Máster en Ingeniería de Sistemas y de Control

HERRAMIENTA DE ANÁLISIS DE VALIDEZ DE PROCESOS DE CLUSTERING

***Autor**: Angel R. Mur Güerri*
***Director**: Raquel Dormido Canto*
Natividad Duro Carralero
***Curso**: 2010-2011*
***Convocatoria**: Septiembre 2011*

Máster en Ingeniería de Sistemas y de Control

HERRAMIENTA DE ANÁLISIS DE VALIDEZ DE PROCESOS DE CLUSTERING

Proyecto Tipo A

***Autor**: Angel R. Mur Güerri*
***Director**: Raquel Dormido Canto*
Natividad Duro Carralero

Autorización

Autorizamos a la Universidad Complutense y a la UNED a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la memoria de este Trabajo Fin de Máster, como el código, la documentación y/o el prototipo desarrollado.

Firmado: **Angel R. Mur Güerri**

Firma del alumno

RESUMEN

Clustering es una técnica de clasificación no supervisada que clasifica un conjunto de datos en diferentes grupos sin que se tenga conocimiento previo de las características de dichos grupos. Cada grupo contiene datos similares que son diferentes a los de los otros grupos.

Existen numerosos campos en donde un proceso de clustering está presente: reconocimiento de formas, minería de datos, aprendizaje automático.

Se han desarrollado numerosos algoritmos que pueden resolver el problema de clustering pero la mayoría son muy sensibles a los parámetros iniciales. Por lo tanto es muy importante evaluar los resultados de estos algoritmos. Sin dicha validación no se podría dar por bueno un determinado resultado. Además es difícil definir si un resultado de un proceso clustering es aceptable o no, y por lo tanto se han desarrollado numerosas técnicas e índices de validación.

En un proceso de validación se evalúa la calidad del resultado y se da respuestas a preguntas como, ¿cuántos clusters hay en un conjunto de datos?, ¿el resultado de un algoritmo de clustering concuerda con los datos?, ¿existe una partición mejor de los datos?.

La evaluación de los resultados de un proceso de clustering es una etapa importante y necesaria. Sin embargo, requiere de mucho tiempo ya que existen varios aspectos que deben ser tratados con cuidado: procesado de los datos, elección del número de clusters, índices de validación etc. Por consiguiente, se hace necesaria una herramienta para analizar los resultados de un proceso de clustering que evalúe dichos resultados en poco tiempo e integre las diferentes técnicas más representativas, con el fin de estandarizar la evaluación y además evitar posibles errores.

En este trabajo de fin de máster se ha realizado una herramienta para automatizar el proceso de clustering y su posterior validación. El proceso de evaluación empieza con la lectura de un conjunto de datos y culmina con la posibilidad de obtener una solución consensuada respecto al número de clusters y su contenido.

***Palabras clave**: Clustering tendency, clustering determination, clustering validación, clustering stability, clustering consensus, criterio interno, criterio relativo, criterio externo, número óptimo de clusters, interfaz gráfica.*

Índice

CAPÍTULO 1: Introducción 1
1.1 Proceso general de validación 1
1.2 Bibliografía 1

CAPÍTULO 2: Tendencia del proceso de clustering 2
2.1 Introducción 2
2.2 Analisis en componentes principales 2
2.3 El mapeo de Sammon 2
2.4 SOM 4
2.5 Vat y reVat 5
2.6 Estadística de Hopkins 6
2.7 Resumen 7
2.8 Bibliografía 8

CAPÍTULO 3: Determinación del proceso de clustering 9
3.1 Introducción 9
3.2 Método K-meams 9
3.3 Método Jerárquico 10
3.4 Método EM 11
3.5 Método Fuzzy C-means 11
3.6 Método SOM 12
3.7 Resumen 16
3.8 Bibliografía 16

CAPÍTULO 4: Técnicas de validación y número óptimo de clusters 17
4.1 Introducción 17
4.2 Criterio Relativo 19
4.2.1 Estadística de Hubert Modificada 20
4.2.2 Índice Silhouette 20
4.2.3 Índice Dunn 21
4.2.4 Índice Davies-Bouldin 21
4.2.5 Índice de Compacidad (CP) 21
4.2.6 Índice Calinski-Harabasz 22
4.2.7 Índice R-Squared 22
4.2.8 Índice SD 23
4.2.9 Índice S_Dbw 24
4.2.10 Criterio de Información Bayesiana 25
4.2.11 Coeficiente de Partición 25
4.2.12 Entropía de Clasificación 26
4.2.13 Índice de Partición 26
4.2.14 Índice de Xie y Beni 26
4.3 Criterio Interno 27
4.4 Criterio Externo 29
4.5 Resumen 30
4.6 Bibliografía 30

CAPÍTULO 5: Estabilidad del proceso de clustering y número óptimo de clusters 32
5.1 Introducción 32
5.2 Estabilidad 32
5.3 Resumen 34
5.4 Bibliografía 34

CAPÍTULO 6: Solución de clustering consensuada 35
6.1 Introducción 35
6.2 Solución consensuada 36
6.2.1 Algoritmo de similitud por parejas 36
6.3 Resumen 37
6.4 Bibliografía 37

CAPÍTULO 7: Interfaz gráfica de integración 38
7.1 Introducción 38
7.2 Interfaz gráfica 38
7.3 Resumen 53

CAPÍTULO 8: Conclusión 55

LISTA DE FIGURAS

Figura 1: Representacion bidimensional de los datos Iris utilizando un PCA. 3
Figura 2: Reprepresentacion bidimensional de los datos Iris utilizando Sammon. 3
Figura 3: Visualización SOM de los datos Iris: Umatrix, planos de los componentes y labels. 4
Figura 4: Matriz de similitud de los datos desordenados. 5
Figura 5: Matriz de similitud tras aplicar el algoritmo VAT. 6
Figura 6: Matriz de similitud después de aplicar el algoritmo reVat. 6
Figura 7: Aprendizaje de las neuronas próximas a la neurona ganadora 14
Figura 8: Descripción de los criterios interno, externo y relativo para analizar la calidad de un algoritmo para agrupar datos. 17
Figura 9: Criterios de validación y algunas de sus estadísticas e índices [Data Clustering]. 18
Figura 10: Proceso de búsqueda de una partición consensuada. 35
Figura 11: Interfaz gráfica para validar un proceso clustering. 38
Figura 12: Sección Número 1 de la interfaz gráfica. 39
Figura 13: Sección Número 2 de la interfaz gráfica. 39
Figura 14: Ventana emergente tras seleccionar el análisis de la estadística de Hopkins. 40
Figura 16: Ventana emergente tras seleccionar el análisis del coeficiente cophenético. 41
Figura 17: Sección Número 4 de la interfaz gráfica. 41
Figura 18: Sección Número 5 de la interfaz gráfica. 42
Figura 19: Sección Número 6 de la interfaz gráfica. 42
Figura 21: Ventana emergente, donde se muestran algunos índices de validación relativos: Silhouete, Dunn, CP y Calinski-Harabasz. 43
Figura 22: Gráficas obtenidas tras seleccionar los índices Hubert y R-squared. 44
Figura 23: Ventana emergente, donde se muestran algunos índices de validación relativos: Davies-Bouldin, SD y S_Dbw. 45
Figura 24: Selección del algoritmo EM en la sección Número 5 de la interfaz gráfica. 45
Figura 25: Ventana emergente, donde se muestra el índice BIC relacionado con el algoritmo EM. 46
Figura 26: Selección en la interfaz gráfica del algoritmo Fuzzy C-means y sus índices. 46
Figura 27: Ventana emergente, donde se muestra algunos índices de validación relativos para Fuzzy C-means: PC, CE, SC y XB. 47
Figura 28: Sección Número 8 de la interfaz gráfica. 47
Figura 29: Ventana emergente, donde se muestra el análisis de estabilidad utilizando los índices: Rand, Adjusted Rand, Jaccard y Folkes. 48
Figura 30: Sección Número 9 de la interfaz gráfica 49
Figura 31: Representacion gráfica del resultado clustering después de aplicar SOM para K=2. 50
Figura 32: Elección particular de SOM en la sección número 9 de la interfaz gráfica (mismo K). 50
Figura 33: Elección particular de SOM en la sección número 9 de la interfaz gráfica (distinto K). 51
Figura 34: Elección particular de K-means en la sección número 9 de la interfaz (mismo K). 51
Figura 35: Elección particular de K-means en la sección número 9 de la interfaz (distinto K). 51
Figura 36: Elección particular de todos los métodos en la sección número 9 de la interfaz gráfica. 51
Figura 37: Ventana emergente tras pulsar el boton Fusión (Análisis del coeficiente cophenético). 52
Figura 38: Ventana emergente tras pulsar el boton Índices Fusión (Análisis de validación relativo). 52
Figura 39: Ventana emergente tras pulsar sobre el boton "ver contenido". 53

LISTA DE TABLAS

Tabla 4.1: Valores del coeficiente cophenético para diferentes combinaciones de distancia y linkage (datos Iris). 28
Tabla 4.2: Índices externos más utilizados. 29

1. INTRODUCCIÓN

El proceso de Clustering es una técnica de clasificación no supervisada. Como consecuencia, los clusters hallados después de aplicar un algoritmo necesitan ser evaluados. El proceso de evaluación de los resultados de un algoritmo clustering es conocido como validación de clusters (CV).

1.1 Proceso general de validación

Una vez se tienen los patrones respectivos de todos los datos/objetos que pretenden ser agrupados se pueden diferenciar 4 etapas en el proceso de validación de un Proceso Clustering:

A) Tendencia: El proceso de validación ya empieza antes de aplicar cualquier algoritmo de agrupamiento. El objetivo es conocer de antemano si existe la posibilidad de que los datos sean agrupados, es decir si los datos poseen una estructura no aleatoria. Con esto se consigue saber si vale la pena aplicar los algoritmos de clustering. Esta etapa se denomina "Clustering tendency (CT)".

B) Determinación: Posterior a CT, se aplica un algoritmo para agrupar los datos. Exiten varios algoritmos aunque ninguno garantiza que los grupos obtenidos sean válidos. Por consiguiente, se aplican diferentes criterios o índices para justificar la validación y poder comparar dichos algoritmos. También se obtiene indirectamente el número óptimo de clusters.

C) Estabilidad: El proceso de validación continúa comprobando la estabilidad de los resultados, es decir que los grupos obtenidos son estables: el resultado no es fruto del azar y por lo tanto estable (pequeños cambios en los datos/objetos producen pequeños cambios en los grupos encontrados).

D) Solución consensuada: En esta última etapa se fusionan los mejores resultados de cada uno de los algoritmos utilizados para obtener un mejor agrupamiento de los datos.

En la presente memoria se va a analizar cada uno de los procesos. Se describirán los métodos más representativos de cada etapa y se desarrollará una interfaz gráfica para integrarlos.

Para ilustrar las capacidades de los diferentes algoritmos se utilizarán los datos Iris [Newman], debido a que son apropiados para agrupamiento de datos y clasificación. Los datos están relacionados con 3 especies de flores (Iris-Setosa, Iris-Versicolor, y Iris-Virginica) caracterizadas por 4 atributos: longitud y anchura del pétalo, longitud y anchura del sépalo. Son un conjunto de 150 observaciones en donde cada clase contiene 50 flores. La clase Iris-Setosa es linealmente separable de las otras 2 clases, pero la Iris-Versicolor y Iris-Virginica no son clases separables linealmente.

1.2 Bibliografía

[Newman] Newman, D. J., Hettich, S., Blake, C. L. and Merz, C. J., 1998. UCI Repository of machine learning databases, http://www.ics.uci.edu/~mlearn/MLRepository.html.

2. TENDENCIA DEL PROCESO DE CLUSTERING

2.1 Introducción

"Clustering" o "Cluster Analysis" es una técnica de clasificación que permite agrupar una colección de datos/objetos $O=[o_1,...,o_n]$ en grupos de datos/objetos similares, utilizando los diferentes valores que los caracterizan (patrones) y una medida de similitud.

Existen numerosos algoritmos para encontrar dichos grupos. Todos ellos descubrirán un número arbitrario de "clusters", incluso si en la realidad éstos no existen. Por lo tanto es importante antes de aplicar cualquier algoritmo de "Clustering" preguntarse si hay o no clusters a determinar.

El problema de analizar si hay clusters presentes es denominado "assessing of clustering tendency".

Lo más sencillo es proyectar los objetos en un espacio de 2 dimensiones. Los métodos de proyección son particularmente útiles ya que facilitan la representación visual de los datos. Las técnicas más representativas son: El Análisis en Componentes Principales (ACP) [Sharma] y el mapeo de Sammon [Sammon]. Otras técnicas más recientes son: "Self Organizing Maps (SOM)" [Vesanto], "Radviz" [Hoffman] y "Star Coodinates" [Kandogan].

Otros métodos visuales que pueden proporcionar más y mejor información son los denominados VAT y reVAT [Bezdek].

También se han desarrollado algunas técnicas estadísticas para valorar la existencia o no de clusters [Jain] [Everit]. De entre ellas, destaca la técnica estadística de Hopkins [Hopkins].

2.2 Análisis en Componentes Principales

PCA es una técnica clásica que transforma datos/objetos multidimensionales en otros con menos dimensiones [Sharma]. Esta transformación intenta preservar la varianza de los datos lo mejor posible. PCA crea nuevas variables (llamadas componentes principales) que son componentes lineales de las variables originales. El máximo número de nuevas variables es igual al número de variables originales, y las nuevas variables son independientes entre ellas. La figura 1 representa la proyección PCA de los datos Iris utilizando las 2 primeras componentes principales.

2.3 El mapeo de Sammon

El mapeo de Sammon [Sammon] intenta encontrar una representacion 2D o 3D de los puntos objeto multidimensionales, tal que la distancia entre puntos en el mapa resultado sea similar lo más posible a la distancia Euclídea entre los datos originales multidimensionales.

La figura 2 muestra la representación de Sammon de los datos Iris.

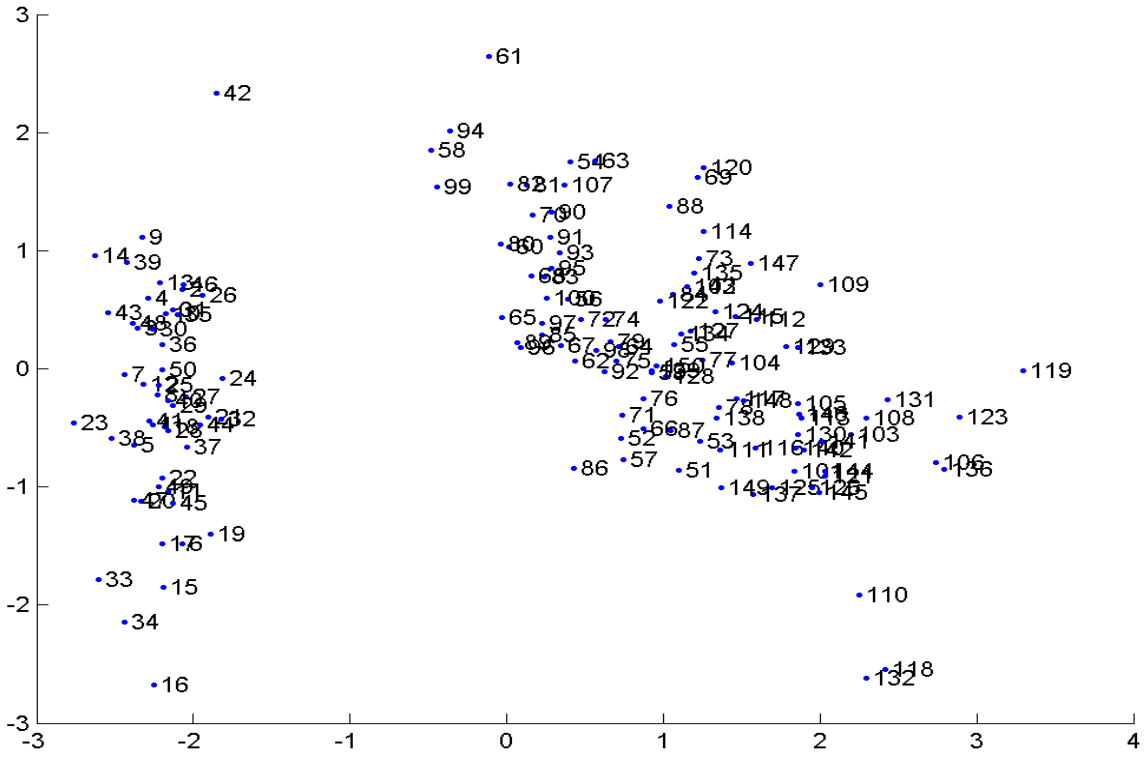


Figura 1: Representacion bidimensional de los datos Iris utilizando un PCA.

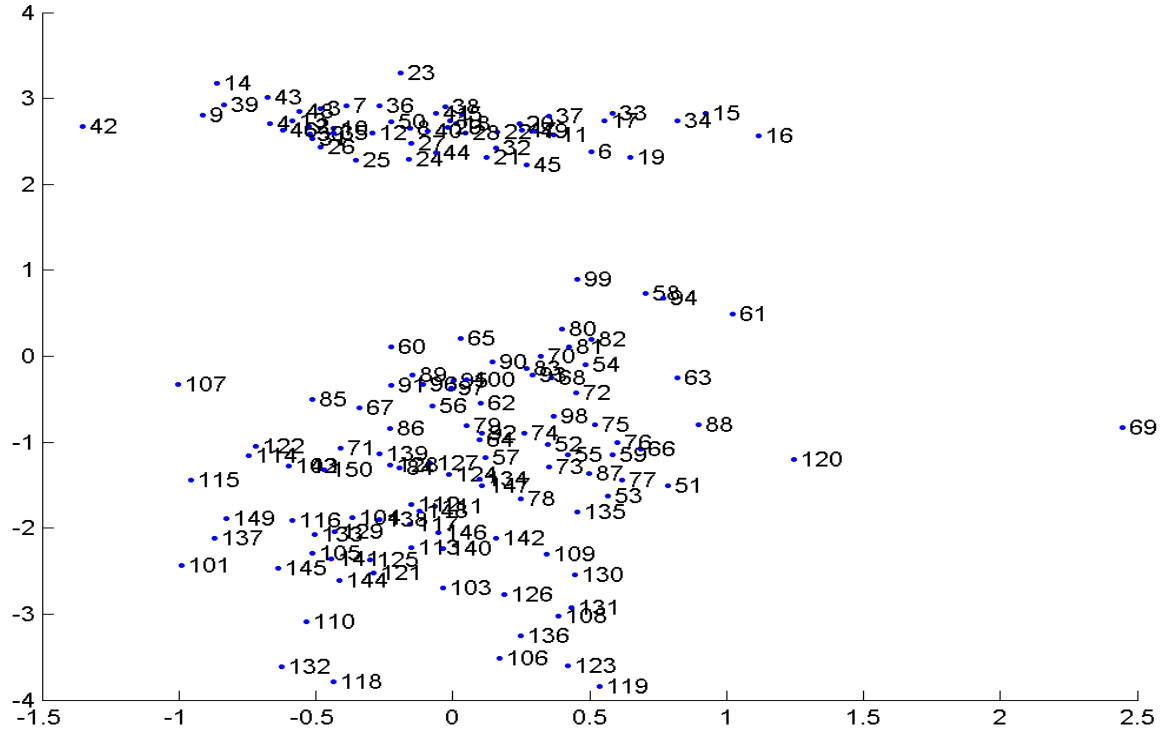


Figura 2: Reprepresentacion bidimensional de los datos Iris utilizando Sammon.

2.4 SOM (Self Organizing Maps)

El método SOM fue propuesto por Kohonen [Kohonen 1], y ha sido ampliamente usado en diversas aplicaciones industriales como reconocimiento de formas, modelización, compresión de datos, procesado de la señal y minería de datos [Kohonen 2]. El éxito del método radica en su simplicidad que hace que sea fácil de entender, simular y pueda ser utilizado en numerosas aplicaciones.

SOM consiste en una colección de neuronas ordenadas sobre una estructura bidimensional de tal forma que hay relaciones de vecindad entre las neuronas. Después del proceso de entrenamiento, cada neurona está caracterizada por un vector de la misma dimensión que los objetos de entrada. Asignando a cada vector objeto la neurona que presenta un vector más cercano, SOM es capaz de dividir el espacio de puntos objeto a agrupar en regiones. Este proceso también se denomina "Vector Quantización" (VQ).

Para mostrar la visualización se ha utilizado el toolbox [SomToolbox](Ver Figura 3).

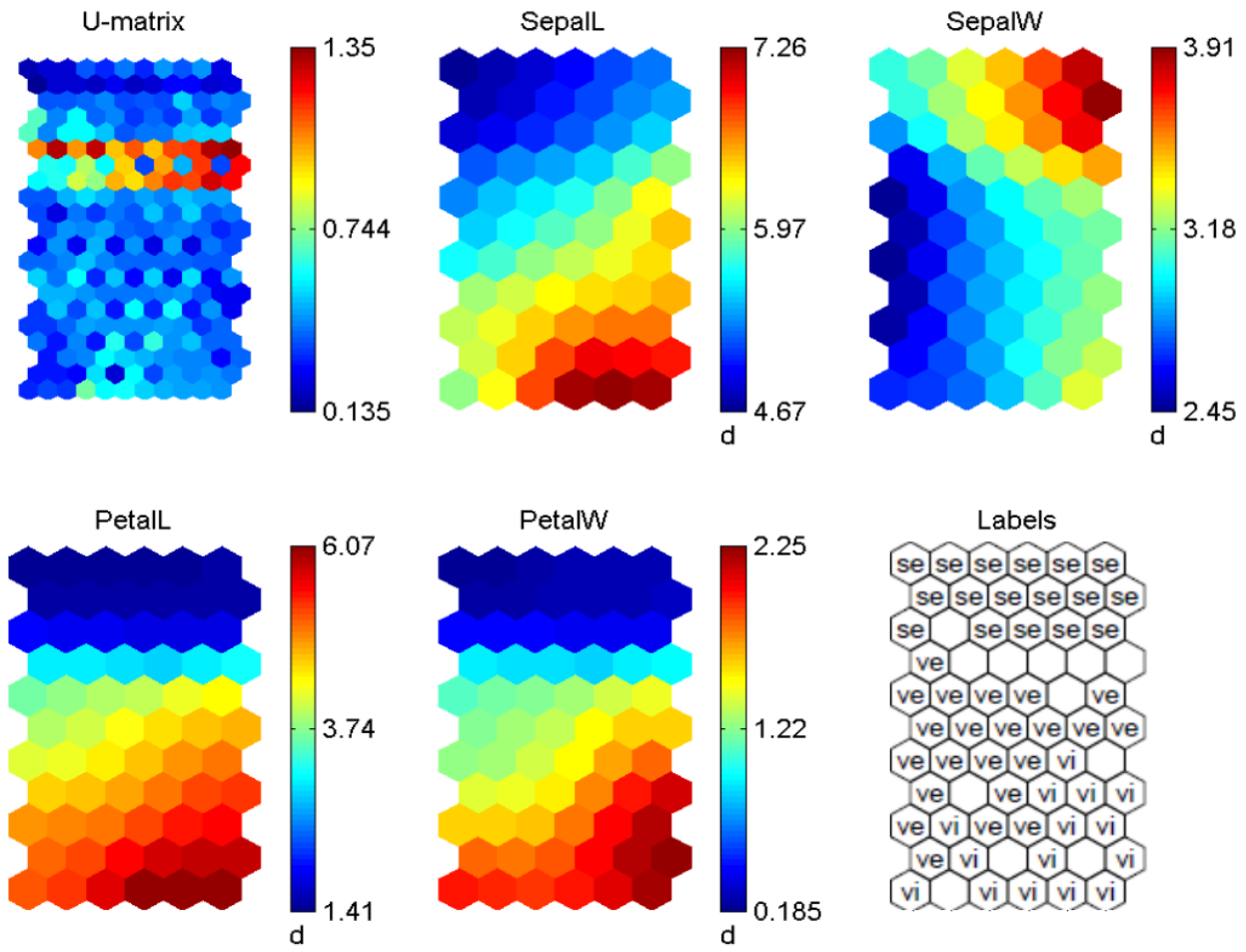


Figura 3: Visualización SOM de los datos Iris: Umatrix, planos de los componentes y labels.

Lo más representativo de la figura 3 es la matriz U (unified distance matrix) que visualiza la distancia entre unidades adyacentes en el mapa.

En lugar de representar con color el valor de una unidad específica, se representa el valor medio de la distancia de esa unidad con otras unidades. Las zonas más rojas muestran que la distancia entre los nodos es importante y por lo tanto hay un agujero. Las zonas con azul oscuro son las de los nodos muy cercanos unos de otros. De esta forma con la matriz U se pueden apreciar visualmente los posibles clusters dando una idea aproximada de cuantos clusters hay.

En el caso de los datos Iris, vemos que su Umatrix diferencia muy bien un cluster en las 3 primeras líneas. Si se examinan las etiquetas, se trata de la especie Setosa. Las otras dos especies Versicolor y Virginica forman el otro cluster. No se muestra una clara separación entre éstas últimas. Respecto a las componentes la longitud del pétalo y su anchura están muy relacionados. También existe una correlación entre éstos y la longitud del sépalo. La especie Setosa presenta pequeños pétalos y cortos pero amplios sépalos. El factor que diferencia Versicolor de Virgínica es que ésta última tiene mayores hojas.

2.5 VAT y reVAT

Bezdek desarrolló el método "Visual Assessment of Tendency (VAT)". El VAT es un método visual y consiste en presentar la información de similitud de cada par de objetos como una imagen cuadrada de n^2 píxeles. Con el fin de que la imagen pueda resalzar una posible estructura formada de varios clusters, los objetos se reordenan inspirandose en el algoritmo de Prim (Véase el anexo 1) para encontrar un mínimo "spanning tree".

El algoritmo reVAT llega a resultados parecidos al VAT pero con menos computación.

La figura 4 muestra la imagen de la matriz de similitud sin aplicar el algoritmo VAT en donde los datos están desordenados.

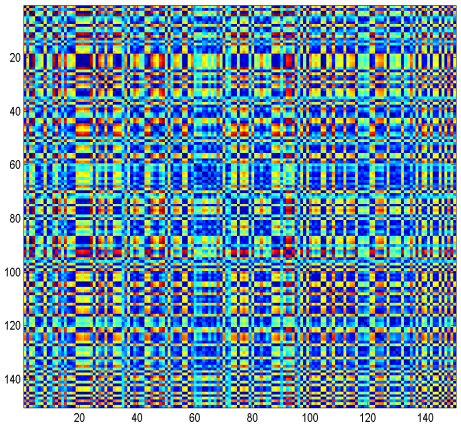


Figura 4: Matriz de similitud de los datos desordenados.

La figura 5 presenta una imagen de la matriz de similitud después de aplicar el algoritmo VAT.

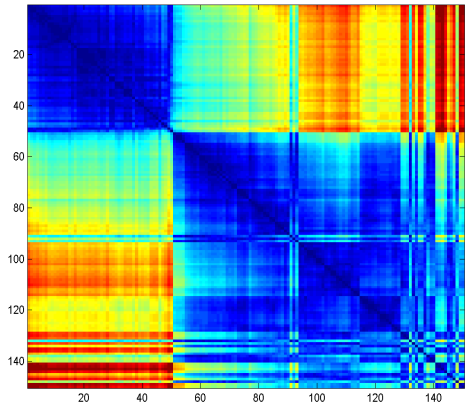


Figura 5: Matriz de similitud tras aplicar el algoritmo VAT.

La figura 6 presenta una imagen de la matriz de similitud después de aplicar el algoritmo reVAT.

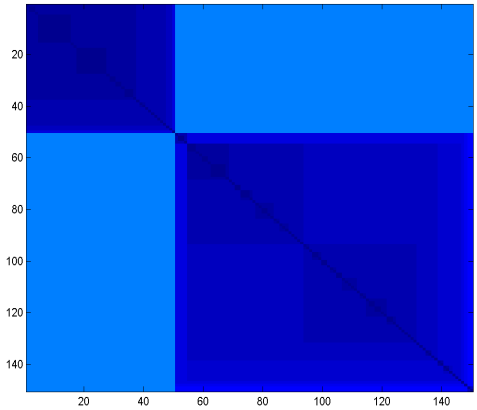


Figura 6: Matriz de similitud después de aplicar el algoritmo reVat.

Visualizando las imágenes se observa si hay cuadrados en la diagonal. Si éstos son identificables entonces es signo de que los objetos se pueden agrupar. Además si es posible contar el número de cuadrados, es una forma sencilla de indicar una posible aproximación del número de clusters a encontrar. Cuando los cuadrados no se forman, es signo de que los datos presentan una estructura aleatoria y no vale la pena aplicar un algoritmo de análisis de Cluster.

De la observación de las figuras 5 y 6, se refleja que los datos Iris utilizados pueden ser agrupados y además el número de clusters será al menos igual o superior a 2.

2.6 Estadística de Hopkins

Una manera más formal de estudiar la tendencia a formar clusters de una colección de objetos es mediante la estadística de Hopkins. Ésta está basada en la hipótesis nula en donde los datos están distribuidos unifórmemente en el espacio. Si ésta hipótesis no puede ser rechazada, el resultado de cualquier análisis de cluster sería una partición aleatoria dependiendo del algoritmo utilizado.

El algoritmo de Hopkins calcula y compara:

- *las distancias entre objetos a agrupar seleccionados al azar y sus vecinos (dW) y*
- *las distacias entre objetos generados artificialmente y sus objetos más cercanos a agrupar (dU).*

Se seleccionan aleatoriamente un número n de objetos a agrupar y objetos y se calculan las distancias dW(i) y dU(i) para cada punto objeto i=1.....n*. La estadística de Hopkins está definida como:*

$$H = \frac{\sum_i dU(i)}{\sum_i dU(i) + \sum_i dW(i)}$$

Si hay clusters a la vista, dW(i) tiende a ser más pequeño que dU(i) y por lo tanto H será más grande que 0.5 y como máximo 1. En la práctica, la estadística de Hopkins se calcula para varias selecciones aleatorias de puntos objeto y se computa la media de H para poder decidir: si la media es mayor a 0.75 entonces la hipótesis nula puede rechazarse de forma muy significativa.

En el caso de los datos/objetos Iris se tiene como resultado un H medio superior a 0.75 con lo que la hipótesis nula se rechaza y se admite que en Iris deberían aparecer grupos de forma significativa después de aplicar un algoritmo de determinación de clusters. El que se encuentren (o no) dichos clusters dependerá del algoritmo utilizado.

2.7 Resumen

En esta sección se han mostrado algunos métodos para evaluar si un conjunto de datos presenta una estructura aleatoria o existen grupos.

Estos métodos nos dicen si vale la pena aplicar los algoritmos para obtener un resultado utilizando técnicas de clustering. También pueden aportar una primera aproximación sobre el número de clusters que puede haber en el conjunto de datos/objetos.

2.8 Bibliografía

[Bezdek] J. C. Bezdek and R J. Hathaway, "VAT: A tool for visual assessment of (cluster) tendency," Proc. IJCNN, IEEE Press, Piscataway, N.J., 2002, pp. 2225 – 2230.

[Everitt] B.S. Everitt, Graphical Techniques for Multivariate Data. New York, NY: North Holland, 1978.

[Hoffman] Hoffman, P., Grinstein, G., and Pinkney, D., 1999. Dimensional Anchors: a Graphic Primitive for Multidimensional Multivariate Information Visualizations, Proc. 1999 Workshop on New Paradigms in Information Visualization and Manipulation, in Conjunction with the 8h ACM Int'l. Conf. Information and Knowledge Management (CIKM '99),pp. 9-16.

[Hopkins] Hopkins, B., Skellam, J. G.: Ann. Bot. 18, 1954, 213–227. A new method for determining the type of distribution of plant individuals.

[Huband] J.M. Huband, J. C. Bezdek and R J. Hathaway,"Revised Visual Assessment of (Cluster) Tendency (reVAT)", 2004.

[Jain] A.K. Jain and R.C. Dubes, Algorithms for Clustering Data. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[Kandogan] Kandogan, E., 2001. Visualizing multi-dimensional clusters, trends, and outliers using star coordinates, Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 107-116.

[Kohonen 1] Kohonen, T.: Self-Organizing Maps. Springer, Berlin, Germany, 1995.

[Kohonen 2] Kohonen, T.: Self-Organizing Maps. Springer, Berlin, Germany, 2001.

[Sammon] Sammon, John W. Jr., "A Nonlinear Mapping for Data Structure Analysis", IEEE Transactions on Computers, vol. C-18, no. 5, pp 401-409, May 1969.

[Sharma] Sharma, S., 1995. Applied Multivariate Techniques (New York, NY: John Wiley & Sons, Inc.).

[SomToolbox] www.cis.hut.fi/projects/somtoolbox/

[Vesanto] Vesanto, J., Himberg, J., Alhoniemi, E., and Parhankangas, J., 1999. Self-organizing map in Matlab: the SOM toolbox, Proceedings of the Matlab DSP Conference, pp. 35-40.

3. DETERMINACIÓN EN UN PROCESO CLUSTERING

3.1 Introducción

Dada la diversidad de algoritmos que permiten determinar los clusters, se eligirán los más representativos. Tendrán como principal parámetro variable el número de clusters a encontrar.

Los métodos seleccionados son: K-means [MacQueen] , Hierarchical clustering [Johnson] [MATLAB], EM clustering [Dempster] [Witten], Fuzzy C-means [Bandemer][Bezdek][Dunn] y SOM [Kohonen].

3.2 Método K-means

La letra K representa el número de clusters. Ese valor es desconocido a priori y es elegido por el usuario. Cada cluster tiene un centroide el cual es calculado como la media de los patrones de los objetos que forman el cluster. La pertenencia de un objeto a un cluster se determina en función de la distancia más cercana de éste a un centroide determinado. Como los centroides no pueden ser calculados antes de formar los clusters, el usuario especifica K valores para los centroides al principio del proceso de clustering.

***K-means** divide una colección de objetos en K clusters utilizando las siguientes etapas [MacQueen]:*

1) *Inicialización de los centroides con K valores.*
2) *Para cada objeto de la colección, encuentra el centroide más cercano (en términos de una distancia euclídea) y se asigna dicho punto al cluster de dicho centroide.*
3) *Calcula los nuevos centroides para cada uno de los clusters formados.*
4) *Se produce una iteración entre las etapas 2 y 3 hasta que una condición de término ocurre.*

Hay varias condiciones que pueden utilizarse para detener el algoritmo. Éste comparará el valor de una medida calculada en la iteración en curso con el valor de la misma medida calculada en la iteración previa. Por ejemplo, estas condiciones podrían ser:

1) *Que los centroides no cambien.*
2) *Que la suma de las distancias de cada punto a sus respectivos centroides no cambien.*
3) *Que los objetos de cada cluster no cambien.*

El algoritmo K-means mueve de forma iterativa los puntos entre los clusters, minimizando la suma de las distancias (que se denotará mediante J), para cada punto con el centroide de su cluster. Si el i-th cluster es C_i, entonces la suma de las distancias para C_i, se denota por J_i, se define como:

$$J_i = \sum_{X \in C_i} d\_Euc(X, Y_i)^2$$

La suma de las distancias para todos los K clusters se denota por J:

$$J = \sum_{i=1}^{k} J_i$$

La inicialización del algoritmo de K-means con diferentes valores puede conducir a encontrar un mínimo local de J. Como el objetivo es encontrar el mínimo global, es posible hacer funcionar el algoritmo K-means varias veces inicializando siempre con valores diferentes y elegir la solución que minimiza J. Es más probable que haciendo funcionar el algoritmo K-means numerosas veces se encuentre la solución que minimiza globalmente a J, o al menos un mínimo local muy próximo al mínimo global.

3.3 Método Jerárquico

Hay dos tipos de métodos jerárquicos: aglomerativo y divisivo. El aglomerativo empieza con cada dato formando un único cluster y de forma iterativa va agrupando los clusters que están más próximos en pares hasta formar un sólo cluster. El divisivo va en la dirección opuesta. Empieza por un único cluster que contiene todos los puntos e iterativamente lo divide en 2 cluster que están lo más lejos el uno del otro.

Dividir un cluster es computacionalmente más intensivo que unir 2 clusters. Por lo tanto el método divisivo no es muy utilizado. La estructura de los clusters encontrada tras utilizar un algoritmo jerárquico puede ser representada gráficamente por un dendograma.

Para llevar a cabo un agrupamiento de datos jerárquico de una colección de n puntos objeto, se necesita generar una matriz de similitud cuyos elementos son la distancia de todas las parejas posibles entre los datos. Las principales etapas de un ***algoritmo jerárquico*** *son:*

1) *Asignar cada punto en un sólo cluster de un sólo punto. Por lo tanto tendremos n clusters.*
2) *Encontrar el par de clusters más cercanos y unirlos para formar un nuevo cluster.*
3) *Calcular las distancias entre el nuevo cluster de la etapa (2) y los demás clusters.*
4) *Iterar entre las etapas (2) y (3) hasta que los n puntos objeto se hayan fusionado en un único cluster.*

En adición a la medida de distancia entre 2 puntos objeto (por ejemplo: Euclídea), se necesita un método para calcular la distancia entre clusters de la etapa (2) para que los 2 clusters más cercanos puedan fusionarse. Este método se denomina linkage [Johnson]. Denotamos $x^{(j)}{}_i$ *y* $x^{(l)}{}_i$ *(para i=1....n puntos objeto), los puntos objeto asignados a los clusters j (de dimensión* n_j *) y l (de dimensión* n_l*). Entonces la distancia entre dos clusters con los índices j y l puede ser determinada por varios métodos:*

$$\text{COMPLETE LINKAGE: } \max_i ||\boldsymbol{x}_i^{(j)} - \boldsymbol{x}_i^{(l)}||$$
$$\text{SINGLE LINKAGE: } \min_i ||\boldsymbol{x}_i^{(j)} - \boldsymbol{x}_i^{(l)}||$$
$$\text{AVERAGE LINKAGE: } \text{average}_i ||\boldsymbol{x}_i^{(j)} - \boldsymbol{x}_i^{(l)}||$$
$$\text{CENTROID METHOD: } ||\boldsymbol{c}_i - \boldsymbol{c}_j||$$
$$\text{WARD'S METHOD: } ||\boldsymbol{c}_i - \boldsymbol{c}_j|| \cdot \frac{\sqrt{2n_j n_l}}{\sqrt{n_j + n_l}}$$

3.4 Método EM

En este método se asume que cada cluster es generado por una distribución normal multivariable. Cada cluster k tiene como parámetros: un vector medio (m_k) y una matriz de covarianza S_k.

El ajuste de los parámetros del modelo requiere alguna medida de su bondad, es decir, cómo de bien encajan los datos sobre la distribución que los representa. Este valor de bondad se conoce como el likelihood de los datos. Se trataría entonces de estimar los parámetros buscados, θ, maximizando este likelihood (este criterio se conoce como MLMaximun Likelihood). Normalmente, lo que se calcula es el logaritmo de este likelihood, conocido como log-likelihood ya que es más fácil de calcular de forma analítica. La función likelihood se denota por L y es la probabilidad de los datos D dados los parámetros, se denota por P(D|θ). La solución obtenida es la misma, gracias a la propiedad de monotonicidad del logaritmo. La forma de esta función log-likelihood es:

$$\mathrm{L} = p(D \mid \theta) = \prod_{i=1}^{n} p(X_i \mid \theta)$$

*donde n es el número de puntos en D, que se suponen independientes entre sí. El algoritmo **EM**, procede en dos pasos que se repiten de forma iterativa:*

1) ***Expectation**: Computa la probabilidad de cada observación que pertenece a cada cluster utilizando los parámetros estimados. Utiliza los valores de los parámetros, iniciales o proporcionados por el paso Maximization de la iteración anterior, obteniendo diferentes formas de la FDP buscada.*
2) ***Maximization**: Obtiene nuevos valores de los parámetros a partir de los datos proporcionados por el paso anterior.*

Después de una serie de iteraciones, el algoritmo EM tiende a un máximo local de la función L. Finalmente se obtendrá un conjunto de clusters que agrupan el conjunto de proyectos original. Cada uno de estos clusters estará definido por los parámetros de una distribución normal.

Aunque el algoritmo EM parece estar condicionado por la obtencion de clusters de forma elíptica como resultado de un modelo de distribución normal multivariable, éste tiene varias ventajas. No requiere de la elección de una medida de distancia, ni la elección de una medida de validación ya que existe la medida BIC (Bayesian Information Criterion) que como veremos en el próximo capítulo puede ser utilizada para este fin. El número de clusters tiene que ser proporcionado para un cierto intervalo y el método automáticamente selecciona el mejor número y tipo de modelo de cluster. Un valor de BIC alto indica una fuerte correspondencia con el modelo.

3.5 Método Fuzzy C-means

Con este método un punto objeto no se asigna exclusivamente a un sólo cluster. Se utiliza coeficientes de pertenencia u_{ij} para cada observación x_i (i=1...n) en cada cluster indexado por j=1...K. Normalmente las funciones de pertenencia son normalizadas entre [0,1] en donde 0 indica que no pertenece al cluster y 1 que pertenece. Además, como los puntos objeto son distribuidos

sobre todos los clusters, los coeficientes de pertenencia deberían sumar 1 para cada objeto:

$$\sum_{j=1}^{k} u_{ij} = 1$$

Existen diversos algoritmos. El más conocido es el ***Fuzzy C-means*** *[Bandemer][Bezdek][Dunn]. La función objetivo es similar a la del método K-means:*

$$\sum_{j=1}^{k} \sum_{i=1}^{n_j} u_{ij}^2 \cdot ||x_i^{(j)} - c_j||^2 \rightarrow \min$$

En lugar de utilizar una distancia Euclídea, otras distancias pueden también ser consideradas. Además, otra potencia diferente a 2 puede ser utilizada sobre los coeficientes de pertenencia con lo que se cambia las características del proceso (grado de Fuzzificación). Similar al método K-means, el número de clusters K, es la variable de entrada del algoritmo que hace uso de los centroides c_j que son computados por

$$c_j = \frac{\sum_i u_{ij}^2 \cdot x_i}{\sum_i u_{ij}^2}$$

De esta forma, los centroides son una media ponderada de todas las observaciones. Los pesos son los coeficientes de pertenencia de los puntos objeto a los clusters correspondientes. Cuando los coeficientes de pertenencia sólo toman como valores 0 y 1, el algoritmo Fuzzy C-means se reduce a K-means.

La búsqueda del mínimo de la función objetivo es realizada por un proceso iterativo de optimización . En cada iteración los coeficientes de pertenencia se recalculan mediante:

$$u_{ij} = \frac{1}{\sum_{l=1}^{k} \left(||x_i - c_j|| / ||x_i - c_l|| \right)^2}$$

y por consiguiente los centroides, con estos nuevos coeficientes, son también actualizados. El proceso se detiene cuando los coeficientes de pertenencia o los centroides cambian muy poco.

3.6 Método SOM

SOM *se introdujo en la sección 2.4 como método de visualización. Ahora se describirá el proceso del algoritmo con algo más de detalle.*

El proceso de aprendizaje en una red neuronal se entiende como la aplicación iterada de un algoritmo de actualización de los vectores de pesos W, con la finalidad de que la red neuronal mejore las respuestas que emite al procesar el conjunto de datos de entrada.

El proceso de aprendizaje de una red neuronal produce una dinámica en los vectores de referencia, ya que los vectores de pesos cambian en cada tiempo de iteración t, en función del dato de entrada x(t).

Una de las formas de aprendizaje no supervisado, es el de las redes de entrenamiento competitivo. En estas redes, las neuronas reciben de manera idéntica la información de entrada sobre la cual compiten. Dicha competencia consiste en determinar cual de las neuronas es la que mejor representa a un estímulo de entrada dado. Como resultado de esta competencia solo una neurona es activada en cada momento. En cada tiempo de iteración t se determina la neurona ganadora c(t).

Una forma bastante común de establecer esta competencia es elegir como neurona ganadora aquella cuyo vector de pesos $w_{c(t)}(t) \in W(t)$, en este caso vector de referencia, es más parecido al dato de entrada x(t), es decir, $w_{c(t)}$ queda determinado de manera que:

$$\left\| x(t) - w_{c(t)}(t) \right\| = \min_{i=1}^{k} \{ \| x(t) - w_i(t) \| \}$$

El proceso de entrenamiento competitivo de una red neuronal es estable, si después de un número finito de iteraciones, ningún patrón en el conjunto de aprendizaje cambia de representante [Atenogenes].

La principal razón de aceptación del método SOM es su capacidad de presentar, de manera automática, un mapa en el cual se puede observar una descripción intuitiva de la similitud entre los datos. El despliegue bidimensional tiene la propiedad de exhibir la información contenida en los datos de manera ordenada y resaltando las relaciones de similitud. A continuación se exponen algunos conceptos generales relativos a la naturaleza y utilidad del algoritmo SOM.

La arquitectura de la red es una retícula con una configuración rectangular o hexagonal. La localización de cada neurona sobre la retícula está representada por su vector de localización $r_i = (p_i, q_i)$.

En el algoritmo SOM básico, la configuración de los nodos (hexagonal o rectangular) y el número K de neuronas se fijan desde el principo. Normalmente se definen las distancias entre las unidades del mapa de acuerdo a la distancia Euclidiana entre los vectores de localización.

Durante el proceso de entrenamiento del SOM, se utiliza un conjunto finito de datos $X = \{x_0, ..., x_{m-1}\} \subset R^n$.

El algoritmo de entrenamiento es el siguiente:

1) *Se define la condición inicial de los vectores de referencia W(0) de manera alealoria y se presenta el dato x(0).*
2) *Para la presentación del dato x(t) se determina la neurona ganadora $\eta_{c(t)}$ de acuerdo a la expresión:*

$$\left\| x(t) - w_{c(t)}(t) \right\| = \min_{i=1}^{k} \{ \| x(t) - w_i(t) \| \}$$

3) *Para toda i ∈ {1, ..., K} se actualiza al vector de referencia $w_i(t)$ de acuerdo a la siguiente regla de aprendizaje:*

$$w_i(t+1) = w_i(t) + \alpha(t) h_{(c,i)}(t)\,[x(t) - w_i(t)]$$

4) *Se presenta el dato x(t + 1) y se repíte el cíclo desde el paso 2.*

Este proceso se repíte hasta un número determinado de iteraciones. En caso de que el índice c(t) no esté bien definido; es decir, cuando para un dato x(t) existan dos η_e, $\eta_d \in N$ tal que:

$$\|x(t) - w_e\| = \min_{i=1}^{k} \{\|x - w_i\|\} = \|x(t) - w_d\|$$

la selección de un único c(t) debe hacerse de manera aleatoria.

Cada vez que se determina una neurona ganadora $\eta_c(t)$, la idea clave en el algoritmo de aprendizaje es que aquellas neuronas que se encuentran dentro de una vecindad de ηc(t) en el arreglo bidimensional también aprenderán de la entrada x(t) (Figura 7).

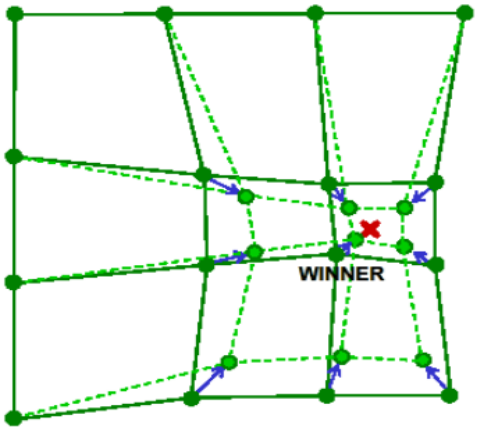


Figura 7: Aprendizaje de las neuronas próximas a la neurona ganadora.

Para determinar la magnitud del aprendizaje de una neurona η_i en términos de la distancia con la neurona ganadora $\eta_c(t)$ se define la denominada función vecindad que es de la forma:

$$h_{(c,i)}(t) = h(\|r_{c(t)} - r_i\|, t) \in [0, 1].$$

Independientemente de cual sea la forma explícita de esta función, debe ser tal que $h_{(c,c)}(t) = 1$ para todo t ; además para cada t fijo, $h_{(c,i)}(t)$ debe ser decreciente en función de $||r_c(t) - r_i||$ y cumplir con $h_{(c,i)}(t) \rightarrow 0$ cuando $||r_c(t) - r_i||$ se incrementa.

Una de las definiciones más simples que se encuentran de la función vecindad es la siguiente:

$$h_{(c,i)}(t) = 1 \text{ si } i \in N_c(t)$$
$$h_{(c,i)}(t) = 0 \text{ si } i \notin N_c(t)$$

en este caso $N_c(t)$ es una vecindad de $\eta_c(t)$ sobre la retícula que se define de la siguiente manera:

$$N_c(t) = \{i \in \mathbb{N} \mid \|r_{c(t)} - r_i\| \leq \rho(t)\}$$

Otra forma común de la función vecindad está dada en términos de la función Gaussiana:

$$h_{(c,i)}(t) = \exp\left(\frac{\|r_c - r_i\|^2}{2\rho^2(t)}\right)$$

en este caso $\rho(t)$ corresponde al ancho promedio de $N_c(t)$.

Para efectos de la convergencia del algoritmo, la variación del radio a través del tiempo debe cumplir que $t_i \leq t_j \Rightarrow \rho(t_i) \geq \rho(t_j)$ y además $\rho(t) \to 0$ cuando $t \to \infty$. Se recomienda que $\rho(1)$ sea más grande que la mitad del diámetro de la red.

La función $\alpha(t)$ es el factor de aprendizaje, y en este caso cumple con la condición $0 < \alpha(t) < 1$ y es no creciente de manera que $\alpha(t) \to 0$ cuando $t \to \infty$.

Tanto $\rho(t)$ como $\alpha(t)$ son componentes autónomas de la regla de aprendizaje y su principal objetivo es garanizar la convergencia del algoritmo a partir de producir cambios cada vez más locales (centrados en la neurona ganadora) y de menor magnitud.

Durante la evolución del proceso de entrenamiento, los vectores de pesos son modificados de manera que cada uno de éstos se vuelva representante de una porción en el espacio de entrada.

Si comparamos SOM con K-means: K-means elige el número de clusters que se adaptarán a los datos/objetos. Para SOM se elige la forma y dimensión de la red de clusters que deberán concordar con los datos/objetos. Los clusters, en SOM, se denominan nodos. Igual que en K-means se debe elegir una dimensión inicial basada en el número esperado de clusters en los datos.

Una importante diferencia entre SOM y K-means es que SOM proporciona automáticamente información sobre la similitud de los nodos, es decir como se parecen el uno al otro.

3.7 Resumen

En este capítulo se han mostrado los principales métodos utilizados para determinar los grupos dentro de un conjunto de datos/objetos. Los algoritmos seleccionados tienen todos el número de clusters como única variable. El método jerárquico la utiliza al final y los demás métodos al principio del análisis. Por otro lado, en todos los métodos, los puntos objeto sólo pueden pertenecer a un sólo cluster a excepción del método Fuzzy C-means donde el grado de pertenecia viene determinado por los coeficientes de pertenencia.

Estos algoritmos pueden encontrar grupos aunque la arquitectura de los datos/objetos sea aleatoria. Es por eso necesario que se apliquen después de un análisis de la tendencia. También es importante evaluar los resultados de clustering obtenidos. Esto se analizará en el próximo capítulo.

3.8 Bibliografía

[Atenogenes] Atenogenes Elio "El Algoritmo SOM: un ejemplo de visualización informétrica". dynamics.unam.edu.

[Bandamer] Bandemer, H., Näther, W.: Fuzzy Data Analysis. Kluwer Academic, Dordrecht, the Netherlands, 1992.

[Bezdek] Bezdek, J. C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York, 1981.

[Dempster] Dempster A, Laird N, Rubin D. Maximum likelihood from incomplete data via the EM algorithm. J. of the Royal Statistical Society, Series B. 1977;39(1):1–38.

[Dunn] Dunn, J. C.: J. Cybern. 3, 1973, 32–57. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters.

[Johnson] Johnson SC. Hierarchical clustering schemes. Psychometrika. 1967;32:241–254.

[Kohonen 1] T. Kohonen, Self-organized formation of topologically correct feature maps, Biol. Cybern. 43 (1982) 59–69.

[Kohonen 2] T. Kohonen, Self-Organizing Maps, Springer, Berlin, Germany, 1997.

[MacQueen] MacQueen J. Some methods for classification and analysis of multivariate observations. Proc. of the 5^{th} Symp. on Mathematical Statistics and Probability. 1967;1:281–297.

[MATLAB] The Math Works Inc. MATLAB version R12.

[Witten] Witten I, Frank E. Data mining: practical machine learning tools and techniques. 2nd Ed. Morgan Kaufmann; 2005. (WEKA implementation of machine learning tools)

4. TÉCNICAS DE VALIDACIÓN Y NÚMERO ÓPTIMO DE CLUSTERS

4.1 Introducción

En los capítulos anteriores se han presentado ciertas técnicas para valorar si los datos a agrupar "merecen ser agrupados" y también los algoritmos más representativos para obtener los clusters.

En este capítulo se pretende valorar los grupos obtenidos después de aplicar los algoritmos de clustering.

Para ello se describirán 3 maneras de validar los algoritmos y los clusters obtenidos [Halkidi 1]. En primer lugar, se determinará la calidad del algoritmo para obtener una partición interesante. Es decir que los grupos obtenidos concuerdan con los datos/objetos (criterio interno). Posteriormente, se analizará la calidad del algoritmo para encontrar clusters significativos (criterio relativo) y por último la calidad del algoritmo para encontrar grupos que se parezcan a grupos conocidos utilizados como referencia (criterio externo).

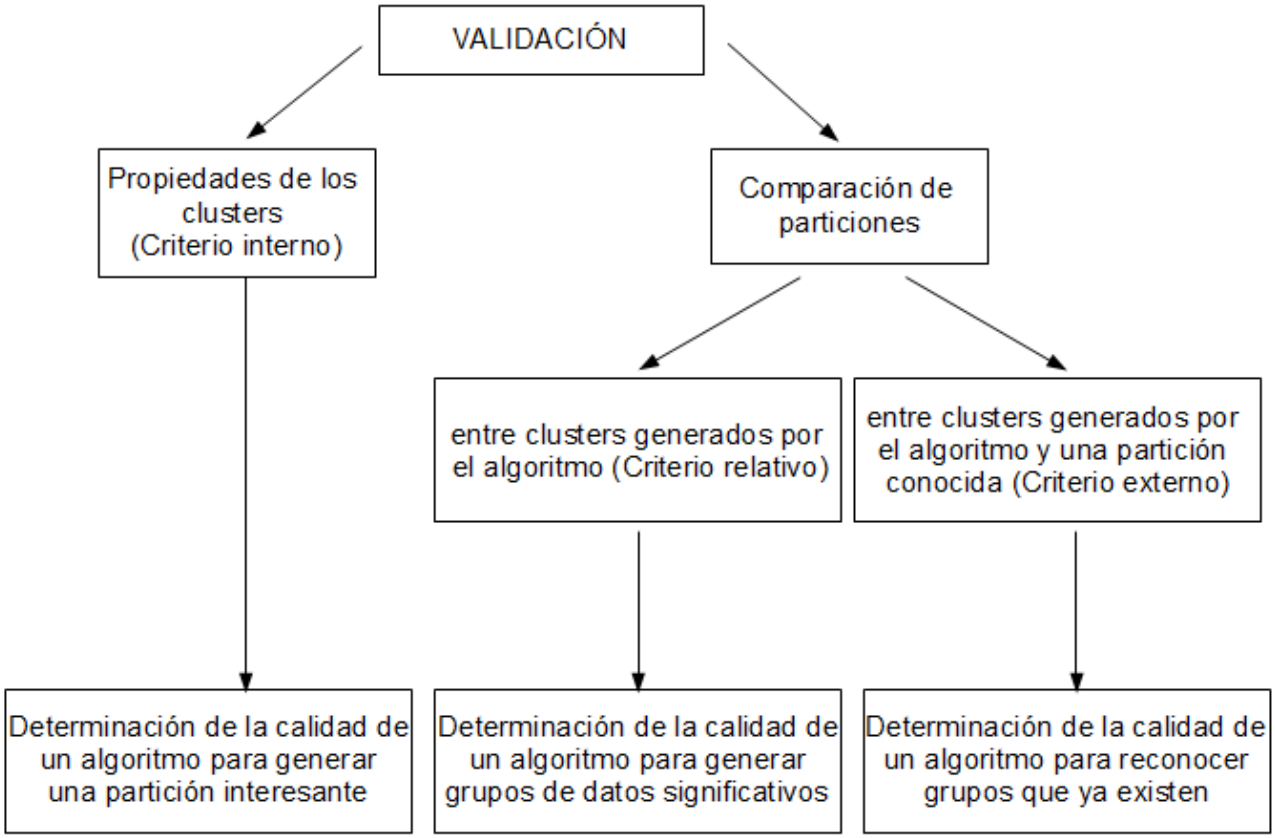


Figura 8: Descripción de los criterios interno, externo y relativo para analizar la calidad de un algoritmo para agrupar datos.

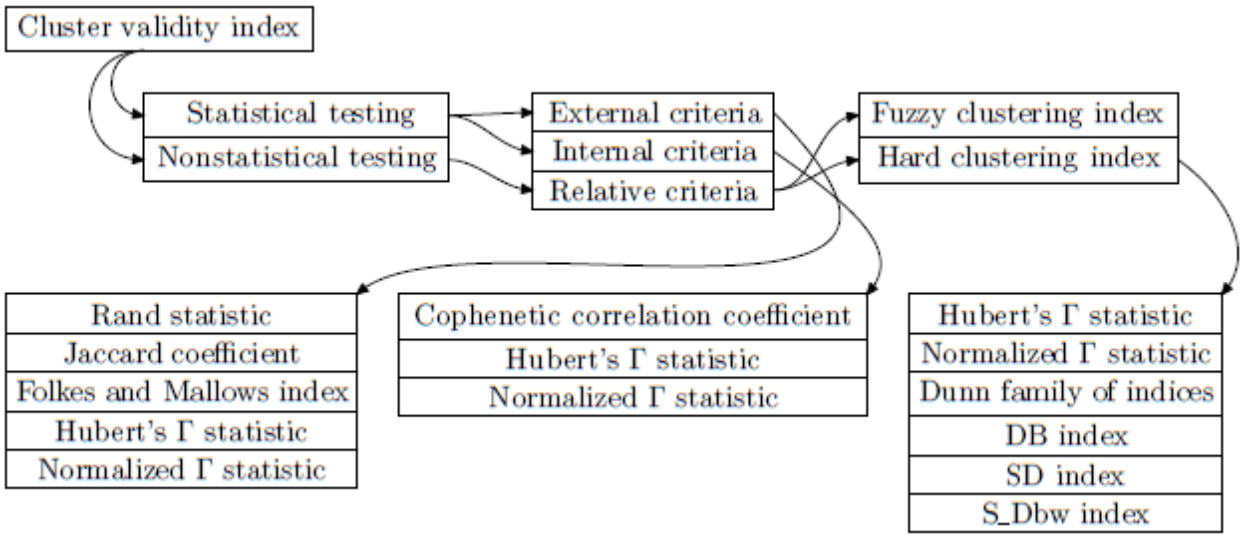


Figura 9: Criterios de validación y algunas de sus estadísticas e índices [Data Clustering].

La figura 9 muestra los principales criterios y su relación con las estadísticas e índices que los describen.

La base de los métodos para los criterios externo (comparar un algoritmo de cluster comparando su resultado con una partición de referencia) e interno (evaluar el método de cluster utilizando sólo cantidades y características de los propios datos a agrupar) es estadística. Y por consiguiente la demanda computacional puede ser importante.

Por el contrario, un critero relativo no se basa en un test estadístico. El principio fundamental de esta aproximación es elegir el mejor resultado de un algoritmo de cluster comparando los diferentes agrupamientos que se obtienen utilizando diferentes valores de las variables de entrada del algoritmo (por ejemplo: el número de clusters a encontrar).

Más precisamente [Halkidi 2], si Palg son el conjunto de parámetros asociados con un algoritmo de clustering (por ejemplo el número de clusters) entonces, hay que elegir entre los diferentes resultados que se obtienen al variar los valores de los Palg, el mejor que más se adapta a los puntos objeto. Para ello se pueden considerar 2 casos:

1) ***Palg no contiene al número de clusters como parámetro**: en este caso se hace funcionar el algoritmo de clustering para un amplio intervalo de los valores de los parámetros. A continuación se elige el mayor intervalo dentro del cual el número de cluster permanece cte. Y se toma los valores medios dentro de ese intervalo como valores de los parámetros. De este modo también se ha identificado el número de clusters.*
2) ***Palg contiene al número de clusters como parámetro**: el proceso para identificar el mejor resultado de un algoritmo clustering está basado en la utilización de un índice de validación.*

En esta memoria no se considerará el primer caso. Los algoritmos de clustering utilizados se corresponden con el segundo caso. A continuación se va a describir con más detalle el uso de los índices de validación.

4.2 Criterio Relativo

Un buen resultado de agrupamiento de datos consiste en encontrar puntos objeto similares en cada cluster pero diferentes de los otros puntos objeto de otros clusters. Para medir como de bueno es el resultado se utilizan los índices de validación. Estos miden principalmente 2 características:

1) ***Compacidad**: Mide cómo de cerca están los puntos objeto en un cluster. Hay índices que evaluan la compacidad basados en la varianza. Cuanto menor es la varianza mejor es la compacidad. También pueden medir la compacidad mediante medidas basadas en la distancia como el máximo (o la media) de las distancias entre parejas, o el máximo (o la media) de la distancia al centro del cluster.*
2) ***Separación**: Mide cómo de separados están los clusters entre sí. Por ejemplo, las distancias entre las parejas de los centros de los clusters, o la mínima distancia entre las parejas de los objetos de diferentes clusters.*

En general, el procedimiento para utilizar un índice de validación y así obtener el mejor resultado de un algoritmo de clustering y por tanto el óptimo número de clusters es el siguiente:

1) *Se elige un intervalo para el parámetro número de clusters entre un mínimo y un máximo.*
2) *Para cada uno de los valores del intervalo número de clusters, se hace funcionar el algoritmo clustering.*
3) *Para cada resultado se calcula el valor del índice de validación elegido. Se suele representar una gráfica de los valores del índice en función del número de clusters.*
4) *Dependiendo de las características del índice de validación, se elige el mejor valor del índice y por consiguiente se obtiene el optimo número de clusters para dicho índice.*

La gráfica obtenida en el paso tercero, permite identificar el número óptimo de clusters. Existen 2 posibilidades para definir el mejor resultado clustering dependiendo del comportamiento del índice en relación al número de clusters:

1) *El índice de validación varía de forma regular conforme el número de clusters K se incrementa (por ejemplo el índice aumenta hasta un cierto valor K y después decrece).*
2) *El índice de validación no varía de forma regular conforme K aumenta.*

En el primer caso, se busca un máximo (o mínimo) en la gráfica. En el segundo caso, se busca el valor de K en donde el índice experimenta un cambio en su valor. Este cambio aparece como la forma de un "codo" en la gráfica y es donde se identifica el valor del número de clusters. Además la ausencia de un "codo" puede ser un indicativo que no existe un resultado con clusters.

Existen en la literatura numerosos índices de validación. A continuación se describirán los más importantes o al menos los más utilizados.

Para el algoritmo jerárquico, K-means y SOM se utilizarán los índices Silhouette, SD, S_Dbw,

Davies-Bouldin, Calinski-Harabasz, R-squared, Dunn, Hubert (modificado y normalizado).

Para el algoritmo EM se utizará un único índice denominado Bayes Information Criteria (BIC).

Para el algoritmo Fuzzy C-means se utilizarán 4 índices: CP, SP, CE, XB.

A continuación se van a describir dichos índices en detalle.

4.2.1 Estadística de Hubert Modificada

La estadística de Hubert modificada [Halkidi 2] se define como:

$$\Gamma = \frac{1}{M}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} P_{ij}\,Q_{ij}$$

donde n es el número de puntos objeto a agrupar, M = n(n−1)/2 , P es la matriz de similitud y Q es la matriz definida por:

$$Q_{ij} = d(\mu_{c_i}, \mu_{c_j}), \quad 1 \le i, j \le n,$$

donde d(·, ·) es una distancia (ejem:euclídea), y μ_{ci}, μ_{cj} son los centroides de los clusters que pertenecen los puntos i,j.

La estadística de Hubert normalizada se define de forma parecida [Halkidi 2]. De la definición de la estadística de Hubert modificada, se deduce que cierto valor de Γ (o el de la normalizada) indica que los clusters son compactos. De las expresiones anteriores se observa que los índices no están definidos cuando el número de clusters es igual a 1 o n.

4.2.2 Índice Silhouette

La técnica de validación de Silhouette [Rousseeuw] calcula la anchura de la silhouette para cada punto objeto. Se calculan también el promedio Silhouette para todos los puntos objeto de un cluster y de todos los clusters.

La silhouette promedio puede ser utilizada para evaluar la validez de un resultado clustering y también para decidir el número de clusters. Para construir las silhouettes S(i) se usa la fórmula:

$$S(i) = \frac{b(i) - a(i)}{\max a(i), b(i)}$$

donde a(i) (disimilitud promedio) del punto objeto i con todos los demás puntos objeto en el mismo cluster; b(i) (mínimo del promedio de disimilitud) del objeto i hacia todos los objetos en otro cluster (el más cercano).
De la fórmula anterior se deduce que $-1 \le S(i) \le 1$. Si Silhouette está próximo de 1, significa que el

punto objeto está asignado a su cluster. Con valor 0, el punto objeto podría pertenecer a otro cluster. Si el valor es próximo a -1, el punto objeto está mal clasificado y por lo tanto está en alguna parte entre los clusters. Para tener un único valor que represente a todos los cluster, se calcula el promedio de la silhouette de todos los puntos objeto.

Por lo tanto, el mejor resultado clustering será el que proporcione un valor máximo del promedio de Silhouette.

4.2.3 Índice Dunn

Esta técnica [Dunn] se basa en la idea de identificar los clusters que son compactos y bien separados. Para una partición de clusters, donde c_j representa el j-cluster de la partición, el índice Dunn, D, puede ser calculado por la siguiente fórmula:

$$D = \min_{1 \le j \le n} \left\{ \min_{\substack{1 \le j \le n \\ i \ne j}} \left\{ \frac{d(c_i, c_j)}{\max_{1 \le k \le n} \{d'(c_k)\}} \right\} \right\}$$

donde $d(c_i,c_j)$ es la distancia entre los clusters c_i y c_j (distancia entre clusters); $d'(c_k)$ es la "intracluster" distancia del cluster c_k, n es el número de clusters. El principal objetivo de la medida es maximizar las distancias entre cluster mientras que se aumenta la compacidad. Por lo tanto, el número de cluster que maximiza D es el que nos indica el núméto óptimo de clusters.

4.2.4 Índice Davies-Bouldin

El índice de [Davies] es el cociente siguiente:

$$DB = \frac{1}{n} \sum_{i=1}^{n} \max_{i \ne j} \left\{ \frac{S_n(Q_i) + S_n(Q_j)}{S(Q_i, Q_j)} \right\}$$

donde n es el número de clusters, S_n es la distancia promedio de todos los objetos de un cluster con su centro, $S(Q_i,Q_j)$ es la distancia entre los centros de los clusters. Por lo tanto, el cociente es pequeño cuando los clusters son compactos y están lejos el uno del otro. DB es un índice que será pequeño cuando se tenga un buen resultado clustering.

4.2.5 Índice de Compacidad (CP)

De acuerdo con [Nguyen], CP mide la distancia media entre cada par de puntos objeto que pertenecen al mismo cluster. Más precisamente se define como:

$$CP = \frac{1}{N} \sum_{k=1}^{K} n_k \left(\frac{\sum_{x_i, x_j \in C_k} d(x_i, x_j)}{n_k(n_k - 1)/2} \right)$$

donde K denota el número de clusters, n_k es el número de puntos objeto que pertenecen al cluster k-th. $d(x_i; xj)$ es la distancia entre los puntos x_i y x_j, y N es el número total de puntos objeto a agrupar. Idealmente, los miembros de cada cluster deberían estar lo más cercanos posible. Esto significa que pequeños valores para CP significa una mejor configuración.

4.2.6 Índice Calinski-Harabasz

El índice de Calinski-Harabasz [Maulik], está definido utilizando las trazas de las matrices "scatter" entre los clusters y dentro de los clusters. Si n es el número de puntos objeto y K el número de clusters, entoces C-H se define como:

$$C-H=\frac{(n-K)\,Tr\,(B)}{(K-1)\,Tr(W)}$$

donde Tr(B) y Tr(W) son las trazas de las matrices B y W respectivamente. Y B y W son las matrices "scatter":

$$\mathrm{Tr}(B)=\sum_{i=1}^{k}|C_i|(\mathbf{z}_i-\mathbf{z})^T(\mathbf{z}_i-\mathbf{z})$$

$$\mathrm{Tr}(W)=\sum_{i=1}^{k}\sum_{\mathbf{x}\in C_i}(\mathbf{x}-\mathbf{z}_i)^T(\mathbf{x}-\mathbf{z}_i)$$

*donde **z** y $\mathbf{z}_i$ son la media de todos los puntos y del cluster C_i,respectivamente.*

4.2.7 Índice R-Squared

Sea $D=\{x_1,...x_n\}$ un conjunto de datos/objetos. Entonces la suma de los cuadrados de D se define como:

$$SS=\sum_{i=1}^{n}(\mathbf{x}_i-\bar{\mathbf{y}})^2=\sum_{i=1}^{n}\sum_{j=1}^{d}(\mathbf{x}_{ij}-\bar{\mathbf{y}}_j)^2$$

*donde **y** es la media de los puntos de D.*

Sea $C=\{C_1,....C_K\}$ una partición de los datos D. Entonces se define:

SS_w= la suma de los cuadrados dentro de un grupo
SS_b= la suma de los cuadrados entre grupos
SS_t= la suma total de cuadrados de todos los datos.

Y el índice RS se expresa como sigue (Halkidi 2]:

$$RS = \frac{SS_b}{SS_t}$$

$$= \frac{SS_t - SS_w}{SS_t}$$

$$= \frac{\sum_{\mathbf{x}\in D}\sum_{j=1}^{d}(\mathbf{x}_j - \bar{\mathbf{y}}_j)^2 - \sum_{i=1}^{k}\sum_{\mathbf{x}\in C_i}\sum_{j=1}^{d}(\mathbf{x}_j - \mu_{ij})^2}{\sum_{i=1}^{n}\sum_{j=1}^{d}(\mathbf{x}_{ij} - \bar{\mathbf{y}}_j)^2}$$

Cuanto más diferencias hay entre grupos, más homogéneos son cada grupo y viceversa. El índice RS puede ser considerado como una medida de similitud entre clusters.

4.2.8 Índice SD

El índice SD [Halkidi 2] se define como la media de la dispersión de los clusters y la separación total separación entre ellos:

$$SD = \alpha S_a + S_t$$

Donde α es un factor de ponderación, S_a es la media de la dispersión entre clusters y S_t es la separación total entre clusters.

La media de la dispersión de los clusters se define como:

$$S_a = \frac{1}{k}\sum_{i=1}^{k}\frac{\|\sigma(v_i)\|}{\|\sigma(X)\|}$$

donde K es el número de clusters.

La separación total entre clusters se define como:

$$S_t = \frac{D_{max}}{D_{min}}\sum_{i=1}^{k}\left(\sum_{j=1}^{k}\|v_i - v_j\|\right)^{-1}$$

El número K que minimiza SD es el óptimo número de clusters que mejor concuerdan con los datos/objetos.

4.2.9 Índice S_Dbw

Similar a SD, el índice S_Dbw está también basado en la compacidad y separación, es decir, dispersión entre clusters y distancia dentro de los clusters.
Se necesita definir la densidad interna y la varianza de los clusters. La densidad es:

$$Dens\_bw(k) = \frac{1}{k(k-1)} \sum_{i=1}^{k} \left(\sum_{j=1, j\neq i}^{k} \frac{density(C_i \cup C_j)}{\max\{density(C_i), density(C_j)\}} \right)$$

donde K es el número de clusters. La funcion de densidad se define por:

$$density(C) = \sum_{i=1}^{|C|} f(x_i, \mu)$$

donde μ es el centro del cluster C, |C| es el número de puntos en el cluster C, y la función f(s,y) está definida por:

$$f(x,u) = \begin{cases} 0 & \text{if } d(x,u) > stdev, \\ 1 & \text{otherwise.} \end{cases}$$

stdev es la media de la desviación estandar de los clusters:

$$stdev = \frac{1}{k} \sqrt{\sum_{i=1}^{k} \|\sigma(C_i)\|}.$$

Si $C = C_i \cup C_j$,podemos tomar μ como punto medio del segmento entre μi and μ_j, los cuáles son los centros de los clusters C_i and C_j, respectivamente.

La varianza de los clusters mide la media de la dispersión de los clusters. Se define como:

$$Scat(k) = \frac{1}{k} \sum_{i=1}^{k} \frac{\|\sigma(C_i)\|}{\|\sigma(D)\|}$$

donde σ(S) es la varianza de los datos del conjunto D. Su pth dimension es:

$$\sigma(S)_p = \frac{1}{n} \sum_{i=1}^{n} \left(x_{ip} - \frac{\sum_{j=1}^{n} x_{jp}}{n} \right)^2, \quad p = 1, 2, \ldots, d,$$

donde n es el número de puntos en D, es decir, D = {x_1, . . . , x_n}, y d es la dimentsion de los datos.

De forma similar, σ(Ci) es la varianza del cluster Ci , es decir,

$$\sigma(C_i)_p = \frac{1}{|C_i|} \sum_{y \in C_i} (y_p - \mu_{ip})^2, \quad p = 1, 2, \ldots, d,$$

donde μi_p es la pth dimensión del centro del cluster C_i y y_p es la pth dimensión del punto y en el cluster Ci.

El índice S_Dbw se define como:

$$S\_Dbw(k) = Scat(k) + Dens\_bw(k).$$

El número K que minimiza S_Dbw es el óptimo número de cluster que concuerdan con los datos/objetos.

4.2.10 Criterio de Información Bayesiana

Para un conjunto de datos/objetos, el algoritmo EM maximiza el likelihood de los parámetros para generar los parámetros de las K distribuciones Gaussianas, donde K es el número de clusters asumidos para los datos.

Cuando se usa un importante número de distribuciones Gaussianas, éstas pueden modelar muy bien los datos y producir una likelihood alta. Sin embargo, en la mayoría de los casos, el objetivo del agrupamiento de datos es describir la población total de puntos objeto. Por lo tanto el fenómeno de sobre-modelar los datos observados y no ser capaz de generalizar a otros datos es llamado Overfitting. Consecuentemente, se tiende a evitar modelos de clustering muy complejos.

El Criterio de Información Bayesiana (BIC) tiene como función evitar el overfitting y está definido por:

$$BIC = 2\ln(L) - v \ln(n)$$

donde n es el número de puntos objeto, L es el likelihood de los parámetros para generar los datos en el modelo y v es el número de parámetros libres (en un modelo mixto el número de parámetros es el vector medio de cada grupo, más el número de elementos de la matriz o matrices de covarianza, más el número de las probabilidades mixtas) en el modelo mixto Gaussiano. BIC tiene en cuenta la concordancia del modelo con los datos y la complejidad del modelo.

4.2.11 Coeficiente de Partición

Este índice, denotado por PC, mide la cantidad de solapamiento entre clusters. Se utiliza para evaluar un algoritmo clustering tipo Fuzzy como Fuzzy C-means. Fue definido por [Bezdek] como:

$$PC(c) = \frac{1}{N}\sum_{i=1}^{c}\sum_{j=1}^{N}(\mu_{ij})^2$$

donde u_{ij} es el coeficiente de pertenencia del punto j en el cluster i. La desventaja de PC es la falta de una conexión directa con las propiedades inherentes en los datos.

El número óptimo de clusters es el que hace PC un valor máximo.

4.2.12 Entropia de Clasificación

Se denota CE y es un coeficiente similar a PC. Se define como:

$$CE(c) = -\frac{1}{N}\sum_{i=1}^{c}\sum_{j=1}^{N}\mu_{ij}log(\mu_{ij})$$

4.2.13 Índice de Partición

El índice de partición (SC) es el cociente entre la suma de compacidad y separación entre clusters [Bensaid]:

$$SC(c) = \sum_{i=1}^{c}\frac{\sum_{j=1}^{N}(\mu_{ij})^m||x_j - v_i||^2}{N_i\sum_{k=1}^{c}||v_k - v_i||^2}$$

El término N_i es el índice de cardinalidad igual a:

$$\sum_{k=1}^{n} u_{ik}.$$

Cuanto más pequeño es el valor de SC mejor es el resultado del proceso de clustering. Se utiliza para evaluar un algoritmo clustering tipo Fuzzy como Fuzzy C-means.

4.2.14 Índice de Xie y Beni

El objetivo del índice de Xie y Beni (XB) es cuantificar el cociente entre la variación total dentro de los clusters y la separación de los clusters [Xie]:

$$XB(c) = \frac{\sum_{i=1}^{c}\sum_{j=1}^{N}(\mu_{ij})^m||x_j - v_i||^2}{Nmin_{i,j}||x_j - v_i||^2}$$

El número óptimo de clusters debería minimizar el valor de este índice. Se utiliza para evaluar un algoritmo clustering tipo Fuzzy como Fuzzy C-means.

4.3 Criterio Interno

El resultado de un algoritmo de clustering se evalúa utilizando cantidades y características inherentes a los datos/objetos.

Existen dos casos dependiendo de la estructura para los que se aplica un criterio interno:

1) Clusters formados por un algoritmo Jerárquico

La matriz cophenética, Pcc, representa el diagrama jerárquico que produce un algoritmo jerárquico. El elemento Pc(i, j) representa el nivel de proximidad relativo a 2 puntos x_i y x_j que se encuentran en el mismo cluster la primera vez.

El coeficiente de correlación cophenético es un índice estadístico que mide el grado de similitud entre la matriz Pcc y la matriz de proximidad P.

$$CPCC = \frac{\frac{1}{M}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} d_{ij}c_{ij} - \mu_P\mu_C}{\sqrt{\left(\frac{1}{M}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} d_{ij}^2 - \mu_P^2\right)\left(\frac{1}{M}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} c_{ij}^2 - \mu_C^2\right)}}$$

Donde M=N(N-1)/2 y N es el número de puntos objeto a agrupar.

$$\mu_P = \frac{1}{M}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} d_{ij},$$

$$\mu_C = \frac{1}{M}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} c_{ij},$$

El rango del CPCC es [−1, 1]. Valores próximos a 1 indica una gran similitud entre P y Pc. Se puede considerar que el resultado de clustering es aceptable si el valor del coeficiente es superior a 0.85.

De esta forma, como existen tantos algoritmos clustering de tipo jerárquico como combinaciones según sea la elección de la distancia y el método de linkage, el coeficiente cophenético se utiliza no sólo para detectar si el resultado es aceptable (el algoritmo es capaz de encontrar un cluster interesante) sino que sirve también para elegir la mejor combinación de entre todas las posibles.

Si consideramos los datos/objetos Iris se tiene como resultado:

Distancia/linkage	*Single*	*Complete*	*Average*	*Centroid*	*Ward*
Euclídea	*0.8300*	*0.7514*	***0.8543***	*0.8537*	*0.8226*
Seuclídea	*0.8300*	*0.7514*	***0.8543***	*0.8537*	*0.8226*
Cityblock	*0.8354*	*0.7325*	*0.8527*	*0.8601*	*0.8449*
Mahal	*0.6450*	*0.4720*	*0.6767*	*0.6515*	*0.5250*
Minkowski	*0.8300*	*0.7514*	***0.8543***	*0.8537*	*0.8226*

Tabla 4.1: Valores del coeficiente cophenético para diferentes combinaciones de distancia y linkage (datos Iris).

2) Cuando se tienen los grupos resultado de un algoritmo clustering

El objetivo es encontrar el grado de acuerdo entre un resultado de clustering dado C formado por K clusters y la matriz de proximidad P.

Para esto se utilizan los índices Hubert's Γ y Normalized Γ. Estos necesitan calcular la matriz:

$$Y_{ij} = \begin{cases} 1 & si\ \ x_i\ \ y\ \ x_j\ \ están\ \ en\ \ clusters\ \ diferentes \\ 0 & otra\ \ situación \end{cases}$$

Hubert's Γ es un índice que mide la correlación entre 2 matrices A y B, de dimensiones NxN [Theodoris]. [Theodoris] analiza la utilización de este índice como un criterio interno y externo para analizar la validez de un resultado de clusters. Para 2 matrices simétricas A y B, la estadística está definida por:

$$Hubert's\ \Gamma = (1/M)\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} A(i,j)B(i,j)$$

Donde A(i,j) y B(i,j) son los (i,j) elementos de las matrices A y B, y M=N(N-1)/2. Valores altos de Γ indica una alta relación entre A y B. La estadística Γ normalizada puede también ser usada:

$$\hat{\Gamma} = \frac{(1/M)\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}(A(i,j)-\mu_A)(B(i,j)-\mu_B)}{\sigma_A \sigma_B}$$

donde el denominador representa la varianzas al cuadrado de A y B. La estadística Γ normalizada varía entre -1 y 1. Valores cercanos a 1 sugieren una buena relación entre A y B.

Aplicando la estadística de Hubert a los datos Iris agrupados con K-means se obtiene un valor alto. Lo que nos dice que el algoritmo es capaz de encontrar un cluster interesante que concuerda con los datos/objetos.

El programa para este criterio interno es iris_hubertgamma.m. Este criterio no se integrará en la interfaz gráfica.

4.4 Criterio externo

Validar un resultado clustering mediante un criterio externo supone comparar el resultado de un algoritmo con una partición de referencia. Esto raramemente sucede en un problema real ya que se desconoce a priori la estructura real de los datos agrupados.

Sin embargo, los métodos implicados en el criterio externo pueden ser útiles para evaluar un algoritmo de clustering determinado. También como se verá en el siguiente capítulo, dichos métodos son útiles para valorar la estabilidad de un resultado clustering.

Existen varios métodos para comparar particiones. En [Haldiki 1] destacan los de la tabla (4.2).

Nombre del índice	***Fórmula***
Estadística Rand	$R=\frac{a+d}{M}$
Coeficiente de Jaccard	$J=\frac{a}{a+b+c}$
Índice de Folkes y Mallows	$FM=\sqrt{\frac{a}{a+b}\cdot\frac{a}{a+c}}$
Estadística de Hubert	$\Gamma=\frac{1}{M}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} X_{ij}Y_{ij}$
Estadística de Hubert Normalizada	$\hat{\Gamma}=\frac{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}(Y_{ij}-u_y)(X_{ij}-u_x)}{M\,\sigma_x\sigma_y}$

Tabla 4.2: Índices externos más utilizados

Sean $C=\{C_1,....C_n\}$ los clusters resultado de un algoritmo clustering y $P=\{P_1...P_s\}$ una particicion de referencia. Se define:

- *SS: número de parejas en donde ambos pertenecen al mismo cluster de C y el mismo grupo de la partición P.*
- *SD: número de parejas que pertenecen al mismo cluster en C pero en diferentes grupos en P.*
- *DS: número de parejas que pertenecen a diferentes clusters en C pero a un mismo grupo en P.*
- *DD: número de parejas que pertenecen a diferentes clusters en C y diferentes grupos en P.*

Asumiento que a,b,c y d son los números SS,SD, DS y DD respectivamente, entonces a+b+c+d=M. M es el máximo número de pares (M=N(N-1)/2 donde N es el número total de puntos a agrupar). El rango de Rand, Jaccard y Folkes está entre 0 y 1. Altos valores de esos índices indican una gran similitud entre C y P. Hubert también varía entre 0 y 1. La estadística normalizada tiene un rango entre -1 y 1. Existe otro índice denominado Adjusted Rand [Hubert] que mejora al índice Rand (pues su valor no es cero cuando se comparan particiones aleatorias). Se define como:

$$ARI = \frac{\binom{n}{2}(a+d) - [(a+b)(a+c)+(c+d)(b+d)]}{\binom{n}{2}^2 - [(a+b)(a+c)+(c+d)(b+d)]}$$

donde n representa el número total de puntos objeto.

4.5 Resumen

En este capítulo se ha mostrado como evaluar un resultado clustering. Indirectamente, dicha evaluación determina el número óptimo de clusters. La evaluación responde a un criterio interno cuando el resultado se corresponde con las características internas de los datos/objetos. La evaluación utiliza un criterio externo cuando compara el resultado clustering con otra partición de referencia. Y hace uso de un criterio relativo cuando se utilizan índices que analizan la compacidad de los clusters y su separación.

Esta forma de evaluar se aplica para cada algoritmo de clustering de forma independiente. Existen otras formas de evaluación diferentes a los criterios descritos. En el próximo capítulo se analizará la estabilidad del resultado y al mismo tiempo se proporcionará un nueva forma de determinar el número óptimo de clusters.

4.6 Bibliografía

[Bensaid] A.M. Bensaid, L.O. Hall, J.C. Bezdek, L.P. Clarke, M.L. Silbiger, J.A. Arrington, and R.F. Murtagh. Validity-guided (Re)Clustering with applica-tions to image segmentation. IEEE Transactions on Fuzzy Systems, 4:112-123, 1996.

[Davies] Davies DL, Bouldin DW (1979). "A Cluster Separation Measure." IEEE Transactions on Pattern Analysis and Machine Intelligence, 1(2), 224-227.

[Data Clustering] Data Clustering. ASA-SIAM Series on Statistics and Applied Probability 2007.

[Dunn] Dunn JC (1974). Well Separated Clusters and Optimal Fuzzy Partitions." Cybernetics and Systems, 4(1), 95-104.

[Halkidi 1] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "Cluster validity methods: Part I," ACM SIGMOD Record, vol. 31, no. 2, June 2002.

[Halkidi 2] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "Cluster validity methods: Part II," ACM SIGMOD Record, vol. 31, no. 3, September 2002

[Hubert] Hubert L, Arabie P (1985). "Comparing Partitions." Journal of Classication, 2, 193-218.

[Jain] Jain AK, Dubes RC (1998). Algorithms for Clustering Data. Prentice-Hall, New Jersey.

[Maulik] U. Maulik and S. Bandyopadhyay, "Performance evaluation of some clustering algorithms and validity indices," IEEE Transaction on Pattern Analysis and Machine Learning, vol. 24, no. 12, pp. 1650-1654, Decem- ber 2002.

[Nguyen] Nguyen N, Caruana R (2007). "Consensus Clusterings." In Proceedings of IEEE International Conference on Data Mining, pp. 607{612. IEEE Computer Society, Washington, DC.

[Rand] Rand WM (1971). "Objective Criteria for the Evaluation of Clustering Methods." Journal of the American Statistical Association, 66, 846-850.

[Rousseeuw] P.J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. 1987. Journal of Computational and Applied Mathematics. 20. 53-65

[Theodoridis] Theodoridis, S. and Koutroumbas, K., 1999. Pattern recognition (Academic Press, San Diego, CA).

[Xie] X. L. Xie and G. A. Beni. Validity measure for fuzzy clustering. IEEE Trans. PAMI, 3(8):841-846, 1991.

5. ESTABILIDAD DEL PROCESO DE CLUSTERING Y NÚMERO ÓPTIMO DE CLUSTERS

5.1 Introducción

La estabilidad de un resultado de análisis de cluster es otro criterio de validación utilizado [Ben-Hur], [Lange].

Se considera una solución clustering estable aquella que mantiene la arquitectura de los grupos de datos/objetos obtenidos aunque se produzcan variaciones en los datos originales [Lange]. Indirectamente el estudio de la estabilidad, también servirá para calcular el número óptimo de clusters. Este será un método complementario a la utilización de los índices de criterio relativo.

5.2 Estabilidad

Hay varios trabajos relacionados con el estudio de la estabilidad de un resultado clustering. Para este trabajo se ha elegido el denominado Replication Analysis [Breckenridge].

Para realizar un "replication analysis" el conjunto de los datos/objetos se divide en 2 subconjuntos iguales. Con los datos de un subconjunto se aplica un algoritmo clustering y el resultado es considerado el "ground truth". Es decir, es utilizado para clasificar los puntos objeto del segundo subconjunto.

La clasificacion es un método supervisado. Cada elemento de un conjunto tiene una etiqueta asignada por un experto o por previo conocimiento. Si todos los elementos son agrupados en diversos grupos denominados clases, entonces un nuevo elemento será clasificado cuando se le asigne a una de las clases.

La clasificación es una etapa importante en un proceso de replicación. Este se define como:

1) *Dos subconjuntos disjuntos A y B son seleccionados de forma aleatoria de un conjunto de puntos objeto D.*
2) *El subconjunto A es agrupado en k clusters disjuntos , $<A_1, A_2 ... A_k>$, tal que $A = A_1 \cup ... \cup A_k$; se denota la partición de A como Clu(A).*
3) *El subconjunto B es agrupado en k clusters disjuntos, $<B_1, B_2 ... B_k>$, tal que $B = B_1 \cup ... \cup A_k$; se denota la partición de B como Clu(B). Se construye un modelo de clasificación para aprender la estructura de clases del subconjunto A, asumiendo que A es el conjunto de entrenamiento y Clu(A) es the "ground trouth".*
4) *Los puntos objeto del subconjunto B son clasificados utilizando el modelo de clasificación utilizando el modelo de clasificación aprendido en la etapa 4. Se denota la partición del subconjunto B por Pred(B).*
5) *El grado de replicación entre A y B es medido por la concordancia entre las dos particiones del subconjunto B, Pred(B) y Clu(B).*

[Breckenridge] encontró que el nivel de replicación del cluster entre A y B indicaba la habilidad

del método clustering para reproducir la estructura real de los clusters.

Otros trabajos hacen uso del resultado de Breckenridge para validar la estabilidad de los resultados clustering variando el número de clusters y eligiendo kopt como el k con una solución más estable [Wu]. En estos estudios de validación de la estabilidad, el procedimiento de replicación descrito, fue repetido numerosas veces para cada número de clusters k. Durante cada repetición, los datos originales son divididos en 2 subconjuntos disjuntos A y B de forma aleatoria (como se ha explicado en la etapa 1).

En la etapa (5), se tienen dos particiones Pred(B) y Clu(B). Para cada k, se calculan numerosas parejas de particiones Pred(B) y Clu(B), y por lo tanto se tiene que realizar múltiples evaluaciones entre ellas. El k que conduce a la mejor concordancia entre Pred(B) y Clu(B) es considerado como el kopt de los datos iniciales. Como hay numerosas evaluaciones de concordancia para cada k, calcular su promedio es una forma de medir la totalidad. El k que produce la más alta concordancia se considera el más apropiado.

Hay varios algoritmos para producir Pred(B) y varios tipos de medidas para valorar la concordancia entre Pred(B) y Clu(B)[Breckenridge][Wu].

En la etapa (4) se hace necesario un algoritmo de clasificación con un pequeño margen de error en la clasificación para que la concordancia entre Pred(B) y Clu(B) pueda estar relacionada con la estabilidad de la solución clustering sin que tenga influencia una pobre clasificación. Sin embargo, no hay un algoritmo de clasificación que se reconozca como el mejor [Breckenridge].

[Lange] sugirió la utilización de un clasificador que imitara al algoritmo clustering utilizado para los subconjuntos A y B. Si no hay una elección válida, también se aconsejaba la utilización del clasificador Pnn, el cual asigna una clase a un nuevo dato examinando los P vecinos más proximos.

Además de elegir un clasificador, hay que seleccionar los índices para medir la concordancia entre Pred(B) y Clu(B). En el capítulo anterior se decribieron los principales índices para comparar particiones (apartado criterio externo).

Entre esos índices ARI se ha mostrado como el más robusto y por ello es ampliamente utilizado [Arabie]. Aunque es una aproximación interesante, la media de numerosos ARI (o de los otros índices) no es necesariamente una medida segura de concordancia. Por lo tanto, se hace necesario la utilización de un test estadístico para mostrar que las medias son significativamente diferentes.

Para estudiar la estabilidad con los datos Iris, se ha empleado como algoritmo de clustering el Jerárquico (con la combinación de distancia y linkage que maximiza el coeficiente cophenético), un algoritmo de clasificación Pnn y como índices para valorar la concordancia entre particiones (Rand, Adjusted Rand, Folkes y Jaccard).

En el capítulo de la descripción de la interfaz gráfica se mostrará los resultados de los índices de estabilidad obtenidos para los datos Iris.

5.3 Resumen

En este capítulo se han explicado los métodos utilizados para evaluar un resultado clustering para un algoritmo determinado. Para estudiar la estabilidad se ha focalizado el estudio utilizando el método jerárquico como método clustering, Pnn como método de clasificación, los métodos (explicados en el criterio externo) para comparar particiones (y por lo tanto medir la estabilidad y el número óptimo de clusters) y por último un test estadístico de Wilcoxon para mostrar las diferencias significativas de los resultados.

En el próximo capítulo se analizará la fusión de varios resultados clustering a los que se aplicará de nuevo (y si fuese necesario) algunos criterios de validación. En lugar de decidir qué algoritmo es el mejor a raíz del proceso de validación, se pretende fusionar los resultados para obtener una única partición. En general dicha partición sería al menos igual o mejor que los resultados clustering obtenidos para cada algoritmo de forma individual.

5.4 Bibliografía

[Arabie] L. J. Arabie, P. Hubert. Comparing partitions. Journal of classification. 2: 193-218. 1985.

[Ben-Hur] A. Ben-Hur, A. Elisseeff, I. Guyon. A stability based method for discovering structure in clustered data. In Aetman, R.B. (Ed.), et al. Pacific Symposium on Biocomputing. New Jersey World Scientific Publishing Co. 2002.

[Breckenridge] J. Breckenridge. Replicating cluster analysis: Method, consistency, and validity. Multivariate Behavioural Research, 24: 147- 161. 1989.

[Lange] T. Lange, V. Roth, M. Braun. Stability-based validation of clustering solutions. Neural Computation. 16:1299–1323. 2004.

[Wu] F. Wu. Computational Methods for Analysis and Modeling of Time-Course Gene Expression Data. Ph. D. Thesis. University of Saskatchewan. 2004.

6. SOLUCIÓN DE CLUSTERING CONSENSUADA

6.1 Introducción

Dada la gran variedad de algoritmos clustering y que muchos requieren una inicialización aleatoria (que puede condicionar el resultado), se hace necesario hacer funcionar el mismo algoritmo varias veces y fusionar el resultado. También puede consensuarse varias soluciones de diferentes algoritmos.

Después de combinar diferentes resultados de distintos algoritmos de clustering sobre el mismo conjunto de datos/objetos aparece una única solución. En la práctica se ha encontrado que dicha solución aumenta la robustez y la calidad de los resultados del proceso de clustering. De esta forma el objetivo de buscar una solución consensuada es mejorar los resultados clustering obtenidos de forma individual.

Si inicialmente se hace un estudio del número óptimo de clusters (por ejemplo utilizando los índices relativos, estabilidad) se puede obtener una solución única kopt o algo dispersa [kopt1 kopt2] (en general kopt1 y kopt2 debieran de ser bastante próximos). Si la mayoría de los índices apuesta por el mismo valor, se puede considerar que se ha obtenido un kopt único. Pero también es posible que no haya una mayoría de índices apostando por un único kopt: puede haber dudas entre 2 o 3 valores para el kopt. Posteriormente, y en ambas situaciones, se pueden generar varias resultados clustering utilizando diferentes algoritmos con repetición (sólo para los kopts seleccionados como variable inicial) y a continuación fusionarlos. En general, la fusión redistribuirá los puntos objeto dentro de los clusters mejor que aplicar una sóla vez un solo algoritmo. Si la fusión se realiza para resultados en donde kopt está entre pocos valores se aplicarán de nuevo algunos índices relativos para decidir un único valor para el número de clusters óptimo.

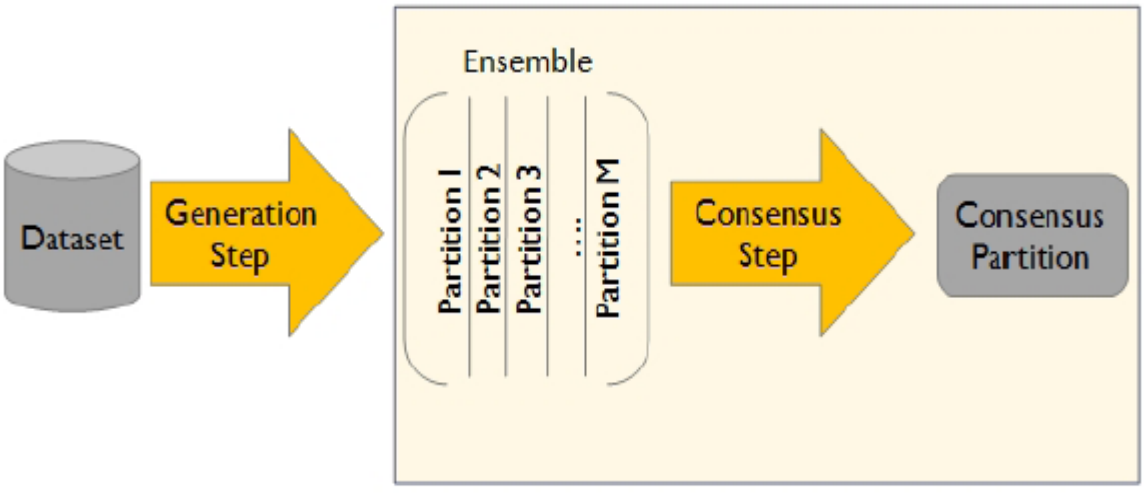


Figura 10: Proceso de búsqueda de una partición consensuada.

6.2 Solución consensuada

Una vez se han obtenido un conjunto de resultados clustering, se pueden aplicar varios algoritmos para encontrar una solución clustering consensuada. Estos métodos se suelen dividir en 3 categorias: (i) similitud por parejas (ii) basado en grafos y (iii) basado en características, respectivamente [Strehl]. En esta memoria sólo se ha utilizado el método de similitud por parejas que se describe a continuación.

6.2.1 Algoritmo de similitud por parejas

Este método está basado en la similitud por parejas entre los puntos objeto. En particular, a partir de una colección de puntos objeto $X=\{x_1,....x_N\}$ se pueden generar M resultados clustering $\{f_1.....f_M\}$. Para cada resultado clustering se construye una matriz de similitud NxN. Por lo que se tendrá M matrices de similitud $\{S_1 ....S_M\}$. Cada coeficiente de una matriz de similitud representa la relación entre dos puntos objeto. Si ambos puntos objeto están en el mismo cluster el coeficiente de la matriz será un 1. El valor será 0 en cualquier otro caso.

Más precisamente, la similitud entre 2 puntos objeto x_i, x_j para el m-th resultado clustering puede ser expresada como:

$$S_m(x_i, x_j) = \begin{cases} 1 & if\ C(x_i) = C(x_j) \\ 0 & cualquier\ otro\ caso \end{cases}$$

Todas las matrices de similitud obtenidas se fusionan para obtener una matriz consensuada CO [Fred 2].

Cada elemento de la matriz CO representa el grado de similitud entre dos puntos objeto. Su valor es el promedio de entre todas las matrices de similitud. Formalmente, la similitud entre x_i, y xj está definida por:

$$CO(x_i, x_j) = \frac{1}{M} \sum_{m=1}^{M} S_m(x_i, x_j)$$

Como CO es una matriz de similitud, se puede aplicar un algoritmo tipo jerárquico para obtener la partición final que representará la fusión de todos los resultados clustering. [Fred 1, 2] emplea un método de linkage single y average para generar la partición final.

En el próximo capítulo se mostrará la interfaz gráfica que incorpora la fusión de resultados clustering.

6.3 Resumen

En este capítulo se ha introducido como fusionar los resultados clustering en busca de una solución consensuada. Existen varios métodos, pero se ha preferido utilizar el algoritmo de similitud por parejas pues es sencillo de comprender y al mismo tiempo aporta flexibilidad en cuanto a su implantación.

En el próximo capítulo se describirá una interfaz gráfica capaz de integrar la validación de los resultados clustering de los diferentes algoritmos de forma individual así como el resultado consensuado después de una fusión de diferentes combinaciones de resultados.

6.4 Bibliografía

[Fred 1] Fred ALN, Jain AK (2003). "Robust Data Clustering." In Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, pp. 128-136. IEEE Computer Society, Los Alamitos.

[Fred 2] Fred ALN, Jain AK (2005). "Combining Multiple Clusterings Using Evidence Accumulation." IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(6), 835-850.

*[Strehl] A. Strehl & J. Ghosh, Cluster ensembles – A knowledge reuse framework for combining multiple partitions, Journal of Machine Learning Research, **3**, 2002, 583-618.*

7. INTERFAZ GRÁFICA DE INTEGRACIÓN

7.1 Introducción

En este capítulo se integrará en una interfaz gráfica todas las técnicas descritas en los capítulos anteriores con el fin de poder aplicar los diferentes algoritmos a cualquier conjunto de datos/objetos automáticamente.

Se explicará la funcionalidad de la interfaz utilizando los datos Iris.

7.2 Interfaz gráfica

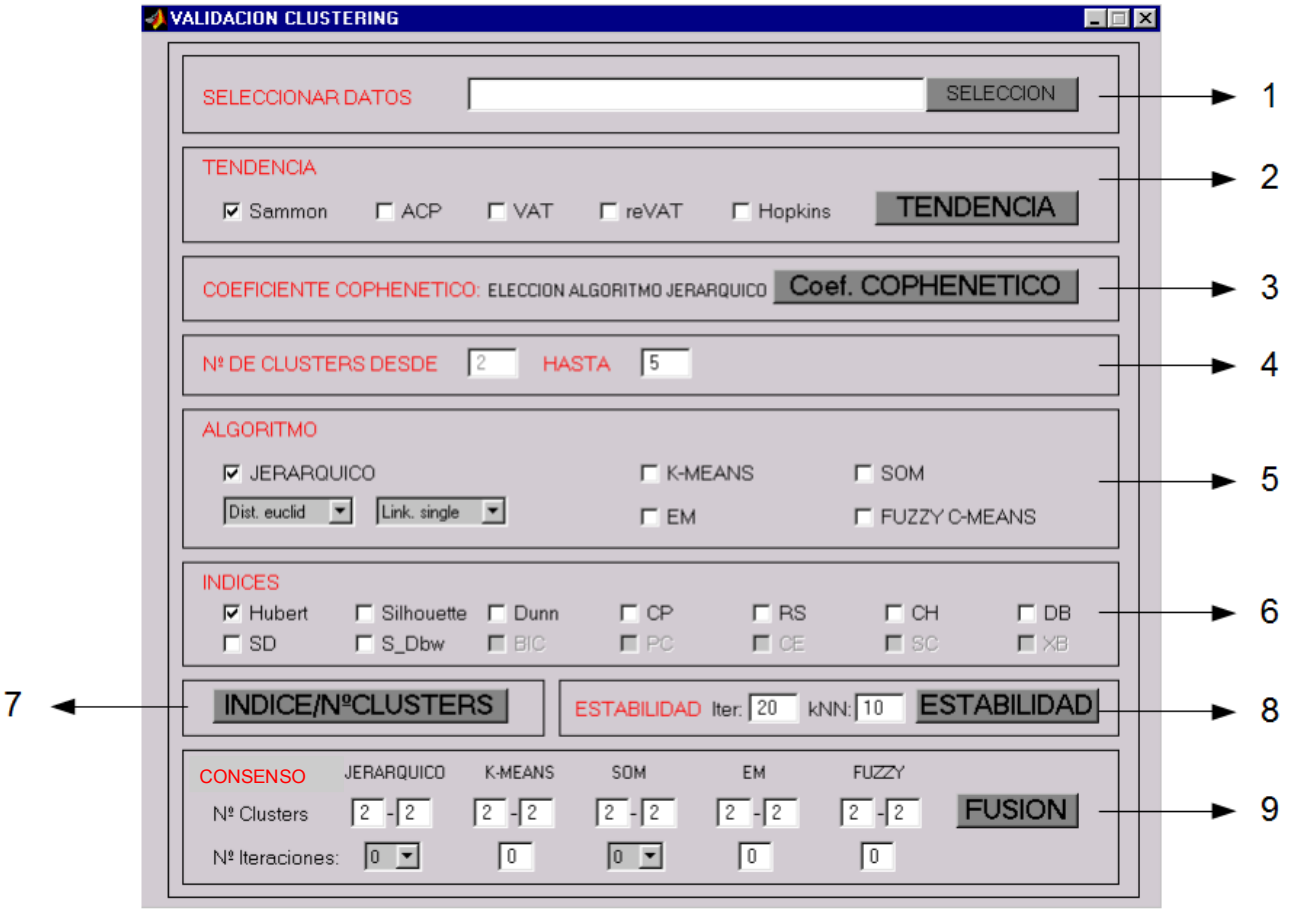


Figura 11: Interfaz gráfica para validar un proceso clustering.

La interfaz gráfica de la Figura 11 se explica siguiendo las secciones numeradas:

***Número 1**: Selección de los datos (Figura 12).*

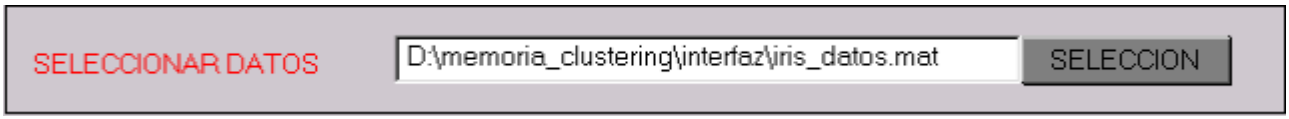


Figura 12: Sección Número 1 de la interfaz gráfica.

En esta sección se elige el fichero que contiene los puntos objeto a agrupar. En este caso, se trata del fichero iris_datos.mat que contiene los datos Iris utilizados durante todo el trabajo.

Cualquier otro fichero deberá ser un fichero tipo .mat. Podrá cambiar de nombre pero es obligatorio que el fichero guarde los datos en una variable cuyo nombre es "datos".

***Número 2:** Tendencia del proceso clustering (Figura 13). Comprobar visualmente y estadísticamente que los datos no presentan una estructura aleatoria y pueden agruparse.*

Figura 13: Sección Número 2 de la interfaz gráfica.

En esta sección se selecciona un método para evaluar la tendencia de los datos a ser agrupados. Es decir, si en los datos existen ciertos grupos que puedan visualizarse o demostrar que no tienen una estructura aleatoria (test de Hopkins).

Para Sammon, ACP, VAT y reVat las gráficas obtenidas son similares a las mostradas en el capítulo 2. Para el caso de Hopkins aparece una ventana como la siguiente (Figura 14):

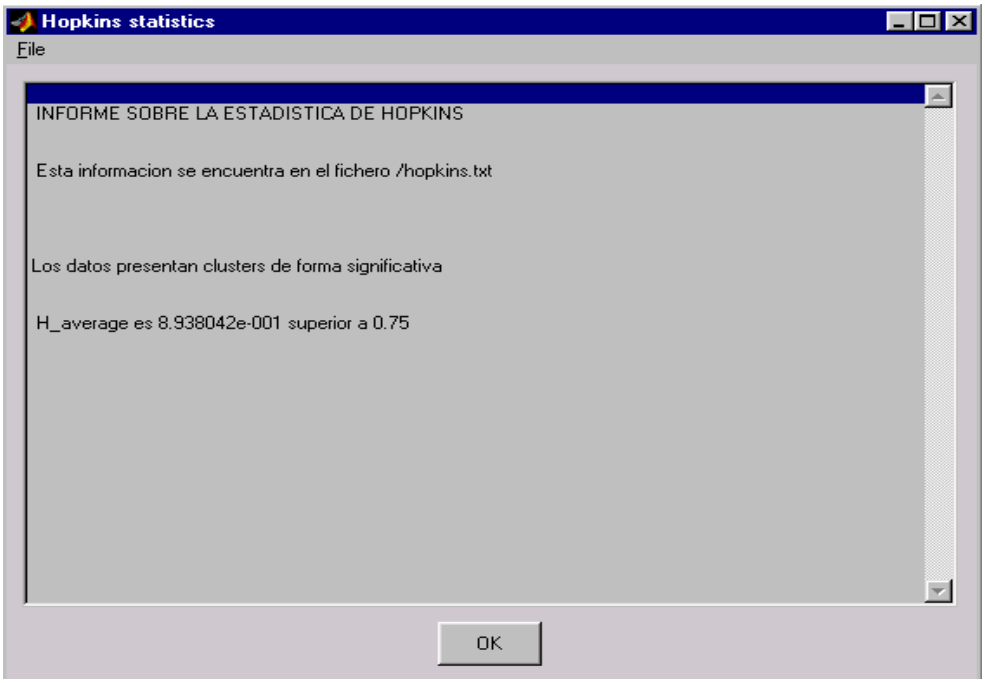


Figura 14: Ventana emergente tras seleccionar el análisis de la estadística de Hopkins.

Número 3: *Coeficiente cophenético (Figura 15).*

Figura 15: Sección Número 3 de la interfaz gráfica.

El objetivo de esta sección es seleccionar la mejor combinación de distancia, linkage para el algoritmo jerárquico. Aparece una ventana con todas las combinaciones posibles (Figura 16):

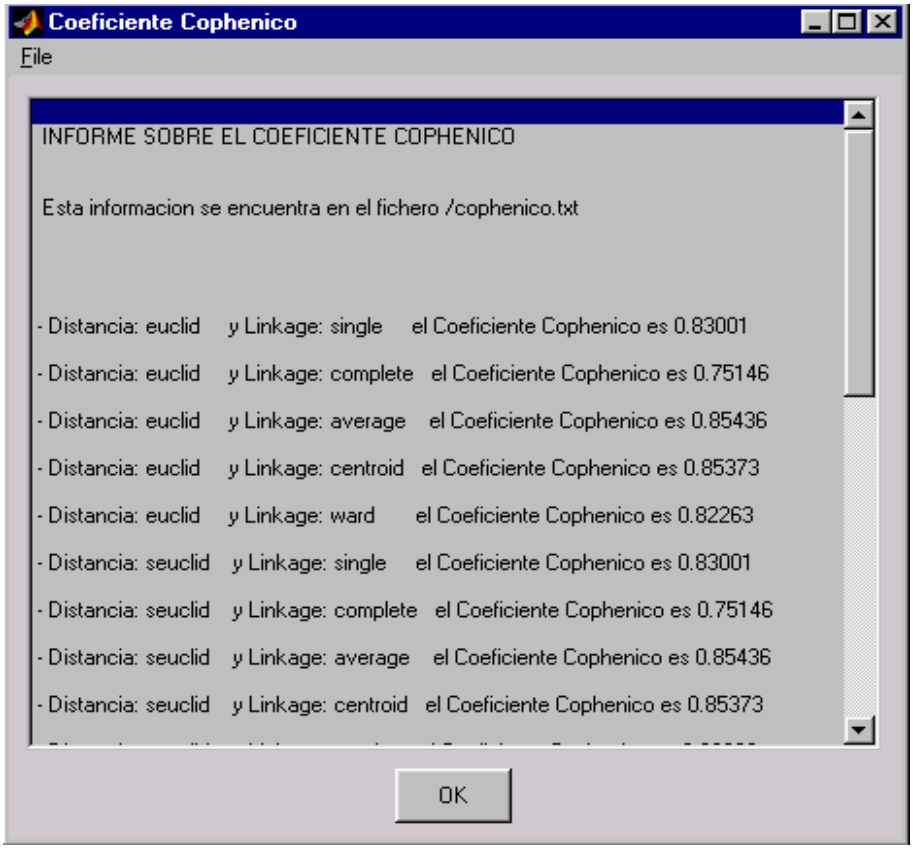


Figura 16: Ventana emergente tras seleccionar el análisis del coeficiente cophenético.

Todos los coeficientes cophenético para Iris se encuentran en la Tabla 4.1.

***Número 4:** Intervalo de valores entre los cuales se va a seleccionar el número óptimo de clusteres (Figura 17).*

Figura 17: Sección Número 4 de la interfaz gráfica.

En este apartado, se elige el intervalo del número de clusters que va a ser objeto de estudio con el fin de encontrar el número de clusters óptimo. Con la ayuda de todo lo expuesto en el capítulo 2 (sección número 2 de la interfaz) es posible detectar visualmente una aproximación del número de clusters. Esto puede servir de ayuda para concretar un intervalo.

Número 5: *Selección del método clustering (Figura 18).*

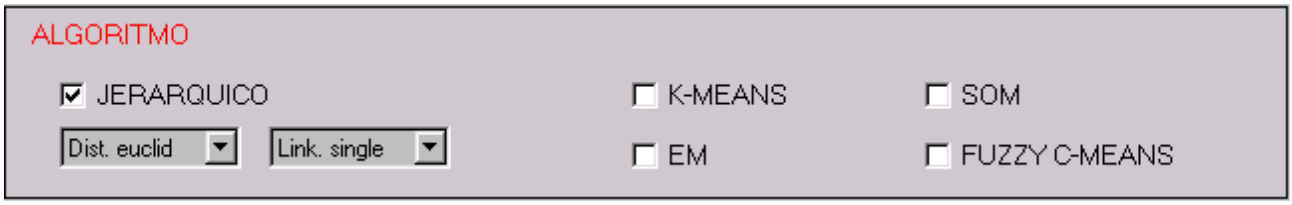


Figura 18: Sección Número 5 de la interfaz gráfica.

En esta sección se elige el algoritmo clustering que se quiere utilizar. En el caso del método jerárquico hay que seleccionar también la mejor combinación obtenida con el coeficiente cophenético (nº 3).

Número 6: *Índices (Figura 19).*

Figura 19: Sección Número 6 de la interfaz gráfica.

El apartado nº 6 muestra 14 índices que van a ser utilizados para evaluar el resultado de un algoritmo clustering. Dependiendo del método clustering seleccionado en (5) se activan o desactivan ciertos índices de validación.

Para el Jerárquico, K-means y SOM se pueden activar los 9 primeros índices desde Hubert hasta S_Dbw. Para EM se activa sólamente el índice BIC. Para Fuzzy C-means sólo los últimos 4 índices.

Número 7: *Botón que permite el estudio del número optimo de clusters utilizando los datos aportados por el intervalo de clusters, el algoritmo clustering y los índices anteriormente seleccionados en las secciones cuarta, quinta y sexta (Figura 20).*

Figura 20: Sección Número 7 de la interfaz gráfica.

Tras pulsar el boton se activa la ventana mostrada en la (Figura 21):

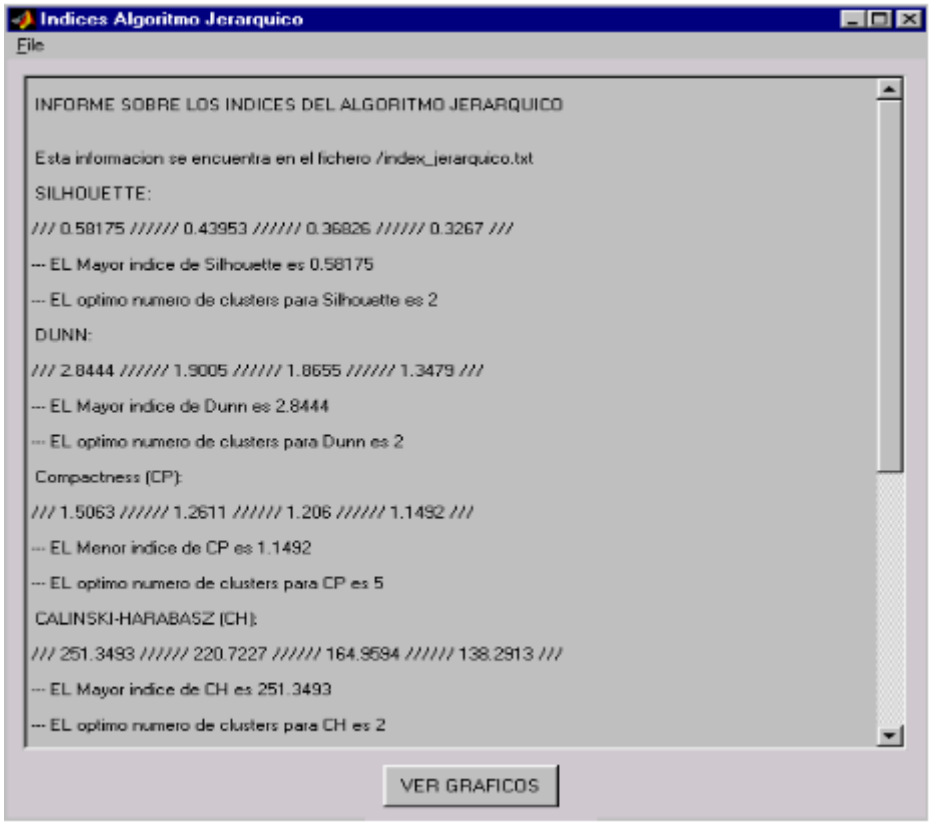


Figura 21: Ventana emergente, donde se muestran algunos índices de validación relativos: Silhouete, Dunn, CP y Calinski-Harabasz.

En esta ventana se observa el análisis para cada uno de los índices seleccionado. En este caso se muestra parte de los índices ya que es una ventana deslizante. Aparecen los índices Silhouette, Dunn, CP y Calinki-Harabasz. Por ejemplo, para Silhouette se observa que los valores obtenidos son 0.58, 0.43, 0.36 y 0.32 para K igual a 2,3,4 y 5 clusters respectivamente. El índice Silhouette mayor es 0.58 con lo que para este índice el óptimo número de clusters es 2.

También se puede observar los valores para apreciar si las diferencias no son significativas. Valores muy próximos nos indicarían que el número de clusters estaría entre los valores K correspondientes.

En la parte baja de la ventana aparece un botón para mostrar los gráficos de la variación de los índices con el número de clusters. Esto es especialmente útil para los índices Hubert, Normalized Hubert y S-squared en donde se tiene que apreciar un codo en la gráfica para determinar el óptimo número de clusters. Por ejemplo, habiendo seleccionado esos índices y pulsado sobre el botón "mostrar gráficos" se obtiene (Figura 22):

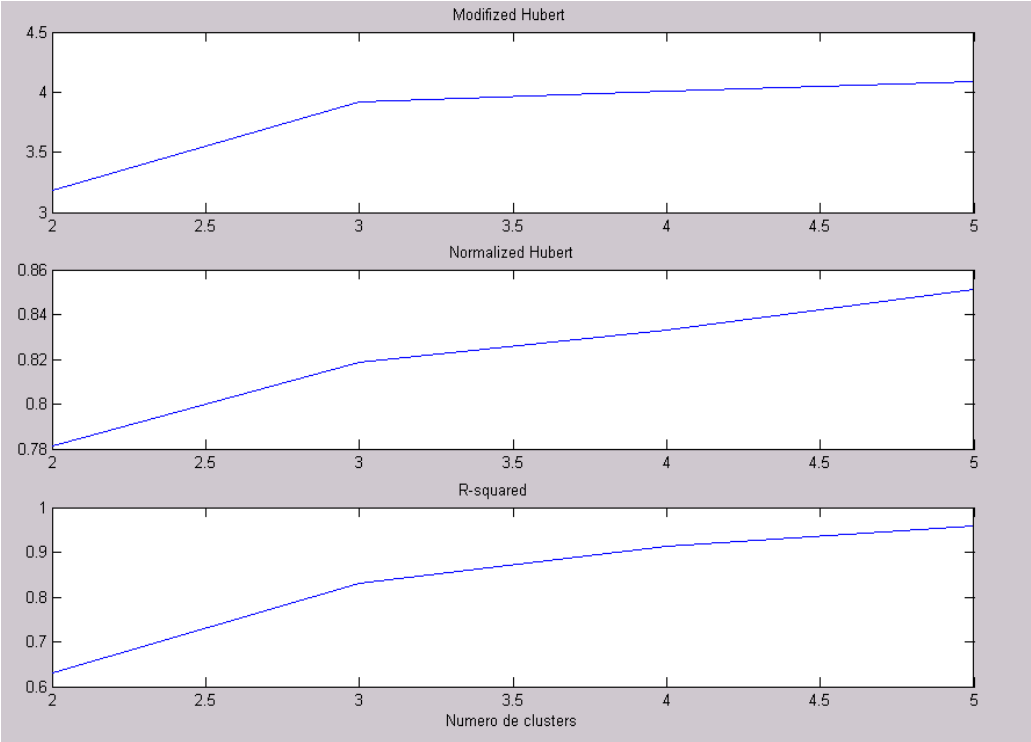


Figura 22: Gráficas obtenidas tras seleccionar los índices Hubert y R-squared.

Para Hubert y R-squared parece claro que el número óptimo de clusters es 3.

Si se selecciona los últimos índices aparecería una ventana con la siguiente información (Figura 23):

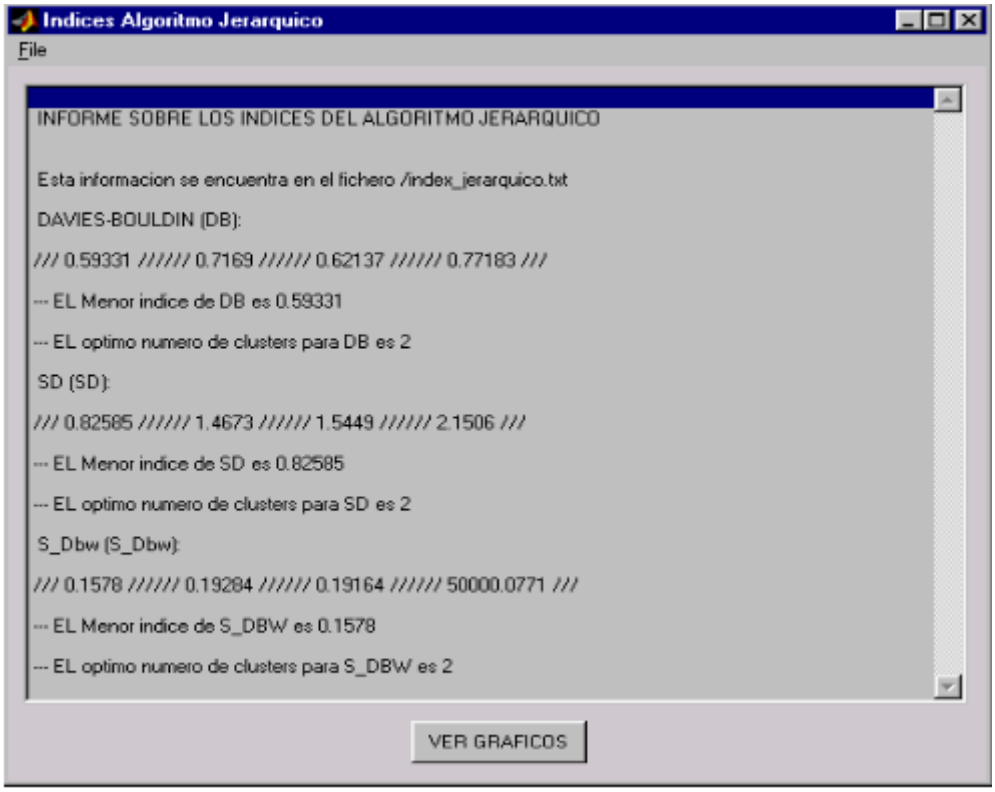


Figura 23: Ventana emergente, donde se muestran algunos índices de validación relativos: Davies-Bouldin, SD y S_Dbw.

Para Davies-Bouldin, SD y D_Dbw el número óptimo de clusters es 2. Sin embargo atendiendo al índice S_dbw no está del todo claro pues los valores para K igual a 2, 3 y 4 no se diferencia mucho.

Si se hubiese elegido el algoritmo EM (Figura 24), en la ventana aparecería información relacionada con el índice BIC (Figura 25):

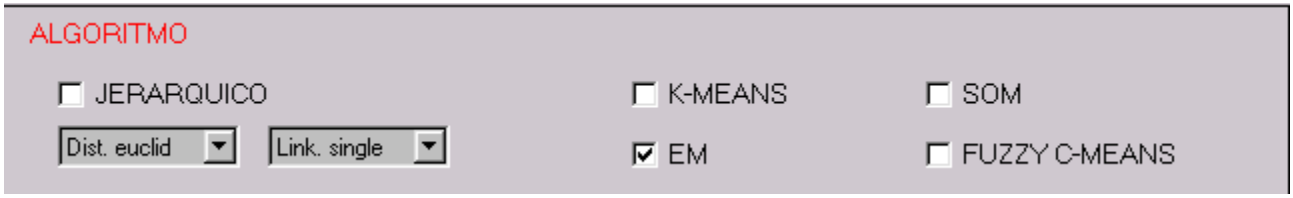


Figura 24: Selección del algoritmo EM en la sección Número 5 de la interfaz gráfica.

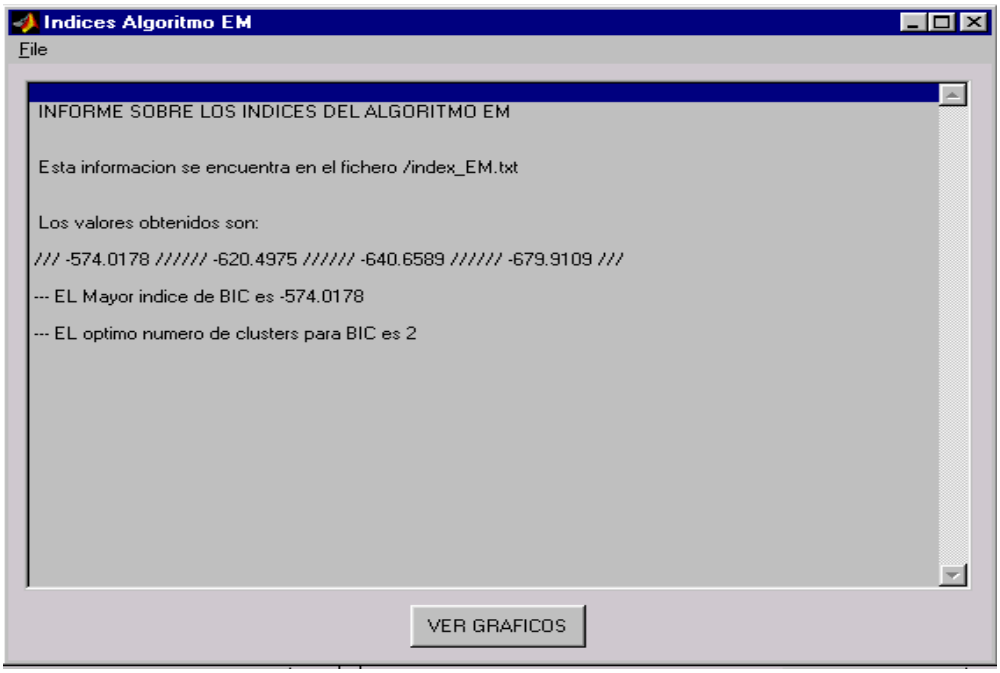


Figura 25: Ventana emergente, donde se muestra el índice BIC relacionado con el algoritmo EM.

Como se observa para BIC el número optimo de clusters es 2.

Si se hubiese elegido el algoritmo Fuzzy C-Means (Figura 26):

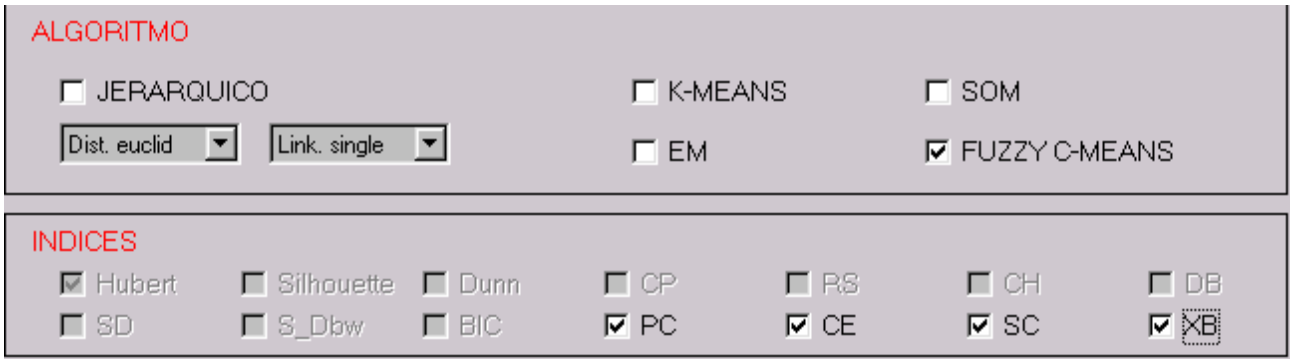


Figura 26: Selección en la interfaz gráfica del algoritmo Fuzzy C-means y sus índices.

en la ventana emergente se tendría información de los índices PC, CE, SC y BX (Figura 27):

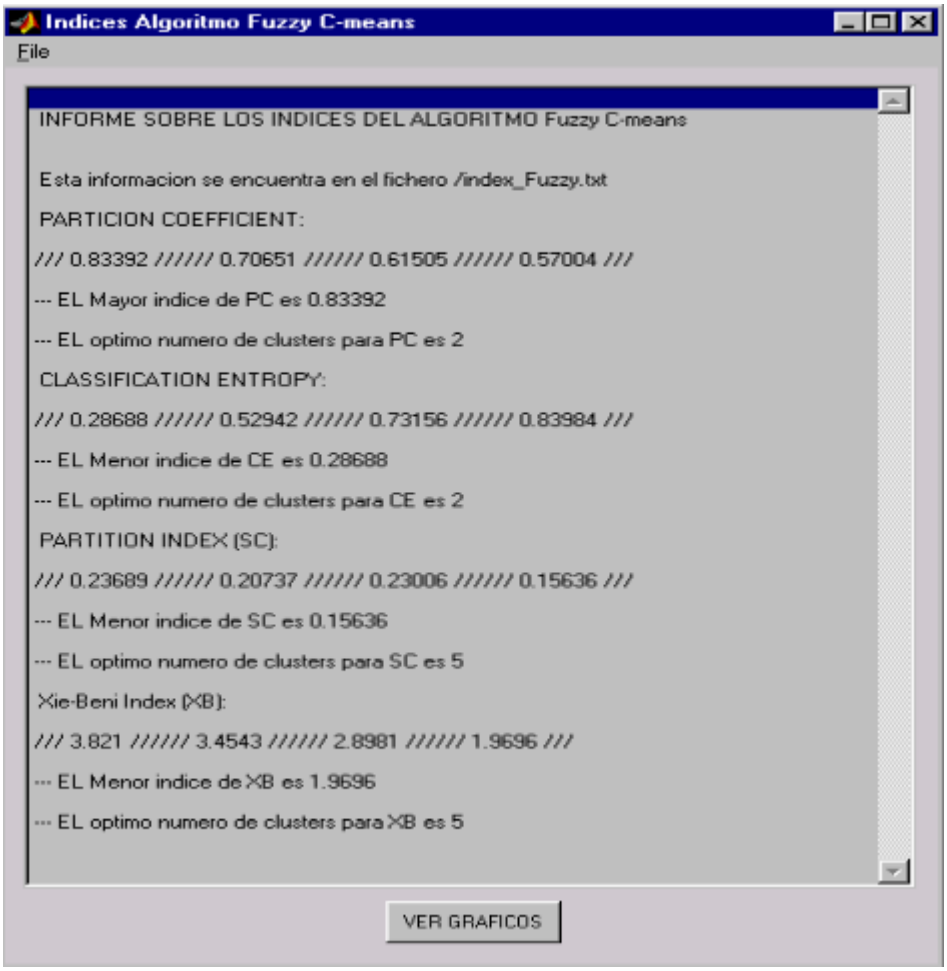


Figura 27: Ventana emergente, donde se muestra algunos índices de validación relativos para Fuzzy C-means: PC, CE, SC y XB.

Para los índices PC y CE el número optimo de clusters es 2 y para SC y XB es 5.

Número 8: *Estudio de la estabilidad y otra estimación del número óptimo de clusters (Figura 28).*

Figura 28: Sección Número 8 de la interfaz gráfica.

En el estudio de la estabilidad se utiliza el método clustering jerárquico. Por lo tanto no hay que olvidarse en mantener la mejor elección de distancia y linkage obtenidos con la comparación del coeficiente cophenético (Sección número 3 de la interfaz).

Es flexible la elección del número de iteraciones y del número de los vecinos más proximos para el algoritmo de clasificación Pnn.

Tras pulsar el boton "Estabilidad" aparece una ventana en donde se muestran los valores de los índices Rand, Jaccard, Folkes y Adjusted Rand (Figura 29). Nótese que se trata de una ventana con cursor deslizante para visualizar todos los índices.

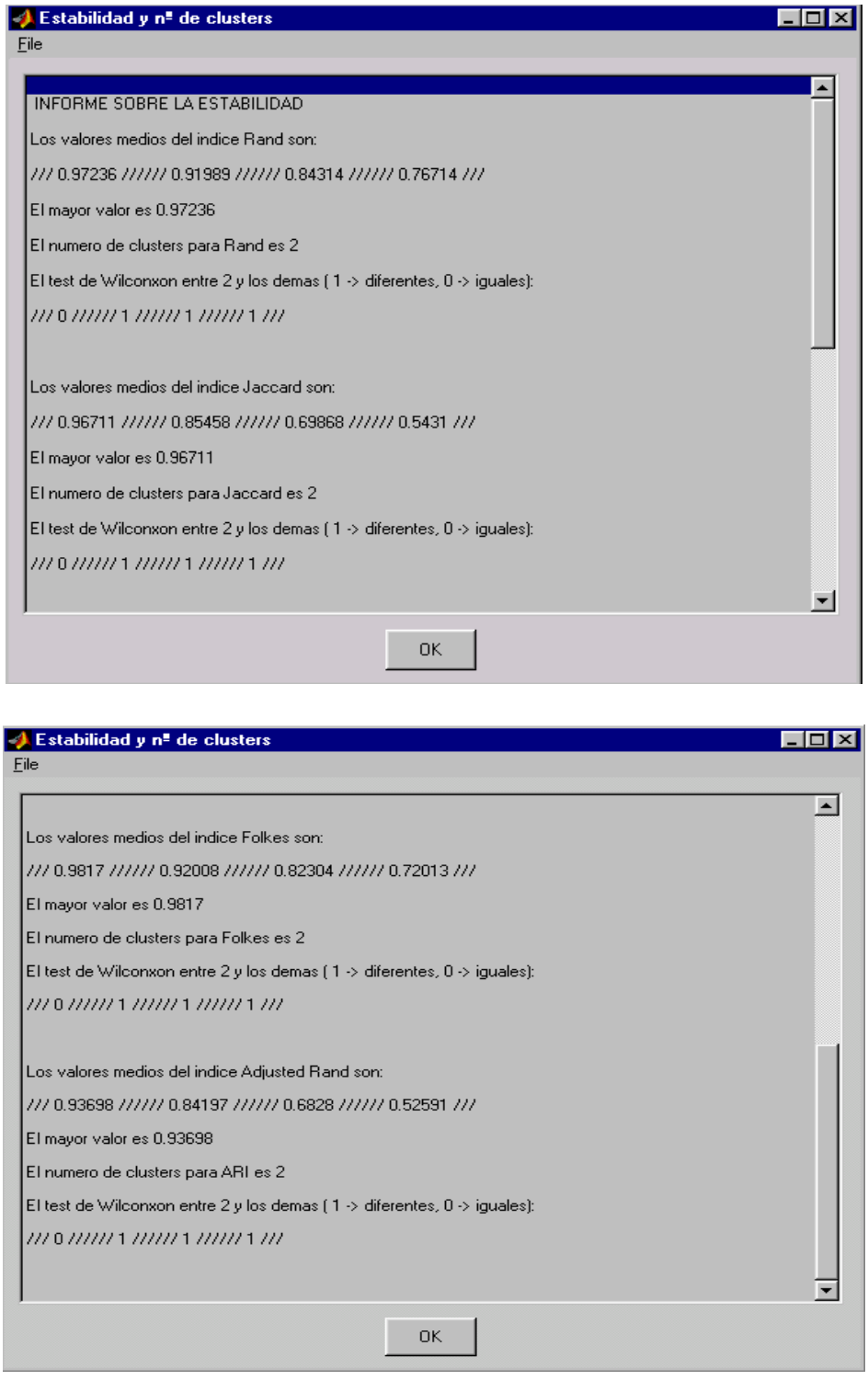


Figura 29: Ventana emergente, donde se muestra el análisis de estabilidad utilizando los índices: Rand, Adjusted Rand, Jaccard y Folkes.

Aparece la elección del número óptimo de clusters para cada índice y además el resultado de aplicar un test de Wilcoxon para mostrar que el valor medio del índice (para el óptimo número de clusters) es significativamente diferente al valor medio para los otros valores de K (numero de clusters). Un 0 indica que son iguales. Un 1 que son diferentes. Evidentemente, el test del óptimo consigo mismo dará siempre 0.

***Número 9:** Encontrar una solución consensuada, decidir de nuevo el óptimo número de clusters (si fuese necesario) y visualización de los resultados (contenido y gráficas de los clusters) (Figura 30).*

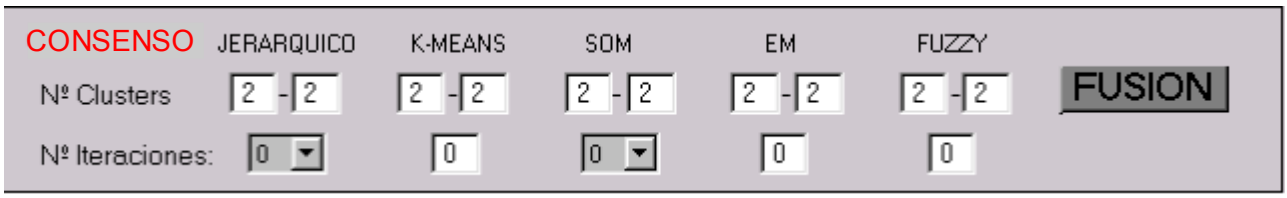


Figura 30: Sección Número 9 de la interfaz gráfica.

Esta sección incluye numerosas formas de realizar la fusión de los resultados clustering y la visualización y análisis de los resultados.

En la interfaz se muestra una tabla que relaciona los métodos clustering con el número de clusters y el número de iteraciones.

Si el número de iteraciones es 0 (como en la gráfica), se está indicando que el algoritmo correspondiente no participa. Si el valor es 1 o superior, el algoritmo sí que participa. Para K-means, EM y Fuzzy existe la posibilidad de incluir el número de iteraciones que se quiera ya que estos algoritmos se inician de forma aleatoria. El algoritmo EM toma su origen en el resultado de aplicar un algoritmo K-means. Por consiguiente, el número de iteraciones indica el número de resultados deseados de cada algoritmo clustering para luego fusionarlos.

Para que todos los algoritmos tengan el mismo peso en el proceso de fusión, se fusionarán los resultados para cada algoritmo y posteriormente los diferentes algoritmos entre si.

Respecto al número de clusters, debajo del nombre de cada algoritmo clustering aparecen 2 cuadros. Es para elegir un intervalo del número de clusters que se quiere estudiar durante la busqueda de la solución consensuada. Por ejemplo, en el estudio anterior, cada algoritmo de forma independiente (no se mostró los resultados para K-means y SOM) los índices indicaban mayoritariamente que el óptimo número de clusters es 2 aunque puede haber dudas con que hubiese 3 (ver índices Hubert , RS y los valores de S_Dbw). Por eso se podría incluir en el estudio de la fusión el intervalo entre 2 y 3. Si el valor en los cuadros es el mismo, indica que el estudio se focaliza en dicho número de clusters.

La fusión dependerá de la combinación entre el número de clusters y el número de iteraciones. Por ejemplo:

1) *Si el número de iteraciones es 0, para cualquier intervalo de número de clusters, no habrá fusión pues no se ha seleccionado ningún algoritmo clustering.*

2) *Si el número de iteraciones es 1 y en el intervalo hay un único valor (ejemplo [2,2]) entonces se activará el algoritmo correspondiente para dicho valor pero no habrá fusión pues no hay más que un resultado clustering. En este caso, tras pulsar el boton "Fusión" aparecerá una ventana con el contenido de los clusters y una visualización gráfica de los mismos (Figura 31).*

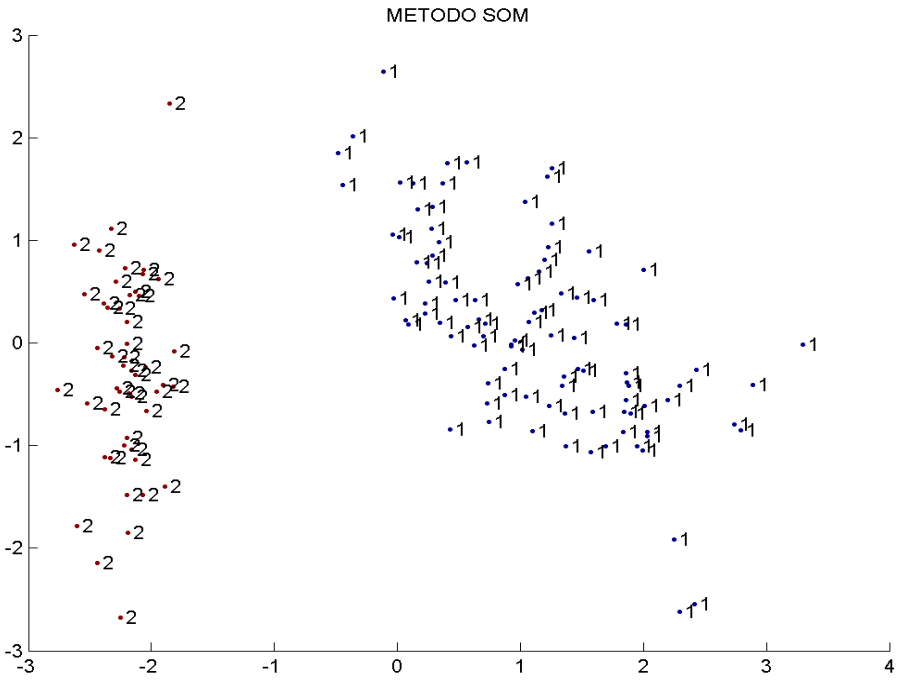


Figura 31: Representacion gráfica del resultado clustering después de aplicar SOM para K=2.

En el caso de la Figura 32, se ha elegido el intervalo [2,2] y el número de iteraciones igual a 1 debajo de SOM. Los demás algoritmos no fueron utilizados:

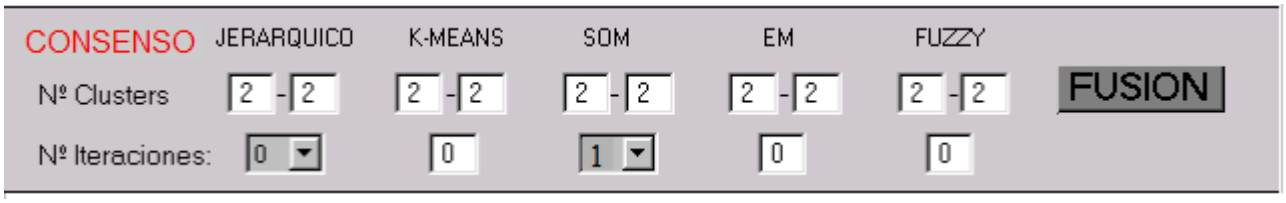


Figura 32: Elección particular de SOM en la sección número 9 de la interfaz gráfica (mismo K).

3) *Si se activa un algoritmo con un 1 en el número de iteraciones y el número de clusters varía (ejemplo:[2,3]) entonces se ejecutará el algoritmo para K=2, para K=3 y se fusionarán ambos resultados (Figura 33).*

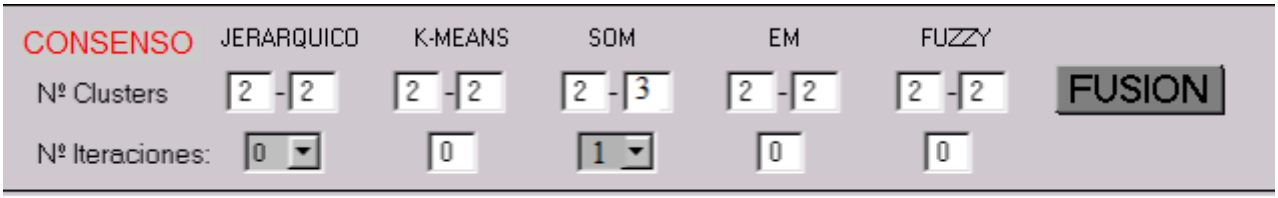


Figura 33: Elección particular de SOM en la sección número 9 de la interfaz gráfica (distinto K).

4) *Si se activa K-means, EM y Fuzzy con un valor del número de iteraciones superior a 1 (por ejemplo: 15) y el intervalo es para un sólo valor (ej: [2,2]) entonces se calcularán 15 resultados clustering para K=2 y posteriormente se fusionarán (Figura 34).*

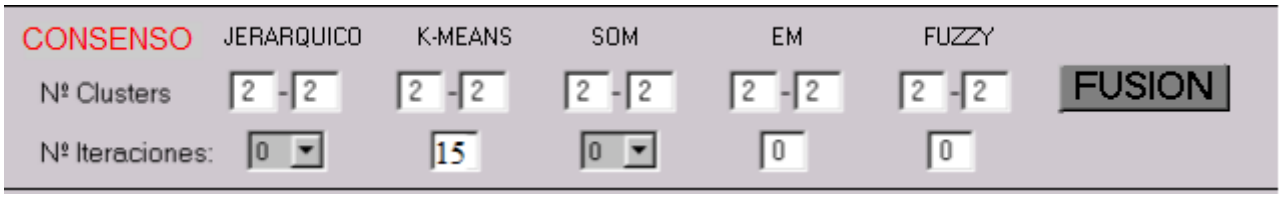


Figura 34: Elección particular de K-means en la sección número 9 de la interfaz (mismo K).

5) *Si se activa K-means, EM y Fuzzy con un valor del número de iteraciones superior a 1 (por ejemplo: 15) y el intervalo varía (ej: [2,3]) entonces se calcularán 15 resultados clustering pero de forma aleatoria se eligirá antes de cada resultado si es para K=2 ó K=3 (Figura 35).*

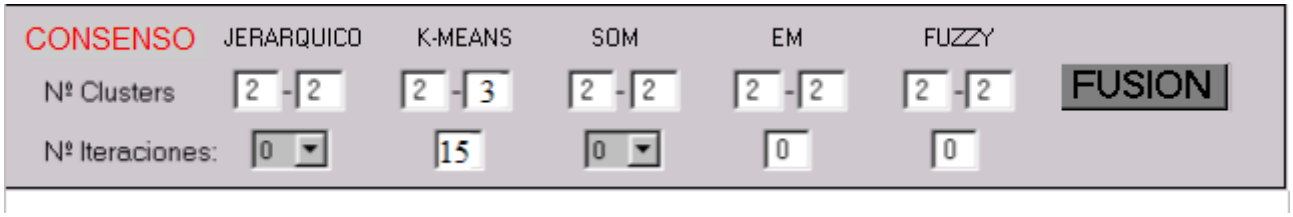


Figura 35: Elección particular de K-means en la sección número 9 de la interfaz (distinto K).

Salvo para el caso 1 y 2 que no hay realmente una fusión, tras pulsar el Boton "Fusión" aparecerá una ventana para el análisis de la solución consensuada. Si por ejemplo se considera una fusión de los 5 métodos con las siguientes características (Figura 36):

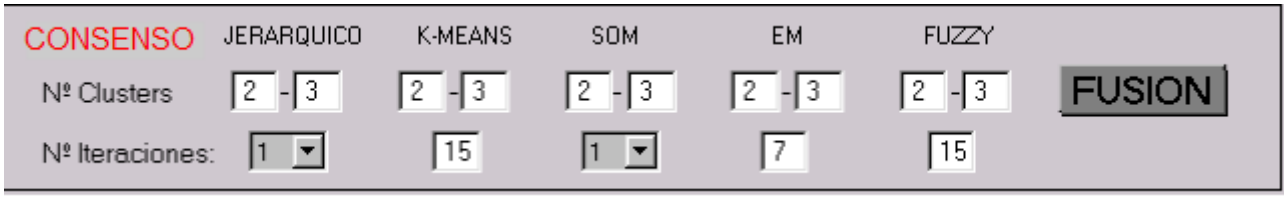


Figura 36: Elección particular de todos los métodos en la sección número 9 de la interfaz gráfica.

se obtendría una ventana con la siguiente configuración (Figura 37):

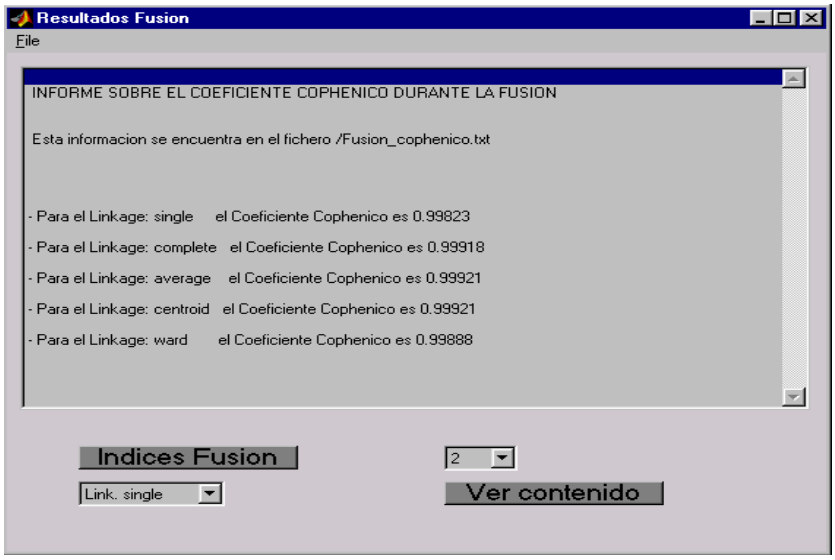


Figura 37: Ventana emergente tras pulsar el boton Fusión (Análisis del coeficiente cophenético).

En esta ventana se muestra como primera información el coeficiente cophenético del método jerárquico para las distintas combinaciones de la matriz de similitud obtenida tras la fusión y diversos métodos de linkage. Se aprecia que el linkage "average" puede ser elegido como uno de los mejores. Por consiguiente debajo del boton "índices Fusión" se eligirá el linkage "average".

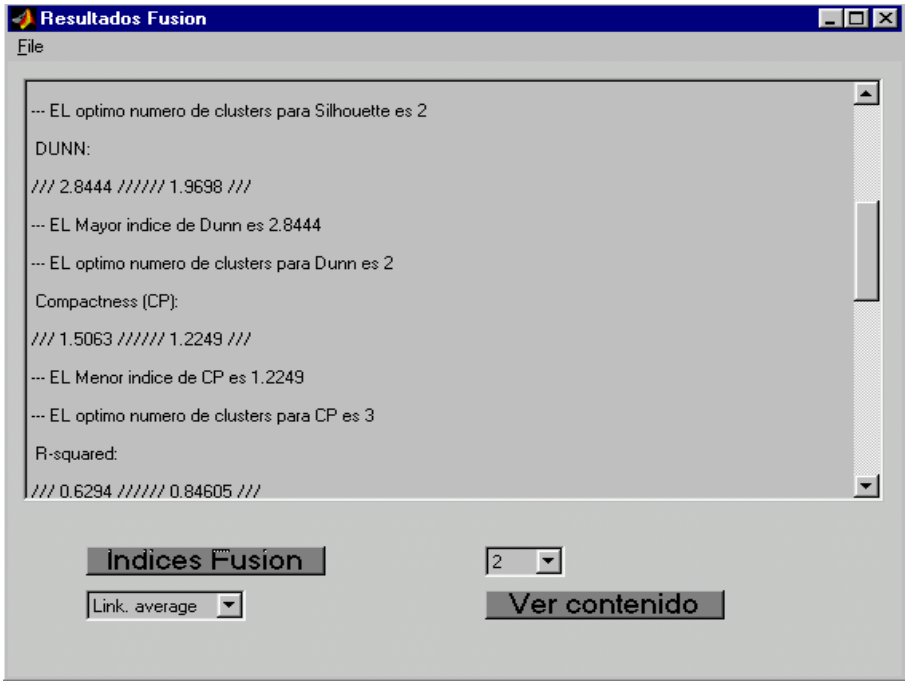


Figura 38: Ventana emergente tras pulsar el boton Índices Fusión (Análisis de validación relativo).

El siguiente paso es pulsar el boton "Índices Fusión". Aparecerá la misma ventana (Figura 38)

pero con distinto contenido: la variación del valor de algunos índices con el número de clusters (ejem: entre 2 y 3). El objetivo es decidir de nuevo cúal es el número óptimo de clusters (también se podría haber elegido un intervalo mayor pero se eligió esos valores a raíz de los resultados anteriores a la fusión).

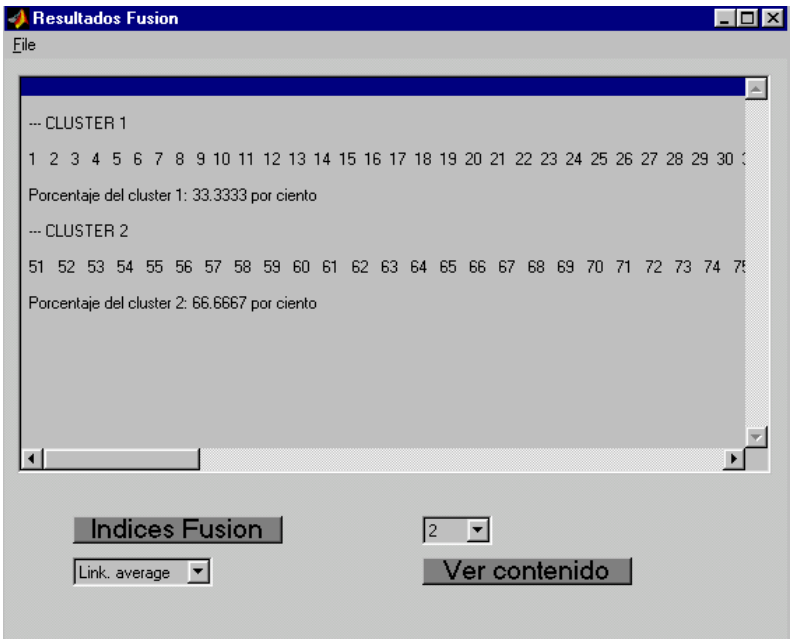


Figura 39: Ventana emergente tras pulsar sobre el boton "ver contenido".

Por último a la derecha de la ventana emergente aparece un "popup" en donde se puede elegir el número de clusters (el que se ha obtenido como óptimo y los otros del intervalo). Con ese valor y pulsando sobre "ver contenido" aparece un nuevo texto en la ventana indicando el contenido de cada cluster (Figura 39) y una gráfica mostrando los clusters con sus colores respectivos y todos los puntos numerados (similar a la mostrada anteriormente en la Figura 31).

7.3 Resumen

Se ha mostrado la utilización de la interfaz gráfica para evaluar un análisis clustering de una colección de puntos objeto. La validación se efectúa tanto para un algoritmo clustering de forma individual como tras la fusión de varios resultados clustering.

Los datos utilizados (Iris) han sido analizados. La validación se inclina por considerar que existen 2 clusters en los datos. Algún índice indica que hay 3. Pero es difícil diferenciar para los índices los dos clusteres que están juntos. Al ser un ejemplo simple, nos ha servido también para comprobar que la fusión de datos proporciona un contenido en los clusteres respectivos, igual o mejor que los resultados de cada algoritmo individualmente.

La interfaz gráfica analiza la validación utilizando el criterio interno y relativo. No utiliza el criterio externo ya que en la práctica no se suele conocer a priori la estructura en grupos de unos puntos objeto dados. Sin embargo se ha utilizado la comparación de particiones en el estudio de la

estabilidad.

Respecto a la elección del número de clusters óptimo, en la práctica, se puede considerar que cuando los valores de los índices presentan pequeñas diferencias, es mejor elegir la partición con el menor número de clusters.

8. CONCLUSIÓN

En este trabajo se han mostrado los pasos para validar un proceso clustering: estudio de la tendencia, criterio interno, determinación de los clusters con métodos de forma individual, criterio relativo, estabilidad y determinación de los clusters mediante una solución consensuada.

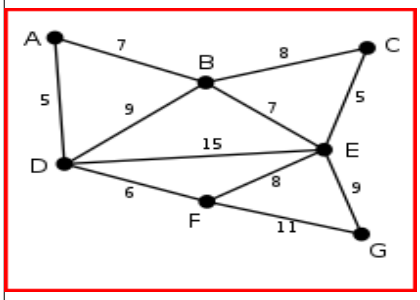
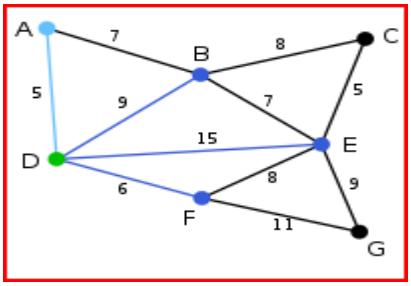
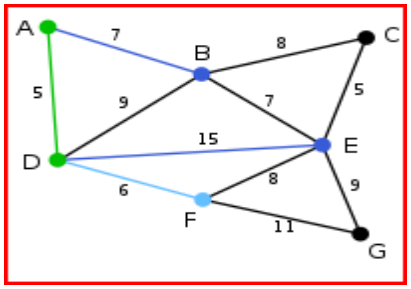
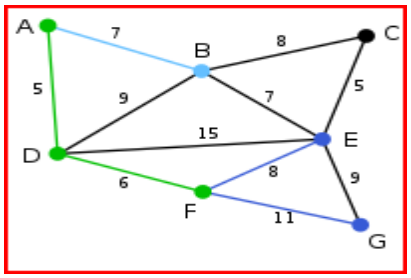
Se han seleccionado los métodos y algoritmos más representativos. Sin embargo, hay que remarcar que la validación de resultados clustering es un campo de investigación activo y por lo tanto, nuevos métodos de validación aparecerán en el futuro. Quizás, más que encontrar el mejor método de validación de un resultado clustering para cualquier conjunto de puntos objeto, el proceso de validación se adaptará a los puntos objeto a analizar.

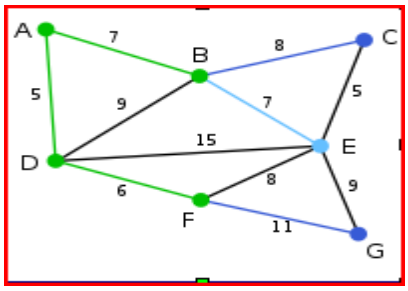
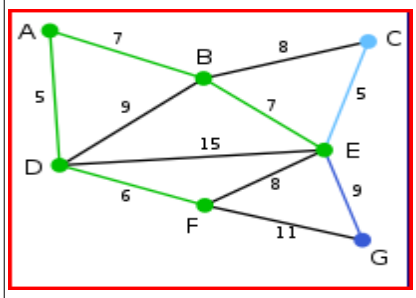
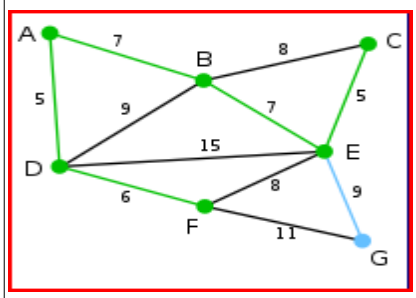
Se han integrado todos los algoritmos en una interfaz gráfica construida en Matlab. Se muestra en ventanas los análisis de validación respectivos (criterio interno, relativo y estabilidad) y los gráficos correspondientes. Destaca la flexibilidad para realizar la fusión de los resultados clustering.

No se han implementado algoritmos para detectar los llamados "outliers" o puntos objeto atípicos. Es decir, puntos objeto con un comportamiento muy diferente al resto del grupo. Como trabajo futuro, éste sería un punto importante a considerar ya que si en el conjunto de puntos objeto hay outliers los resultados pueden ser erróneos o menos precisos.

ANEXO 1

Ejemplo de ejecución del algoritmo de PRIM (http://es.wikipedia.org/wiki/Algoritmo_de_Prim).

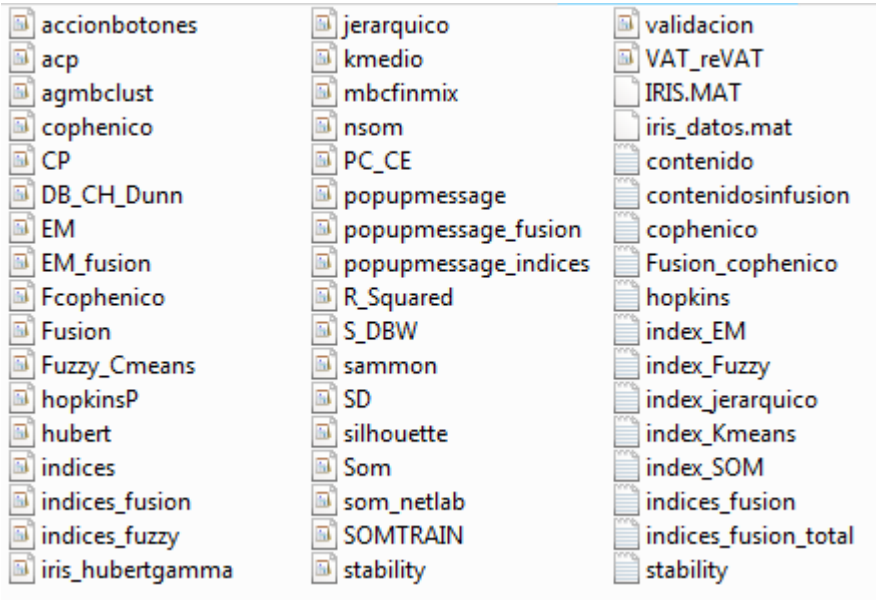
Imagen	*Descripción*	*Árbol*
	*Este es un grafo ponderado de partida. No es un árbol ya que requiere que no haya ciclos y en este grafo los hay. Los números cerca de las aristas indican el peso. Ninguna de las aristas está marcada, y el vértice* ***D*** *ha sido elegido arbitrariamente como el punto de partida.*	*D*
	*El segundo vértice es el más cercano a* ***D****:* ***A*** *está a 5 de distancia,* ***B*** *a 9,* ***E*** *a 15 y* ***F*** *a 6. De éstos, 5 es el valor más pequeño, así que marcamos la arista* ***DA****.*	*A, D*
	*El próximo vértice a elegir es el más cercano a* ***D*** *o* ***A****.* ***B*** *está a 9 de distancia de* ***D*** *y a 7 de* ***A****,* ***E*** *está a 15, y* ***F*** *está a 6. 6 es el valor más pequeño, así que marcamos el vértice* ***F*** *y a la arista* ***DF****.*	*A, D, F*
	*El algoritmo continua. El vértice* ***B****, que está a una distancia de 7 de* ***A****, es el siguiente marcado. En este punto la arista* ***DB*** *es marcada en rojo porque sus dos extremos ya están en el árbol y por lo tanto no podrá ser utilizado.*	*A, D, F, B*

	*Aquí hay que elegir entre **C**, **E** y **G**. **C** está a 8 de distancia de **B**, **E** está a 7 de distancia de **B**, y **G** está a 11 de distancia de **F**. **E** está más cerca, entonces marcamos el vértice **E** y la arista **EB**. Otras dos aristas fueron marcadas en rojo porque ambos vértices que unen fueron agregados al árbol.*	*A, D, F, B, E*
	*Sólo quedan disponibles **C** y **G**. **C** está a 5 de distancia de **E**, y **G** a 9 de distancia de **E**. Se elige **C**, y se marca con el arco **EC**. El arco **BC** también se marca con rojo.*	*A, D, F, B, E, C*
	***G** es el único vértice pendiente, y está más cerca de **E** que de **F**, así que se agrega **EG** al árbol. Todos los vértices están ya marcados, el árbol de expansión mínimo se muestra en verde. En este caso con un peso de 39.*	*A, D, F, B, E, C, G*

ANEXO 2

Para lanzar el programa hay que colocar sobre el mismo directorio todos los programas desarrollados así como el fichero .mat: iris_datos.mat. A continuación escribir ***validación*** *en la ventana principal para lanzar la interfaz gráfica.*

Los programas son:



Aparte de esos programas se ha añadido el programa iris_hubertgamma.m para valorar el criterio interno de un algoritmo no jerárquico. Para su uso hay que cambiar los labels por los obtenidos utilizando un algoritmo clustering.