International Journal of Innovation and Applied Studies
ISSN 2028-9324 Vol. 24 No. 1 Aug. 2018, pp. 17-30

http://www.ijias.issr-journals.org/

Algoritmos de agrupación difusos

[Fuzzy Clustering Algorithms]

Gary Reyes Zambrano[1] and Christopher Crespo[2]

[1]Facultad de Ciencias Matemáticas, Universidad de Guayaquil, Guayaquil, Ecuador

[2]Facultad de Ciencias Administrativas, Universidad de Guayaquil, Guayaquil, Ecuador

ABSTRACT: In recent years advances in technology have led to the generation of large volumes of data, mainly numerical data, highlighting the interest in processing them to extract knowledge and information from them. The main objective is to make more efficient the systems from which these data have been obtained and help in decision making. The information in a database is implicit in the values that represent the different states of the systems, whereas the knowledge is implicit in the relations between the values of the different attributes or present characteristics. These relationships are identified by groups to be discovered and describe the relationships between the input and output states. One of the main human functions is to classify, differentiate and group different objects according to their attributes. The article investigates how to apply fuzzy grouping algorithms, which allow an element to belong to more than one group by a degree of membership, in order to obtain relevant characteristics or recognize patterns of a set of data. We discuss a study that involved 4 main fuzzy algorithms where each algorithm is explained and how they are related, as well as with each new algorithm solves problems that the previous one did not solve efficiently.

KEYWORDS: Diffuse grouping, Fuzzy logic, Data mining and Diffuze technology.

RESUMEN: En los últimos años, los avances en la tecnología han llevado a la generación de grandes volúmenes de datos, principalmente datos numéricos, destacando el interés en procesarlos para extraer conocimiento e información de ellos. El objetivo principal es hacer más eficientes los sistemas a partir de los cuales se han obtenido estos datos y ayudar en la toma de decisiones. La información en una base de datos está implícita en los valores que representan los diferentes estados de los sistemas, mientras que el conocimiento está implícito en las relaciones entre los valores de los diferentes atributos o características presentes. Estas relaciones son identificadas por grupos para ser descubiertas y describen las relaciones entre los estados de entrada y salida. Una de las principales funciones humanas es clasificar, diferenciar y agrupar diferentes objetos según sus atributos. El artículo investiga cómo aplicar algoritmos de agrupamiento difusos, que permiten que un elemento pertenezca a más de un grupo por un grado de membresía, con el fin de obtener características relevantes o reconocer patrones de un conjunto de datos. Discutimos un estudio que involucró 4 algoritmos difusos principales donde cada algoritmo se explica y cómo se relacionan, así como con cada nuevo algoritmo resuelve problemas que el anterior no resolvió de manera eficiente.

PALABRAS CLAVE: Agrupación difusa, lógica difusa, extracción de datos y tecnología Difsa.

Corresponding Author: Gary Reyes Zambrano

1 INTRODUCCIÓN

La gran cantidad de datos y el elevado volumen de información que se tienen actualmente ha hecho necesario contar con técnicas automáticas que permitan indagar, organizar y extraer información implícita presente en las enormes bases de datos que contienen información variada, la cual extraer de forma manual resulta prácticamente imposible a medida que va creciendo el tamaño de las bases de datos. El problema con que se enfrentan las principales funciones humanas es la de clasificar, diferenciar y agrupar diversos objetos según sus atributos o características en comportamiento de los sistemas. Esta selección de atributos para tareas de clasificación y segmentación es encontrar un lenguaje de representación adecuado a los objetos que estudian y analizan.

Para formalizar lo que entendemos por agrupamiento necesitamos establecer dos conceptos fundamentales, el primero, qué se entiende propiamente por agrupar, y segundo, qué se entiende por asignar a un grupo, proceso que es denominado en la literatura por clasificar. Se establece una definición matemática formal al concepto de agrupamiento, de manera que sea útil para el objeto de las definiciones y aplicaciones posteriores. Desde un punto de vista formal, la tarea de un procedimiento de agrupar es asignar los individuos u objetos del conjunto $X = \{x_1,\ldots, x_n\}$ a c subconjuntos, denominados clases o grupos. Cada uno de esos subconjuntos es representado por un prototipo v_i $i \in \{1,\ldots, c\}$, y se calcula para el caso de agrupamiento con lógica difusa el grado de pertenencia μ_{ik} del objeto x_k al grupo c_i. [1]

Entre los métodos para agrupar se encuentra los métodos clásicos de la estadística, como el método k-means o jerarquizados, y otros desarrollados en las áreas de inteligencia artificial, como las redes neuronales, el aprendizaje de máquina, y los métodos de agrupamiento difuso.

Uno de los principales tipos de algoritmos de data mining son los de “clustering” de datos, los que agrupan datos por similitud. Podemos considerar que el proceso de clustering emula una de las funciones básicas del hombre, su capacidad de agrupar objetos. De ahí su importancia y la extensa literatura publicada al respecto.

La lógica difusa ha demostrado ser de gran utilidad para representar el comportamiento o dinámica de los sistemas mediante reglas difusas del tipo “Si-Entonces”.

Los primeros sistemas basados en reglas difusas se basaban en la información suministrada por expertos; sin embargo, para el caso de sistemas complejos las reglas así construidas no permitían una simulación aceptable del sistema. La búsqueda de sistemas difusos que aproximen de manera aceptable la dinámica de sistemas complejos ha conllevado al desarrollo de investigación en técnicas de extracción de reglas difusas a partir de datos de entrada y salida; es decir, al desarrollo de técnicas de identificación difusa. Sin embargo, estas técnicas de identificación deben generar modelos difusos que cumplan con dos características fundamentales: una acertada precisión, lo cual se puede determinar mediante una métrica del error; y una buena interpretabilidad para que el sistema difuso resultante se asemeje a la forma como el ser humano toma decisiones. La interpretabilidad requiere de, al menos, un número no muy alto de reglas, preferiblemente no mayor a nueve; no solapamiento de más de dos funciones de pertenencia.

Una de las tareas frecuentes en el área de reconocimiento de patrones, estadística aplicada y minería de datos, consiste en aplicar algoritmos de agrupamiento en el tratamiento de datos, con el fin de obtener características relevantes o reconocer patrones. En general, por patrones se designa alguna representación abstracta de un conjunto de datos. Desde un punto de vista abstracto, el problema del análisis de agrupamiento se define frecuentemente como el de encontrar “grupos naturales”. El agrupamiento es el proceso de asignar un ítem, un individuo, un objeto, o una observación a su lugar propio, es decir, al lugar natural que debe ocupar. En un sentido más concreto, el objetivo es reunir un conjunto de objetos en clases tales que el grado de “asociación natural” para cada individuo es alto con los miembros de su misma clase y bajo con los miembros de las otras clases. Lo esencial del análisis de agrupamiento se enfoca entonces a cómo asignar un significado a los términos “grupos naturales” y “asociación natural”, donde “natural” usualmente se refiere a estructuras homogéneas y “bien separadas”.

Las técnicas de lógica difusa permiten manejar datos en los cuales existe una transición suave entre categorías distintas, por lo que algunos datos pueden tener propiedades de clases diferentes, estando parcialmente en más de un grupo con un grado específico de pertenencia. Por lo que cabe realizar la pregunta de ¿cuándo usar la tecnología fuzzy o difusa?:

- En procesos complejos, si no existe un modelo de solución sencillo.
- En procesos no lineales.
- Cuando haya que introducir la experiencia de un operador “experto” que se base en conceptos imprecisos obtenidos de su experiencia.
- Cuando ciertas partes del sistema a controlar son desconocidas y no pueden medirse de forma fiable (con errores posibles).

- Cuando el ajuste de una variable puede producir el desajuste de otras.
- En general, cuando se quieran representar y operar con conceptos que tengan imprecisión o incertidumbre (como en las Bases de Datos Difusas).

La misma además tiene aplicación en varias áreas como: Control de sistemas, Predicción y optimización, Reconocimiento de patrones y Visión por ordenador y Sistemas de información o conocimiento.

Una de las tareas de la minería de datos es la identificación de grupos o clusters naturales en los conjuntos de datos.

El clustering o agrupamiento es la tarea de agrupar datos a partir de una medida de similitud [3,4]. Estos métodos permiten clasificar y correlacionar datos en el espacio (ver Fig. 1). La clasificación puede manejarse en forma supervisada y no supervisada. Un aprendizaje es supervisado, cuando se realiza un reconocimiento de patrones, es decir, se conoce la salida esperada de la clasificación. Un aprendizaje es no supervisado, cuando se construyen clases desconociendo la salida esperada del agrupamiento. En el caso de aprendizaje no supervisado se habla de clustering. [2]

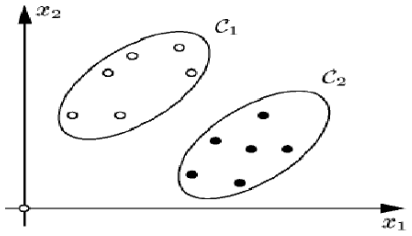

Fig. 1. Representación geométrica de dos clases en el espacio de datos

La importancia de desarrollar algoritmos de agrupamiento de carácter dinámico, proviene a partir de que éstos tienen la capacidad para enfrentar situaciones derivadas de actualizaciones o agregación de nuevos objetos, y segundo que pueden reutilizar las estructuras de clases y la información conocida en el pasado. Este potencial es equivalente a tener la capacidad de reutilizar nuevamente la información de las clases obtenidas en una etapa anterior en un nuevo agrupamiento sobre actualizaciones de los objetos o individuos que se agregan, por ejemplo, utilizarlos en aplicaciones de Data Mining que pueden ser bastante costosas en tiempo de análisis, refrescándolas con la actualización o agregación de objetos aprovechando el conocimiento ya obtenido.

Los algoritmos de agrupamiento difuso representan la técnica más adecuada para la obtención de modelos difusos, siendo los métodos de Fuzzy C-Means, K-Mean y de GustafsonKessel los más empleados. [6]

2 ANÁLISIS TEÓRICO

2.1 K-MEANS

Uno de los algoritmos de clustering más conocido es el K-Means. Este recibe como parámetro un número K de clusters a encontrar, y luego mediante iteraciones respecto a los datos, encuentra K “centroides” o “centros de masa” de los datos, además de asignar cada dato al centroide más cercano. K-Means (el más simple) agrupa los datos en hiperesferas, y su entrenamiento depende del número de clases (K) y de los centroides iniciales [4]. Con base en un criterio de optimización el método permite agrupar los datos en función de la similitud entre ellos, en este caso la distancia mide la separación de un dato con respecto al centro de una clase.

K-Means clustering tiene la intención de dividir n objetos en k clusters en los que cada objeto pertenece al clúster con la media más cercana. Este método produce exactamente k diferentes clusters de mayor distinción posible. El mejor número de grupos k que conducen a la mayor separación (distancia) no se conoce como a priori y se debe calcular a partir de los datos. El objetivo del agrupamiento de K-Means es minimizar la varianza intra-agrupación total o la función de error cuadrático. Además, procesa los patrones secuencialmente (por lo que requiere un almacenamiento mínimo).

K-Means es relativamente un método eficiente. Sin embargo, tenemos que especificar el número de clústeres, por adelantado y los resultados finales son sensibles a la inicialización y, a menudo termina en un óptimo local. Desafortunadamente no hay un método teórico global para encontrar el número óptimo de grupos. Un enfoque práctico es comparar los resultados de múltiples ejecuciones con diferentes k y elegir el mejor basado en un criterio predefinido. En general, un k grande probablemente disminuye el error pero aumenta el riesgo de overfitting[1]. [2]

Aunque dicho algoritmo está sesgado por el orden de presentación de los patrones y su comportamiento depende enormemente del parámetro K si se selecciona adecuadamente el número de agrupamientos el algoritmo se comporta como un buen clasificador, ya que los elementos internos son cercanos y los elementos externos se alejan. En este algoritmo la distancia cuadrática Euclideana es usada como medida discriminante:

$$d(x_i, x_{i'}) = \sum_{j=1}^{n} (x_{ij} - x_{i'j})^2 = \|x_i - xi'\|^2 \quad (1)$$

De igual forma los puntos de dispersión pueden ser escritos como:

$$W(C) = \sum_{k=1}^{K} N_k \sum_{C(i)=k} \|x_i - xi'\|^2 \quad (2)$$

Donde $\bar{x_k} = (\bar{x}_{1k}, \ldots, \bar{x}_{nk})$ es el vector de medias asociado con el k-ésimo grupo, y $Nk = \sum_{i=1}^{N} I(C(i) = k)$.

El algoritmo puede presentar mejor convergencia que otros algoritmos, dado que se evalúa la actualización de cada centroide de forma independiente, aunque, a su vez, podría representar un mayor costo computacional.

Asumiendo que un elemento x_i perteneciente a un grupo C_l en la solución actual es reasignado a algún otro grupo C_j, la actualización de los centroides puede realizarse aplicando [8]:

$$\boldsymbol{q}_l \leftarrow \frac{n_l \boldsymbol{q}_l - \boldsymbol{x}_i}{n_l - 1}, \quad \boldsymbol{q}_j \leftarrow \frac{n_j \boldsymbol{q}_j + \boldsymbol{x}_i}{n_j + 1} \quad (3)$$

donde $n_i = n_e(C_i)$ y $l \neq j$.

Seudocódigo del algoritmo [8]:

1. Inicialización: escoger un valor de k y una partición inicial C(0) con centroides Q(0), fijar número máximo de iteraciones N_{iter}, inicializar el contador: r = 1
   Mientras r < Niter
   Desde j = 1 **hasta** k **hacer**
2. Mover los centroides:
   $$\boldsymbol{q}_l^{(r)} \leftarrow \frac{n_l \boldsymbol{q}_l^{(r-1)} - \boldsymbol{x}_i}{n_l - 1}, \quad \boldsymbol{q}_j^{(r)} \leftarrow \frac{n_j \boldsymbol{q}_j^{(r-1)} + \boldsymbol{x}_i}{n_j + 1}$$
3. Calcular el cambio de la función objetivo:
   $$v_{ij} = \frac{n_j}{n_j + 1} \|\boldsymbol{q}_j^{(r)} - \boldsymbol{x}_i\|^2 - \frac{n_l}{n_l - 1} \|\boldsymbol{q}_l^{(r)} - \boldsymbol{x}_i\|^2, \quad \boldsymbol{x}_i \in \boldsymbol{C}_l^{(r)}$$
   Si $v_{ij} \geq 0$ (i = 1,..., n y j = 1,..., k)
4. El proceso termina y la solución es $C^{(r)}$
   en caso contrario

   r ← r + 1

   Termina Si

   Termina Desde

   Termina Mientras

2.2 C-MEANS CLÁSICO

C-means es un algoritmo iterativo que hace parte de las técnicas de agrupamiento no supervisado y tiene como objetivo encontrar patrones o grupos interesantes en un conjunto de datos dado, de tal manera que tales patrones, estructuras o grupos encontrados sirvan para la clasificación, el diseño de estrategias, el soporte de decisiones o la organización de la información [10].

Al igual que otras técnicas clásicas deagrupamiento C-means realiza una partición dura del conjunto de datos, que se caracteriza porque cada dato pertenezca exclusivamente a un cluster (grupo o clase) de la partición, además, los clusters deben cubrir totalmente el conjunto de datos, es decir, cada dato tiene que pertenecer a alguno de los clusters. Para lo cual la cantidad de clusters debe ser definida para inicializar el algoritmo. Una partición dura se define formalmente de la siguiente manera:

Sea X un conjunto de datos y x_i un elemento perteneciente a X. se dice que una partición $P = \{C_1, C_2,..., C_C\}$ donde c es un número entero no negativo que indica la cantidad de clusters, es una partición dura de X si y solo si:

$\forall x_i \in X \; \exists C_j \in P$ tal que $x_i \in C_j$

$\forall x_i \in X \; x_i \in C_j \Rightarrow x_i \notin C_k$

Donde $k \neq j$, C_k, C_j, $\in P$

La primera condición asegura que la partición cubra todos los puntos de X, la segunda garantiza que todos los clusters sean mutuamente excluyentes.

Los objetivos del algoritmo c-means convencional son: encontrar el centro de cada cluster (este punto central es conocido con el nombre de prototipo del cluster) y determinar cuál es el único cluster al que pertenece cada punto del conjunto de datos.

Para lograr el objetivo de hallar el centro de cada cluster se establece un criterio de búsqueda de dicho centro. Uno de tales criterios es la suma de la distancia entre los puntos de cada cluster y su centro, de la siguiente manera:

$$J(P,V) = \sum_{j=1}^{c} \sum_{x_i \in X} \left\| x_i - v_j \right\|^2 \quad (4)$$

Donde V es un vector de los centros de cada cluster a ser identificados. Este criterio es útil porque un conjunto de centros de los clusters adecuado o correcto brindará un valor mínimo de la función J.

Como primer paso el algoritmo C-means calcula la partición actual con base a los prototipos actuales, como segundo paso modifica los prototipos actuales usando un método de optimización (Ej. Gradiente óptimo) para minimizar la función J, luego estos dos pasos se repiten iterativamente hasta alcanzar algún criterio de parada que usualmente es la diferencia de los prototipos entre dos ciclos consecutivos; cuando el algoritmo alcanza su criterio de parada significa que la función J llegó a un mínimo local.

2.3 FUZZY C-MEANS (FCM)

En muchas situaciones cotidianas ocurre el caso que un dato está lo suficientemente cerca de dos grupos de tal manera que es difícil etiquetarlo en uno o en otro, esto se debe a la relativa frecuencia con la cual un dato particular presenta características pertenecientes a grupos distintos y como consecuencia no es fácilmente clasificado.

En agrupamiento difuso, los puntos de datos pueden pertenecer a más de un grupo, y asociado con cada uno de los puntos son los grados de miembros que indican el grado en que los puntos de datos pertenecen a los diferentes grupos.

Una técnica difusa bastante conocida y que ha cobrado importancia en la tarea de clustering o agrupamiento es el algoritmo Fuzzy C-Means (FCM), es una técnica difusa de minería de datos para el clustering que se basa en el algoritmo clásico C-Means.

FCM es un algoritmo que se desarrolló con el objetivo de solucionar los inconvenientes de la técnica de K-means. El algoritmo FCM asigna a cada dato un valor de pertenencia dentro de cada grupo y por consiguiente un dato específico puede pertenecer parcialmente a más de un grupo. A diferencia del algoritmo K-means clásico que trabaja con una partición dura, FCM realiza una partición suave del conjunto de datos, en tal partición los datos pertenecen en algún grado a todos los grupos; una partición suave se define formalmente como sigue: Sea X conjunto de datos y x_i un elemento perteneciente a X, se dice que una partición P = (C1, C2,..., Cc) es una partición suave de X si y solo si las siguientes condiciones se cumplen:

$\forall x_i \in X \forall C_j \forall P 0 \leq \mu_{cj}(x_i) \leq 1$

$\forall x_i \in X \exists C_j \forall P / 0 \leq \mu_{cj}(x_i) \leq 1$

Donde $\mu_{cj}(x_i)$ denota el grado en el cuál x_i pertenece al grupo C_j.

El algoritmo asigna a cada dato un grado de pertenencia dentro de cada cluster y como consecuencia un dato puede pertenecer parcialmente a más de un grupo. Es una técnica de minería de datos que permite encontrar grupos naturales en un conjunto de datos y puede ser aplicado en diversos campos como organización y clasificación de datos, reconocimiento de patrones, estudio del clima, diagnóstico de enfermedades, bioinformática, genética, cancelación de ruido e interferencia de una señal, estudio de series de tiempo, estudio de la rentabilidad económica de una empresa, soporte de la decisión, segmentación de mercados y clientes (weber).

Pasos para aplicar el algoritmo:

- Agrupar los datos en k grupos donde k está predefinido.
- Seleccionar k puntos al azar como centros de agrupación.
- Asignar objetos a su centro de clúster más cercano de acuerdo con la función de distancia euclidiana.
- Calcular el centroide o la media de todos los objetos en cada grupo.
- Repetir los pasos 2, 3 y 4 hasta que los mismos puntos se asignen a cada grupo en rondas consecutivas.

Para calcular el grado de pertenencia de cada caso, respecto a cada conjunto, y para cada variable de entrada se utiliza la siguiente ecuación:

$$\mu_{ci}(x) = \frac{1}{\sum_{j=1}^{k} \left(\frac{\|x_i - c_i\|^2}{\|x_i - c_j\|^2} \right)^{\frac{1}{m-1}}} \quad (5)$$

FCM se basa en la minimización de la siguiente función objetiva:

$$J_m = \sum^{N} \sum^{C} u_{ij}^{m} \|x_i - c_j\|^2 \quad (6)$$

Donde m es cualquier número real mayor que 1 el cual es un peso que determina el grado en el cuál los miembros parciales de un conjunto afectan el resultado, xi es el i-ésimo caso de datos, μ_{ij} es el grado de pertenencia de x_i en cada conjunto j, c_j es el centro del conjunto j, y $|| * ||$ es cualquier norma que expresa la similitud entre un caso y el centro del conjunto difuso.

Una partición difusa $\{C_1, C_2, \ldots, C_k\}$ puede ser un mínimo local de la función objetivo J_m solo si las siguientes condiciones se cumplen:

$$\mu_{Ci}(x) = \frac{1}{\sum_{j=1}^{k} \left(\frac{\|x - v_i\|^2}{\|x - v_j\|^2} \right)^{\frac{1}{m-1}}} \quad 1 \leq i \leq k, x \in X \quad (7)$$

$$v_i = \frac{\sum_{x \in X} (\mu_{Ci}(x))^m \, x}{\sum_{x \in X}^{n} (\mu_{Ci}(x))^m} \quad 1 \leq i \leq k \quad (8)$$

ALGORITMO FCM:

Considerando a U^t como el valor de la matriz de partición y v^t como el valor de la matriz de prototipos en la iteración t-ésima del método se presenta el siguiente algoritmo:

1. Inicializar U^0 y v^0 aleatoriamente.
2. If $\| v^t - v^{t-1} \| < \epsilon \rightarrow$ Fin
3. Else:

- Actualizar U^t con v^{t-1} y la ecuación (7).
- Actualizar v^t con U^t y la ecuación (8).

2.4 WEIGHTED FUZZY C-MEANS (WFCM)

El algoritmo Weighted Fuzzy C-Means (WFCM), que representa una adaptación particular de FCM para incorporar el proceso de selección de atributos.

El algoritmo FCM clásico nos entrega como salida un conjunto de vectores prototipos v y a una matriz de partición U que asigna a cada uno de los patrones, distintos grados de pertenencia a cada uno de los respectivos prototipos o clúster. Sin embargo, el método clásico no entrega información respecto a la relevancia de los atributos en este proceso de clustering este algoritmo que se explica a continuación implementa un aprendizaje de los pesos de los atributos en un proceso de clustering.

Antes de entrar en el algoritmo primero se debe presentar la métrica de distancia que incorpora los pesos, luego la derivación de la regla de aprendizaje de pesos y finalmente la nueva estructura del algoritmo WFCM.

MÉTRICA DE DISTANCIA

Uno de los componentes más importantes en un proceso de clustering donde se busca agrupar a los símiles y separar a los disimiles es la forma en que se define la función de distancia entre objetos. Como se ha presentado el algoritmo original de FCM se basa en la función de distancia euclideana. La primera modificación planteada consiste en considerar una ponderación de cada atributo por un conjunto de pesos.

Sea $w = [w_1,\ w_2, \dots, w_p]$ un vector de pesos asociado a los p atributos de un conjunto de datos particular X. Se define la distancia entre un patrón $x_k \in X$ al i-ésimo prototipo v_i con pesos w mediante:

$$d_{ik}^{w} = \sqrt{\sum_{j=1}^{p} \frac{w_j^{\alpha}(x_{kj} - v_{ij})^2}{\sigma_j^2}} \quad (9)$$

Donde σ^2_j representa la varianza del j-ésimo atributo.

Esta función de distancia es muy similar a la euclideana utilizada en FCM. Sin embargo, se puede notar que hay dos diferencias relevantes. En primer lugar se considera un factor de escalamiento por atributo asociado a la varianza. Este factor proviene de la distancia de Mahalanobis que considera un escalamiento por la matriz (Σ) de varianzas y co-varianzas. En nuestro caso no se considera la correlación entre atributos, lo cual elimina las co-varianzas de la matriz Σ de manera que esta sea diagonal. De esta forma la función distancia escala cada atributo dividiendo por su varianza σ^2_j

La segunda diferencia está en la ponderación de pesos $w_j^{\propto}$ asociada a los atributos. Esta es la ponderación que se desea aprender durante el proceso de clustering. El parámetro α representa una suavización respecto a la discriminación de los atributos, similar a la función del parámetro m en FCM.

Además de considerar esta nueva función de distancia, se desea que $w_j \in [0,1]$ y que $\sum_{j=1}^{p} w_j = 1$ para que los pesos representen una ponderación. De esta manera, se obtendrá una interpretación más directa de los pesos resultantes.

NORMALIZACIÓN E INVARIANZA DE ESCALA

Es importante notar que la distancia Mahalanobis planteada se puede aplicar a FCM, incluso sin los pesos w adicionales. Además el efecto de usar esta función de distancia es equivalente a una normalización de los datos. Consideremos una normalización gaussiana de los datos x de la siguiente manera:

$$x^n = \frac{x - \bar{x}}{\sigma} \quad (10)$$

Donde $\bar{x}$ y σ representan la media y desviación estándar de los datos respectivamente, y x^n corresponde a los datos normalizados. Ahora considerando la distancia euclideana entre dos datos normalizados:

$$\begin{aligned} d_{ik} &= \sqrt{\sum_{j=1}^{p}(x_{ij}^n - x_{kj}^n)^2} \\ &= \sqrt{\sum_{j=1}^{p}((\frac{x_{ij}-\bar{x}_j}{\sigma_j}) - (\frac{x_{kj}-\bar{x}_j}{\sigma_j}))^2} \\ &= \sqrt{\sum_{j=1}^{p}(\frac{(x_{ij}-x_{kj})^2}{\sigma_j^2}} \end{aligned} \quad (11)$$

Se obtiene justamente la distancia mahalanobis escalar que se considera en la ecuación (9) pero sin pesos.

Por otra parte, es importante tener en cuenta que el algoritmo FCM no tiene la propiedad de invarianza de escala, i.e., al cambiar la escala de los datos, no necesariamente se obtienen los mismos clusters. Como la normalización mencionada es justamente un escalamiento de los datos, debemos tener en cuenta que ello afecta al algoritmo de FCM.

En WFCM se plantea este escalamiento aprendido como un aporte al proceso de clustering. La consideración de la distancia de Mahalanobis escalar permite evitar sesgos del proceso relacionados con la varianza de los datos y de esta forma el algoritmo WFCM tiene invarianza de escala.

APRENDIZAJE DE LOS PESOS DE LOS ATRIBUTOS

En esta sección se deriva una regla de aprendizaje para los nuevos pesos de los atributos introducidos en la función de distancia. Para ello se modifica la función objetivo de FCM para incorporar dichos pesos:

$$J_m^w = \sum_{k=1}^{n}\sum_{i=1}^{c} u_{ik}^m \sum_{j=1}^{p} \frac{w_j^{\alpha}(x_{kj}-v_{ij})^2}{\sigma_j^2} = \sum_{k=1}^{n}\sum_{i=1}^{c} u_{ik}^m d_{ik}^{w2} \quad (12)$$

Con esta función objetivo se define el problema de Weighted Fuzzy C-Means (WFCM) de la siguiente manera:

$$min J_m^w(U,v,w) = \sum_{k=1}^{n}\sum_{i=1}^{c} u_{ik}^m d_{ik}^{w2}$$

S.A.

$$\begin{aligned} \sum_{i=1}^{c} u_{ik} &= 1 \qquad (\forall\, k) \\ \sum_{j=1}^{p} w_j &= 1 \\ u_{ik}, w_j &\in [0,1] \end{aligned} \quad (13)$$

Se puede ver que el problema planteado es el mismo que el de FCM, considerando la nueva función de distancia propuesta y la restricción sobre los pesos de los atributos. Por ello, se sigue un análisis y resolución similar al planteado por Bezdek [2] para el algoritmo original. Para la minimización de esta función se consideran multiplicadores de Lagrange asociadas a las primeras dos restricciones. Ello permite incorporar dichas restricciones a la función objetivo. Las columnas de la matriz U son independientes, lo que permite plantear el lagrangeano:

$$L_k = \sum_{k=1}^{n}\sum_{i=1}^{c} u_{ik}^m d_{ik}^{w2} - \lambda(\sum_{i=1}^{c} u_{ik} - 1) - \eta(\sum_{j=1}^{p} w_j - 1) \quad (13)$$

Para encontrar el óptimo, se resuelve $\nabla L_k = 0$ con respecto a los parámetros U , v y w. Al derivar respecto a u_{ik} y v_i se obtienen las ecuaciones originales de actualización de Bezdek [2]:

$$u_{ik} = \frac{1}{\sum_{l=1}^{c} \left(\frac{d_{ik}^w}{d_{lk}^w}\right)^{\frac{2}{m-1}}} \quad (14)$$

$$v_i = \frac{\sum_{k=1}^{n} u_{ik}^m x_k}{\sum_{k=1}^{n} u_{ik}^m} \quad (15)$$

La ecuación (15) de actualización de los centros v_i, no cambia respecto de FCM al agregar los pesos. Sin embargo, la ecuación de actualización (14) de la matriz de pertenencia U, sí cambia al considerar la relación entre las distancias $ponderadas$ por nuestra nueva función de distancia.

Al resolver $\frac{\partial L_k}{\partial w_j} = 0$ se obtiene:

$$\frac{\partial L_k}{\partial w_j} = \sum_{k=1}^{n} \sum_{i=1}^{c} u_{ik}^m 2 d_{ik}^w \frac{\partial d_{ik}^w}{\partial w_j} - \eta = 0 \quad (16)$$

Ya en una ecuación anterior (13) se encuentra la derivada de la función distancia respecto al peso w_j :

$$\begin{aligned} \frac{\partial d_{ik}^w}{\partial w_j} &= \frac{1}{2}\left(\sum_{j=1}^{p} \frac{w_j^\alpha (x_{kj} - v_{ij})^2}{\sigma_j^2}\right)^{-\frac{1}{2}} \frac{\alpha w_j^{\alpha-1} (x_{kj} - v_{ij})^2}{\sigma_j^2} \\ &= \frac{\alpha w_j^{\alpha-1} (x_{kj} - v_{ij})^2}{2\sigma_j^2 d_{ik}^w} \end{aligned}$$

Poniendo esto en 10:

$$\begin{aligned} \Rightarrow \frac{\partial L_k}{\partial w_j} &= \sum_{k=1}^{n} \sum_{i=1}^{c} u_{ik}^m 2 d_{ik}^w \frac{\alpha w_j^{\alpha-1} (x_{kj} - v_{ij})^2}{2\sigma_j^2 d_{ik}^w} - \eta = 0 \\ &= \sum_{k=1}^{n} \sum_{i=1}^{c} u_{ik}^m \frac{\alpha w_j^{\alpha-1} (x_{kj} - v_{ij})^2}{\sigma_j^2} - \eta = 0 \end{aligned}$$

De donde se obtiene:

$$w_j = \left(\frac{\eta}{f_j}\right)^{\frac{1}{\alpha-1}} \quad (17)$$

donde f_j se ha definido como:

$$f_j = \frac{\alpha}{\sigma_j^2} \sum_{k=1}^{n} \sum_{i=1}^{c} u_{ik}^m (x_{kj} - v_{ij})^2 \quad (18)$$

De la segunda restricción en 6 se tiene que $\sum_{l=1}^{p} w_{l=1}$, y en ecuación 17 ellos nos entregan:

$$\sum_{l=1}^{p} w_l = \sum_{l=1}^{p} (\frac{\eta}{f_l})^{\frac{1}{\alpha-1}} = 1$$

$$= \eta^{\frac{1}{\alpha-1}} \sum_{l=1}^{p} (\frac{1}{f_l})^{\frac{1}{\alpha-1}} = 1$$

$$\Rightarrow \eta^{\frac{1}{\alpha-1}} = \frac{1}{\sum_{l=1}^{p} \frac{1}{f_l^{\frac{1}{\alpha-1}}}}$$

Reemplazando $n^{\frac{1}{\alpha-1}}$ en 11,

$$w_j = \frac{1}{f_j^{\frac{1}{\alpha-1}}} \frac{1}{\sum_{l=1}^{p} \frac{1}{f_l^{\frac{1}{\alpha-1}}}}$$

de donde finalmente se obtiene:

$$w_j = \frac{1}{\sum_{l=1}^{p} (\frac{f_j}{f_l})^{\frac{1}{\alpha-1}}} \quad (18)$$

El algoritmo WFCM sigue el mismo esquema que el de FCM planteado en el que se basa en el concepto de Optimización Alternante [11], a su vez basado en la idea de iteraciones de Picard como método de aproximación numérica a una solución. En general la idea consiste en fijar todos los parámetros menos, el que se optimiza mediante las ecuaciones de actualización mencionadas anteriormente. Ello se repite hasta que se cumpla un criterio de convergencia, ya sea en torno a uno de los parámetros o asociado a un número fijo de iteraciones. En este contexto el algoritmo se puede resumir de la siguiente manera:

1. Inicializar U^0, v^0 *y* w^0.
2. $If\ \|v_t - v_{t-1}\| < \in \rightarrow Fin$
3. $Else$:

   - Actualizar U^t con v^{t-1} y w^{t-1} según la ecuación 15.
   - Actualizar v^t con U_t según ecuación 16.
   - Actualizar w^t con U^t y v^t según ecuación 18.

En el algoritmo FCM la inicialización es aleatoria y sólo afecta a uno de los 2 parámetros dado que las actualizaciones son interdependientes. Vale decir, si en el paso 3 del algoritmo, primero se actualiza U^1, y el valor de U^0 no es relevante. En el caso de WFCM la interdependencia es respecto a 3 parámetros. En el esquema sólo nos interesan los valores iniciales para v^0 y w^0. En el caso de v^0 cada prototipo se inicializa aleatoriamente entre los rangos mínimos y máximos por dimensión. En el caso de w^0 el vector se inicializa con un valor constante igual a $1/p$, valor que permite no sesgar el movimiento inicial de los otros parámetros y además cumplir con las condiciones planteadas en el 13.

2.5 GUSTAFSONKESSEL

La agrupación difusa utiliza la teoría de conjuntos difusos y las técnicas de agrupamiento basadas en funciones objetivas en distancias entre clusters y distancias entre datos y clusters. Los primeros trabajos sobre agrupamiento difuso [14, 15] se basaron sobre la distancia euclidiana [16] que no pudo capturar la correlación entre los datos. Este problema ha sido resuelto en GK-clutering [17], utilizando Mahalanobis distancia [18] para considerar la distribución de los datos mediante la incorporación de la Covarianza de datos. La distancia de Mahalanobis tiene en cuenta la correlación del conjunto de datos y es invariante en escala, es decir, no depende de la escala de las mediciones. Una versión mejorada de fuzzy GK-clustering se introdujo en [19].

Para afrontar la singularidad y la sobrealimentación.

Gustafson y Kessel extendieron los medios c difusos estándar utilizando una norma adaptativa de distancia, con el fin de para detectar agrupamientos de diferentes formas geométricas en un solo dato Conjunto [13]. Cada grupo tiene su propia matriz inductora de normas A_1. Las matrices A_i se utilizan como variables de optimización en la C-means funcional, lo que permite a cada grupo adaptar la distancia a la estructura topológica local de los datos. Dado el conjunto de datos Z, elija el número de clústeres como $1 < c < N$, el exponente de ponderación $m > 1$ y la terminación Tolerancia $0 > \varepsilon$. Inicializar la matriz de partición aleatoriamente tal que $U^{(0)} \in M_{fc}$.

El algoritmo G-K es parecido que el C-means, sólo el cálculo de la distancia medida se calcula utilizando matrices de covarianza como sigue:

$$F_i = \frac{\sum_{k=1}^{N} (\mu_{ik}^{(l-1)})^m (Z_k - V_i^{(l)})(Z_k - V_i^{(l)})^T}{\sum_{k=1}^{N} (\mu_{ik}^{(l-1)})^m}, \quad (19)$$

La aplicación del algoritmo GK-clustering se representa en los siguientes pasos [17, 19]:

Paso 1: Determine el número de clusters en términos de conocimiento previo sobre el sistema. A continuación, inicialice la partición $Matrix\ u = [B_{ik}]$ con valores aleatorios tomados del intervalo [0,1]. Las filas de matriz de partición corresponden a clústeres mientras que sus columnas se asocian con los datos. Por lo tanto, cada fila de matriz de partición tiene que incluir al menos una entrada distinta de cero y la suma de cada columna de esta matriz debe ser 1;

Paso 2: Calcule los prototipos usando

$$v_i^l = \frac{\sum_{k=1}^{N} (\beta_{ik}^{l-1})^e Z_k}{\sum_{k=1}^{N} (\beta_{ik}^{l-1})^e}; \quad (20)$$

Paso 3: Calcular $D_{ik\,A_i}^2$ valores usando la ecuación $D_{ikA_i}^2 = (Z_k - V_i)^T A_i (Z_k - V_i)$ y ecuación 13.

Paso 4: Actualizar la matriz de particiones basada en

$$\beta_{ik}^l = \frac{1}{\sum_{j=1}^{C} \left(\frac{D_{ikA_i}}{D_{jkA_j}} \right)^{2/(e-1)}}; \quad (21)$$

Paso 5: Vaya al paso 2 y repita los procedimientos, hasta que la desigualdad de convergencia $\left\| u^{(l+1)} - u^{(l)} \right\| <$ ∈ está satisfecho donde $\left\| u^{(l+1)} - u^{(l)} \right\|$ es el cambio en la norma de la matriz de partición, l el número de iteraciones y ε es un pequeño valor real positivo.

El algoritmo GK-clustering es una potente técnica de clustering; Sin embargo, los problemas numéricos pueden ocurrir con frecuencia frecuentemente en la agrupación GK estándar cuando el número de datos en algún grupo es pequeño o cuando los datos dentro de un grupo están casi linealmente correlacionados. En tales casos, la matriz de covarianza de clúster se vuelve singular y no puede invertirse para calcular el determinante de matriz así como la matriz inductora de normas. Babuska en [19] presentó un método para superar el problema de la singularidad mediante la fijación de la relación entre los valores propios máximos y mínimos de la matriz de covarianza. El otro problema es la superposición en la que algunos grupos se forman extremadamente largos en la dirección de los autovalores más grandes. En este caso, la matriz de covarianza no refleja la distribución real de los datos y, en consecuencia, se obtendrá un modelo deficiente. La siguiente modificación para el algoritmo GK se ha propuesto en [19] para superar este problema:

$$F_i^{new} = (1 - \gamma) F_i + \gamma \det(F_0)^{1/n} I \quad (22)$$

Donde ∈ $[0,1]$ es un parámetro de ajuste y F_0 es la matriz de covarianza de todo el conjunto de datos. El parámetro de escala γ debe ser lo suficientemente grande para superar el problema de ajuste excesivo. Por otra parte, cuando está cerca de 1, todas las matrices de covarianza convergen a la misma matriz $\det(F_0)^{1/n}\ I$. En este caso, GK-clustering se reduce a K-significa agrupación en la que no se toma en cuenta la matriz de covarianza.

Un método de agrupación fuzzy particional basado en las distancias cuadráticas adaptativas se ha introducido en [20]. Utiliza el método de Babuska [19] para superar la singularidad. Aunque este método es capaz de modificar las distancias de agrupación de forma adaptativa en cada iteración, sufre del problema de sobre equipamiento para la aplicación de la estimación de parámetros de distribución K, especialmente cuando se experimenta con un número pequeño de muestras. Además, utiliza la distancia euclidiana que tiene una limitación para capturar una distancia justa debido al hecho de que la correlación de conjunto de datos no se considera.

Para beneficiar la expresividad de la distancia de Mahalanobis, así como superar la singularidad y la sobre ejecución, Karimoddini et al. En [21], propuso una mejora en la ecuación (22) para superar el problema de ajuste excesivo, así como para mantener la capacidad del algoritmo GK-clustering. La idea principal es que el parámetro de escala tiene que ser aumentado cuando el número de datos en un grupo es demasiado pequeño para hacer frente al problema de sobre equipamiento mientras que es necesario disminuir cuando el cluster es rico suficiente para mantener el efecto de la matriz de covarianza en la distancia de Mahalanobis. Por lo tanto, el parámetro de escala se requiere actualizar sujeto al valor de la contribución de datos en cada grupo como:

$$\rho_i = \frac{\sum_{i=1}^{N} u_{ij}}{N} \times 100 \quad (23)$$

Donde ρ_i representa el valor de la contribución (densidad de presencia o probabilidad) del $i-th$ grupo de datos, u_{ij} denota el valor de pertenencia de los $j-th$ grupos de datos $i-th$ y N es número de todos los datos.

Ahora, la ecuación (22) se puede modificar a:

$$F_i^{new} = (1-\gamma^{\rho_i})F_i + \gamma^{\rho_i} \det[F_0]^{1/n} I \quad (24)$$

Donde $\gamma \in [0,1]$ se elige para estar cerca de 1 (por ejemplo $\gamma = 0.9$). De esta manera, el algoritmo GK-clustering se facilita para realizar la agrupación independientemente del número de datos en cada clúster.

3 RESULTADOS

En estudios realizados se debe destacar que desde que comenzó la utilización de estos algoritmos difuso se han ido modificando y surgiendo otro mejorando dificultades de los anteriores, contextualizándolo a resolver un problema dado. De ahí la gran gamificación de estos algoritmo y fusión de los mismos. Por ejemplo se ha discutido que FCM tiene una limitación de tendencia de partición igual para conjuntos de datos, resultado de agrupación óptima de algoritmo FCM podría no ser válido para la demarcación de conjuntos de datos. A partir de lo planteado surgen modificaciones de este algoritmo dando lugar al WFCM que siguiendo con las idea del FCM, el algoritmo FCM ponderado (WFCM) utilizando la distancia euclídena e introduciendo la densidad de la muestra como el coeficiente de peso obtiene gran mejora en la agrupación. Con este algoritmo han surgidos otras modificaciones, se ha propuesto un nuevo algoritmo de C-means fuzzy ponderado alternativo (AWFCM) en vista de una nueva métrica y el peso basado en la densidad de los puntos para reducir la debilidad del FCM. Otras modificaciones a mismo algoritmo NERFCM Ponderado (WNERFCM) consiste en reducir el conjunto de datos original a uno más pequeño, asignando a cada dato seleccionado un peso que refleje el número de datos cercanos, agrupando el conjunto de datos reducido ponderado utilizando una versión ponderada del algoritmo de FCM de datos de características o relacionales y, los datos reducidos vuelven a los datos originales. Se proporcionan varios métodos para cada una de las tareas de selección de subconjuntos de datos, asignación de peso y extensión de los resultados de agrupación ponderada.

Fuzzy C-Means y K-Means: son algoritmos muy similares en el concepto de encontrar un número predefinido de centroides. La principal diferencia está en que K-Means sólo entrega la pertenencia discreta. FCM entrega una pertenencia por cluster, no obstante, ella se puede discretizar considerando que cada dato pertenece a su centroide más cercano. En términos de calidad de clustering, FCM discretizado y K-Means son muy similares. La principal ganancia de FCM está en el modelamiento de la densidad del espacio de los datos. Si sólo se desea una partición discreta del espacio, es más eficiente utilizar el algoritmo de K-Means.

4 CONCLUSIONES

Podemos considerar que el proceso de clustering emula una de las funciones básicas del hombre, su capacidad de agrupar objetos. De ahí su importancia y la extensa literatura publicada al respecto. Uno de los algoritmos de clustering más conocido

es C-Means. Este trabajo se enfoca en el área de algoritmos de agrupamiento difuso, donde cada objeto puede pertenecer a más de un grupo a la vez. En particular se trabaja con los algoritmos de K-Means, C-Means Clásico, Fuzzy C-Means y GustafsonKessel.

Con los algoritmos planteados, hemos podido comprobar algunas respuestas a preguntas como: ¿Hay atributos que sean más relevantes que otros para la separación de clusters?, ¿existen atributos irrelevantes para el proceso de clustering?, ¿existen atributos redundantes para el proceso de clustering?, ¿cambia la calidad (validez) del clustering resultante?

Hay atributos que resultan ser más importantes que otros para el proceso de clustering, los cuales obtienen pesos más altos de manera consistente en el tiempo. Se puede plantear además que los atributos que tienen pesos nulos o muy bajos dentro de un conjunto son potenciales atributos irrelevantes, es decir, el resultado de correr el mismo algoritmo sin esos atributos ha de ser el mismo. La tercera pregunta plantea la existencia de 2 o más atributos que están correlacionados de tal forma de que su contribución o relevancia al proceso de clustering sea igual. Es decir, si solo se dejara uno de esos atributos redundantes, el resultado del clustering debiera mantenerse o incluso mejorar.

No existen actualmente esquemas o algoritmos que permitan encontrar pesos de atributos y la cantidad de clusters de manera simultánea. El poder trabajar con menos atributos o con una representación de datos mediante menos atributos tiene importantes aplicaciones computacionales para efectos de procesamiento y análisis. De ahí la importancia de buscar eliminar atributos irrelevantes, o reducir la cantidad de atributos redundantes.

Estos algoritmos de agrupamiento difuso pueden ser aplicados en diferentes áreas del conocimiento como: organización y clasificación de datos, reconocimiento de patrones, estudio del clima, diagnóstico de enfermedades, bioinformática, genética, cancelación de ruido e interferencia de una señal, estudio de series de tiempo, estudio de la rentabilidad económica de una empresa, soporte de la decisión, segmentación de mercados y clientes (weber).

REFERENCIAS

[1] ROMERO, Fernando Alexis Crespo; PARA OPTAR AL TÍTULO, Memoria. Agrupamiento dinámico con lógica difusa. 2001. Tesis Doctoral. Master's thesis, Santiago de Chile.

[2] SARMIENTO, Henry O., et al. Agrupamiento difuso en el monitoreo térmico de líneas de transporte de energía eléctrica. Revista Politécnica, 2015, vol. 8, no 14, p. 49-55.

[3] Ali, A., Karmakar, G. and Dooley, L.; Review on Fuzzy Clustering Algorithms, IETECH Journal of Advanced Computations, vol. 2 (no. 3), pp. 169–181, 2008.

[4] Marroquin, J., & Girosi, F.; Some extensions ofthe k-means algorithm for image segmentation andpattern recognition. (AI Memo 1390). Massachusetts Institute of Technology, Cambridge, MA, 1993.

[5] COFRE, Sebastian Beca. Clustering difuso con selección de atributos. 2007. Tesis Doctoral. Tesis de Maestría, Universidad De Chile, Facultad De Ciencias Físicas y Matemáticas

[6] CONTRERAS, J.; MISA, R.; URUETA, L. Algoritmos para identificación de modelos difusos interpretables. IEEE Latin American Transactions, 2007, vol. 5, no 5.

[7] BROUWER, Roelof K. A method for fuzzy clustering with ordinal attributes. International Journal of Intelligent Systems, 2007, vol. 22, no 6, p. 599-620

[8] PELUFFO ORDOÑEZ, Diego Hernán, et al. Estudio comparativo de métodos de agrupamiento no supervisado de latidos de senales ECG. 2009. Tesis Doctoral. Universidad Nacional de Colombia-Sede Manizales.

[9] TOSCANO, Ruth; AROBA, Javier; PEREGRÍN, Antonio. UNA MÉTODOLOGÍA PARA GENERAR BASES DE CONOCIMIENTO DIFUSAS BASADA EN AGRUPAMIENTO DIFUSO, SELECCIÓN Y AJUSTE EVOLUTIVOS

[10] DIAZ, Jeronimo Rojas; PORRAS, Julio Cesar Chavarro; LAVERDE, Ricardo Moreno. Tecnicas de logica difusa aplicadas a la mineria de datos. Scientia et technica, 2009, vol. 3, no 4

[11] Bezdek, J. C. Pattern Recognition with Fuzzy Objective Function Algorithms. Kluwer Academic Publishers, Norwell, MA, USA, 1981.

[12] Ali, A., Karmakar, G. and Dooley, L.; Review on Fuzzy Clustering Algorithms, IETECH Journal of Advanced Computations, vol. 2 (no. 3), pp. 169–181, 2008.

[13] R. Babuska, Fuzzy Modeling for Control, Boston, MA: Kluwer, 1998.

[14] J.C. Dunn, A fuzzy relative to the ISODATA process and its use in detecting compact, well-separated clusters, Journal of Cybernetics 3 (1974) 32–57.

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

[16] K. Jajuga, L1-norm based fuzzy clustering, Fuzzy Sets and Systems 39 (2001) 63–83

[17] D.E. Gustafson, W.C. Kessel, Fuzzy clustering with a fuzzy covariance matrix, in: Proceedings of IEEE CDC, San Diego, CA, USA, 1979, pp. 761–766.

[18] P.C. Mahalanobis, On the generalised distance in statistics, Proceedings of the National Institute of Sciences of India 2 (1) (1936) 49–55 Retrieved on 2008-11-05.
[19] R. Babuska, P.J. van der Veen, U. Kaymak, Improved covariance estimation for Gustafson–Kessel clustering, in: Proceedings of the 2002 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE apos, vol. 2, 2002, pp. 1081–1085.
[20] F.de A.T. de Carvalho, C.P. Tenório, N.L. Cavalcanti Jr, Partitional fuzzy clustering methods based on adaptive quadratic distances, Fuzzy Sets and Systems 157 (21) (2006) 2833–2857.