Clustering difuso con pertenencias intervalares

Aranzazu Jurio
Departamento de Estadística,
*I*nformática y Matemáticas
*I*nstituto de Smart Cities
Universidad Pública de Navarra
Pamplona, España
aranzazu.jurio@unavarra.es

Humberto Bustince
Departamento de Estadística,
*I*nformática y Matemáticas
*I*nstituto de Smart Cities
Universidad Pública de Navarra
Pamplona, España
bustince@unavarra.es

Vicenç Torra
School of Informatics
University of Skövde
Skövde, Sweden
vtorra@ieee.org

***Resumen*—En este trabajo estudiamos cómo solucionar el problema que presentan los outliers a la hora de realizar un proceso de agrupamiento. Para ello, presentamos una función objetivo que extiende a la del algoritmo Fuzzy Cluster Means, mediante el uso de conjuntos intervalo-valorados difusos. En este caso, las pertenencias de cada dato a cada grupo se representan mediante intervalos. Posteriormente, presentamos un algoritmo que minimiza la función objetivo propuesta y mostramos cómo se comporta ante diferentes conjuntos de datos.**

***Index Terms*—clustering, intervalo, pertenencia**

I. Introducción

El problema de clustering o agrupamiento es un problema de clasificación no supervisada cuyo objetivo es encontrar los grupos naturales existentes en un conjunto de datos. Para ello, se basa en la idea de que los datos que pertenecen a un mismo grupo deben compartir características similares, mientras que los datos que pertenecen a diferentes grupos deben diferenciarse en dichas características [4].

Los algoritmos de agrupamiento se pueden dividir de manera general en dos tipos: algoritmos jerárquicos y algoritmos de particiones. Los algoritmos jerárquicos crean un árbol (dendograma) que mide las similitudes entre los datos [5] [9]. Por su parte, los algoritmos de particiones separan los datos en un número prefijado de grupos. Cada uno de esos grupos se representa mediante un centroide, que es el punto cuya suma de distancias desde todos los datos del grupo a sí mismo es mínima [6] [7] [8]. En este trabajo nos centramos en los algoritmos de particiones.

Dentro de los algoritmos de particiones uno de los más conocidos y utilizados es el k-means [3] [8]. En este algoritmo se separan todos los datos en c grupos y se calculan los centroides de cada grupo. El objetivo es minimizar la suma de las distancias de cada dato a su centroide correspondiente.

$$J=\sum_{i=1}^{c}\sum_{x_k\in cluster_i}||x_k-v_i||_A^2$$

Este trabajo ha sido parcialmente financiado por el Ministerio de Educación, Cultura y Deporte mediante el programa José Castillejo para estancias de movilidad en el extranjero de jóvenes doctores. También ha sido parcialmente financiado por el Ministerio de Economía, Industria y Competitividad del Gobierno de España mediante el proyecto TIN2016-77356-P (AEI/FEDER, UE).

donde $||x||_A=\sqrt{x^tAx}$ es cualquier norma asociada a un producto escalar.

Uno de los problemas que presenta el algoritmo k-means, así como muchos algoritmos de agrupamiento de particiones, es que no son capaces de manejar situaciones en las que los grupos de datos se encuentran solapados. En esos casos, los datos que se encuentran en la zona solapada entre dos o más grupos deberían pertenecer a todos esos grupos y no solo a uno de ellos.

Una de las maneras de solucionar este problema es utilizando la lógica difusa [10]. De este modo, cada uno de los datos puede pertenecer a más de un grupo, con diferente valor de pertenencia. Estos valores de pertenencia son números entre 0 y 1. El algoritmo Fuzzy Cluster Means (FCM) [1] extiende la idea del algoritmo k-means empleando la lógica difusa.

En el FCM, cada dato tiene una pertenencia total de 1, que reparte entre todos los grupos. De esta manera, un mismo dato puede pertenecer a todos los grupos existentes, siempre y cuando la suma de sus valores de pertenencia sea 1. El objetivo es minimizar la suma ponderada de las distancias de cada dato a todos los centroides. Los pesos de esta suma vienen dados por el valor de pertenencia de cada dato a cada grupo.

$$J=\sum_{k=1}^{n}\sum_{i=1}^{c}(u_{ik})^m||x_k-v_i||_A^2$$

Este algoritmo consigue solucionar el problema del solapamiento entre grupos. Sin embargo, cuando existen outliers entre los datos a clasificar, el FCM no es capaz de detectarlos y por ello sus resultados se ven distorsionados.

Para solucionar este problema, en este trabajo presentamos un nuevo algoritmo de agrupamiento que extiende el FCM, de tal forma que es capaz de detectar los datos que no pertenecen a los grupos naturales existentes en el conjunto de datos.

De la misma forma que el uso de los conjuntos difusos permite aportar nueva información al proceso de agrupamiento, en nuestra propuesta utilizamos una extensión de los conjuntos difusos: los conjuntos intervalo-valorados difusos. En este trabajo, utilizamos dichos conjuntos para cuantificar las pertenencias, por lo que cada dato va a pertenecer a todos los grupos con un valor de pertenencia que es un intervalo en [0,1]. Utilizamos la amplitud de dicho intervalo para representar la seguridad que tenemos de que ese dato pertenezca a los grupos presentes en el dataset que estamos clasificando. De esta forma, si estamos totalmente seguros de que un dato pertenece a uno o varios de los grupos existentes, las pertenencias de dichos datos a todos los grupos tendrán amplitud 0. Por el contrario, si estamos totalmente seguros de que un dato no pertenece a ninguno de los grupos presentes, las pertenencias de ese dato a todos los grupos tendrán amplitud 1, que es el máximo permitido.

El resto de este trabajo está organizado de la siguiente forma: en la Sección II repasamos brevemente el algoritmo Fuzzy Cluster Means; en la Sección III explicamos en detalle el algoritmo que proponemos en este trabajo y en la Sección IV vemos cómo se comporta mediante el uso de varios ejemplos de datos. Finalmente, en la Sección V mostramos las conclusiones obtenidas.

II. FCM

El Fuzzy Cluster Means (FCM) [1] es uno de los algoritmos de agrupamiento difuso más utilizados. Al utilizar conjuntos difusos, permite que cada uno de los datos pertenezca a más de un grupo al mismo tiempo. De hecho, basa su idea en que cada dato debe pertenecer a todos los grupos presentes con un grado de pertenencia dado. Estos grados de pertenencia son valores entre 0 y 1, de tal forma que la suma de los valores de pertenencia de cada dato a todos los grupos siempre sea 1.

Con esta premisa, el algoritmo FCM trata de minimizar la suma de distancias ponderadas de cada dato a todos los centroides, haciendo que los pesos de ponderación sean proporcionales al valor de pertenencia del dato a ese grupo.

$$J = \sum_{k=1}^{n} \sum_{i=1}^{c} (u_{ik})^m ||x_k - v_i||_A^2$$

donde x_k es el k-ésimo dato a clasificar, v_i es el centroide del grupo i, u_{ik} es el grado de pertenencia del dato k al grupo i y m es un valor real mayor que 1. Además, se deben cumplir tres restricciones:

- $u_{ik} \geq 0, \quad k = 1..n,\ i = 1..c$
- $\sum_{k=1}^{n} u_{ik} > 0, \quad i = 1..c$
- $\sum i = 1^c u_{ik} = 1, \quad k = 1..n$

La solución a este problema es un proceso iterativo que comienza con unos centroides elegidos aleatoriamente. A partir de los centroides se pueden calcular los nuevos valores de pertenencia de cada dato a cada grupo:

$$u_{ik} = \left(\sum_{j=1}^{c} \left(\frac{||x_k - v_i||_A}{||x_k - v_j||_A} \right)^{2/(m-1)} \right)^{-1}$$

$k = 1..n$, $i = 1..c$. Y a partir de los valores de pertenencia se pueden calcular los nuevos centroides:

$$v_i = \frac{\sum_{k=1}^{n} (u_{ik})^m x_k}{\sum_{k=1}^{n} (u_{ik})^m}$$

$i = 1..c$. El proceso termina cuando los cambios en los valores son suficientemente pequeños.

Uno de los problemas que presenta el algoritmo FCM se produce cuando el conjunto de datos a clasificar presenta ruido, o outliers. En estos casos, todos los datos se asignan a los diferentes grupos, por lo que los datos alejados pueden modificar erróneamente sus centroides. En la Figura 1 podemos ver un conjunto de datos marcados con asteriscos negros. Claramente podemos ver dos grupos de datos que se solapan y un dato que no pertenece a ninguno de los grupos. Marcados con círculos rojos se ven los centroides de los dos grupos detectados por el algoritmo FCM.

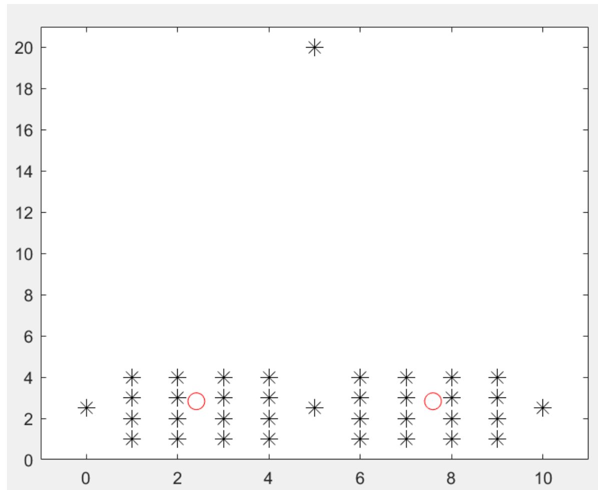

Figura 1. Ejecución del algoritmo FCM sobre un conjunto de datos con un outlier. Conjunto de datos a clasificar marcados con asteriscos negros. Centroides obtenidos por el FCM marcados con círculos rojos.

Los dos grupos de datos están centrados verticalmente en el punto 2.5. Sin embargo, como el dato situado en el punto $(5, 20)$ pertenece a ambos grupos, los centroides se desplazan hacia arriba hasta el valor vertical 2.83.

Además, si analizamos los valores de pertenencia de cada dato a los dos grupos, vemos que el dato situado en el punto (5, 2.5) tiene un valor de pertenencia de 0.5 a cada uno de los grupos. Por su parte, el dato situado en el punto (5, 20) también tiene un valor de pertenencia de 0.5 a cada grupo. Por tanto, el algoritmo nos indica que ambos datos son iguales a la hora de formar los dos grupos. Sin embargo, observando la figura, vemos que uno de los dos datos pertenece a la zona solapada de los dos grupos mientras que el otro dato no pertenece a ninguno de los grupos.

A partir de este ejemplo, podemos afirmar que el algoritmo Fuzzy Cluster Means no funciona adecuadamente cuando en el conjunto de datos a clasificar hay datos que no pertenecen a ninguno de los grupos existentes.

III. Nuevo algoritmo de clustering intervalar

En esta sección explicamos nuestra propuesta de algoritmo de agrupamiento. Su principal novedad es que hace uso de los conjuntos intervalo-valorados difusos para representar las pertenencias de los datos a cada cluster.

Llamamos $L([0,1])$ al conjunto de todos los subintervalos cerrados en $[0,1]$, es decir,

$$L([0,1]) = \{\mathbf{x} = [\underline{x}, \overline{x}] | (\underline{x}, \overline{x}) \in [0,1]^2 \ \text{ y } \ \underline{x} \leq \overline{x}\}$$

Un conjunto intervalo-valorado difuso Z en el universo $U \neq \emptyset$ es una función $Z : U \rightarrow L([0,1])$.

Una de las interpretaciones existentes de los conjuntos intervalo-valorado difusos, la cual utilizamos en este trabajo, es la siguiente: "el grado de pertenencia de un elemento al conjunto es un valor dentro del intervalo de pertenencia considerado. No conocemos exactamente el valor, por lo que proporcionamos sus extremos" [2].

Siguiendo esta idea, podemos asumir que la amplitud del intervalo representa la ignorancia que tenemos a la hora de asignar el valor de pertenencia del elemento al conjunto.

Aplicándolo sobre nuestro problema de agrupamiento, queremos que si el algoritmo está completamente seguro de que un dato pertenece a un grupo, entonces la amplitud de su intervalo de pertenencia será mínima. No importa si la pertenencia es $[1,1]$ a un grupo y $[0,0]$ a los demás, o si pertenece $[0{,}5, 0{,}5]$ a dos grupos. Por el contrario, si el algoritmo no está seguro de que un dato pertenezca a los grupos que se han creado en los datos, entonces la amplitud de sus intervalos de pertenencia deberá ser mayor. En el caso extremo, si un dato parece no pertenecer a ninguno de los grupos existentes, los intervalos de pertenencia a todos los grupos pueden ser $[0,1]$.

Por tanto, a diferencia del algoritmo Fuzzy Cluster Means, en nuestra propuesta la suma de los extremos inferiores y superiores de las pertenencias de un dato a todos los grupos tiene que ser un valor entre 2 y c, siendo c el número de grupos. En el caso en el que no existe ninguna duda sobre la pertenencia de los datos a los grupos, los extremos inferiores de todos los valores de pertenencia son iguales a los extremos superiores. Manteniendo la misma restricción que existía en el FCM, estos deben sumar 1, por lo que la suma total es 2. En el caso de que la duda sobre la pertenencia sea máxima, los intervalos de pertenencia a todos los grupos serán $[0,1]$, por lo que la suma total será igual al número de grupos.

De forma análoga al k-means y al FCM, con este nuevo algoritmo queremos minimizar la suma ponderada de distancias entre cada dato y los centroides de cada grupo, utilizando como pesos los valores de pertenencia. En este caso, esos valores de pertenencia son intervalos. Cuando no existe duda sobre el valor de pertenencia, el extremo inferior y superior del intervalo son muy parecidos y representan el valor que debe tomar el peso. Por el contrario, cuando existe una gran duda de que un dato pertenezca a un grupo, no queremos que su información modifique en gran medida el valor del centroide, por lo que queremos que su peso sea pequeño. Al ser la amplitud del intervalo grande, eso significa que su extremo inferior tiene que ser pequeño. Por tanto, en ambos casos podemos utilizar un peso para la suma ponderada proporcional al extremo inferior del intervalo de pertenencia.

También es necesario restringir la suma total de las amplitudes de los intervalos de pertenencia. De no hacerlo, nuestro sistema se minimizaría al decir que tenemos una duda máxima sobre la pertenencia de todos los datos existentes.

Por tanto, la función objetivo que queremos minimizar en nuestra propuesta es la siguiente:

$$J = \frac{1}{a} \sum_{k=1}^{n} \sum_{i=1}^{c} (\underline{u_{ik}})^m ||x_k - v_i||_A^2 + \sum_{k=1}^{n} \sum_{i=1}^{c} (\overline{u_{ik}} - \underline{u_{ik}})^m$$

donde x_k es el k-ésimo dato a clasificar, v_i es el centroide del grupo i, $[\underline{u_{ik}}, \overline{u_{ik}}]$ es el intervalo de pertenencia del dato k al grupo i y m es un valor real mayor que 1.

El parámetro $1/a$ permite ajustar la importancia relativa de los dos términos de la ecuación. Hay que tener en cuenta que ambos términos no tienen por que estar en la misma escala: el primer término depende de las distancias entre los datos y el segundo es siempre un valor entre 0 y 1. Mediante este parámetro podemos conseguir que la solución obtenida se parezca más a la obtenida por el FCM si hacemos que el segundo término tenga mucha importancia, o que la solución presente en general amplitudes muy grandes, si es el primer término el más importante.

Esta función está sujeta a las siguientes restricciones:

- Los intervalos tienen que estar bien formados. $\overline{u_{ik}} \geq \underline{u_{ik}}$, $k = 1..n$, $i = 1..c$
- Todos los grupos tienen que tener por lo menos un dato con extremo inferior de pertenencia positivo. $\sum_{k=1}^{n} \underline{u_{ik}} > 0, \quad i = 1..c$
- La suma de los extremos de las pertenencias de un dato a todos los grupos tiene que estar entre 2 y c. $2 \geq \sum_{i=1}^{c} (\underline{u_{ik}} + \overline{u_{ik}}) <= c, \quad k = 1..n$

Cuando el número de grupos es 2, esta función se puede minimizar utilizando multiplicadores de Lagrange. De esta forma, obtenemos un algoritmo iterativo análogo al FCM. A partir de una inicialización aleatoria, podemos actualizar los intervalos de pertenencia basándonos en los datos de los centroides.

$$\underline{u_{ik}} = \frac{2(2a)^{1/m-1}}{||x_k - v_i||^{2/m-1} \left[c + 2(2a)^{1/m-1} \sum_{j=1}^{c} \frac{1}{||x_k - v_j||^{2/m-1}} \right]}$$

$$\overline{u_{ik}} = \frac{2 \left[||x_k - v_i||^{2/m-1} + (2a)^{1/m-1} \right]}{||x_k - v_i||^{2/m-1} \left[c + 2(2a)^{1/m-1} \sum_{j=1}^{c} \frac{1}{||x_k - v_j||^{2/m-1}} \right]}$$

para $k = 1..n$, $i = 1..c$. A partir de los datos de los intervalos de pertenencia, podemos actualizar los centroides.

$$v_i = \frac{\sum_{k=1}^{n} (\underline{u_{ik}})^m x_k}{\sum_{k=1}^{n} (\underline{u_{ik}})^m}$$

para $i = 1..c$. El proceso termina cuando los cambios en los valores son suficientemente pequeños.

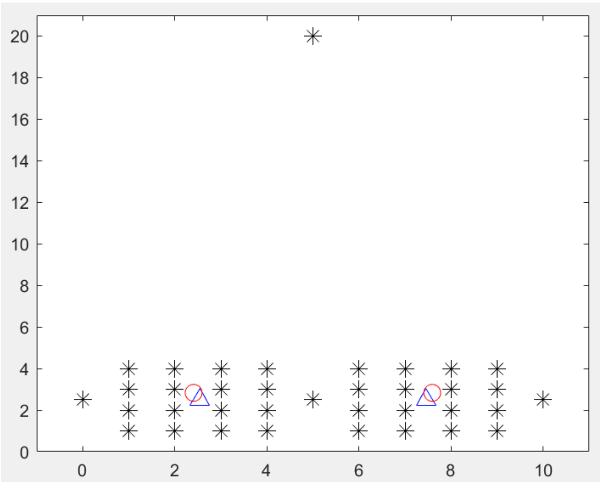

Figura 2. Ejecución del algoritmo propuesto y del FCM sobre un conjunto de datos con un outlier. Conjunto de datos a clasificar marcados con asteriscos negros. Centroides obtenidos por nuestro algoritmo marcados con triángulos azules. Centroides obtenidos por el FCM marcados con círculos rojos.

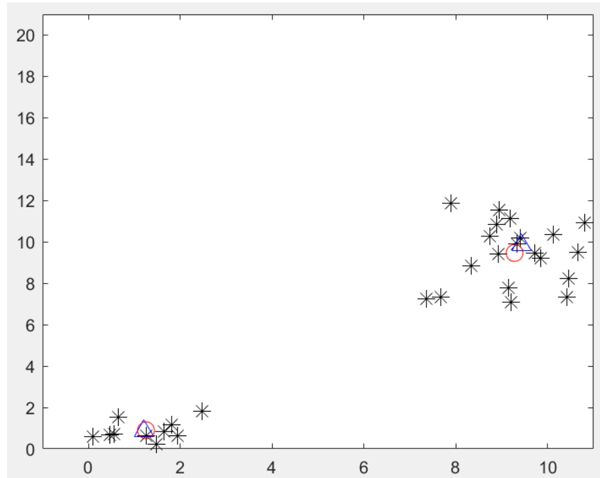

Figura 3. Ejecución del algoritmo propuesto y del FCM sobre un conjunto de datos sin outliers. Conjunto de datos a clasificar marcados con asteriscos negros. Centroides obtenidos por nuestro algoritmo marcados con triángulos azules. Centroides obtenidos por el FCM marcados con círculos rojos.

IV. Ejemplos numéricos

En esta sección mostramos el comportamiento del algoritmo que hemos propuesto. Para poder visualizar los ejemplos con mayor simplicidad y que sean más fáciles de entender, en todos ellos utilizamos sólo dos dimensiones.

Comenzamos con el mismo ejemplo que hemos visto en la Figura 1. Como ya hemos comentado, el FCM no es capaz de resolver este ejemplo correctamente. Al aplicar nuestro algoritmo, los centroides obtenidos ya no se desplazan hacia arriba de donde deberían estar y se colocan en el centro real de los grupos existentes. En la Figura 2 hemos marcado con triángulos azules los centros obtenidos por nuestro algoritmo y con círculos rojos aquellos obtenidos por el FCM.

Si analizamos los intervalos de pertenencia resultantes de este ejemplo, el dato situado en la intersección de los dos grupos (punto (5, 2.5)) tiene pertenencia [0.3488, 0.6512] a cada uno de los grupos, con una amplitud de 0.3024. Por su parte, el dato outlier (punto 5, 20)) tiene pertenencia [0.0004, 0.9996] a cada uno de los grupos, con una amplitud de 0.9991. Podemos ver claramente como el algoritmo es capaz de identificar que estos dos datos no son iguales a la hora de hacer la clasificación. Al tener una amplitud tan grande el outlier, esto hace que casi no se tenga en cuenta a la hora de calcular los centroides, y por eso estos se sitúan en el centro geométrico del grupo correspondiente.

En la Figura 3 podemos ver un nuevo conjunto de datos y cómo se comportan sobre él nuestro algoritmo (centroides marcados con triángulos azules) y el FCM (centroides marcados con círculos rojos). Se puede comprobar que ambos algoritmos obtienen una solución muy similar.

Si a ese mismo conjunto de datos le añadimos tres outliers, entonces vemos que los dos algoritmos ya cambian su comportamiento. Esta nueva ejecución se puede ver en la Figura 4, siguiendo la misma leyenda.

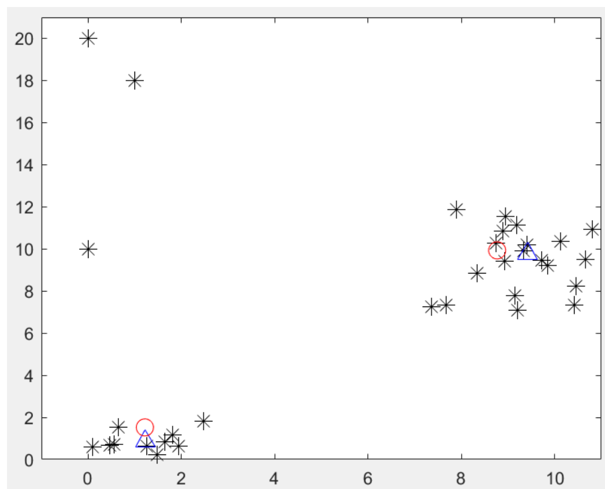

Figura 4. Conjunto de datos igual al de la figura 3 al que se le han añadido tres outliers en los puntos (0,20), (1,18) y (0,10). Ejecución del algoritmo propuesto y del FCM. Conjunto de datos a clasificar marcados con asteriscos negros. Centroides obtenidos por nuestro algoritmo marcados con triángulos azules. Centroides obtenidos por el FCM marcados con círculos rojos.

Como podemos comprobar visualmente, la adición de tres outliers casi no ha modificado los valores de los centroides obtenidos por nuestro algoritmo. Analíticamente, estos valores han pasado de ser

- Grupo1$\rightarrow$ (1.2107, 0.8496)
- Grupo2$\rightarrow$ (9.4268, 9.8496)

a valer

- Grupo1$\rightarrow$ (1.2223, 0.8670)
- Grupo2$\rightarrow$ (9.4149, 9.7710)

Sin embargo, en la ejecución del FCM los nuevos centroides se ven claramente influenciados por los outliers, y se desplazan hacia allí. El centroide del grupo 1 se desplaza hacia arriba y el centroide del grupo 2 lo hace hacia la izquierda. De esta forma, pasan de valer

- Grupo1→ (1.2531, 0.8985)
- Grupo2→ (9.2796, 9.4647)

a valer

- Grupo1→ (1.2181, 1.5217)
- Grupo2→ (8.7661, 9.9312)

Uno de los aspectos más importantes a la hora de aplicar nuestro algoritmo de agrupamiento es el ajuste del parámetro $1/a$. Este parámetro permite ajustar la amplitud de los intervalos de pertenencia. Si el parámetro $1/a$ toma valores muy grandes, entonces el algoritmo obtiene soluciones donde las amplitudes de todas las pertenencias son grandes. Por el contrario, si el parámetro $1/a$ toma valores pequeños, entonces el algoritmo encuentra soluciones donde las amplitudes de los intervalos de pertenencia son pequeñas. Para cada uno de los problemas es necesario ajustar este parámetro, para obtener las soluciones deseadas.

En la Figura 5 vemos un nuevo conjunto de datos marcado con asteriscos negros y tres pares de puntos que representan los centros obtenidos con tres valores del parámetro $1/a$ distintos. Mediante cuadrados rojos los centros obtenidos cuando $1/a = 1/0{,}0001$, triángulos azules cuando $1/a = 1/10000$ y círculos verdes cuando $1/a = 1/1$.

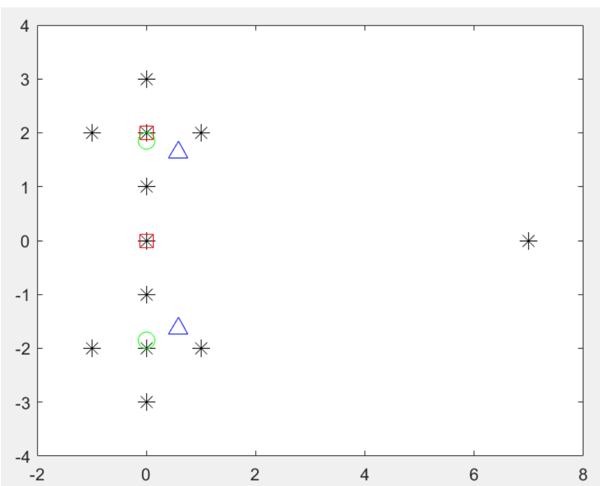

Figura 5. Conjunto de datos marcado mediante asteriscos negros. Centros obtenidos mediante el algoritmo propuesto cuando $1/a = 1/0{,}0001$ (cuadrados rojos), $1/a = 1/10000$ (triángulos azules) y $1/a = 1/1$ (círculos verdes).

Como se puede observar, los resultados varían en gran medida dependiendo del valor del parámetro. En el Cuadro I se muestran los valores de los intervalos de pertenencia de cada dato a cada grupo. Cuando $1/a = 1/0{,}0001$, el algoritmo encuentra dos centroides aleatorios, cuyas pertenencias son $[0, 0]$ y $[1, 1]$ a los dos grupos. El resto de datos tienen una pertenencia de $[0, 1]$ a los dos grupos, es decir, la máxima amplitud posible. En este ejemplo de ejecución, los datos elegidos para convertirse en centroides son el $(0, 2)$ y el $(0, 0)$. Por el contrario, cuando $1/a = 1/10000$, todos los intervalos de pertenencia tienen amplitud 0 y, por ello, pueden ser considerados como valores puntuales. En este caso, el resultado obtenido es muy parecido al obtenido por el FCM.

Por último, el caso en el que $1/a = 1/1$ muestra un ejemplo de buen balance entre los dos extremos. En este caso, los datos que están muy cercanos a algún centroide y que, por lo tanto, el algoritmo está seguro de su pertenencia a los grupos existentes, tienen intervalos de pertenencia con amplitudes pequeñas. El dato situado en el punto $(0, 0)$, justo en la intersección de los dos grupos, tiene una amplitud mayor que el resto de datos, puesto que está más alejado de los centroides, pero mucho menor que el dato $(7, 0)$, que no debería pertenecer a ninguno de los dos grupos. Este dato tiene una pertenencia de $[0{,}0015, 0{,}9985]$ a ambos grupos, casi la máxima posible.

Por tanto, es muy importante ajustar el parámetro $1/a$, que balancea la importancia relativa de las distancias entre los datos y los centroides con las amplitudes de las pertenencias. Según nuestras pruebas hechas sobre varios conjuntos de datos, un valor que funciona en la mayoría de los casos es cuando a toma el valor del percentil 10 o 15 de las distancias entre los datos del conjunto de datos.

V. Conclusiones

En este trabajo hemos presentado una extensión del algoritmo Fuzzy Cluster Means. Mediante nuestra propuesta, utilizamos una extensión de los conjuntos difusos, los conjuntos intervalo-valorados difusos, para permitir que el algoritmo detecte los datos considerados outliers y que estos no interfieran en el proceso de agrupamiento del resto de datos. Mediante varios ejemplos ilustrativos hemos comprobado que el algoritmo se comporta de manera correcta, tanto cuando existen outliers como cuando no. Además, hemos comprobado que es muy importante la elección de un buen valor para el parámetro que pondera los dos términos de la función objetivo, ya que este parámetro permite una variedad total entre soluciones con la máxima y la mínima amplitud en las pertenencias de todos los datos a todos los grupos.

Referencias

[1] J.C. Bezdek, “Pattern Recognition with Fuzzy Objective Function Algorithms” Plenum Press, 1981.

[2] H. Bustince, E. Barrenechea, M. Pagola, J. Fernandez, Z. Xu, B. Bedregal, J. Montero, H. Hagras, F. Herrera, B. De Baets, “A historical account of types of fuzzy sets and their relationships” IEEE Transactions on Fuzzy Systems 24, 179-194, 2016.

[3] E.W. Forgy, “Cluster analysis of multivariate data: efficiency versus interpretability of classifications”, Biometrics 21, 768–769, 1965.

[4] A.K. Jain, M.N. Murty, P.J. Flynn, “Data clustering: A review” ACM Computing Surveys 31, 264-323, 1999.

[5] S.C. Johnson, “Hierarchical clustering schemes” Psychometrika 32, 241-254, 1967.

[6] L. Faufman, P.J. Rousseeuw, “CLustering by means of Medoids”, in Statistical Data Analysis Based on the L_1–Norm and Related Methods, edited by Y. Dodge, North-Holland, 405–416, 1987.

[7] S.P. Lloyd, “Least square quantization in PCM”, IEEE Transactions on Information Theory 28, 129-137, 1982.

[8] J.B. MacQueen, “Some Methods for Classification and Analysis of Multivariate Observations”, Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability 1, 281-297, 1967.

[9] J.H. Ward Jr., “Hierarchical Grouping to Optimize an Objective Function”, Journal of the American Statistical Association, 58, 236–244, 1963.

[10] L.A. Zadeh, “Fuzzy sets”, Information and Control 8, 338–353, 1965.

Cuadro I
Resultados de los intervalos de pertenencia obtenidos por el algoritmo propuesto sobre el conjunto de datos de la Figura 5. En cada fila se muestra la información referente a un dato. Para él se indican los intervalos de pertenencia a los dos grupos existentes, cuando ejecutamos el algoritmo con diferentes valores del parámetro $1/a$: $1/0{,}0001$, $1/10000$ y $1/1$.

Dato	$1/a = 1/0{,}0001$		$1/a = 1/10000$		$1/a = 1/1$	
	Grupo 1	**Grupo 2**	**Grupo 1**	**Grupo 2**	**Grupo 1**	**Grupo 2**
(-1, -2)	[0.0000, 1.0000]	[0.0000, 1.0000]	[0.9725, 0.9725]	[0.0275, 0.0275]	[0.7882, 0.9967]	[0.0033, 0.2118]
(0, -1)	[0.0000, 1.0000]	[0.0000, 1.0000]	[0.9897, 0.9897]	[0.0103, 0.0103]	[0.8775, 0.9930]	[0.0070, 0.1225]
(0, -2)	[0.0000, 1.0000]	[0.0000, 1.0000]	[0.9988, 0.9988]	[0.0012, 0.0012]	[0.9999, 1.0000]	[0.0000, 0.0001]
(0, -3)	[0.0000, 1.0000]	[0.0000, 1.0000]	[0.9899, 0.9899]	[0.0101, 0.0101]	[0.6956, 0.9978]	[0.0022, 0.3044]
(1, -2)	[0.0000, 1.0000]	[0.0000, 1.0000]	[0.9995, 0.9995]	[0.0005, 0.0005]	[0.7925, 0.9967]	[0.0033, 0.2075]
(-1, 2)	[0.0000, 1.0000]	[0.0000, 1.0000]	[0.0275, 0.0275]	[0.9725, 0.9725]	[0.0033, 0.2118]	[0.7882, 0.9967]
(0, 1)	[0.0000, 1.0000]	[0.0000, 1.0000]	[0.0103, 0.0103]	[0.9897, 0.9897]	[0.0070, 0.1225]	[0.8775, 0.9930]
(0, 2)	[1.0000, 1.0000]	[0.0000, 0.0000]	[0.0012, 0.0012]	[0.9988, 0.9988]	[0.0000, 0.0001]	[0.9999, 1.0000]
(0, 3)	[0.0000, 1.0000]	[0.0000, 1.0000]	[0.0101, 0.0101]	[0.9899, 0.9899]	[0.0022, 0.3044]	[0.6956, 0.9978]
(1, 2)	[0.0000, 1.0000]	[0.0000, 1.0000]	[0.0005, 0.0005]	[0.9995, 0.9995]	[0.0033, 0.2075]	[0.7925, 0.9967]
(0, 0)	[0.0000, 0.0000]	[1.0000, 1.0000]	[0.5000, 0.5000]	[0.5000, 0.5000]	[0.2024, 0.7976]	[0.2024, 0.7976]
(7, 0)	[0.0000, 1.0000]	[0.0000, 1.0000]	[0.5000, 0.5000]	[0.5000, 0.5000]	[0.0015, 0.9985]	[0.0015, 0.9985]