



# Una generalización del algoritmo gravitacional de clústering utilizando funciones de overlap

I. Rodríguez, J. Ruiz-Aranguren, J. Fernández, H. Bustince  
 Departamento de Estadística, Informática y Matemáticas  
 Universidad Pública de Navarra, Pamplona  
 bustince@unavarra.es

Javier Armentia  
 Planetario de Pamplona  
 Pamplona, España

**Abstract**—En este trabajo consideramos una modificación del algoritmo de clúster gravitacional propuesto por Wright y utilizamos el concepto de función de overlap para mejorar su eficacia. Nuestra generalización recupera el algoritmo original. Comparamos los resultados de nuestra propuesta con los obtenidos con algoritmos usuales como el K-means o el FCM, con el objetivo de descubrir los puntos fuertes y las carencias de la nueva versión.

**Keywords**—Clúster; overlap; fuerza gravitacional

## I. INTRODUCCIÓN

La variedad en los problemas a tratar por las técnicas de clúster ha fomentado la aparición de decenas de algoritmos distintos, capaces de adaptarse a distintas situaciones y ofrecer resultados óptimos para cada caso [1]. Uno de estos algoritmos es el propuesto por W. E. Wright en 1977 [2]. Este algoritmo trata cada uno de los ejemplos sobre los que trabaja como una partícula que afecta al resto del sistema, según una adaptación de la ley de gravitación universal de Newton. Cada partícula compone inicialmente un clúster, que se van combinando a lo largo del tiempo hasta que solo queda uno.

El objetivo de este trabajo es generalizar el algoritmo de Wright utilizando funciones de overlap [3]. Nuestra generalización modifica la función fundamental que simula el movimiento de las partículas sustituyendo un término concreto por una función de overlap. Las funciones de overlap se introdujeron inicialmente en el campo difuso para tratar con el problema de determinar a cuál de dos conjuntos difusos pertenece en mayor medida una entrada dada. Sin embargo, desde sus orígenes se han aplicado con éxito en campos como los del procesamiento de imagen [4], la clasificación [5, 6] o la optimización [7].

En particular, en [7], una generalización muy competitiva del GSA, un algoritmo de optimización que también hace uso de la fuerza gravitacional, ver [8], fue propuesto, reemplazando el producto en la expresión de la fuerza por funciones de overlap (y otro tipo de funciones).

La estructura del trabajo es la siguiente. En la Sección II presentamos algunos conceptos preliminares. En la Sección III, presentamos el algoritmo de clúster gravitacional original y nuestra propuesta de generalización. La Sección IV se dedica a un estudio experimental de nuestra propuesta. Terminamos con algunas conclusiones y referencias.

## II. PRELIMINARES

### A. Funciones de Agregación

**Definición. [9]** Una función de agregación n-dimensional es una función  $M:[0,1]^n \rightarrow [0,1]$  que verifica las dos condiciones siguientes:

- M es no decreciente en cada una de sus variables;
- $M(0, \dots, 0) = 0$  y  $M(1, \dots, 1) = 1$ .

**Definición.** Sea  $M:[0,1]^n \rightarrow [0,1]$  una función de agregación n-dimensional.

- $a \in [0,1]$  es un aniquilador de M si  $M(x_1, \dots, x_n) = a$  siempre que  $a \in \{x_1, \dots, x_n\}$ .
- Si M no tiene aniquilador, M se dice estrictamente creciente si lo es en el dominio  $[0,1]^n$  como función real de n variables. Si a es un aniquilador de M, si es estrictamente creciente en el dominio  $([0, 1] \setminus \{a\})^n$ .
- M tiene divisores de cero si existen  $x_1, \dots, x_n \in [0, 1]$  tales que  $M(x_1, \dots, x_n) = 0$ .
- M es idempotente si  $M(x_1, \dots, x_n) = x$  para todo  $x \in [0, 1]$ .

Nos vamos a centrar en el caso de funciones de agregación en dimensión 2. Por ello, revisamos algunas propiedades que nos van a resultar de interés.

**Definición.** Sea M una función de agregación de dos variables.

- M es simétrica si  $M(x, y) = M(y, x)$  para cualesquiera  $x, y \in [0, 1]$ .
- M es asociativa si  $M(M(x, y), z) = M(x, M(y, z))$  para cualesquiera  $x, y, z \in [0, 1]$ .

La asociatividad permite extender las funciones de agregación bidimensionales a dimensiones mayores de una manera “razonable”. En particular, es una de las propiedades fundamentales que se demandan en la construcción de normas triangulares (t-normas).

**Definición .** Una norma triangular o t-norma es una función de agregación  $T:[0, 1]^2 \rightarrow [0, 1]$  asociativa y simétrica tal que  $T(1, x) = x$  para todo  $x \in [0, 1]$ .

Las t-normas permiten modelar intersecciones entre conjuntos difusos. Ahora bien, si estamos trabajando con solo dos elementos, la asociatividad no es en principio una propiedad que deba ser requerida. Esta idea, la de poder representar la intersección sin necesariamente requerir la asociatividad nos lleva al concepto de función de solapamiento (función de overlap) que estudiamos a continuación.

### B. Funciones de Overlap

En esta subsección recordamos algunos conceptos y resultados presentados en [3].

**Definición** Una función  $O:[0, 1]^2 \rightarrow [0, 1]$  es una función de solapamiento si

- O es simétrica;
- $O(x, y) = 0$  si y solo si  $xy = 0$ ;
- $O(x, y) = 1$  si y solo si  $xy = 1$ ;
- O es no decreciente;
- O es continua.

Podemos presentar el siguiente resultado sobre la relación entre funciones de solapamientos y t-normas.

**Teorema.** Sea O una función de solapamiento asociativa. Entonces O es una t-norma.

En cuanto a la construcción de funciones de overlap, es posible hacerlo de forma general tal y como muestra el siguiente resultado.

**Teorema.** La función  $O:[0, 1]^2 \rightarrow [0, 1]$  es una función de solapamiento si y solo si

$$O(x, y) = f(x, y) / (f(x, y) + h(x, y))$$

Con  $f, h: [0, 1]^2 \rightarrow [0, 1]$  tales que

- f y h son simétricas;
- f es no decreciente y h es no creciente;
- $f(x, y) = 0$  si y solo si  $xy = 0$ ;
- $h(x, y) = 0$  si y solo si  $xy = 1$ ;
- f y h son continuas;

## III. EL ALGORITMO DE CLÚSTER GRAVITACIONAL

### A. El Algoritmo Original

El algoritmo de clúster gravitacional pretende hacer uso de la Ley de Gravitación Universal de Newton para llevar a cabo un proceso de clasificación no supervisado [2].

El esquema del algoritmo, tal y como figura en [2], es el siguiente:

Supongamos que tenemos n partículas  $p_1, \dots, p_n$ , para las que conocemos sus posiciones  $s_1, \dots, s_n \in \mathbb{R}^n$ . Asignamos a cada partícula  $p_i$  una masa  $m_i = 1$  para comenzar el algoritmo.

1. Fijamos dos parámetros reales positivos  $\delta$  y  $\epsilon$ .
  - Utilizamos  $\delta$  para determinar la longitud de cada paso temporal  $dt$ . En concreto, en el intervalo  $[t, t + dt]$  la partícula que se mueva más deprisa debe desplazarse una distancia  $\delta$ .
  - Si dos partículas están en algún momento a distancia menor que  $\epsilon$ , las unimos en una partícula con masa igual a la suma de las masas original y posición dada por el centro de masas de las partículas originales.
2. Inicializamos el tiempo con  $t = 0$ .
3. Repetimos los pasos (i)-(iv) siguientes hasta que solo quede una partícula.

- (i). En cada intervalo de tiempo  $[t, t + dt]$ , para cada partícula  $i$  calculamos el efecto gravitacional de las demás partículas sobre ella. En concreto:

$$g(i, t, dt) = \frac{1}{2} G \sum_{j \neq i} \frac{m_i(t)m_j(t)}{m_i(t)} \frac{s_j(t) - s_i(t)}{|s_j(t) - s_i(t)|} \frac{1}{|s_j(t) - s_i(t)|^2} dt^2 \quad (1)$$

donde G es una constante positiva.

- (ii). Para cada partícula  $i$ , su nueva posición es:  $s_i(t + dt) = s_i(t) + g(i, t, dt)$ .
- (iii). Incrementamos el tiempo de  $t$  a  $t + dt$ .
- (iv). Si dos partículas  $i$  y  $j$  están a distancia (euclídea) menor que  $\epsilon$ , se funden en una partícula como se ha explicado anteriormente.

Al final tenemos una sola partícula y, si el proceso ha durado un tiempo T, la siguiente evolución:

- De  $t_n = 0$  a  $t_{n-1}$  había n partículas.
- De  $t_{n-1}$  a  $t_{n-2}$  había n-1 partículas.
- ...
- De  $t_3$  a  $t_2$  había 3 partículas.
- De  $t_2$  a  $t_1$  había 2 partículas.

Definimos la vida relativa de la configuración de k clusters como

$$R_k = (t_k - t_{k-1}) / T$$

Tomamos como solución aquella configuración  $R_{k_0}$  con la mayor vida relativa.

Nótese que en una iteración dada puede haber varias fusiones de partículas. El caso en que en  $t_k$  se tienen k partículas y en  $t_{k-1}$  pasa a haber k-2 partículas, se puede ver como que de  $t_k$  a  $t_{k-1}$  había k partículas, de  $t_{k-1}$  a  $t_{k-2}$  pasa a haber k-1 partículas y  $t_{k-1} = t_{k-2}$ , luego  $R_{k-1} = 0$ .

Este modelo puede generalizarse reemplazando (1) por



$$g(i, t, dt) = \sum_{j \neq i} \frac{m_i(t)^p m_j(t)^q}{m_i(t)} \frac{s_j(t) - s_i(t)}{|s_j(t) - s_i(t)| |s_j(t) - s_i(t)|^2} dt^2 \quad (2)$$

con  $p, q \geq 0$ . En particular, los mejores resultados se obtienen cuando  $p = q = 0$ , obteniéndose el denominado modelo unitario de Markov

$$g(i, t, dt) = \frac{1}{m_i(t)} \sum_{j \neq i} \frac{s_j(t) - s_i(t)}{|s_j(t) - s_i(t)| |s_j(t) - s_i(t)|^2} dt^2 \quad (3)$$

Nótese que la constante  $G/2$  se ha eliminado por tratarse únicamente de un valor heredado a partir de la ley de gravitación universal de Newton que no tiene un impacto útil en nuestro algoritmo, más allá del de escalar los valores obtenidos por la función  $g$ .

### B. Generalización Usando Overlaps

En esta sección presentamos el contenido principal del trabajo: un nuevo algoritmo de clúster gravitacional que generaliza el mostrado en la sección anterior, reemplazando el factor  $1/m_i(t)$  en la ecuación (3) por un término de la forma  $O(1/m_i(t), 1)$  donde  $O: [0,1]^2 \rightarrow [0,1]$  es una función de overlap general. Ahora bien, dado que la función de overlap está definida sobre el cuadrado unidad  $[0,1]^2$ , en principio es necesaria una normalización. Esto puede hacerse considerando la función  $O(m_i^{p-1}/n^{p-1}, m_j^q/n^q)$ , con  $p, q > 0$ .

Sin embargo, dado que estamos interesados en el caso  $p, q$  próximos a cero (que es el que mejor funciona en el trabajo original), se tiene que esta función siempre es mayor o igual que  $O(1/m_i(t), 1)$ . La ecuación final a utilizar será la siguiente:

$$g(i, t, dt) = O\left(\frac{1}{m_i(t)}, 1\right) \sum_{j \neq i} \frac{s_j(t) - s_i(t)}{|s_j(t) - s_i(t)| |s_j(t) - s_i(t)|^2} dt^2 \quad (4)$$

Los overlaps que ofrecen resultados más prometedores, y que por tanto se utilizarán en la fase de experimentación son los siguientes:

$$O(x, y) = (x * y)^p \rightarrow O\left(\frac{1}{m_i}, 1\right) = \frac{1}{m_i^p} \quad (5)$$

$$O(x, y) = \frac{\sqrt{x}}{\sqrt{x} + \max(1-x, 1-y)} \rightarrow O\left(\frac{1}{m_i}, 1\right) = \frac{\sqrt{\frac{1}{m_i}}}{\sqrt{\frac{1}{m_i}} + \max\left(1 - \frac{1}{m_i}, 0\right)} \quad (6)$$

## IV. EXPERIMENTACIÓN

### A. Pruebas

Para las pruebas se utilizaron una serie de datasets sintéticos bidimensionales con el fin de comprobar en qué casos se obtienen los mejores resultados para cada uno de los overlaps. Los datasets utilizados se representan visualmente en las Figs. 1-6.

Las pruebas se realizaron utilizando tanto métricas de evaluación interna (Dunn-Index [10]) como de evaluación externa (Purity, Rand-Index [11]). Se compararon los resultados del algoritmo empleando cada uno de los overlaps

con el modelo unitario de Markov, el algoritmo K-means [12] y el Fuzzy c-Means (FCM) [13].

Para estos dos últimos algoritmos se realizaron 100 ejecuciones y se tomaron los mejores valores obtenidos, buscando evitar el problema de caer en óptimos locales. Para el algoritmo gravitacional se asignaron valores tomados de 0.01 en 0.01 en  $[0.1, 1]$  para  $\delta$ , y de  $\epsilon = 2\delta$ . Como parámetro  $p$  para (5) se utilizó el valor 20, con el objetivo de que las masas grandes se vieran muy penalizadas. En las tablas se reflejan tanto los valores medios obtenidos tras todas las ejecuciones, como los valores máximos. La columna izquierda hace referencia al dataset para el cuál se han obtenido los resultados (Figs. 1-6).

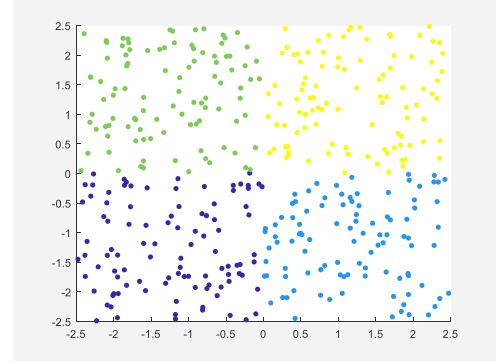


Fig. 1. Dataset "4 squares"

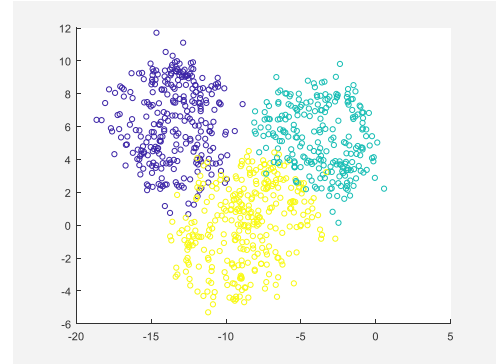


Fig. 2. Dataset "3 close circles"

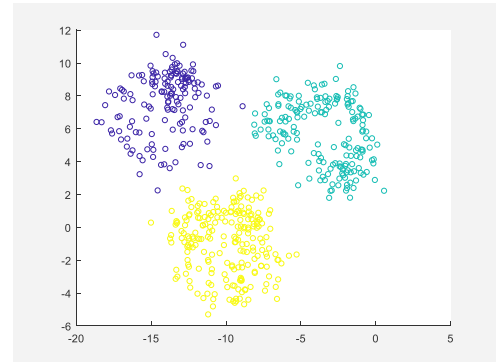


Fig. 3. Dataset "3 circles"

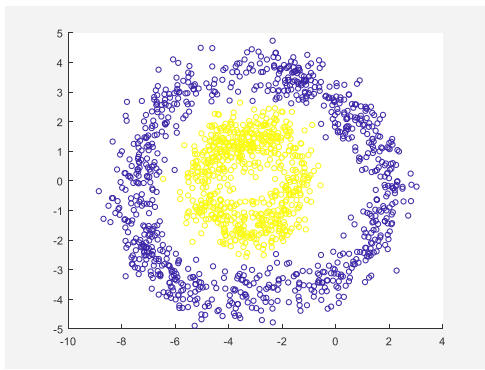


Fig. 4. Dataset "2 concentric rings"

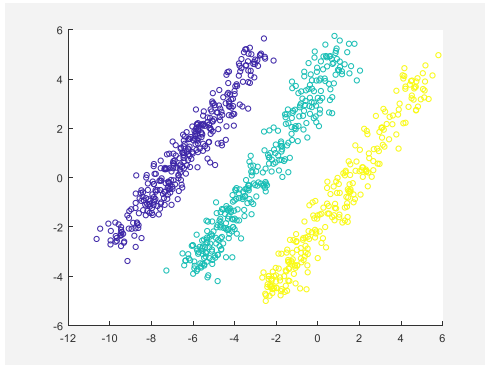


Fig. 5. Dataset "3 stripes"

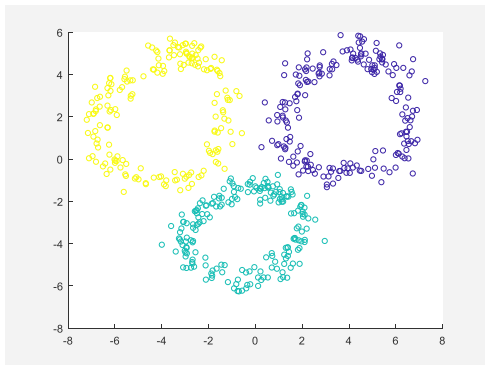


Fig. 6. Dataset "3 rings"

|   | Dunn-Index |       |       |              |                 |              |              |              |
|---|------------|-------|-------|--------------|-----------------|--------------|--------------|--------------|
|   | (5)        |       | (6)   |              | Modelo unitario |              | K-means      | FCM          |
|   | Mean       | Max   | Mean  | Max          | Mean            | Max          |              |              |
| 1 | 0,508      | 0,682 | 0,577 | 0,682        | 0,578           | 0,682        | <b>0,743</b> | 0,736        |
| 2 | 0,633      | 0,690 | 0,621 | <b>0,691</b> | 0,641           | <b>0,691</b> | 0,688        | 0,685        |
| 3 | 0,848      | 0,861 | 0,859 | 0,860        | 0,858           | 0,860        | <b>0,932</b> | <b>0,932</b> |
| 4 | 0,386      | 0,464 | 0,468 | <b>0,497</b> | 0,464           | <b>0,497</b> | 0,464        | 0,438        |
| 5 | 0,378      | 0,579 | 0,442 | <b>0,596</b> | 0,432           | 0,586        | 0,467        | 0,485        |
| 6 | 0,519      | 0,831 | 0,631 | 0,758        | 0,621           | 0,715        | <b>0,874</b> | 0,848        |

<sup>a</sup>. Tabla 1 – Comparación de índice Dunn entre los algoritmos para los datasets 1-6

|   | Purity |              |       |              |                 |              |              |              |
|---|--------|--------------|-------|--------------|-----------------|--------------|--------------|--------------|
|   | (5)    |              | (6)   |              | Modelo unitario |              | K-means      | FCM          |
|   | Mean   | Max          | Mean  | Max          | Mean            | Max          |              |              |
| 1 | 0,784  | 0,893        | 0,807 | 0,910        | 0,810           | 0,900        | <b>0,988</b> | 0,983        |
| 2 | 0,859  | 0,946        | 0,856 | <b>0,948</b> | 0,852           | 0,920        | 0,901        | 0,901        |
| 3 | 0,991  | <b>0,998</b> | 0,996 | <b>0,998</b> | 0,996           | <b>0,998</b> | <b>0,998</b> | <b>0,998</b> |
| 4 | 0,577  | 0,628        | 0,622 | <b>0,713</b> | 0,624           | 0,705        | 0,570        | 0,570        |
| 5 | 0,670  | <b>0,928</b> | 0,564 | 0,642        | 0,569           | 0,690        | 0,697        | 0,690        |
| 6 | 0,806  | <b>0,998</b> | 0,788 | 0,916        | 0,784           | 0,916        | 0,960        | 0,960        |

<sup>b</sup>. Tabla 2 – Comparación de Purity entre los algoritmos para los datasets 1-6

|   | Rand-Index |              |       |              |                 |              |              |              |
|---|------------|--------------|-------|--------------|-----------------|--------------|--------------|--------------|
|   | (5)        |              | (6)   |              | Modelo unitario |              | K-means      | FCM          |
|   | Mean       | Max          | Mean  | Max          | Mean            | Max          |              |              |
| 1 | 0,822      | 0,901        | 0,834 | 0,917        | 0,838           | 0,907        | <b>0,988</b> | 0,983        |
| 2 | 0,843      | <b>0,931</b> | 0,840 | <b>0,931</b> | 0,836           | 0,902        | 0,884        | 0,884        |
| 3 | 0,989      | <b>0,998</b> | 0,995 | <b>0,998</b> | 0,994           | <b>0,998</b> | <b>0,998</b> | <b>0,998</b> |
| 4 | 0,507      | 0,532        | 0,533 | <b>0,591</b> | 0,533           | 0,584        | 0,501        | 0,501        |
| 5 | 0,718      | <b>0,905</b> | 0,626 | 0,702        | 0,633           | 0,719        | 0,762        | 0,754        |
| 6 | 0,812      | <b>0,998</b> | 0,741 | 0,900        | 0,735           | 0,900        | 0,949        | 0,949        |

<sup>c</sup>. Tabla 3 – Comparación de índice Rand entre los algoritmos para los datasets 1-6

**B. Resultados**

Lo primero que se puede observar es que no siempre existe una relación directa entre los mejores resultados para la evaluación interna y los de la externa. Esto supone un problema, dado que dificulta la búsqueda de los parámetros óptimos para el algoritmo en los casos en que carecemos de información sobre las clasificaciones a priori, que serán la mayoría.

Si nos fijamos en las medidas de evaluación externa, sin embargo, vemos que el modelo usando (6) tiende a mejorar el mejor resultado del modelo clásico, mientras que suele ofrecer valores similares en media. Por tanto, parece una alternativa razonable al mismo. En concreto, obtiene los mejores resultados de cualquier método cuando opera con datasets circulares, del mismo modo que cuando lo hace con el dataset de la Fig. 4, aunque en este caso los resultados son mediocres.

En el caso del overlap indicado en (5) tenemos que se obtienen valores con mucha varianza para la mayoría de datasets. No obstante, se obtienen resultados más que aceptables al trabajar con los dos últimos, casos en los que ninguna de las alternativas del algoritmo ofrece buenos resultados. Esto parece deberse a la gran penalización de las partículas con masas mayores. El modo de trabajo del algoritmo asegura que en todas las iteraciones, una partícula se desplazará la distancia  $\delta$ . En este caso también sucede así, pero el resto de partículas de masa mayor se desplazan distancias ínfimas, de modo que cuando se tiene un número elevado de partículas, en cada iteración tiende a moverse una sola, obteniéndose un funcionamiento similar al del algoritmo "single-linkage" [15], que opera bien con conjuntos de datos de esta forma.



## V. CONCLUSIONES Y FUTUROS TRABAJOS

Las pruebas realizadas parecen apuntar a una mejoría del funcionamiento del algoritmo cuando se utilizan los overlaps propuestos en ciertos casos concretos. Además, lo dotan de una mayor capacidad de adaptación a conjuntos de datos de forma no circular, problema con el que no pueden tratar otro tipo de técnicas.

En cualquier caso, resulta importante recalcar que nuestro caso sigue siendo una generalización del algoritmo original, por lo que siempre se puede recuperar el funcionamiento clásico.

Esta generalización abre la puerta a un estudio en mayor profundidad que permita extender la generalización empleando funciones de overlap a funciones de otro tipo, consiguiendo posiblemente comportamientos mejores. Del mismo modo, puede haber alternativas a la distancia euclidiana a tener en cuenta a la hora de calcular la distancia entre partículas. Ambas posibilidades se tendrán en cuenta para futuros trabajos.

Este trabajo ha sido parcialmente financiado por el proyecto de investigación TIN2016-77356-P (MINECO/UE/FEDER) del Gobierno de España.

## REFERENCIAS

- [1] A. Saxena, M. Prasad, A. Gupta, N. Bharill, O. Prakash Patel, A. Tiwari, E. M. Joo, D. Weiping, L. Chin-Teng, "A review of clustering techniques and developments," in *Neurocomputing*, vol. 267, pp.664-681, 2017
- [2] W. E. Wright, "Gravitational clustering", in *Pattern Recognition*, Pergamon Press 9, 1977, pp.151-166
- [3] H. Bustince, J. Fernandez, R. Mesiar, J. Montero, R. Orduna, "Overlap functions," in *Nonlinear Analysis*, vol. 72, pp.1488-1499, 2010
- [4] A. Jurio, H. Bustince, M. Pagola, A. Pradera, R. R. Yager, "Some properties of overlap and grouping functions and their application to image thresholding," in *Fuzzy Sets and Systems*, vol. 229, pp.69-90, 2013
- [5] D. Gómez, J. T. Rodríguez, J. Montero, H. Bustince, E. Barrenechea, "n-Dimensional overlap functions," in *Fuzzy Sets and Systems*, vol. 287, pp.57-75, 2016
- [6] D. Paternain, H. Bustince, M. Pagola, P. Sussner, A. Kolesrov, R. Mesiar, "Capacities and overlap indexes with an application in fuzzy rule-based classification systems," in *Fuzzy Sets and Systems*, vol. 305, pp.70-94, 2016
- [7] M. Minárová, D. Paternain, A. Jurio, J. Ruiz-Aranguren, Z. tak, H. Bustince, "Modifying the gravitational search algorithm: A functional study," in *Information Sciences*, vol. 430, pp.87-103, 2018
- [8] E. Rashedi, H. Neamabadi-Pour, S. Sariazdi, "GSA: a gravitational search algorithm," in *Information Sciences*, vol 179, pp.2232-2248, 2009
- [9] M. Grabisch, J. Marichal, R. Mesiar, E. Pap. *Aggregation Functions, Encyclopedia of Mathematics and its Applications*, vol. 127, Cambridge University Press, Cambridge, 2009.
- [10] J. C. Dunn, "Well-separated clusters and optimal fuzzy partitions," in *Journal of Cybernetics*, vol. 4, pp.95-104, 1973
- [11] D. L. Davies, D. W. Bouldin, "A cluster separation measure," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, pp.224-227, 1979
- [12] D. Arthur, S. Vassilvitskii, "K-means++: the advantages of careful seeding," in *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, New Orleans, 2007
- [13] J. C. Bezdek, R. Ehrlich, W. Full, "FCM: the fuzzy c-means algorithm," in *Computer & Geosciences*, vol. 10, pp.191-203, 1984
- [14] R. Sibson, "SLINK: an optimally efficient algorithm for the single-link cluster method," in *The Computer Journal*, vol. 16, pp.30-34, 1973