario (K), se mezclan los agrupamientos por tamaño (se mezclan los agrupamientos con menos del T% de M miembros con sus agrupamientos más cercanos). Si aún quedan demasiados agrupamientos, se mezclan los agrupamientos más cercanos hasta obtener el número deseado de agrupamientos K.

El proceso de mezcla nos asegura que al final obtenemos el número deseado de agrupamientos y no más (como solía suceder en el método adaptativo o en el algoritmo de Batchelor y Wilkins).

## 2.7 Algoritmo ISODATA

ISODATA es el acrónimo de *Iterative Self-Organizing Data Analysis Techniques* (con la A añadida para hacer pronunciable el nombre), un iterativo método de agrupamiento que, como ya sucedía con el método de agrupamiento secuencial, requiere un considerable esfuerzo para ajustar adecuadamente todos sus parámetros. Además, éstos pueden modificarse en cada iteración del algoritmo.

| Parámetro | Descripción  |
|-----------|--|
| <b>K</b>  | Número deseado de agrupamientos                                    |
| <b>A</b>  | Número inicial de agrupamientos                                    |
| <b>n</b>  | Umbral del número de patrones para la eliminación de agrupamientos |
| <b>s</b>  | Umbral de desviación típica para la división de un agrupamiento    |
| <b>c</b>  | Umbral de distancia para la unión de agrupamientos                 |
| <b>L</b>  | Máximo número de mezclas en una iteración                          |
| <b>I</b>  | Máximo número de iteraciones permitidas                            |

---

---

*Inicialmente se seleccionan los centros de A agrupamientos.*

*En cada iteración*

*Se fijan los valores de los distintos parámetros del algoritmo*

*Se asigna cada patrón al agrupamiento más cercano*

*Se eliminan los agrupamientos con menos de n patrones*

*Si el número actual de agrupamientos es pequeño (menor o igual que  $K/2$ ), dividimos los agrupamientos más dispersos (siendo la dispersión de un agrupamiento la distancia media de sus patrones al centroide del cluster) por la componente de máxima dispersión (respetando el umbral mínimo s).*

*En las iteraciones pares o cuando el número actual de agrupamientos es elevado ( $>2K$ ), unimos como máximo L pares de agrupamientos cuya separación entre ellos quede por debajo del umbral de distancia c*

## 2.8 Métodos basados en grafos: Matriz de similitud

El principal inconveniente de la mayor parte de los algoritmos heurísticos es su dependencia del orden en que se le presentan los patrones. Los métodos basados en grafos, igual que los algoritmos GRASP, intentan evitar este hecho pero su coste computacional los hace inaplicables en muchas ocasiones.

### *La matriz de similitud*

La matriz de similitud se emplea para mostrar el grado de similitud entre un conjunto de patrones. Se construye una matriz  $S$  simétrica de tamaño  $N \times N$ , siendo  $N$  el número de patrones del conjunto de entrenamiento.  $S[i,j]$  toma el valor 1 si la distancia entre los patrones  $i$  y  $j$  queda por debajo de un umbral preestablecido  $\theta$ . En caso contrario,  $S[i,j]$  vale 0. Por lo tanto, con un bit por celda podemos almacenar la matriz de similitud.

### *Algoritmo de agrupamiento basado en la matriz de similitud*

---

---

Mientras queden patrones en la matriz de similitud  $S$

Selecciónar la fila  $i$  de la matriz de similitud  $S$  que contenga más unos. Si hay varias, escoger una al azar.

Crear un agrupamiento con los patrones  $j$  tales que  $S[i,j] = 1$

Añadir al agrupamiento todos aquellos patrones  $k$  tales que  $S[j,k] = 1$ , siendo  $j$  un patrón incluido en el nuevo agrupamiento hasta que no se puedan añadir más patrones a dicho agrupamiento.

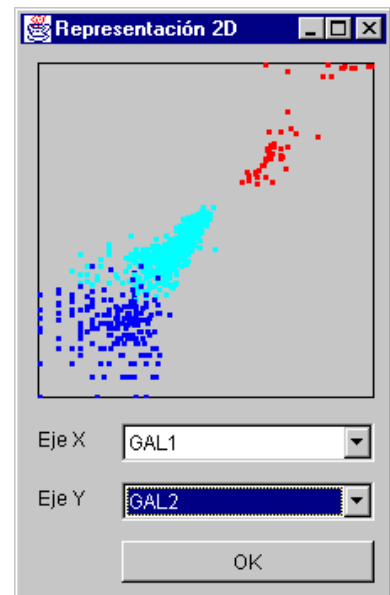
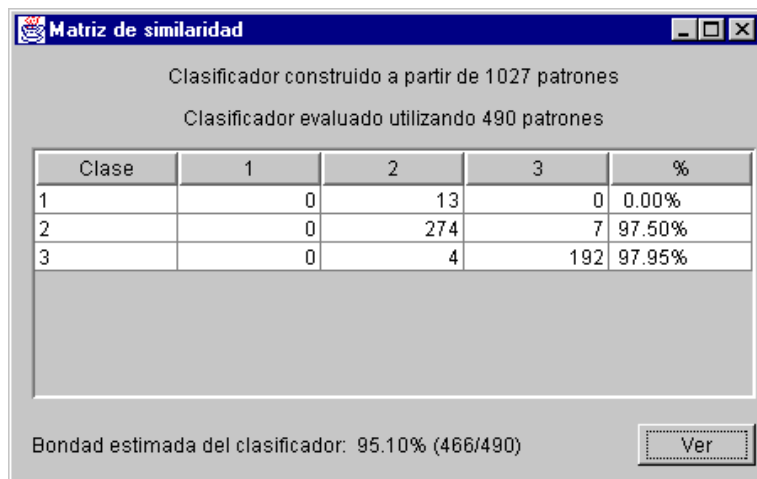
Reducir la matriz de similitud: Eliminar de  $S$  todas las filas y columnas correspondientes a patrones incluidos en el agrupamiento recién creado.

---

---

*Ejemplo: Galaxia espiral*

| Umbral de distancia | Agrupamientos | Bondad estimada |
|---------------------|---------------|-----------------|
| 100                 | 86            | 92.04 %         |
| 500                 | 10            | 95.10 %         |
| 1000                | 4             | 93.06 %         |
| 1500                | 3             | 91.83 %         |
| 2000                | 1             | 57.34 %         |



*Resultado obtenido con el umbral de distancia igual a 500*

El método de agrupamiento basado en la matriz de similitud produce un resultado curioso, si no sorprendente, cuando es aplicado a la base de datos de la galaxia espiral. A partir de un umbral de distancia relativamente pequeño, no se consigue ningún agrupamiento correspondiente a la clase 1 debido a la distribución de las muestras en el conjunto de entrenamiento (que se puede visualizar en la imagen de la derecha).

### 3. Bibliografía

*Métodos de agrupamiento clásicos (vg: DHB, DHF, algoritmo de Forgy):*

- A.K. Jain & R.C. Dubes: “Algorithms for Clustering Data”. Prentice Hall, 1988.
- M.A. Ismael & M.S. Kamel: “Multidimensional Data Clustering Utilizing Hybrid Search Strategies”. Pattern Recognition 22, 1989, pgs. 75-89.

*Metaheurísticas (enfriamiento simulado y GRASP):*

- Adenso Díaz (coordinador), Fred Glover, Hassan M. Ghaziri, J.L. González, Manuel Laguna, Pablo Moscato, Fan T. Tseng: “Optimización Heurística y Redes Neuronales”. Madrid: Editorial Paraninfo, 1996.

*Métodos de agrupamiento mediante enfriamiento simulado:*

- S.Z. Selim & K. Alsultan: “A Simulated Annealing for the Clustering Problem”. Pattern Recognition 24, 1991, pgs. 1003-1008.
- R.W. Klein & R.C. Dubes: “Experiments in Projection and Clustering by Simulated Annealing”. Pattern Recognition 22, 1989, pgs. 213-220.