Clustering a través de técnicas bio-inspiradas

Diego Alejandro Ingaramo[†],Guillermo Leguizamón[†], Marcelo Errecalde[†]

[†]Laboratorio de Investigación y Desarrollo en Inteligencia Computacional (LIDIC)
Departamento de Informática
Universidad Nacional de San Luis
Ejército de los Andes 950 - Local 106
(5700) - San Luis - Argentina
Tel: (02652) 420823 / Fax: (02652) 430224
e-mail: {daingara,legui,merreca}@unsl.edu.ar

Resumen

Este artículo describe los trabajos de investigación y desarrollo que se están llevando a cabo en el Laboratorio de Investigación y Desarrollo en Inteligencia Computacional (LIDIC), relacionados a la aplicación de técnicas bio-insipiradas a problemas de minería de datos, y en particular, a tareas de *clustering*. Intuitivamente, una tarea de *clustering* consiste en la clasificación no supervisada de patrones (observaciones, datos, vectores, etc.) en grupos. Este problema ha sido analizado en varios contextos y por investigadores de distintas disciplinas, reflejando su amplia utilidad. Si bien se han propuesto distintas alternativas para abordar las tareas de *clustering*, existe un área particularmente interesante y novedosa que ha planteado distintos enfoques bio-inspirados que incluyen los algoritmos genéticos y algoritmos basados en la metáfora del comportamiento de las hormigas. En este trabajo, describimos brevemente algunos de los trabajos que se están llevando a cabo en el LIDIC referidos a la utilización de algoritmos basados en el comportamiento de hormigas en la Minería de Datos, y más específicamente, a la tarea de *clustering*. Entre estos algoritmos podemos mencionar a AntTree, con el cual se ha experimentado utilizando distintas instancias del problema de *clustering*, reportándose algunas de las ventajas y desventajas observadas en este algoritmo en el trabajo experimental. También se proponen extensiones a este algoritmo que permitirían flexibilizar el proceso del descubrimiento de *clusters* dentro de los datos a analizar.

1. Introducción

La tarea de *clustering* [3] es la clasificación no supervisada de patrones (observaciones, datos, vectores, etc.) en grupos. Este problema ha sido analizado en varios contextos y por investigadores de distintas disciplinas, reflejando su amplia utilidad. Es un problema de gran dificultad combinatoria y, dado a que se ha utilizado en diferentes áreas, los métodos obtenidos carecen de generalidad.

La tarea de *clustering* consiste en la organización de una colección de patrones (usualmente representados como vectores de atributos o puntos en un espacio multidimensional) en *clusters* (o grupos) basándose en la similitud que existe entre los mismos. Intuitivamente, patrones de un mismo *cluster* son más similares a patrones que se encuentran fuera del mismo. Los humanos resuelven de manera competitiva problemas de *clustering* en dos dimensiones, pero la mayoría de los problemas reales implican dimensiones más grandes. Además, la distribución de los datos muy difícilmente siga una forma definida. Por ello, encontramos una gran cantidad de algoritmos que se comportan de mejor o peor manera dependiendo de la distribución del conjunto de datos.

La variedad de técnicas que existen difieren en la representación de los datos, en la medida de proximidad (o similitud) entre elementos, y en la manera que agrupan los elementos. El método más simple que resuelve el problema de *clustering* se denomina *K-Mean*, en donde debemos definir la cantidad de grupos (llamados centroides) que existen en los datos. Cada centroide define un grupo de datos y se asocia cada dato al centroide más cercano. Luego, iterativamente se recalculan estos centroides de tal forma que en cada iteración se minimiza la función SSE (Suma de los errores al cuadrado). Cuando no existe mejora, el algoritmo finaliza su ejecución. Este método posee desventajas muy importantes. Una de las principales es que se debe especificar el número de *clusters* desde el principio, dato que generalmente no se conoce. Ésto significa que para aplicar dicho algoritmo, se presupone un conocimiento previo de la distribución de los datos.

Debido a la gran importancia de este problema en diferentes campos, distintos métodos se han propuesto en la literatura para resolverlo. Recientemente, los enfoques *bio-inspirados* tales como los algoritmos genéticos (un ejemplo es el algoritmo ACODF[4], o el método propuesto en [6]) y metaheurísticas tales como tabu search, simulated annealing han sido aplicados exitosamente a este problema [1]. Otra de las metaheurísticas utilizada es el enfoque *Ant Colony Optimization* (ACO) [5], adaptando una versión al problema de *clustering* como se describe en [9]. Sin embargo, en la actualidad ha surgido un importante grupo de algoritmos basados en la metáfora del comportamiento de las hormigas reales los cuales son aplicados a *clustering*. Entre dichos algoritmos se encuentran: *Ant-Class* [8] un algoritmo de *clustering* que usa los principios exploratorios y estocásticos del enfoque ACO combinados con los principios determinísticos y heurísticos de K-Mean. El entorno simulado es una grilla bidimensional en la cual las hormigas recogen o depositan objetos en una parva de acuerdo a su similitud. *Ant-Tree* [2] algoritmo inspirado en las posibilidades de auto-ensamblaje de las hormigas reales. Por ejemplo, realizar construcciones de puentes vivientes en beneficio de la colonia. Finalmente, *Ant-Clust* [7] un algoritmo inspirado en el reconocimiento químico de las hormigas para formar grupos o *clusters* diferenciados por sus respectivas propiedades químicas u olores.

2. Tareas en Progreso

Los trabajos realizados en el grupo de investigación en este área incluyen un estudio pormenorizado sobre los principios de Minería de Datos y la tarea de *clustering*. En particular, nos hemos concentrado en las distintas versiones existentes de los algoritmos basados en el comportamiento de la colonias de hormigas (BCH).

La propuesta consiste en desarrollar una nueva versión de un algoritmo de la clase de algoritmos BCH orientado a la tarea de *clustering*. La nueva versión está diseñada en base al algoritmo Ant-Tree y tiene como característica principal la incorporación de aspectos dinámicos de las hormigas en el proceso de construcción del árbol respectivo. La idea original del algoritmo Ant-Tree consiste en construir una estructura de árbol a través de un proceso de auto-ensamblaje por parte de las hormigas (datos a analizar). Dicho proceso toma en consideración medidas de distancia entre los datos (hormigas) de manera tal que cada nuevo dato ingresado a la estructura bajo construcción se ubique como una nueva hoja de una rama del árbol cuyos nodos se encontrarían en el mismo *cluster*. La aridad del nodo raíz (ocupado por un dato u hormiga) representa el número de *cluster* encontrados por el algoritmo. Los nodos de cada uno de dichos subárboles son los miembros de cada uno de los *clusters* resultantes. Nuestra propuesta consiste en permitir la movilidad de las hormigas dentro del árbol, esto es, que puedan cambiar de posición dentro del árbol y de esta manera incrementar el grado de precisión del algoritmo en cuanto al número de *clusters* descubiertos. La metodología incluye también un estudio comparativo entre este nuevo algoritmo y el AntTree, y así demostrar las potenciales ventajas.

Los algoritmos son implementados en el lenguaje JAVA dentro del paquete de software de dominio público denominado WEKA[10]. Dicho paquete provee de utilidades específicas para *clustering* lo cual permite una fácil realización del estudio comparativo propuesto.

3. Trabajos Futuros y consideraciones finales

La presente propuesta tiene por objetivo la investigación y desarrollo de métodos alternativos para determinadas tareas de minería de datos. La utilización de algoritmos BCH es uno de los temas a profundizar por el grupo de investigación. También está bajo consideración la posibilidad de ampliar este tipo de desarrollos a otras técnicas bio-inspiradas, tal como algoritmos evolutivos y optimización por cúmulo de partículas respecto de las cuales el LIDIC tiene una importante experiencia.

Referencias

[1] Hussein A. Abbass, Ruhul A. Sarker, and Charles S. Newton. *Data Mining: A Heuristic Approach*. University of New South Wales, Australia, 2002.

[2] H. Azzag, N. Monmarche, M. Slimane, G. Venturini, and C. Guinot. Anttree: A new model for clustering with artificial ants. In Ruhul Sarker, Robert Reynolds, Hussein Abbass, Kay Chen Tan, Bob McKay, Daryl Essam, and Tom Gedeon, editors, *Proceedings of the 2003 Congress on Evolutionary Computation CEC2003*, pages 2642–2647, Canberra, 8-12 December 2003. IEEE Press.

[3] Morven Leese Brian S. Everitt, Sabine Landau. *Cluster Analysis*. Institute of Psychiatry, Kings College, London, 2001.

[4] Han-Chang Wu Tzer Yang Cheng-Fa Tsai, Chun-Wei Tsai. Acodf: a novel data clustering approach for data mining in large databases. *Journal of Systems and Software*, pages 133–145, 2004.

[5] M. Dorigo and L. M. Gambardella. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1):53–66, 1997.

[6] A.A. Freitas. *A survey of evolutionary algorithms for data mining and knowledge discovery*, pages 819–845. Springer-Verlag, 2002.

[7] Nicolas Labroche, Nicolas Monmarché, and Gilles Venturini. Antclust: Ant clustering and web usage mining. In E. Cantú-Paz, J. A. Foster, K. Deb, D. Davis, R. Roy, U.-M. O'Reilly, H.-G. Beyer, R. Standish, G. Kendall, S. Wilson, M. Harman, J. Wegener, D. Dasgupta, M. A. Potter, A. C. Schultz, K. Dowsland, N. Jonoska, and J. Miller, editors, *Genetic and Evolutionary Computation – GECCO-2003*, volume 2723 of *LNCS*, pages 25–36, Chicago, 12-16 July 2003. Springer-Verlag.

[8] N. Monmarché, M. Slimane, and G. Venturini. Antclass: discovery of clusters in numeric data by an hybridization of an ant colony with the kmeans algorithm. Rapport interne 213, Laboratoire dÍnformatique de lÚniversité de Tours, E3i Tours, Janvier 1999. 21 pages.

[9] B. D. Kulkarni P.S. Shelokar, V. K. Jayaraman. An ant colony approach for clustering. Technical report, Chemical Engineering & Process Division, National Chemical Laboratory, India, 2003.

[10] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, San Francisco, 2000.