SACO: Un algoritmo de clustering espacial con hormigas inteligentes

José I. Dib Ashur, Jorge G. Vallón, Cristian A. Martínez y Carlos G. Said

Departamento de Informática, Facultad de Ciencias Exactas,
Universidad Nacional de Salta
{nachodibashur,gonzalo.vallon}@gmail.com, cmartinez@unsa.edu.ar y
carlossaid@gmail.com

Abstract. Bio-inspired algorithms have received much attention in recent years, because they allow to discover, quickly and efficiently, knowledge and patterns in large databases. In this work, a new algorithm based on the behaviour of ant colonies to discover clusters in spatial databases is presented. The algorithm proposed was evaluated using several well-known test instances and compared its performance considering other proposals from the literature.

Keywords: ACO, Algorithms, Datamining, DBSCAN, Spatial Clustering.

Resumen ejecutivo. Los algoritmos bioinspirados han recibido mucha atención en los últimos años, ya que permiten, de manera eficiente y eficaz, el descubrimiento de conocimiento y patrones en bases de datos grandes. En este trabajo, se propone un nuevo algoritmo basado en el comportamiento de las colonias de hormigas para descubrir clústers en bases de datos espaciales. En concreto, se presenta un algoritmo oportunamente evaluado sobre conjuntos de datos comúnmente empleados en la literatura. Se comparan los resultados que se obtienen con otros algoritmos conocidos.

Palabras claves: ACO, Algoritmos, Clustering Espacial, DBSCAN, Minería de Datos.

1 Introducción

El clustering es un proceso que busca agrupar un conjunto de objetos en múltiples grupos, donde los objetos dentro de cada uno tienen una alta similaridad, pero son muy disímiles con los objetos de otros. Las disimilitudes y similitudes se evalúan en base a los valores de los atributos que describen los objetos y a menudo involucran medidas de distancia. Tradicionalmente, los métodos de clustering se han clasificado en dos grupos: jerárquicos y particionales. Sin embargo, en los últimos años, han surgido nuevas aproximaciones de mayor eficacia y eficiencia para conjuntos de datos grandes y complejos.

En este trabajo se presenta un nuevo algoritmo bioinspirado de clustering, denominado SACO, que permite descubrir patrones y grupos con formas arbitrarias requiriendo sólo dos parámetros fáciles de determinar.

El resto del trabajo se organiza de la siguiente manera. En la sección 2 se discuten nuevos métodos de agrupación utilizados para clustering espacial. En la sección 3 se desarrolla la propuesta SACO. Se realizaron evaluaciones experimentales que se resumen en la secciones 4 y 5, que permiten destacar la eficacia y eficiencia sobre datasets ampliamente conocidos. Finalmente, la sección 6 indica un resumen del trabajo y pautas de trabajo futuro.

2 Clustering espacial

La minería de datos espaciales es el descubrimiento de conocimiento implícito y previamente desconocido en bases de datos espaciales. Sus métodos pueden ser usados para entender los datos espaciales, descubrir relaciones entre datos espaciales y no espaciales, reorganizar los datos en bases de datos espaciales y determinar sus características generales de manera simple y concisa.

Hay cuatro metodologías tradicionales que se pueden implementar para el proceso de clustering espacial. A los métodos jerárquicos y no jerárquicos se añaden los métodos basados en redes o grillas y, principalmente, los métodos basados en densidad [4]. Para mayor información se recomienda [11].

Los algoritmos basados en densidad determinan grupos como regiones densas de puntos, separadas por otras regiones poco densas. La agrupación espacial evalúa la similitud de acuerdo a las características espaciales de los datos y, por lo tanto, en vez de hablar de la "similitud entre dos objetos" solemos referirnos a la "proximidad espacial de dos objetos". De esta forma se pueden obtener grupos con formas irregulares que se entrelazan [3] [9]. Existen varios algoritmos basados en densidad, entre los cuales se destaca **DBSCAN** propuesto por Ester [7].

Además, en los últimos años, la inteligencia de enjambres ha sido introducida en el análisis de clústers para resolver diversos problemas existentes. La misma consiste en la modelización del comportamiento grupal de animales sociales, tales como las hormigas. Dichas sociedades son sistemas distribuidos que, a pesar de la sencillez de sus individuos, presentan una organización social altamente estructurada. Como resultado de esta organización, las colonias pueden realizar tareas complejas que en algunos casos superan con creces las capacidades individuales de un solo individuo.

Particularmente, entre los algoritmos basados en colonias de hormigas para clustering podemos destacar dos modelos [13]:

1. Modelo de amontonamiento: Inspirados en el comportamiento de las hormigas que agrupan los cadáveres y clasifican las larvas de la colonia. Los algoritmos de clustering que siguen este enfoque permiten encontrar clústers circulares, no siendo muy útiles para reconocimiento de patrones.
2. Modelo de búsqueda: Algoritmos inspirados en el comportamiento de las hormigas que buscan la ruta más corta entre su hormiguero y una fuente

de alimento. El libre movimiento de la hormigas permite conectar objetos no neceseramiente de forma elíptica. En este grupo se destaca el algoritmo **CACO** propuesto por Chu [5].

3 Smart Ant Colony Optimization

Basado en el algoritmo CACO, se presenta **SACO** (Smart Ant Colony Optimization).

Dado un conjunto de datos, nuestra propuesta busca obtener agrupaciones mediante una colección de hormigas que inician sus recorridos sobre diferentes elementos y construyen sus caminos considerando información de aprendizaje y específica del problema propias de algoritmos ACO, pero también incluyendo aspectos distintivos relacionados con longitudes de recorridos condicionados por umbrales dinámicos e información de densidad. Es importante destacar que la colonia colabora con la construcción de una única solución a partir de los recorridos construidos, la cual es luego mejorada.

A continuación se presentan las características distintivas de la propuesta:

- Se calcula un umbral estadístico de longitud de tramo en tiempo real (1) permitiendo la movilidad de la hormiga k-ésima entre dos objetos siempre que la longitud de dicho tramo no supere el valor del umbral.

$$L_{ts}^{k} = AvgL_{tramo}^{k} + StDevL_{tramo}^{k} \tag{1}$$

$$AvgL_{tramo}^{k} = \frac{\sum L_{ij}^{k}}{E} \tag{2}$$

$$StDevL_{tramo}^{k} = \sqrt{\frac{\sum (L_{ij}^{k} - AvgL_{tramo}^{k})^2}{E}} \tag{3}$$

  Siempre que el tramo (X_i, X_j) sea visitado por la hormiga k-ésima, y donde E es el número de tramos recorridos por dicha hormiga. Se puede notar que se obtienen rutas que no necesariamente poseen la misma longitud.
- Como consecuencia de la característica anterior, se logra evitar en gran medida la necesidad de realizar cortes a causa de rutas mal formadas, pero no totalmente. Esto se debe a que si la posición inicial de la hormiga, que es aleatoria, coincide con un punto ruido, el umbral se ve gravemente condicionado. Por tal motivo, se lleva a cabo una revisión y corrección de tales rutas.
- Teniendo en cuenta el funcionamiento de las hormigas reales, se puede observar que con el transcurso del tiempo y la repetición rutinaria, adquieren mayor conocimiento del medio. Traducido al algoritmo, a medida que avanzan las iteraciones, se espera que las hormigas sean capaces de obtener rutas más largas. Por lo tanto el número de hormigas requerido para visitar todos los objetos del dataset debería disminuir. De esta manera, se finalizan las iteraciones cuando la cantidad de hormigas requeridas en una iteración $i + 1$ es mayor que las requeridas previamente en la iteración i.

- Si bien en todo algoritmo de búsqueda existe un parámetro de decaimiento para evaporación de feromonas, las rutas obtenidas en la primera iteración resultan condicionantes para las futuras iteraciones. Es decir, en muchos casos, las hormigas imitan rápidamente los caminos previamente recorridos por otras, causando así soluciones de mala calidad. En busca de corregir este comportamiento, en lugar de determinar aleatoriamente posiciones iniciales para una cierta cantidad de hormigas individuales, se determinan Z puntos de partida para conjuntos de R hormigas exploradoras que caminan en diferentes direcciones desde dichos puntos.
- La información de densidad también puede ser útil para determinar si es necesario o no la definición de puntos extras a los Z ya escogidos. Se define un umbral de densidad según la siguiente ecuación:

$$Density_{ts} = AvgDensity + StDevDensity \tag{4}$$

$$AvgDensity = \frac{\sum_{i=1}^{T} SKNND(X_i)}{T} \tag{5}$$

$$StDevDensity = \sqrt{\frac{\sum_{i=1}^{T}(SKNND(X_i) - AvgDensity)^2}{T}} \tag{6}$$

  Donde T es la cantidad de objetos del dataset. De esta manera, luego de definir aleatoriamente las posiciones iniciales y obtener las rutas correspondientes, se constituye un punto extra inicial sobre un objeto X_j aún no visitado si se verifica que $SKNND(X_i) < Density_{ts}$.
- En base a lo anterior, las capacidades de las hormigas se ven incrementadas respecto a los algoritmos ACO tradicionales, por lo que se requiere de menor información heurística para obtener buenas rutas.
- Es de vital importancia el análisis de rutas que no se interconectan pero espacialmente definen regiones superpuestas. Estas rutas determinan clústers diferentes, cuando lo esperable es que no fuera así. Por consiguiente se lleva a cabo la unión de dichos clústers.

Un pseudocódigo básico para SACO se muestra en el algoritmo 1.

4 Experimentos y resultados

A continuación, se indican diferentes experimentos realizados para evaluar la calidad de resultados.

Se realizaron pruebas sobre 8 datasets sintéticos bidimensionales [1] [8] y se compararon los resultados alcanzados con los obtenidos por DBSCAN, el cual es considerado uno de los algoritmos más potentes y conocidos de la literatura. Debido a la falta de resultados numéricos y de información sobre los datasets usados, no se realizaron comparaciones con el algoritmo CACO.

Respecto a la configuración de par'ametros del algoritmo DBSCAN, las pruebas evidenciaron la dificultad para obtener valores apropiados. Se ejecutó

Algoritmo 1 SACO

Require: Z, R
while (No se cumpla condición de finalización) **do**
  Elegir Z posiciones iniciales de grupos de hormigas exploradoras
  for each Posición inicial **do**
    Establecer R hormigas exploradores
    Obtener rutas
  end for
  for each punto P del dataset que no haya sido visitado y sea denso **do**
    Obtener rutas
  end for
  Actualizar feromonas entre objetos
end while
Cortar aquellas rutas que conecten puntos que parecieran pertenecer a clústers diferentes
Conectar las rutas que posean elementos en común formando clústers
Unir clústers superpuestos

el algoritmo 100 veces sobre cada dataset, iniciando los parámetros de forma aleatoria dentro de los rangos [0,15] y [0,30] para $MinPts$ y Eps respectivamente. Oportunamente se guardaron los valores de los parámetros para los que se obtuvieron los mejores aciertos.

En referencia a SACO, Z se estableció a partir de las sugerencias de cantidad de hormigas iniciales expuestas en CACO y se probó con valores para R en el rango [2,6]. SACO fue implementado en JAVA, y todos los experimentos se realizaron sobre un equipo Intel I7 4CPU @ 2.7Ghz y Memoria RAM 8192MB.

En la tabla 1 se indica información de las instancias (nombre, cantidad de elementos, número de grupos) y de los algoritmos SACO y DBSCAN (parámetros usados y porcentajes de acierto alcanzado). Los resultados listados por SACO corresponden al mejor de 100 corridas.

Tabla 1. Parámetros y porcentajes de aciertos de SACO y DBSCAN según mejor agrupación conocida.

				SACO			**DBSCAN**		
Dataset	**Número de objetos**	**Número de clústers**	**Tiene ruido**	R	Z	**% acierto**	$MinPts$	Eps	**% acierto**
Complex9	3031	9	NO	4	20	**0,9584**	2	12,1191	0,9978
Complex9RN8	3273	9	SI	3	20	**0,9059**	9	16,0012	0,8851
Complex9RN16	3515	9	SI	3	20	**0,8481**	11	15,9997	0,8259
Complex8	2551	8	NO	4	20	**0,9177**	2	15,4818	0,9933
Aggregation	788	7	NO	3	20	**0,9927**	13	1,9500	0,9759
Compound	399	5	SI	2	20	**0,9599**	3	1,5000	0,9724
Jain	373	2	NO	4	20	**0,9973**	3	2,4604	0,9276
Pathbased	300	3	NO	3	20	**0,8833**	2	1,9321	0,8066

4.1 Evaluación de rendimiento

A partir de los experimentos realizados se puede afirmar que SACO es capaz de reconocer grupos de diferentes densidades debido a que las hormigas poseen la capacidad de trabajar de manera local. Esto les permite determinar rutas independientemente de lo que otras hormigas construyan, razón por la cuál pueden obtenerse clústers de características diferentes; lo que resulta una gran ventaja respecto a otros algoritmos basados en densidad, tal como DBSCAN. En adición, SACO logra detectar fronteras difusas entre grupos de elementos. Estas cuestiones pueden observarse en las agrupaciones obtenidas en el dataset Pathbased (véase figura 1).

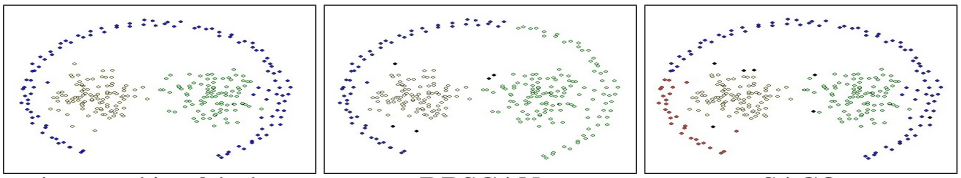

Figura 1. Dataset Pathbased.

Otro ventaja de SACO es que el agrupamiento final no se ve fuertemente afectado por puntos outliers o ruido, lo cuál permite una mejor agrupación y un mejor porcentaje de acierto. Esto queda reflejado en los resultados obtenidos con dos versiones del dataset Complex9 que poseen diferentes cantidades de puntos outliers aleatoriamente distribuidos, conocidos como Complex9RN8 y Complex9RN16 (veáse figura 2).

4.2 Comparación con otros algoritmos de la literatura

A los efectos de lograr una mejor apreciación de la calidad de resultados obtenidos por SACO, se realizaron comparaciones con otras propuestas de la literatura, considerando las mismas instancias de prueba. Finalmente, se comparó SACO con otros algoritmos presentes en la literatura. Dicha comparación se puede observar en la tabla 2, no incluyéndose resultados visuales.

Los algoritmos Multi-edit y Wilson Editing son los que otorgan los mejores valores. Ambos fueron analizados por Zeidat [12] sobre un total de 14 datasets. Los resultados provistos se corresponden con los obtenidos tras 3 ejecuciones, de manera que la fuerza probatoria no resulta demasiado elevada.

En el otro extremo se encuentran HAC-RAND y HAC-MO, los algoritmos que proporcionan los resultados más bajos. En el trabajo de Bartoň [2] los mismos son evaluados sobre 13 datasets de 2 o 3 dimensiones. Desafortunadamente no se hace referencia al plan de prueba que se llevó a cabo.

En contraposición, SACO se ejecutó 100 veces sobre cada dataset, de manera que los resultados obtenidos poseen un sustento muy fuerte. En algunos casos

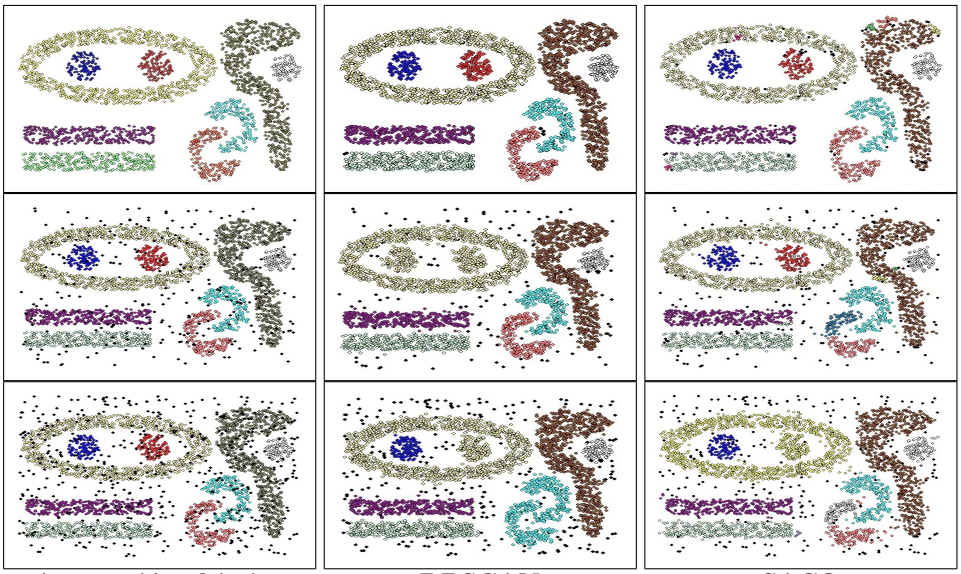

Figura 2. Fila 1: Complex 9 - Fila 2: Complex 9RN8 - Fila 3: Complex9RN16

SACO proporciona porcentajes de acierto inferiores a otros algoritmos, mientras que en otros casos ocurre lo contrario. Sin embargo, todos los trabajos citados a los efectos de la comparación vieron la luz en la última década. De esta manera, se concluye que SACO resulta competitivo según otras alternativas de la literatura.

5 Conclusiones

Se realizó una breve introducción al clustering espacial y se hizo hincapié en los algoritmos basados en colonias de hormigas. La importancia de estos algoritmos radica en la contemporaneidad de los mismos y el potencial públicamente reconocido que poseen. Se presentó el algoritmo bioinspirado SACO, el cual requiere sólo dos parámetros, que pueden determinarse con facilidad. Cabe destacar el hecho de que SACO ha sido evaluado sobre bases de datos sintéticas. Los experimentos y comparaciones con otros algoritmos de la literatura demuestran que SACO es significativamente más efectivo descubriendo clústers de formas arbitrarias.

La investigación futura tendrá que considerar las siguientes cuestiones. Primero, la optimización del algoritmo SACO, buscando mayor robustez en los resultados e introducirlo en la bibliografía corriente. Segundo, la aplicación y evaluación de la utilidad de incorporar algoritmos espaciales en diferentes áreas, como criminalística, geología y organización territorial.

Tabla 2. Comparación de SACO con otros algoritmos de la literatura.

	Dataset					
Algoritmo	**Complex9**	**Complex9RN8**	**Complex9RN16**	**Complex8**	**Aggregation**	**Jain**
SACO	0,9584	0,9059	0,8481	0,9177	0,9927	0,9973
SCAH [6]	0,9740					
SCHG [6]	0,9740					
SCMRG [6]	0,9570					
SCEC [10]	0,9890			0,9510		
MOSAIC [10]	0,9890			0,9520		
HAC-RAND [2]	0,7100			0,7200	0,7400	0,6100
HAC-MO [2]	0,7000			0,7100	0,7100	0,6000
1-NN [12]	1,0000	0,9210	0,8430	1,0000		
Wilson Editing [12]	1,0000	0,9410	0,8870	1,0000		
Multi-edit [12]	1,0000	0,9410	0,8890	0,9990		
Citation Editing [12]	1,0000	0,9390	0,8700	1,0000		
SC Editing [12]	0,9880	0,8220	0,7620	0,9870		
K-means	0,6775	0,6523	0,6101	0,4895	0,7860	0,7791

Referencias

[1] "Clustering datasets," accedido el 10-08-2015. URL: https://cs.joensuu.fi/sipu/datasets

[2] T. Bartoň y P. Kordík, "Using multi-objective optimization for the selection of ensemble members," *Information Technologies - Applications and Theory (ITAT)*, 2015, charles University in Prague, Prague, J. Yaghob.

[3] F. Berzal, "Clustering basado en densidad," accedido en 29-04-2015. URL: http://elvex.ugr.es/idbis/dm/slides/43%20Clustering%20-%20Density.pdf

[4] Chandra y Anuradha, "A survey on clustering algorithms for data in spatial database management systems," *International Journal of Computer Applications*, vol. 24, no. 9, 2011.

[5] S.-C. Chu, J. F. Roddick, C.-J. Su, y J.-S. Pan, "Constrained ant colony optimization for data clustering," *Proc of 8th Pacific Rim International Conference on Artificial Intelligence (PRICAI)*, 2004, auckland, New Zealand, LNAI 3157.

[6] Eick, Vaezian, Jiang, y Wang, "Discovering of interesting regions in spatial data sets using supervised clustering," *The 10th European Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD 06)*, vol. 9, no. 1, 2006.

[7] M. Ester, H.-P. Kriegel, J. Sander, y X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," *Proc of 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*, 1996, portland, OR, USA.

[8] U.-D. Group, "Dataset repository," accedido el 20-03-2016. URL: http://www2.cs.uh.edu/~ml_kdd

[9] J. Han, M. Kamber, y A. K. H. Tung, "Spatial clustering methods in data mining: A survey," school of Computing Science, Simon Fraser University, Burnaby, BC Canada V5A 1S6.

[10] R. Jiamthapthaksin, J. Choo, C. sheng Chen, O. U. Celepcikay, C. Giusti, y C. Eick, "Mosaic: Agglomerative clustering with gabriel graphs," 2009.

[11] R. Xu y D. W. II, *Clustering*. John Wiley and Sons, 2008, iSBN 10: 0470276800.

[12] Zeidat, Wang, y Eick, "Dataset editing techniques: a comparative study," 2005.

[13] G. Zhe, L. Dan, A. Baoyu, O. Yangxi, C. Wei, N. Xinxin, y X. Yang, "An analysis of ant colony clustering methods: Models, algorithms and applications," *International Journal of Advancements in Computing Technology*, vol. 3, no. 11, 2011.