MetaCLAS: A Prototype Evolutionary Proposal to Automatically Suggest Clustering Methods and their Parameters

Macarena A. Latini, Rocío L. Cecchini, and Jessica A. Carballido

Instituto de Ciencias e Ingeniería de la Computación (UNS-CONICET),
Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur
San Andrés 800, Bahía Blanca, Argentina
{macarena.latini,rlc,jac}@cs.uns.edu.ar

Resumen Uno de los principales problemas al que nos enfrentamos al momento de realizar agrupamiento de datos consiste en elegir cuál es el mejor método de clustering para clasificarlos, y cuál es la cantidad ideal (k) de *grupos* en los que se deberían separar esos datos. En este trabajo presentamos una primera aproximación de un método que, a partir de un conjunto de datos estandarizados, sugiere el método de clustering y el valor de k que mejor los agrupa. Para esto considera cuatro índices de evaluación de la estructura final de clusters: *Dunn*, *Silueta*, *Entropía* y *Widestgap*. El algoritmo está implementado como un algoritmo genético en el cual los individuos son posibles configuraciones de métodos de clustering y sus parámetros. En este primer prototipo, el algoritmo sugiere entre los métodos de partición K-means, PAM, CLARA y Fanny. Asimismo, además de sugerir el método que presentó mejor desempeño, también se obtiene como resultado el valor de los parámetros para ejecutarlo. El prototipo fue desarrollado en un entorno de R y se pudo corroborar que sus resultados son consistentes con una combinación de resultados provistos por otros métodos con objetivos similares. La idea de este trabajo es que sirva de base inicial para un desarrollo que incorpore opciones para reducción de la matriz de datos, evaluación de más métodos de agrupamiento y optimización de los operadores genéticos del algoritmo.

Keywords: algoritmos genéticos, clustering particional, computacion evolutiva

1. Introducción

Este artículo presenta una propuesta, basada en algoritmos evolutivos, para la resolución de uno de los principales problemas conocidos en el ámbito del *Clustering*, el cual consiste en la identificación de los métodos más adecuados para llevar a cabo el agrupamiento no supervisado de datos y la estimación de sus correspondientes parámetros. Además, el artículo pretende ser una guía básica de pasos a seguir al momento de realizar *Clustering* de datos, por lo que

se inicialmente mencionarán algunos aspectos ampliamente conocidos en el área.

¿Qué es el Clustering? El clustering o agrupamiento de datos consiste en la tarea de encontrar grupos de objetos de tal forma que quienes pertenecen a un grupo se parezcan más entre sí de lo que se parecen con objetos de otros grupos. La definición formal de *Clustering* (o agrupamiento), dependerá inevitablemente de la definición formal de *Cluster* y ambos términos parecen estar definidos de diversas maneras en la literatura. En [5] podemos encontrar una definición bastante simple para *Clustering*: *"regiones contínuas del espacio que contienen una densidad relativamente alta de puntos, separadas de otras regiones del espacio por regiones cuya densidad de puntos es relativamente baja"*. Si bien esta descripción parece bastante natural, cuantificar la cercanía de dichos puntos dependerá de los tipos de clusters presentes en el conjunto de datos sobre el que se trabaja. Ya que los datos pueden agruparse en formas compactas, alargadas o formando algún tipo de trazo dentro de cada cluster.

Interpretación y preparación de los datos. Como medida de interpretación general, en este artículo vamos a considerar que cada fila de la matriz de datos que se va a procesar es una *observación*, mientras que cada columna es una *variable*. Es decir, si la matriz de datos tiene dimensión $n \times m$, estaremos trabajando con n *observaciones* y m *variables*. Es importante tener en cuenta que los datos deben ser *estandarizados* previo a la aplicación del método de *clustering*, para lograr que las variables sean comparables entre sí.

2. Antecedentes

2.1. Conceptos básicos

Existen varias cuestiones a tener en cuenta al momento de elegir la estrategia con la cual se llevará a cabo el agrupamiento. Por ejemplo, si se tratará de un agrupamiento estricto, en cuyo caso cada elemento pertenecerá a un único grupo, o no. Si se podrá aplicar un método supervisado, para el cual se contará con casos etiquetados, o si sólo se podrá aplicar un método no supervisado.
Uno de los factores más importantes a considerar es cuál será la métrica que se usará para medir la distancia. El agrupamiento de las observaciones requerirá de algún método que permita evaluar la distancia (o similitud/dissimilitud) entre los elementos que se están intentando agrupar (en nuestro caso observaciones). Por medio de estas distancias se podrá conformar una matriz de distancias que es requerida por algunos de los métodos de clustering. Dos de las métricas más comunes para medir la distancia entre dos observaciones x e y son:

- *Distancia Euclídea*, definida como: $D_e = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$
- *Distancia Manhattan*, definida como: $D_m = \sum_{i=1}^{n} |(x_i - y_i)|$

En ambos casos x_i e y_i corresponden respectivamente a la i-ésima componente de los vectores x e y de longitud n. Existen otras métricas para medir distancia, como *Pearson*, *Kendall* o *Spearman* [17,18,10,11,21], las cuales están basadas en *correlación*.

Matriz de Distancia. Una vez definida la métrica de distancia a utilizar se puede calcular la *matriz de distancia*, M_d, de $n \times n$, tal que la componente $M_d[i, j]$ indicará la distancia (o similitud/dissimilitud) entre el vector (fila) i y el j de nuestra matriz original. Como es de suponer, esta matriz será simétrica y con *ceros* (o *unos*) en la diagonal. No todos los métodos utilizan esta matriz.

2.2. Estrategias clásicas para realización de Clustering

Como mencionamos anteriormente, las estrategias clásicas para intentar identificar las características de los grupos de datos subyacentes dentro de un conjunto de datos particular se dividen en dos grupos principales: *Métodos jerárquicos* y *Métodos no jerárquicos*.

Métodos jerárquicos. Estos métodos permiten trabajar con distintos tipos de variables y son útlies cuando no se conoce previamente el número de clusters, siempre y cuando el conjunto de datos no sea muy grande. Estos algoritmos pueden ser, a su vez, *Aglomerativos*, en los que en cada etapa se van agrupando los clusters calculados hasta el momento de tal forma de obtener un número menor de clusters, o *Divisivos*, que trabajan de manera inversa. En ambos casos se trabaja minimizando alguna función de distancia (o maximizando alguna función de similitud), para lo cual se utiliza la matriz de distancias (o similitudes).

Métodos no jerárquicos (o de particionamiento). Estos métodos, a diferencia de los *jerárquicos*, trabajan procurando alcanzar la mejor partición posible de los datos para un número dado de clusters (k) . Este número debe ser determinado de manera previa a la ejecución del algoritmo. En general, estos métodos no trabajan sobre una matriz de distancia sino sobre los datos originales.
Existe una gran variedad de algoritmos no jerárquicos, entre los más comunes podemos mencionar *K-means* [14], *K-medoids* (PAM) [9], *CLARA*, *DBSCAN* [4]. A continuación se explicarán de forma resumida en qué consisten los métodos que se utilizaron en este trabajo.

K-means [14]. Este método, propuesto en 1967, es uno de los algoritmos no supervisados más utilizados para realizar el particionamiento de un conjunto de datos en k subconjuntos. La idea básica consiste en lograr determinar grupos de objetos de tal forma que se minimice la variación total dentro de los clusters (*within-cluster variation*) medida por el error cuadrado total (E_d) de la siguiente manera: $E_d = \sum_{i=1}^{k} \sum_{j=1}^{n_i} d(x_{i_j}, c_i)^2$, donde, d es la métrica de distancia utilizada, k es el número de clusters, x_{i_j} es la j-ésima obervación del cluster i, y n_i y c_i son la cardinalidad y el controide del cluster i.
La forma y el momento en que se recalcula dicho centroide ha dado origen a distintas variantes del algoritmo, como *MacQueen* [14], *Forgy/Lloyd* [6,13] y *Hartingan* [7].

PAM [9,12]. PAM proviene de Partitioning Around Medoids. Este método, propuesto en 1987, es una de las implementaciones más conocidas del algoritmo K-medoids. El objetivo del método es encontrar una secuencia de objetos (o

medoides) tales que se encuentren ubicados de manera central en los grupos o *clusters*. Es más robusto ante el ruido y *outliers* que K-means porque minimiza una suma de disimilaridades (entre pares de puntos) en vez de una suma de distancias euclídeas cuadradas. Algunos resultados experimentales han mostrado que PAM funciona relativamente bien para conjuntos de datos pequeños [12], pero se vuelve ineficiente con conjuntos de datos grandes debido a que el tiempo de ejecución de cada iteración aumenta en forma cuadrática con N [21].

CLARA [12,21]. En el método clara (Clustering LARge Aplications) fue desarrollado con el objetivo de resolver los problemas de PAM frente a conjuntos de datos grandes. La idea subyacente de CLARA es procesar aleatoriamente una muestra X_i de tamaño N_i del conjunto de datos entero, X, y determinar el conjunto de *medoides*, θ_i que representa mejor a X_i utilizando PAM. CLARA utiliza PAM sobre un número de subconjuntos de X, denotados como $X_1, ..., X_m$, de tal forma que en cada ejecución de PAM se obtiene un conjunto de *medoides*, $\theta_1, ..., \theta_m$. Luego, se evalúa la calidad del agrupamiento asociado a cada uno de los conjuntos θ_i teniendo en cuenta el conjunto de datos entero. Trabajando de esta manera, el algoritmo logra requisitos de tiempo y almacenamiento lineales.

Fanny [12]. Fanny (Fuzzy Analysis clustering) es un algoritmo de particionamiento difuso ("fuzzy"). Es decir, a diferencia del resto de algoritmos vistos hasta ahora, las observaciones no se clasifican en un único cluster sino que se les asigna las probabilidades de pertenecer a cada grupo. Estas probabilidades no pueden ser negativas y deben sumar 1. Si u_{ij} es la probabilidad del elemento i de pertenecer al cluster j, el objetivo del algoritmo Fanny es minimizar la función $\sum_{v=1}^{k} \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} u_{iv}^r u_{jv}^r d(i,j)}{2\sum_{j=1}^{n} u_{jv}^r}$, donde n es el número de observaciones, $d(i,j)$ es la disimilitud entre las observaciones i y j y k es el número de clusters. El valor r se denomina *exponente de membresía o pertenencia* y si es cercano a 1 produce como resultado grupos menos difusos, mientras que si tiende a infinito se logra una difusión completa.

2.3. Validación de los Clusters

Si bien es claro que existen algunas características básicas que se desea alcanzar con el agrupamiento encontrado, como por ejemplo que la densidad entre los elementos del grupo sea superior a la densidad entre esos elementos y otros elementos externos, no es trivial evaluar qué tan bueno es dicho agrupamiento de la manera más adecuada posible. Además, se debe tener en cuenta que la bondad de cada grupo será relativa a la aplicación o problema que se está intentado resolver. Existe un gran número de índices que permiten evaluar diferentes aspectos del resultado de un algoritmo de clustering y, si bien se han clasificado de varias formas [1,21,2], la categorización más ampliamente utilizada es la que los agrupa en *internos* y *externos* [1,21]. La principal diferencia está en si la medida usa o no información externa para la validación, es decir, información que no es producto de la técnica de clustering. En general, de una u otra manera todas las medidas internas buscan analizar dos características principales de la

estructura: *cohesión* y *separación*. La primera busca que el miembro de cada cluster sea lo más cercano posible a los otros miembros del mismo cluster y la segunda apunta a tener clusters ampliamente separados. Los índices de validación interna más utilizados son *Dunn* y *Silueta*. En este trabajo, se eligieron dos índices adicionales, *Entropía* y *Widestgap*, que permiten un análisis más global del resultado final del algoritmo.

Dunn [3,15]. El objetivo principal de este índice es dar un valor sobre la cohesión de los clusters analizando la varianza entre los miembros del cluster y la separación entre los clusters. La distancia entre los miembros de un cluster debe ser lo más pequeña posible y entre los clusters lo más grande posible. Su valor va desde cero hasta infinito y se calcula como:

$$D = \min_{1 \leq i \leq n} \left\{ \min_{1 \leq j \leq n, i \neq j} \left\{ \frac{d(i,j)}{\max_{1 \leq k \leq n} d'(k)} \right\} \right\} \quad (1)$$

donde $d(i,j)$ representa la distancia entre los clusters i y j, y $d'(k)$ mide la distancia dentro del cluster k.

Silueta [15,19]. Este índice se utiliza para evaluar la compactitud (inter-cluster) y separabilidad (intra-cluster) de una estructura de clusters. El coeficiente de *Silueta* para un agrupamiento se calcula a partir del valor $s(i)$ de cada elemento de la siguiente manera:

$$S = \frac{1}{N} \sum_{i=1}^{N} s(i) \quad (2)$$

donde $s(i)$ se calcula a partir de $a(i)$ (la distancia media entre el objeto y todos los otros objetos del mismo cluster) y $b(i)$ (la distancia media entre el objeto y todos los otros objetos del cluster más próximo) como: $s(i) = \frac{b(i)-a(i)}{max(a(i),b(i))}$. Un valor de $s(i)$ cercano a cero indica que el objeto i está en la frontera de dos clusters. Un valor de $s(i)$ cercano a uno indica que el elemento se ha agrupado adecuadamente y, por el contrario, si el valor de $s(i)$ es negativo, es probable que el elemento haya sido incorrectamente agrupado.

Entropía. El tercer índice analizado mide la entropía de la distribución de los elementos de un cluster. En todos los ámbitos en que se analice, la entropía se concibe como una *medida del desorden*. Por esta razón, cuánto más chico sea este valor, mejor será la calidad de la estructura del clustering. Si bien esta medida suele ser utilizada como una herramienta para comparar dos *agrupamientos* diferentes, por medio de la medición de la *información mutua* existente entre los mismos, en este artículo sólo se utiliza como una forma de evaluar el desorden de un agrupamiento particular, funcionando como una métrica de validación interna. Según la definición de *entropía asociada a un agrupamiento C* dada en [16] se calcula como:

$$H(C) = -\sum_{k=1}^{K} P(k) log P(k) \quad (3)$$

donde K es el número de clases del agrupamiento C y $P(K)$ es la probabilidad de que un elemento pertenezca al grupo k, dada por $P(k) = \frac{n_k}{n}$, donde n_k es

el número de elementos en el grupo k y n es el número de observaciones en el conjunto de datos.

Widestgap *[22,8].* Finalmente, la cuarta medida considerada calcula la brecha más grande dentro de cada cluster, y retorna la mayor de todas. El valor del índice de *widesgap* para un cluster C se calcula como:

$$Wg = \max_{c \in C} wg(c), \quad donde \quad wg(c) = \max_{\substack{c \in C \\ para\ D,E: \\ D \cup E = C \\ D \cap E \neq \emptyset}} \left\{ \min_{\substack{x \in D \\ y \in E}} d(x,y) \right\} \tag{4}$$

Como puede verse en esta ecuación, una vez calculados los valores de los espacios (o brechas) más grandes encontrados dentro de cada cluster (wg_c), Wg se define como el máximo de dichos espacios. Este espacio debería ser minimizado pues implica clusters más compactos.

3. Algoritmo propuesto: MetaCLAS

MetaCLAS es un algoritmo genético implementado en R que, a partir de una matriz de datos, propone un método de clustering y sus correspondientes parámetros. Hasta ahora, los posibles métodos son K-means, PAM, CLARA y Fanny, cuya elección depende de la calidad de los resultados obtenidos para la matriz de entrada. Los índices utilizados para evaluar la calidad del resultado del método de clustering son: Dunn, Silueta, Entropía y Widestgap.

Representación de los individuos. Cada individuo se compone por los campos "***MC***, **K**, ***Algoritmo***". ***MC*** es un entero que representa el método de clustering (K-means, PAM, CLARA o Fanny). **K** es la cantidad de clusters a obtener (fanny, clara) o una lista de centroides (K-means) o una lista de medoides (PAM) según corresponda para el método de clustering elegido en ***MC***. Por último el campo ***Algoritmo*** guarda, en el caso de K-means, una referencia al nombre del algoritmo (*Hartigan-Wong*, *Lloyd*, *Forgy* o *MacQueen*), y en los otros tres casos (PAM, CLARA o Fanny) guarda el método utilizado para calcular la distancia entre dos observaciones.

Al momento de crear el individuo es importante mantener consistencia entre el método de clustering y sus parámetros. Por esto se realizó la validación del k según las restricciones del método siguiendo estas reglas: los métodos K-means, PAM y CLARA requieren que $0 < k < n$. El método Fanny requiere que $0 < k < (n/2) - 1$. Esto es considerado a lo largo de todo el algoritmo para conservar la factibilidad de los individuos.

La población inicial se crea de manera aleatoria respetando en cada caso las restricciones de los individuos, acorde a los parámetros requeridos por cada método. El método de selección utilizado es el de *torneo binario.*

Cruzamiento. Para el cruzamiento de dos individuos, se seleccionan los padres con una probabilidad $PC = 0{,}7$, y se cruzan aleatoriamente usando una de las siguientes opciones:

- Opción 1: Los hijos heredan el valor de k y reciben intercambiado el método de clustering de los padres.

- Opción 2: Los hijos heredan el método de clustering del padre y reciben intercambiado el valor de k.

Este proceso requiere una corrección del valor de k para preservar la propiedad de individuos factibles mencionada anteriormente.

Mutación. Para mutar a un individuo, utilizamos una técnica de reemplazo total. El individuo es seleccionado con una probabilidad $PM = 0{,}2$ y en su posición se genera un individuo nuevo.

Evaluación de la aptitud de cada individuo. La evaluación del fitness de un individuo determinado se hace en dos pasos. En el primer paso, se ejecuta el método correspondiente al individuo, utilizando los parámetros especificados para el mismo, para lo cual se utilizaron los paquetes *stats* y *cluster* de R. Luego, se utiliza la función *cluster.stats* {fpc} para validar el resultado del método. Dicha función calcula varias estadísticas de validez para una estructura de clustering y una matriz de disimilitud. En este caso, de todas las estadísticas que devuelve *cluster.stats*, se analizaron los valores de los índices Dunn, Silueta, Entropía y Widestgap. El objetivo final de la función de aptitud es maximizar los primeros dos índices y minimizar los últimos dos.
Dado que nos encontramos ante un problema de varios objetivos, en esta primera aproximación se decidió utilizar la forma más directa de abordarlo que es mediante una función de agregación. La misma consiste en combinar todas las funciones objetivo $f_i(x)$ en una única función $F(f_1(x), ..., f_k(x))$. La función más utilizada es la combinación o agregación lineal de los objetivos en base a la siguiente ecuación:

$$F = \sum_{i=1}^{k} w_i f_i(x) \tag{5}$$

donde w_i son los pesos de cada función objetivo, siendo común que sean normalizadas, tal que la suma de todos los pesos sea igual a 1. En este trabajo todos los objetivos tienen igual peso. Los objetivos a maximizar se suman y lo demás se restan. Más específicamente, las funciones objetivo fueron determinadas como: $f_1 = D$, $f_2 = S$, $f_3 = -H$ y $f_4 = -W_g$ en base a las ecuaciones dadas en (1), (2), (3) y (4).

4. Evaluación

Para comprobar el desempeño del algoritmo se utilizó un ejemplo tradicional en el análisis de conglomerados. Se trata del conjunto de datos introducido por [20] que está compuesto por 75 observaciones sobre dos variables, x e y.
En la Figura 1 se pueden ver dos posibles soluciones de clustering encontradas con diferentes números de clusters. Se puede ver que en el grupo de cuatro clusters, la separación entre clusters es visualmente reconocible, mientras que a medida que nos movemos al caso de cinco grupos, la interpretación e incluso la definición de los grupos es menos clara.

Figura 1: Posibles agrupamientos encontrados para los datos *Ruspini*, para 4 y 5 grupos respectivamente.

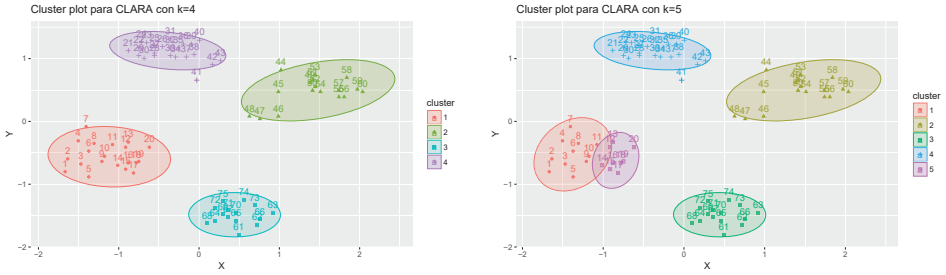

Fig. a Fig. b

4.1. Diseño de experimentos

La experimentación se organizó en 100 corridas independientes del MetaCLAS. Para cada corrida se registra la configuración de clustering sugerida por el mejor individuo utilizando los índices de evaluación presentados en la sección anterior. Una vez finalizadas las corridas, se coteja el resultado utilizando la función *NbClust* del paquete con el mismo nombre. Esta función utiliza 30 índices para determinar el mejor número de clusters. Sin embargo, a diferencia de nuestro método, no propone el algoritmo que arroja los mejores resultados. Sólo realiza el análisis usando un algoritmo no jerárquico, el kmeans, y uno jerárquico, el HAC (Hierarchical Agglomerative Clustering), sin dar la posibilidad de poder modificar estos métodos.

4.2. Análisis de resultados

En la tabla 1 mostramos un resumen de los valores obtenidos en las 100 corridas del MetaCLAS. Cabe destacar que solo una de las 100 veces, el algoritmo sugirió una configuración con 2 clusters. El resto de las corridas sugirieron entre 4 y 5 clusters. Es importante recordar que nuestro método decide cuándo una configuración es mejor que otra en términos de los índices Dunn, Silueta, Entropía y Widestgap. De acuerdo a lo que podemos ver en los resultados, nuestro algoritmo sugiere que utilizando el método CLARA con $k = 5$ obtenemos los mejores valores para los objetivos perseguidos en términos generales. No se ve una gran diferencia entre las métricas utilizadas para este método. Sí podemos ver que el método PAM con distancia Euclídea también demuestra una buena performance, para un $k = 4$. En este punto es cuando se pone en evidencia la importancia de incorporar medidas de evaluación externas que, en vista de estos resultados, ayuden a completar el análisis. Si sólo nos concentramos en el valor de k a sugerir, tendremos como resultado que nuestro algoritmo prefiere estructuras con 4 clusters. Cabe destacar que cada vez que se sugirió $k = 4$, la estructura fue idéntica para todos los casos, mientras que para $k = 5$ había distintas variantes.

Como dijimos anteriormente, validamos los resultados de este caso de estudio con la función *NbClust*. Al invocar dicha función con el dataset de *ruspini* y una

Tabla 1: Algoritmo más adecuado sugerido y número de *clusters* correspondiente para cada ejecución del MetaCLAS. En donde, H-W:Hartigan-Wong, L:Lloyd, F:Forgy, McQ:MacQueen, E:Euclídea, M:Manhattan.

	K-means				PAM		Fanny		CLARA		
	H-W	L	F	McQ	E	M	E	M	E	M	
k=4	5	1	3		14	8	9	8	**8**	**12**	68
k=5					5				**14**	**12**	31
	9				27		17		46		

variación de k desde 2 hasta 8, el resultado obtenido es que entre todos los índices:

- 1 propone que el mejor número de clusters es 2,
- 3 proponen que el mejor número de clusters es 3,
- 6 proponen que el mejor número de clusters es 4,
- 1 propone que el mejor número de clusters es 5,
- 2 proponen que el mejor número de clusters es 8,

y la conclusión a la que llega es que "*de acuerdo con la regla de mayoría, el mejor número de clusters es 4*".
Este resultado nos da dos indicios: el primero es que en términos de la cantidad de clusters, estamos sugiriendo lo mismo que este método que es bien conocido y usado ampliamente en la literatura. El segundo es que elegimos correctamente cuatro índices que resumen las características deseables de una estructura de clusters. En cuanto al método que estamos sugiriendo (clara) no podemos constatar este resultado ya que no existe, hasta donde sabemos, un algoritmo cuyo objetivo sea también proponer el método más adecuado para una estructura determinada.

5. Conclusiones

En este artículo presentamos el algoritmo MetaCLAS, un algoritmo genético cuyos individuos representan distintas configuraciones de métodos de clustering, parámetros de los mismos y distintos valores de k. El algoritmo fue validado con el dataset *Ruspini*, el cual es ampliamente usado en la bibliografía de testeo de clustering. Los métodos entre los cuales nuestro algoritmo sugiere el que mejor se desempeña son los métodos de partición K-means, CLARA, PAM y Fanny. Para seleccionar la mejor configuración de método/parámetros/k, usamos los índices de validación interna *Dunn*, *Silueta*, *Entropía* y *Widestgap*. Luego de realizar 100 corridas independientes de nuestro algoritmo, pudimos comprobar usando el paquete *NbClust* de R que el método propuesto sugiere las mejores configuraciones. Por esto consideramos que el método presentado en este trabajo es un buen punto de partida para una futura implementación en la cual, tomando esta primera implementación como base, vamos a agregar la posibilidad de evaluar distintas reducciones de la matriz como parte del individuo para manejar grandes volúmenes de datos. Además planeamos incluir más métodos de clustering y como objetivo adicional, resta optimizar los operadores de cruzamiento y mutación del algoritmo genético.

Agradecimientos. Agradecemos a CONICET (Consejo Nacional de Investigaciones Científicas y Técnicas), y a los subsidios PIP 112-2012-0100471, y UNS (Universidad Nacional del Sur) PGI 24/N042.

Referencias

1. Charu C. Aggarwal and Chandan K. Reddy. *Data Clustering: Algorithms and Applications*. Chapman & Hall/CRC, 1st edition, 2013.
2. Guy Brock, Vasyl Pihur, Susmita Datta, and Somnath Datta. clvalid: An r package for cluster validation. *Journal of Statistical Software, Articles*, 25(4):1–22, 2008.
3. J. C. Dunn. Well-separated clusters and optimal fuzzy partitions. *Journal of Cybernetics*, 4(1):95–104, 1974.
4. M. Ester, H. P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on KDDM*, pages 226–231. AAAI Press, 1996.
5. Brian S. Everitt, Sabine Landau, and Morven Leese. *Cluster Analysis*. Wiley Publishing, 4th edition, 2009.
6. E. Forgy. Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *Biometrics*, 21:768–780, 1965.
7. J. A. Hartigan and M. A. Wong. A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.
8. Christian Hennig. Cluster validation by measurement of clustering characteristics relevant to the user, 2017. arXiv:1703.09282v1.
9. L. Kaufman and P. J. Rousseeuw. Clustering by means of medoids, 1987.
10. M. Kendall. A new measure of rank correlation. *Biometrika*, 30(1-2):81–93, 1938.
11. Maurice G. Kendall. *Rank Correlation Methods*. Griffin, London, England, 1970.
12. L. Kaufman and P. J. Rousseeuw *Finding groups in data: an introduction to cluster analysis*. Wiley-Interscience, 9th edition, 1990.
13. S. Lloyd. Least squares quantization in PCM. *IEEE Trans. Inf. Theor.*, 28(2):129–137, 1982.
14. J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pages 281–297, 1967.
15. U. Maulik *et al. Multiobjective Genetic Algorithms for Clustering - Applications in Data Mining and Bioinformatics.* Springer, 2011.
16. Marina Meil. Comparing clusteringsan information based distance. *Journal of Multivariate Analysis*, 98(5):873–895, 2007.
17. Karl Pearson. Notes on the history of correlation. *Biometrika*, 13(1):25–45, 1920.
18. J.L. Rodgers and W.A. Nicewander. Thirteen ways to look at the correlation coefficient. *The American Statistician*, 42(1):59–66, 1988.
19. P. J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.
20. Enrique H. Ruspini. Numerical methods for fuzzy clustering. *Inf. Sci.*, 2(3):319–350, July 1970.
21. Sergios Theodoridis and Konstantinos Koutroumbas. *Pattern Recognition, Fourth Edition*. Academic Press, 2009.
22. B. S. Villanueva, K. Gibert, and M. Sànchez-Marrè. Using CVI for understanding class topology in unsupervised scenarios. *CAEPIA*, volume 9868 of *Lecture Notes in Computer Science*, pages 135–149. Springer, 2016.