Análisis de medidas no-supervisadas de calidad en clusters obtenidos por K-means y Particle Swarm Optimization

Andrea Villagra, Ana Guzmán, Daniel Pandolfi
Universidad Nacional de la Patagonia Austral, Unidad Académica Caleta Olivia,
Laboratorio de Tecnologías Emergentes (LabTEm)
Caleta Olivia, Santa Cruz, Argentina, 9011
{avillagra,aguzman,dpandolfi}@uaco.unpa.edu.ar

y

Guillermo Leguizamón
Universidad Nacional de San Luis,
Laboratorio de Investigación y Desarrollo en Inteligencia Computacional (LIDIC)
San Luis, Argentina, 5700
legui@unsl.edu.ar

Abstract

Data clustering helps in discerning the structure and simplifing the complexity of massive quantities of data. It is a common technique used in many fields, including machine learning, data mining, image analysis, and bioinformatics, in which the distribution of information can be of any size and shape. The efficiency of clustering algorithms is strongly required with very large databases and high-dimensional data types. This paper presents an evaluation study, from different perspectives, of several important unsupervised quality measures including quantization error, intra- and inter-cluster distances, obtained by the well-known K-means algorithm and a population-based metaheuristic called Particle Swarm Optimization (PSO) and a hybrid algorithm that combines the characteristics of both algorithms, called PSO+K-means. Results show that in general the PSO+K-means algorithm obtains better results in each measure and generates higher compact and separates clustering than either PSO or K-means alone.

Keywords: Clustering, K-means, Particle Swarm Optimization, Unsupervised quality measures.

Resumen

El *clustering* de datos ayuda a discernir la estructura y simplifica la complejidad de cantidades masivas de datos. Es una técnica común y se utiliza en diversos campos como, aprendizaje de máquina, minería de datos, reconocimiento de patrones, análisis de imágenes y bioinformática, donde la distribución de la información puede ser de cualquier tamaño y forma. La eficiencia de los algoritmos de *clustering* es extremadamente necesaria cuando se trabaja con enormes bases de datos y tipos de datos de grandes dimensiones. Este trabajo presenta una evaluación desde distintas perspectivas de una serie de medidas relevantes no-supervisadas de calidad como por ejemplo, cuantización del error, distancias intra- e inter- cluster, de los *clusters* obtenidos por el conocido algoritmo de *K-means*, una metaheurística poblacional denominada Particle Swarm Optimization (PSO) y un algoritmo híbrido, que combina las características de los dos algoritmos anteriores, denominado PSO+*Kmeans*. De los resultados obtenidos se observa que en general el algoritmo PSO+*K-means* obtiene mejores resultados en cada una de las medidas generando *clusters* más compactos y separados entre ellos que los obtenidos por los otros algoritmos.

Palabras claves: Clustering, K-means, Particle Swarm Optimization, Medidas no-supervisadas.

1. Introducción

Los datos son una fuente crítica en varias organizaciones y por lo tanto la eficiencia de acceso a ellos, el compartir, extraer información y hacer uso de esa información se ha convertido en una imperiosa necesidad. Actualmente, la minería de datos conforma uno de los campos de investigación y aplicación más reconocidos para llevar a cabo dichas tareas. En términos generales, podemos decir que la minería de datos es el proceso de extraer información útil, patrones y tendencias previamente desconocidas, de grandes bases de datos. La minería de datos ha recibido un gran impulso en los últimos tiempos motivado por distintas causas: a) el desarrollo de algoritmos eficientes y robustos para el procesamiento de grandes volúmenes de datos, b) un poder computacional más barato que permite utilizar métodos computacionalmente intensivos, y c) las ventajas comerciales y científicas que han brindado este tipo de técnicas en las más diversas áreas.
Clustering es una de las tareas más utilizadas en el proceso de minería de datos para descubrir grupos e identificar interesantes distribuciones y patrones en los datos. Es un método exploratorio que ayuda a resolver problemas de clasificación. Su uso es adecuado cuando se conoce poco o nada sobre la estructura de los datos. El objetivo del *clustering* es organizar una muestra de casos en consideración en grupos de forma tal que el grado de asociación es alto entre los miembros del mismo grupo y bajo entre los miembros de diferentes grupos. También se llama al *clustering* clasificación no-supervisada, donde no hay clases predefinidas [7].
Hay dos técnicas principales de *clustering* "Particional" y "Jerárquica" [1], [2]. En las ultimas décadas, debido al desarrollo de la inteligencia artificial se han presentado métodos de *clustering* basados en otras teorías o técnicas [9], [8].
Un algoritmo de *clustering* muy conocido es el algoritmo de *K-means* y sus variantes. Este algoritmo es simple, directo y se basa en análisis de varianzas. La principal desventaja del algoritmo de *K-means* es que el *cluster* resultante es sensible a la selección inicial de los centroides y puede converger a un óptimo local. Por lo tanto, la selección inicial de los centroides dirige el proceso de *K-means* y la partición resultante está condicionada a la elección esos centroides. *K-means* realiza una búsqueda local en la vecindad de la solución inicial y va refinando la partición resultante, por esta razón se puede utilizar algún algoritmo de búsqueda global para generar los centroides iniciales. El algoritmo de cúmulo de partículas (PSO - *Particle Swarm Optimization*) es una técnica de optimización estocástica que puede utilizarse para encontrar una solución óptima o cercana al óptimo, ha sido aplicado en *clustering* de datos y de texto con muy buenos resultados [6],[10], [3]. PSO puede utilizarse para generar buenos centroides iniciales para el *K-means*. En este trabajo se utiliza un algoritmo híbrido PSO+K-means que combina la búsqueda global de PSO y la búsqueda local de *K-means*, obteniendo mejoras en las medidas de calidad analizadas.
El resto del trabajo esta organizado de la siguiente manera, en la sección 2 se describe el algoritmo de *K-means*, en la sección 3 se muestra el algoritmo de PSO, en la sección 4 se describe la aplicación de PSO a *clustering* y el algoritmo híbrido PSO+*K-means*, en la sección 5 de muestran las medidas de calidad utilizadas, en la sección 6 se detallan los experimentos y resultados obtenidos y por último en la sección 7 las conclusiones.

2. K-means

El algoritmo de *K-means* fue propuesto por MacQueen en el año 1968 [5] es simple, directo y está basado en el análisis de las varianzas. Agrupa un conjunto de datos en un número predefinido de *clusters*. Comienza con un conjunto aleatorio de centroides de cada uno de los *clusters* y continúa reasignando los datos del conjunto de datos a los centroides, basándose en la similitud entre el dato y el centroide. El proceso de reasignación no se detiene hasta que se converge al criterio de parada (por ejemplo, se alcanzó un número fijo de iteraciones o los *clusters* encontrados no cambian luego de cierto número de iteraciones).
El algoritmo de *K-means* puede resumirse de la siguiente manera:

1. Selecciona un conjunto aleatorio de centroides iniciales.
2. Asigna cada elemento del conjunto de datos al centroide más cercano.
3. Recalcula los centroides usando:
$$c_j = \frac{1}{\mid C_j \mid} \sum_{\forall x \in C_j} z \tag{1}$$
donde z representa un elemento del conjunto de datos, que pertenece al *cluster* C_j; c_j es un centroide y $\mid C_j \mid$ corresponde al número de elementos en el *cluster* C_j.
4. Repetir los pasos 2 y 3 hasta que se alcance la condición de parada.

Una desventaja de este algoritmo es que el resultado obtenido es dependiente de la selección inicial de los centroides de los *clusters* y puede converger a óptimos locales [11]. Por lo tanto, la selección de los centroides iniciales

afecta el proceso principal de *K-means* y la partición resultante de este proceso. No obstante, si se obtienen buenos centroides iniciales con alguna técnica alternativa, *K-means* refinaría esos centroides de los *clusters* obteniendo mejores resultados.

3. Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) es una técnica de computación evolutiva desarrollada por Kennedy y Eberhart en 1995 [4], inspirada en el comportamiento social del vuelo de las bandadas, movimiento de los cardúmenes, entre otros sistemas sociales altamente cohesionados y que permite, simulando este modelo de comportamiento, obtener métodos eficientes para resolver problemas de optimización. En el algoritmo, los cardúmenes se representan simbólicamente como partículas. Estas partículas se consideran simples agentes que se "mueven" a través de un espacio de problema. La ubicación de una partícula en un espacio de problema multi-dimensional representa una posible solución. Cuando la partícula se mueve a una nueva ubicación, se genera una nueva solución. Esta solución se evalúa a través de una función de aptitud o *"fitness"* proveyendo un valor cuantitativo de la utilidad de esa solución. La velocidad y la dirección de cada partícula se ajusta de acuerdo a su propia experiencia p_{id} y a la experiencia de sus vecinos p_{gd}. Los valores aleatorios r_1 y r_2 y los coeficientes c_1 y c_2 controlan la influencia estocástica de las componentes cognitivas y sociales sobre la velocidad de la partícula. Para cada generación, la nueva ubicación de la partícula se calcula agregando a la posición actual de la partícula, vector x, la nueva velocidad, vector v. Matemáticamente, en un espacio de problema multi-dimensional, la $i-$ésima partícula cambia su velocidad de acuerdo a la siguiente ecuación:

$$v_{id} = w * v_{id} + c_1 * r_1 * (p_{id} - x_{id}) + c_2 * r_2 * (p_{gd} - x_{id}) \tag{2a}$$

$$x_{id} = x_{id} + v_{id} \tag{2b}$$

donde w es un factor de peso de inercia; p_{id} es la memoria de la partícula, es decir su mejor posición; p_{gd} es la mejor posición global, es decir la ubicación de la partícula con el mejor valor de aptitud; c_1 y c_2 son los coeficientes de aceleración; d corresponde a la dimensión del espacio del problema; r_1 y r_2 son valores aleatorios en el rango $(0, 1)$. La ecuación 2a implica que cada partícula debe registrar su ubicación actual x_{id}, su velocidad v_{id} y los vectores p_{id} y p_{gd} representan los mejores valores de aptitud encontrados. Estos valores de aptitud se actualizan en cada generación, de acuerdo a:

$$p_i(t+1) = \begin{cases} p_i(t) & f(x_i(t+1)) \leq f(x_i(t)) \\ x_i(t+1) & f(x_i(t+1)) > f(x_i(t)) \end{cases} \tag{3}$$

donde el símbolo f representa la función de *fitness*; $p_i(t)$ representa los mejores valores de *fitness* y las coordenadas donde fueron encontrados; y t una nueva iteración en la generación.

4. Aplicación de PSO para *Clustering*

Es posible ver al problema de *clustering* como un problema de optimización que localiza los centroides óptimos de los *clusters* en lugar de encontrar la partición óptima. A diferencia de la búsqueda localizada del algoritmo de *K-means*, el algoritmo de *clustering* con PSO realiza una búsqueda global del espacio de búsqueda. Con el fin de combinar ambas características se utiliza un algoritmo de *clustering* híbrido PSO+*K-means*. En el algoritmo de PSO+K-means se combina la habilidad de la búsqueda globalizada del algoritmo de PSO con la rápida convergencia del algoritmo de *K-means*. Con la búsqueda global se garantiza que cada partícula busque en forma amplia cubriendo todo el espacio del problema y con la búsqueda local se trata de que todas las partículas converjan al óptimo cuando una partícula se acerca a la vecindad de la solución óptima.
En este contexto, una partícula representa los k centroides de los *clusters*. Es decir que cada partícula x_i se construye de la siguiente manera:

$$x_i = (c_1, ..., c_j, ..., c_K) \tag{4}$$

donde c_j representa el $j-$ésimo centroide de la $i-$ésima partícula en el *cluster* C_j. El fitness de cada partícula se calcula en ambos algoritmos (PSO y PSO+K-means) de la siguiente forma:

$$f(x) = \frac{\sum_{j=1}^{K} [\sum_{z \in C_j} d(z, c_j)]}{n} \tag{5}$$

donde d es la distancia Euclideana y n es la cantidad de elementos del conjunto de datos, z representa un elemento del conjunto de datos y K es la cantidad de *clusters*.

4.1. PSO+*K-means*

A continuación se describe en detalle el procedimiento de PSO+*K-means*. Para el caso del algoritmo PSO simple para *clustering*, el algoritmo es el mismo, excepto que éste no realiza los pasos *(d)* y *(e)* correspondientes a la hibridación con el *K-means*.

1. Inicializa cada partícula con k centroides seleccionados aleatoriamente dentro del conjunto de datos.
2. Para $t = 1$ a t_{max} hacer
   - *a*) Para cada partícula x_i hacer
     - Para cada vector z del conjunto de datos
       1) Calcular la distancia Euclideana $d(z, c_j)$ a todos los centroides C_j
       2) Asigna z a su *cluster* C_j más cercano tal que la distancia a ese *cluster* sea la mínima.
       3) Calcula el fitness usando la ecuación 5.
   - *b*) Actualiza la mejor posición global y la mejor posición local.
   - *c*) Actualiza la velocidad y posición de la partícula usando la ecuación 2a y 2b, respectivamente.
   - *d*) La mejor partícula encontrada es el vector de centroides para el proceso de *K-means*.
     1) asigna cada elemento a su *cluster* más cercano.
     2) recalcula cada centroide $c_j = \frac{1}{|C_j|} \sum_{\forall z \in C_j}$; donde $\mid C_j \mid$ corresponde al número de elementos en el *cluster* C_j.
     3) Repetir los pasos 2*d*1 y 2*d*2 hasta alcanzar la condición de parada.
   - *e*) Reemplazar la mejor partícula global con los centroides obtenidos por *K-means*.
3. t_{max} es el número máximo de iteraciones. Finalizar cuando se alcance t_{max}.

5. Medidas de Calidad de *Clustering*

A los efectos de establecer la calidad de los *clusters* encontrados por los distintos algoritmos estudiados, hemos considerado una serie de medidas no-supervisadas. Las medidas no-supervisadas de validez de un *cluster* pueden dividirse en dos clases. Medida de cohesión de los *clusters* (compacto, apretado) que determina cuán cercanos están los objetos dentro del *cluster* y la separación (aislamiento) que determina lo distinto y bien separado que está un *cluster* con respecto a los otros. Estas medidas a menudo son llamadas índices internos debido a que usan sólo información presente en el conjunto de datos. La calidad de los *clusters* obtenidos por cada algoritmo se analizó según los siguientes índices internos:

- Cuantización del error:

$$Error - C = \frac{\sum_{i=1}^{K} (\sum_{z \in C_i} d(c_i, z) / \mid C_i \mid)}{K} \tag{6}$$

- Distancia Intra-Cluster (cohesión). Distancia entre todos los elementos de un *cluster*, el objetivo es minimizar esta distancia:

$$Intra - C = \frac{\sum_{i=1}^{K} (\sum_{z,t \in C_i} d(z, t) / \mid C_i \mid)}{K} \tag{7}$$

- Distancia Inter-Cluster (separación). Distancia entre los centroides de los *clusters*, donde el objetivo es maximizar la distancia entre ellos:

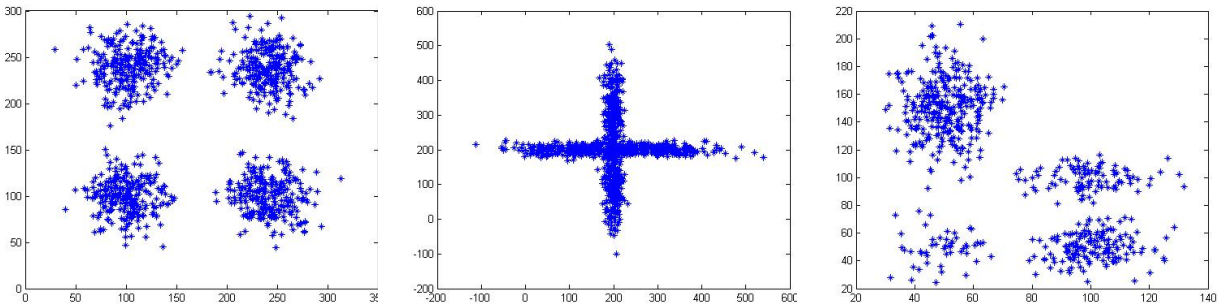
$$Inter - C = \frac{\sum_{i=1}^{K-1} \sum_{j=i+1}^{K} d(c_i, c_j)}{\sum_{i=1}^{K-1} i} \tag{8}$$

6. Experimentos y Resultados

6.1. Conjunto de Datos

Los algoritmos se probaron con tres tipos de conjuntos de datos generados artificialmente y con tres conjuntos de datos provistos por *Donald Bren School of Information and Computer Sciences* (http://mlean.ics.uci.edu/MLSummary.html). Los datos artificiales tienen las siguientes características:

- Se utilizó una distribución Normal y se generaron números aleatorios a partir del método de Box-Müller. Se generaron instancias con elementos de 2, 3, 10, 50 y 100 dimensiones, con el fin de poder trasladar estos datos a problemas reales con dimensiones grandes que representen distintos atributos de un elemento. Además de trabajar con tamaños de conjuntos de datos de 600, 800 y 1000 elementos. Cada conjunto de datos estaba formado por cuatro *clusters* donde los elementos se distribuyeron de forma tal de obtener tres tipos diferentes de *clusters*:
  - *Clusters* de igual cantidad de elementos, bien globulares y separados (figura 1(a)).
  - *Clusters* de igual cantidad de elementos, con forma elíptica y entremezclados (figura 1(b)).
  - *Clusters* de diferente cantidad de elementos, bien globulares y separados (figura 1 (c)).



(a) (b) (c)

Figura 1: Conjunto de tres tipos de datos artificiales

La Figura 1 muestra tres conjuntos de datos artificiales de dimensión 2, con 1000 elementos cada uno (Instancias denominadas I26Ba2-1000, I41Bb2-1000, I56Bc2-1000; el nombre de cada instancia se forma de la siguiente manera por ejemplo I24Ba50-800 representa lo siguiente: *I24* identificador de la instancia *Ba* se refiere a la distribución de los elementos en el conjunto de datos tipo 1(a), *50* es la dimensión y *800* es la cantidad de puntos en el conjunto de datos). Con las características anteriormente descritas se generaron 45 instancias que para la generación de las gráficas que se muestran en la siguiente sección se agruparon de la siguiente manera:

- Grupo (a) está formado por 15 instancias con los datos distribuidos con la forma de la figura 1 (a) donde cinco instancias tienen dimensiones 2, 3, 10, 50 y 100 con 600 elementos del conjunto de datos cada una, otras cinco instancias tienen dimensiones 2, 3, 10, 50 y 100 con 800 elementos del conjunto de datos cada una y las cinco últimas instancias tienen dimensiones 2, 3, 10, 50 y 100 con 1000 elementos del conjunto de datos cada una.
- Grupo (b) las 15 instancias están agrupadas de manera similar al grupo (a) pero en este grupo los datos están distribuidos como se muestra en la figura 1(b).
- Grupo (c) las 15 instancias están agrupadas de manera similar al grupo (a) pero en este grupo los datos están distribuidos como se muestra en la figura 1(c).

El conjunto de datos reales está formado por las siguientes Bases de datos:

- "*Iris plants*": es una base de datos con 4 atributos numéricos, 3 clases y 150 elementos.
- "*Glass identification*": esta base de datos contiene 6 tipos de vidrio definidos en términos de su contenido de óxido (es decir Na, Fe, K, etc).
- "*Teaching assistant evaluation*"(TAE): esta base de datos contiene 151 elementos, 6 atributos, 3 clases. Los datos corresponden a la performance de la enseñanza de tres semestres regulares y dos semestres de verano de 151 asistentes de enseñanza en la Universidad de Wisconsin - Madison.

6.2. Configuración de Algoritmos y Resultados

Para todos los algoritmos se realizaron 30 corridas independientes. En PSO y PSO+*K-means* se utilizó para espacios de problema de dimensiones pequeñas (< 10) un cúmulo de partículas de 15 y para dimensiones mayores (≥ 10) el cúmulo utilizado fue de 50 partículas. Los valores de las contantes c_1 y c_2 se fijaron en 1.49 y $w = 0{,}72$.
A continuación se comparan los resultados de los algoritmos de PSO, *K-means* y PSO+*K-means*, de 45 instancias artificiales se seleccionaron tres instancias de cada grupo y las tres instancias que pertenecen al conjunto de datos reales.
En la tablas 1, 2 y 3 se muestran los resultados obtenidos en cada algoritmo de los promedios de 30 corridas independientes para las instancias analizadas, donde la columna: "Instancia" corresponde a la instancia elegida, "C-Error" muestra la cuantización del error, "Intra-C" corresponde a la distancia Intra-Cluster, "Inter-C" es la distancia Inter-Cluster y la última columna, $\frac{Intra-Cluster}{Inter-Cluster}$ a la razón entre las dos medidas anteriores. Por consiguiente, un buen *clustering* debería ser aquel que contenga cada uno de los *clusters* individuales homogéneos y entre todos los *clusters* tan heterogéneos como sea posible. Donde un valor pequeño de $\frac{Intra-Cluster}{Inter-Cluster}$ representa homogeneidad dentro de los *clusters* y heterogeneidad entre los *clusters*.
Se puede observar que en todos los casos seleccionados del conjunto de datos artificiales y para la instancia *Iris-Plants* el algoritmo PSO+*K-means* obtiene mejores resultados para cada una de las medidas de calidad analizadas. En estas instancias se obtiene una menor cuantización del error, una menor distancia Inter-Cluster (más separación) y una distancia Intra-Cluster menor (más agrupamiento en los elementos) con lo cual significa que este algoritmo híbrido obtiene *clusters* más cohesionados y separados entre ellos.

Tabla 1: Resultados obtenidos con PSO

Instancia	C-Error	Intra-C	Inter-C	$\frac{Intra-C}{Inter-C}$
I22Ba3-800	36,96	46,31	161,74	0,2864
I23Ba10-800	81,04	89,65	282,26	0,3176
I24Ba50-800	153,91	199,38	654,94	0,3044
I38Bb10-800	166,84	213,90	316,29	0,6763
I39Bb50-800	407,95	481,96	741,84	0,6497
I43Bb10-1000	176,35	208,69	304,80	0,6847
I58Bc10-1000	50,05	55,16	139,46	0,3955
I59Bc50-1000	118,49	119,64	311,24	0,3844
I60Bc100-1000	169,12	168,68	446,02	0,3782
Iris-Plants	0,72	0,93	3,16	0,2931
Glass	1,46	1,97	3,65	0,5390
TAE	11,21	14,22	22,19	0,6409

Tabla 2: Resultados obtenidos con *K-means*

Instancia	C-Error	Intra-C	Inter-C	$\frac{Intra-C}{Inter-C}$
I22Ba3-800	31,94	45,25	159,85	0,2831
I23Ba10-800	67,40	93,85	289,76	0,3239
I24Ba50-800	161,31	222,10	623,29	0,3563
I38Bb10-800	156,46	220,65	334,98	0,6587
I39Bb50-800	369,07	518,64	713,08	0,7273
I43Bb10-1000	151,10	213,17	341,13	0,6249
I58Bc10-1000	38,98	54,67	140,82	0,3883
I59Bc50-1000	95,78	133,88	314,64	0,4255
I60Bc100-1000	138,56	194,35	426,41	0,4558
Iris-Plants	0,65	0,92	3,27	0,2825
Glass	1,58	2,29	4,34	0,5286
TAE	10,15	13,96	20,98	0,6657

Tabla 3: Resultados obtenidos con PSO+*Kmeans*

Instancia	C-Error	Intra-C	Inter-C	$\frac{Intra-C}{Inter-C}$
I22Ba3-800	31,67	44,90	160,95	0,2790
I23Ba10-800	62,33	88,32	298,37	0,2960
I24Ba50-800	151,19	199,38	659,56	0,3023
I38Bb10-800	151,27	214,48	356,83	0,6011
I39Bb50-800	361,26	481,96	788,53	0,6112
I43Bb10-1000	146,40	192,03	445,42	0,4311
I58Bc10-1000	36,64	51,43	147,21	0,3493
I59Bc50-1000	90,11	115,02	334,93	0,3434
I60Bc100-1000	136,52	161,93	469,57	0,3449
Iris-Plants	0,64	0,92	3,28	0,2816
Glass	1,44	1,96	3,84	0,5117
TAE	9,83	13,71	20,73	0,6610

Adicionalmente, en la figura 2 se muestran los respectivos valores de $\frac{Intra-Cluster}{Inter-Cluster}$ de los tres algoritmos para el grupo de 15 instancias denominado grupo (a). Se puede observar que con PSO+*K-means* se obtienen mejores resultados en 14 de las 15 instancias. La figura 3 muestra los valores de $\frac{Intra-Cluster}{Inter-Cluster}$ obtenidos por los tres algoritmos para las instancias del grupo (b). Se puede observar que con PSO+K-means se obtienen mejores resultados en 9 de las 15 instancias. Por su parte, en la figura 4 se muestran los valores $\frac{Intra-Cluster}{Inter-Cluster}$ de los tres algoritmos para las instancias del grupo (c). En este caso se puede observar que con esta distribución de los datos con el algoritmo PSO+*K-means*

se obtienen mejores resultados en todas y cada una de las 15 instancias. Por último, para el conjunto de datos reales figura 5 se obtienen mejores resultados en dos de las 3 instancias.

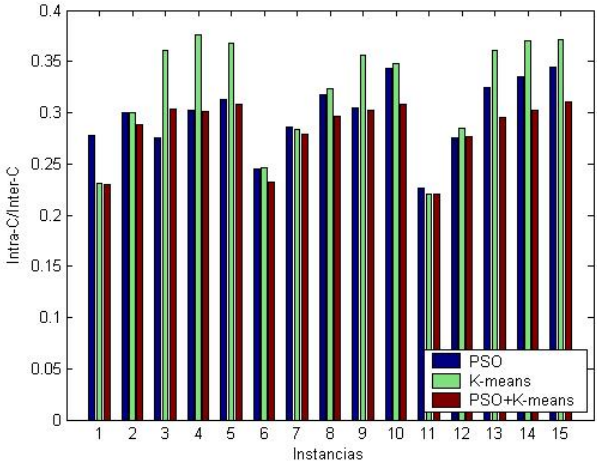


Figura 2: Valores $\frac{Intra-Cluster}{Inter-Cluster}$ para el grupo **a**

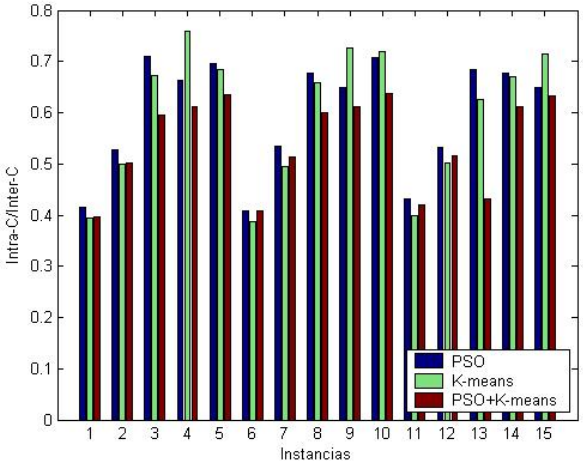


Figura 3: Valores $\frac{Intra-Cluster}{Inter-Cluster}$ para el grupo **b**

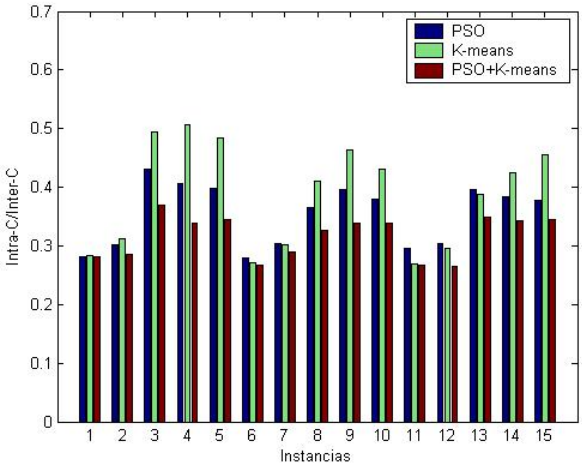


Figura 4: Valores $\frac{Intra-Cluster}{Inter-Cluster}$ para el grupo **c**

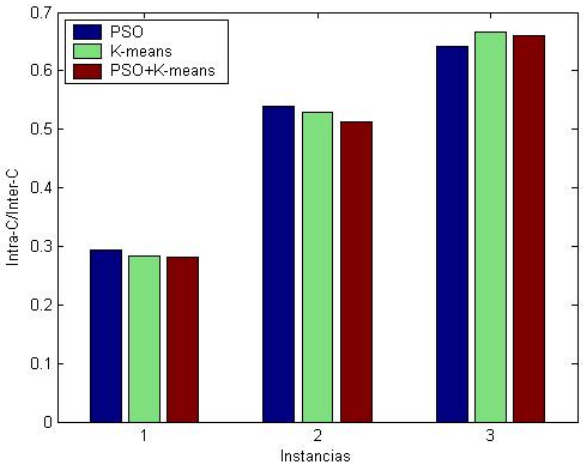


Figura 5: Valores $\frac{Intra-Cluster}{Inter-Cluster}$ para el conjunto de datos reales, donde 1=Iris, 2=Glass y 3=TAE

7. Conclusiones

PSO es una técnica de optimización que realiza una búsqueda globalizada de las soluciones y puede aplicarse a problemas de *clustering*. K-means es un algoritmo simple, directo y muy utilizado en *clustering* pero puede quedar atrapado en óptimos locales. En este trabajo se analizaron medidas de calidad de los *clusters* obtenidos por tres algoritmos PSO, *K-means* y PSO+*K-means*, este último combina la habilidad de la búsqueda globalizada de PSO con la búsqueda local de *K-means*. Se mostró que PSO+*K-means* logró encontrar en la mayoría de los casos analizados mejores resultados comparados por los obtenidos por *K-means* y PSO de manera individual con respecto a la cuantización del error, y las distancias inter-cluster e intra-cluster. Es decir que este algoritmo híbrido en general logró encontrar *clusters* más cohesionados y separados. En futuros trabajos se analizará la calidad y el número *clusters* obtenidos por este algoritmo híbrido así como también el efecto de usar diferentes parámetros en la performance de la convergencia de PSO+K-means.

Agradecimientos

Los autores agradecen a la Universidad Nacional de la Patagonia Austral por su apoyo al grupo de investigación y además, la cooperación de los integrantes del proyecto que continuamente proveen de nuevas ideas y críticas con-

structivas. El cuarto autor agradece además, el constante apoyo brindado por la Universidad Nacional de San Luis y la ANPYCIT que financian sus actuales investigaciones.

Referencias

[1] Jain A.K., Murty M. N., and Flynn P. J. Data clustering: A review. *ACM Computing Survey*, 31(3):264–323, 1999.

[2] Jain A.K. and Dubes R.C. *Algorithms for Clustering Data*. Englewood Cliffs, N.J.:Prentice Hall, 1988.

[3] Chui-Yu Chui, Yi-Feng Chen, I-Ting Kou, and He Chun Ku. An intelligent market segmentation system using k-means and particle swarm optimization. *Expert Systems with Applications*, 2008.

[4] Kennedy J. and Eberhart R. *Swarm Intelligence*. Morgan Kaufmann, San Francisco, California, 2001.

[5] MacQueen J. Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symp. Math. Statist, Prob*, pages 281–297, 1968.

[6] Omran M., Salman A., and Engelbrecht A.P. Image classification using particle swarm optimization. In *Conference on Simulated Evolution and Learning, Computational intelligence for the E-age*, 2002.

[7] Tou J. T. and Gonzalez R. C. *Pattern recognition principles*. Addison-Wesley, 1974.

[8] Fayyad U., Piatetsky-Shapiro G., and Smith P. From data mining to knowledge discovery in database. In *American Association for Artificial Intelligence*, pages 37–54, 1996.

[9] Backer U.E. *Computer-assisted reasoning in cluster analysis*. Prentice-Hall, 1995.

[10] Huang T. W. Application of clustering analysis for reducing smt setup time- a case study on avantech company. Master's thesis, Department of National Taipei University of Technology, 2006.

[11] Selim S. Z. and Ismail M.A. K-means type algorithms: A generalized convergence theorem and characterization of local optimality. *IEEE Trans. Pattern Anal. Mach. Intell.*, (6):81–87, 1984.