Agrupamiento de Datos Linealmente Separables Mediante un Algoritmo Genético Celular

Migdaled López-Juárez, Ricardo Barrón Fernández, Salvador Godoy-Calderón

Centro de Investigación en Computación del Instituto Politécnico Nacional, México
migdaled.lopez@gmail.com, barron2131@gmail.com,
sgodoyc@gmail.com.

Resumen. El uso de algoritmos genéticos es amplio, ya que se permite encontrar soluciones óptimas con una función aptitud que así lo permita, tal es el caso de el agrupamiento particional de objetos, considerando particularmente aquellos que son linealmente separables. En este trabajo se propone un algoritmo genético celular para encontrar el número de particiones óptimo considerando el valor DB (índice que mide el agrupamiento de datos realizado por un algoritmo, introducido por David L. Davies y Donald W. Bouldin) como función de aptitud.

Palabras clave: Algoritmos genéticos, algoritmos genéticos celulares, particionamiento.

1 Introducción

Los algoritmos genéticos (GA) son métodos de búsqueda para problemas de optimización y están inspirados en los principios de selección natural de Charles Darwin [1]. Estos algoritmos tienen como objetivo evolucionar un conjunto de posibles soluciones (población) a un problema, hacia un óptimo global [2], permitiendo ser aplicados en una amplia gama de problemas. Tal es el caso del agrupamiento particional de datos, el cual forma parte de una de las técnicas de aprendizaje no supervisado (clustering). Dicho agrupamiento puede emplear el criterio de selección de centroides u holotipos que permitan particionar los datos [2]. Generalmente es necesario validar que tan adecuadas son las particiones creadas, es decir evaluar si es mejor tener 2 particiones que 3 ó 5, es por ello que existen índices de validación. Estos índices toman en cuenta los criterios de compactación y separación entre las particiones [3].

Como se ha mencionado, los GA trabajan con una población conformada por las posibles soluciones a un problema, en cuyo caso es necesario codificar la información de cada individuo (cromosoma) en números reales, números enteros, etc. A cada uno de los cromosomas se le aplicarán los operadores genéticos de selección, cruza (recombinación) y mutación con el objetivo de encontrar un individuo que represente una solución óptima global.

Advances in Computer Science and Engineering
Research in Computing Science 63, 2013 pp. 79–88
Paper Recived 22-09-2013 and Acepted 21-10-2013

En este trabajo se propone el uso de una familia de algoritmos que forman parte de los GA: algoritmos celulares genéticos (cGA). En los cGA cada individuo obteniene información de los individuos vecinos (vecindario) durante la selección y recombinación [7].

2 Revisión del estado del arte

El trabajo de Sanghamitra Bandyopadhyay y Ujjwal Maulik [4] propone el agrupamiento de datos de manera tal que, no necesita un número de particiones o bien centroides preestablecidos. De hecho, la población está conformada por individuos que no tienen la misma longitud (cromosomas de longitud variable). El cromosoma lo conforman números reales los cuales representan las coordenadas de los centroides en un espacio n-dimensional. El índice I desempeña el papel de función de aptitud.

Ge Xiufeng y Xing Changzheng [5], trabajan de igual forma con cromosomas de longitud variable codificados en números reales, tal como en [4]. Lo más sobresaliente de este trabajo es que dividen a la población en un número arbitrario de subpoblaciones llamadas islas, y éstas evolucionan de manera independiente. El índice DB es usado como función de aptitud [2,3].

De igual forma Venkatesh Katari et al. [6] trabajo con cromosomas de longitud variable como [4,5] pero ahora los cromosomas se encuentran codificados en números enteros positivos representando los niveles de RGB (por sus siglas en inglés, Red Green Blue) en una imagen digital. La función de aptitud es de nueva cuenta el índice DB.

3 Preliminares

3.1 Agrupamiento por particionamiento

El agrupamiento (clustering) de patrones es una técnica de clasificación no supervisada de patrones en la cual se particionan los datos en K regiones. El criterio empleado para dicho particionamiento es la medida de distancia (o bien, similitud) con respecto a un objeto arbitrario (holotipo o centroide).

Es necesario proporcionar el número K de grupos para calcular los holotipos o centroides [8], para este trabajo no es necesario indicar el número de particiones. Esto es, el algoritmo se encargará de evolucionar los holotipos para poder particionar los datos de manera óptima.

3.2 Algoritmos celulares genéticos (cGA)

Los cGA como se ha mencionado, forman parte de una familia de los GA. La población está usualmente estructurada en una rejilla (o malla) n-dimensional de individuos. Este modelo simula la evolución natural desde el punto de vista del individuo, los individuos sólo realizan la selección y recombinación con sus vecinos. Con base en los experimentos realizados [7] se observa que la lenta difusión de soluciones in-

ducida en la población con los vecindarios solapados permite la exploración (diversificación), mientras que la explotación (intensificación) se logra con la interacción de cada uno de los vecindarios (ver figura 1).

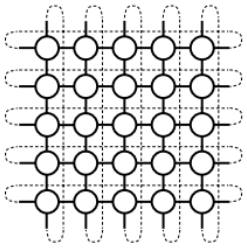

Fig. 1. Malla toroidal de 2 dimensiones donde se representa la estructura de los individuos

4 Propuesta

A continuación se resumen los pasos principales que conforman el algoritmo propuesto (ver tabla 1).

Tabla 1. Algoritmo genético celular propuesto.

1. Inicializar población
2. Evaluar la población inicial
3. Verificar si la condición de paro se cumple
   (a) Finalizar el algoritmo en caso de que se cumpla
   (b) Continuar con el paso 4
4. Determinar el vecindario de cada individuo
5. Seleccionar la pareja de recombinación para cada individuo
6. Recombinar a cada individuo con la pareja previamente seleccionada para generar un nuevo individuo
7. Mutar a cada uno de los nuevos individuos generados
8. Reemplazar cada uno de los individuos, sólo si el nuevo individuo que ha generado es mejor que él
9. Regresar a paso 2

En las siguientes sub secciones se mencionan más a detalle las etapas que intervienen para llevar a cabo la ejecución del algoritmo propuesto.

4.1 Estructura de la población

En este caso se utilizan rejillas de *nxn*, donde *n* es el número máximo de particiones que se desean probar. Para obtener el valor de *n* es necesario considerar el núme-

ro de datos que se tienen, obtener su raíz cuadrada, finalmente el número cuadrado superior cercano a la raíz cuadrada será el resultado.

Es decir, en caso de tener 30 datos, *n* será igual a 9.

La política empleada para formar al vecindario es la selección de los 4 vecinos más cercanos, considerando que el individuo se encuentre en las coordenadas *i,j* dentro de la malla los vecinos serán: *(i,j-1), (i+1,j),(i-1,j),(i,j-1).* Este tipo de vecindario fue propuesto por Von Neumann en la década de los 50's.

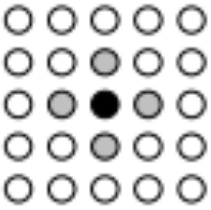

Fig. 2. Selección de vecinos para cada individuo

4.2 Codificación de los individuos

La población que conforma nuestro espacio de búsqueda son coordenadas de números reales (holotipo) para formar un determinado número de particiones.

El cromosoma que se representa para cada individuo es de longitud variable, la longitud depende del número de particiones que se desean formar y la dimensión de los holotipos (R^n), como se muestra en la tabla 2.

$(z_{11}, z_{21}, \ldots, z_{i1})$	$(z_{12}, z_{22}, \ldots, z_{i2})$	...	$(z_{i1}, z_{i2}, \ldots, z_{ij})$

Tabla 2. Codificación de los individuos

4.3 Operadores genéticos

Para llevar a cabo la recombinación entre individuos es necesaria una previa selección de aquellos individuos (padres) que generarán una nueva descendencia. En este caso, tenemos la a un individuo dentro de la población como primer padre (P1) y tomando en cuenta a los individuos más cercanos a él (vecinos), se selecciona al mejor (P2) por medio del método de ruleta, donde básicamente la selección es proporcional a la función de desempeño [1].

La recombinación se logra tomando como base a P1, posteriormente seleccionar de un punto de cruza para poder emplear información de P2. Para este caso nos referimos a coordenadas de holotipos. La cantidad de coordenadas a ser intercambiadas queda en función de una probabilidad de recombinación.

Durante la mutación el nuevo individuo que se formó con P1 y P2 será alterado por incrementos aleatorios.

$$x_{t+1} = x_t(1 \pm 2\delta) \quad (1)$$

Donde,

δ: un número en el rango [0, 1] generado con una distribución uniforme
z: indica el valor de una de las coordenadas del holotipo
t: indica el número de generación

El reemplazo de z_t por z_{t+1} sólo en el caso que z_{t+1} tenga un mejor valor de aptitud.

4.4 Evaluación de los individuos

El cálculo de aptitud de los individuos está basado en el índice DB. Este índice fue seleccionado dado que responde adecuadamente a la evaluación de los agrupamientos en los datos que se manejan (hiperesféricos)

$$\text{Disp Ci} = \sqrt{\frac{1}{|\text{Ci}|}\sum_{\text{oi,oj}\in\text{Ci}} \|\text{oi} - \text{oj}\|} \quad (2)$$

$$\text{DB} = \frac{1}{\text{k}}\sum_{\text{i=1}}^{\text{k}} \max_{\text{j=1...k,i}\neq\text{j}} \left\{\frac{\text{Disp(Ci)} + \text{Disp(Cj)}}{\text{d(Ci, Cj)}}\right\} \quad (3)$$

Donde,

i, j: indican el índice de los holotipos que conforman la partición
$Disp\ Ci$: indican la dispersión de los elementos de una partición
k: número de particiones

4.5 Condición de paro

La condición de paro establecida es llegar a un máximo de *w* generaciones, este dato es arbitrario.

5 Experimentos

Para poner en práctica el algoritmo genético celular propuesto en la sección 4, se implementó el código en C empleando gcc 4.7.1 sobre una plataforma OpenSuse.

6 Experimentos

Para poner en práctica el algoritmo genético celular propuesto en la sección 4, se implementó el código en C empleando gcc 4.7.1 sobre una plataforma OpenSuse Linux 12.2. Se emplearon dos conjuntos de datos. Uno de los conjuntos es sintético y está constituido por tres clases linealmente separables entre ellas. El otro conjunto es real [10] y cuenta con 3 clases, sólo una de ellas es linealmente separable de las otras 2 (estas dos últimas no son linealmente separables una de la otra.)

El objetivo de los experimentos es validar que se pueda obtener el número óptimo de clases para cada uno de los conjuntos de datos, por lo que se toma en consideración la frecuencia relativa en las pruebas ejecutadas para dicho número. El número de ejecuciones para cada experimento es 30.

6.1 Datos sintéticos

Se empleó un conjunto de 300 datos hiperesféricos en R^2 con 100 puntos en cada clase, el conjunto de datos se puede observar en la Figura 2. En la Tabla 3 se muestran los parámetros empleados en la ejecución.

Tabla 3. Parámetros para la ejecución del experimento con 300 datos hiperesféricos en R^2.

Probabilidad de cruza: 0.65
Probabilidad de mutación: 0.01
Número límite de generaciones: 25

A continuación, se muestra los resultados obtenidos con la ejecución del algoritmo, el número de clases que se encuentra como el óptimo y la frecuencia relativa. El resultado es satisfactorio, dado que el conjunto es de tres clases.

Tabla 4. Resultados con un conjunto 300 datos hiperesféricos.

Número óptimo de clases	Frecuencia relativa (%)
3	80%
2	20%

En la Figura 3 se observa el valor promedio de aptitud (fitness) obtenido, se puede observar que la difusión del óptimo local dentro de cada vecindario permite llegar a un óptimo global en 25 generaciones.

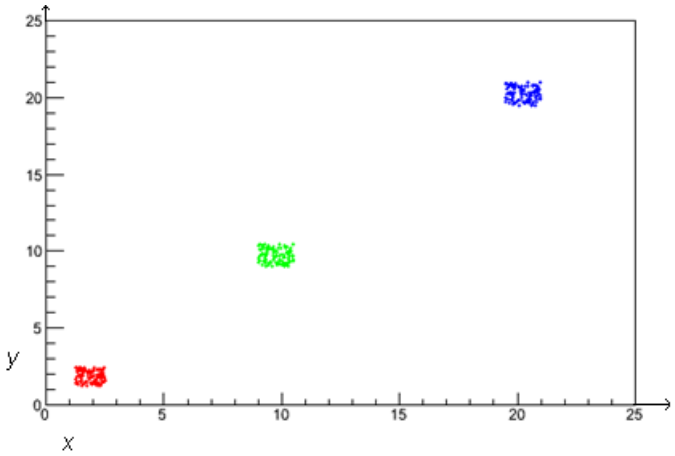

Figura 1. Datos en espacio R^2

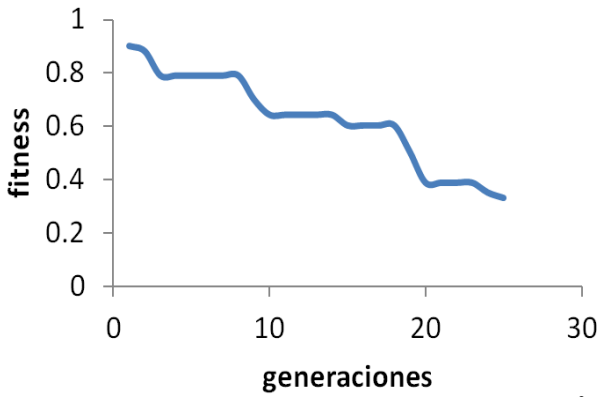

Figura 2. Gráfica de convergencia para 300 datos en R^2.

6.2 Datos reales

Se empleó el conjunto Iris [10] en R^4, el conjunto de datos se puede observar en la Figura 4. En la Tabla 5 se muestran los parámetros empleados en la ejecución.

Tabla 5. Parámetros para la ejecución del experimento con el conjunto Iris.

Probabilidad de cruza: 0.65
Probabilidad de mutación: 0.01
Número límite de generaciones: 45

A continuación, se muestra los resultados obtenidos con la ejecución del algoritmo, el número de clases que se encuentra como el óptimo y la frecuencia relativa del evento. El resultado es aceptable en comparación con [4,11-12] pues encuentran que el número de clases detectadas son dos.

Tabla 6. Resultados con el conjunto Iris Data.

Número óptimo de clases	Frecuencia relativa (%)
2	60%
3	40%

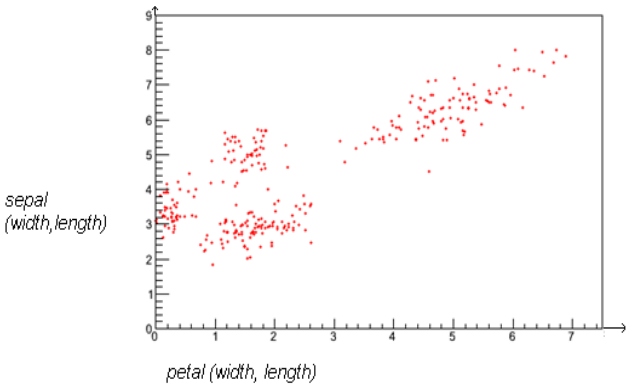

Figura 3. Conjunto Iris Data, en el eje de las abscisas se consideran las medidas de los pétalos y en el eje de las ordenadas los sépalos.

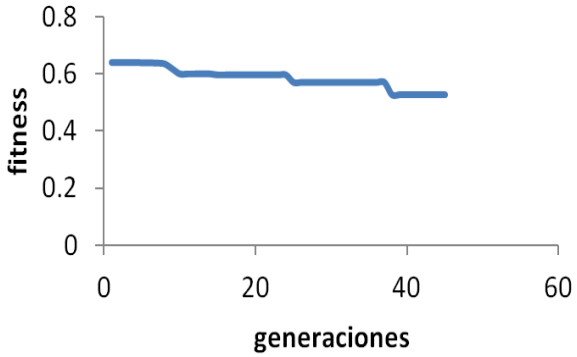

Figura 4. Gráfica de convergencia para Iris Data.

7 Conclusiones

Luego de implementar el algoritmo genético celular y realizar pruebas, se obtiene un resultado satisfactorio, dato que es posible encontrar un número óptimo de clases empleando el índice DB para medir la aptitud (fitness) de cada individuo. Para el conjunto de datos sintético se encuentran tres, y para el conjunto real dos.

Así como también es importante señalar que, en menos de 50 generaciones es posible determinar un valor óptimo para los dos conjuntos de datos probados.

Agradecimientos

Agradecemos el apoyo otorgado por el Instituto Politécnico Nacional a través del proyecto 20131210.

Referencias

1. Holland, J.H. (1975). Adaptation in Natural and Artificial Systems. Ann Arbor, University of Michigan Press.
2. Naldi, MariloCoelho and Carvalho, André C.P.L.F. and Campell, Ricardo José GabrielliBarreto and Hruschka, Eduardo Raul. (2008). Soft Computing for Knowledge Discovery and Data Mining, Genetic Clustering for Data Mining. Springer US.
3. Halkidi, Maria and Batistakis, Yannis and Vazirgiannis, Michalis. (2001). On Clustering Validation Techniques. Kluwer Academic Publishers.
4. Bandyopadhyay, S., Maulik, U., (2001), Nonparametric genetic clustering: Comparison of validity indices. Systems, Man and Cybernetics, Part C, IEEE Transactions on: Applications and Reviews.
5. Ge Xiufeng, Xing Changzheng. (2010). K-means Multiple Clustering Research Based on Pseudo Parallel Genetic Algorithm. Information Technology and Applications (IFITA), 2010 International Forum on.
6. Venkatesh Katari, Suresh Ch, Ra Satapathy, Member Ieee, Jvr Murthy, Pvgd Prasad Reddy. (2007). Hybridized Improved Genetic Algorithm with Variable Length Chromosome for Image Clustering Abstract. IJCSNS International Journal of Computer Science and Network Security.
7. Alba, Enrique and Dorronsoro, Bernabè. (2008). Cellular Genetic Algorithms. Introduction to Cellular Genetic Algorithms. Universidad de Málaga. Springer US.
8. Xiu Riu, Wunsch II Donald C. (2009). Clustering. IEEE Press Series on Computational Intelligence.
9. Niloy Gangul, Biplab K Sikdar, Andreas Deutsch, Geoffrey Canright, P Pal Chaudhuri. (2001). A Survey on Cellular Automata.
10. Iris Dataset, http://archive.ics.uci.edu/ml/datasets/Iris.
11. J. C. Bezdek and N. R. Pal. (1998). Some new indexes of cluster validity. IEEE Trans. Syst., Man, Cybern.
12. R. Kothari and D. Pitts. (1999). On finding the number of clusters. Pattern Recognit. Lett.