Algoritmos genéticos y su aplicación en la visualización de un mapa auto-organizado

Dante Giovanni Sterpin[*]

Resumen

Los algoritmos genéticos fueron propuestos durante la década de los setenta como una gran herramienta computacional para la solución artificial de problemas complejos. Su diseño y funcionamiento están fundamentados en la capacidad que tienen los seres vivos para adaptarse a las exigencias de su medio ambiente, con lo cual se logra artificialmente que un conjunto de soluciones se adapte a las exigencias de un determinado problema. En este artículo se presenta una aplicación de dichos algoritmos, enfocada a visualizar las distancias entre las neuronas cognitivas en un mapa auto-organizado de Kohonen. Este último supone una mejor manera de interpretar su aprendizaje, en contraste con las limitaciones del método conocido como matriz-unificada de distancias.

Palabras clave: algoritmo genético, clusterización, mapa auto-organizado

Abstract

Genetic algorithms were proposed during the 1970's as a great computational tool for the artificial solution of complex problems. Its design and operation are based on the ability of living beings to adapt themselves to the demands of their environment. This capacity artificially achieves that a certain problem adapts a set of solutions. This article presents an application of these algorithms, focused on visualizing the distances between cognitive neurons on a self-organized map of Kohonen. This method is a better way of interpreting their learning, in contrast to the limitations of the method known as matrix-unified distances (or U-matrix).

Keywords: Clustering, Genetic Algorithm, Self-Organized Map

* Ingeniero Electrónico, Universidad Santo Tomás. Especialista en Docencia Universitaria, Universidad Militar Nueva Granada. Estudiante de maestría en Ingeniería de Sistemas y Computación, Pontificia Universidad Javeriana. Contacto: dante_sterpin@javeriana.edu.co

Introducción

Los algoritmos genéticos se inspiran en la genética y la selección natural para hacer sistemas *inteligentes* artificiales, capaces de resolver autónomamente ciertos problemas en optimización y de aprendizaje mecánico (Goldberg, 1989), en robótica móvil (Floreano y Mattiussi, 2008) y hasta en composición de música (Jacob, 1995; Matić, 2010).

Fueron originalmente propuestos por Holland (1975) y, en general, definen la estructura de los llamados *algoritmos evolutivos* (Hart *et al.*, 2005). Dicha estructura está conformada por una población de individuos numéricos que deben adaptarse a las exigencias de un ecosistema matemático. Esto significa que un conjunto de simbolizaciones representa posibles soluciones y debe mejorar para resolver algún tipo de problema, mediante la adaptación de los individuos al ecosistema.

Los individuos que simbolizan las posibles soluciones están conformados por información genética numérica y el ecosistema que los evalúa es alguna función matemática capaz de calcular la capacidad de un individuo para satisfacer las exigencias del problema. Así, algunos individuos sobrevivirán y otros perecerán en beneficio de la evolución poblacional, es decir, en pro de lograr, al cabo de varias generaciones, al menos una configuración genética capaz de satisfacer lo mejor posible dichas exigencias, con lo que se habría encontrado una buena solución al problema.

En esta medida, puede considerarse que los algoritmos genéticos simulan las capacidades adaptativas de los seres vivos con el fin de realizar agentes capaces de resolver problemas, semejante a como lo hacen los seres humanos, pues implican cierta creatividad (Bentley y Corne, 2002).

En este artículo, primero se detallan los elementos y procedimientos típicos de un algoritmo genético y, luego, se describe su utilización para resolver el problema de visualizar en 2D la clusterización de un conjunto de datos nD, realizada con un mapa auto-organizado de Kohonen para hacer minería de datos.

Agente evolutivo artificial

En términos generales, un agente tiene la capacidad de representar, adquirir y emplear conocimiento con el propósito de comportarse racionalmente, es decir, actuar para alcanzar autónomamente sus metas, según lo que perciba y lo que aprenda (Russell y Norvig, 1995).

Cada ser humano es un tipo de agente, con procesos biológicos, psicológicos y sociales muy complejos, por supuesto, pero en su mecánica de supervivencia cumple con el mismo principio básico. El propósito de este artículo no es reflexionar al respecto, sino señalar que, como agente, cada ser humano puede verse como un único individuo, dentro de su respectiva población, y en su interior hay un cerebro que procesa la información perceptiva y cognitiva para decidir las actuaciones mediante las cuales busca garantizar la consecución de sus metas. Sin embargo, dentro de cada cerebro hay

neuronas individuales interactuando entre sí y, en conjunto, exhiben el tipo de racionalidad acá mencionada.

Así mismo, un único agente artificial basado en un algoritmo genético tiene una población de ciertos individuos, con cierta manera de interacción entre ellos, y así, está dotado con la capacidad de encontrar soluciones autónomamente, en ciertos contextos de aplicación, para responder en consecuencia: mostrar datos en una pantalla o mover los motores de un *cuerpo* robot.

Individuos numéricos

La simbolización de alguna posible solución para un problema que requiera resolverse artificialmente corresponde a la caracterización genética de cada individuo en la población del algoritmo genético. Dicho código genético puede interpretarse como una especie de cromosoma artificial, o bien, como un conjunto de varios cromosomas, donde cada cromosoma es una secuencia de símbolos numéricos, normalmente en binario, pero también puede emplearse otra base numérica, si se considera que el ADN natural es de base 4.

Además del concepto de *cromosoma artificial*, también puede haber genes artificiales, que son subsecuencias numéricas en los cromosomas; así se tiene que una posible solución puede ser la interacción de varios genes. Los alelos artificiales son formas posibles por las que pueden optar los genes y, por ende, los cromosomas. De esta manera, cada individuo tiene un genotipo, a manera de caracterización genética específica, y exhibe un fenotipo, a manera de poner-en-marcha la solución propuesta por el individuo, de cara al problema.

Ecosistema matemático

La valoración de la calidad del código genético de los individuos debe realizarse con una función matemática que represente la idoneidad de las soluciones que simbolizan, porque de esto depende que realmente resuelvan el problema en cuestión. En lo posible, debe calcularse de manera normalizada: [0.0; 1,0]. Esto no sólo da a conocer la calidad de una solución que se obtiene como resultado manifiesto, sino que permite monitorear el desarrollo evolutivo correspondiente al proceso cognitivo del algoritmo genético en busca de dicha solución.

Evolución artificial

El mecanismo mediante el cual la población evoluciona es reiterativo y en cada iteración, típicamente denominada generación, se valora numéricamente la calidad de los individuos, se distribuye una probabilidad de supervivencia entre los individuos, se establece probabilísticamente una población de individuos útiles –sobrevivientes de la selección natural simulada–, se realizan intercambios genéticos entre dichos sobrevivientes, se hacen modificaciones

genéticas aleatorias a los individuos resultantes, y los individuos resultantes constituyen la siguiente nueva población.

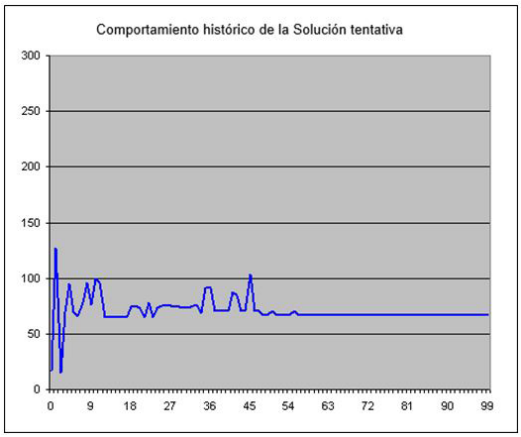
Este mecanismo evolutivo debe iniciar con una población típicamente aleatoria y dichos pasos se repiten hasta que al menos uno de los individuos proponga una solución que satisfaga el problema de la mejor manera posible, lo cual es determinado por cierto valor crítico, o de tolerancia, basado en la función de idoneidad genética. La distribución de probabilidad para poder calcular la supervivencia de los individuos se calcula en función de la idoneidad genética individual $F(g_i)$ y la comunal, empleando la ecuación 1:

$$P_i = \frac{F(g_i)}{\sum_j F(g_j)}$$

Ecuación 1. Probabilidad de supervivencia, para una población de j individuos

Al realimentar continuamente la nueva generación de individuos se procura mejorar las soluciones halladas o quizás se encuentren mejores, tal como puede observarse en los resultados de un algoritmo genético sencillo –detallado en las figuras 1 y 2, en donde el problema en cuestión consiste en la maximización de cierta función H(x), empleada tal cual como función de idoneidad genética–.

Figura 1. Solución entregada por el algoritmo durante 100 generaciones



Fuente: elaboración propia.

Figura 2. Registro histórico de la idoneidad genética promedio de la población

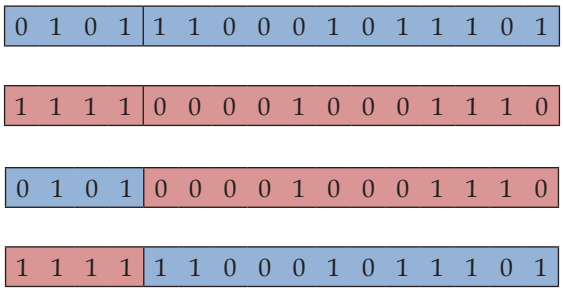
Fuente: elaboración propia.

En la figura 2 puede observarse mucha incertidumbre al comienzo del proceso evolutivo y que el algoritmo genético debe desordenarse en la vigésima generación para encontrar la mejor solución.

Operadores genéticos

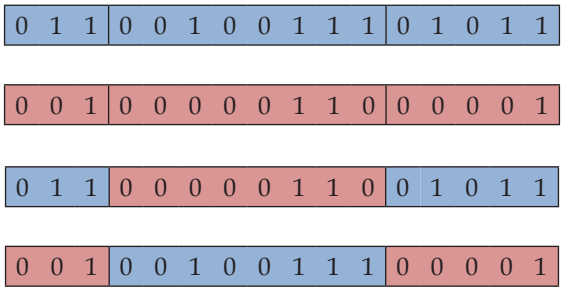
Al proceso de intercambio genético se le conoce como cruzamiento. Este consiste en fraccionar cada cromosoma en uno o varios puntos –denominados *locus* de cruce y definidos al azar–, con el fin de recombinar el material genético, como herencia de progenitores a sucesores. En la figura 3 se muestra un ejemplo del cruzamiento entre los cromosomas de dos progenitores que *engendran* dos sucesores empleando un solo *locus* de cruzamiento. Por su parte, en la figura 4, los sucesores se *engendran* tras usar dos *locus* de cruzamiento.

Figura 3. Progenitores y sucesores en un cruzamiento genético simple



Fuente: elaboración propia.

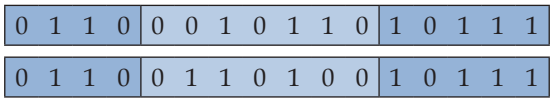
Figura 4. Progenitores y sucesores en un cruzamiento genético doble



Fuente: elaboración propia.

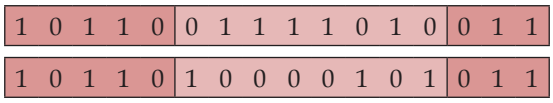
El proceso de modificación genética se conoce como mutación. Este consiste en escoger aleatoriamente una sección genética y modificarla según el tipo de código numérico empleado. En la figura 5, la mutación sucede al intercambiar el orden de los *locus* genéticos escogidos. En la figura 6, la mutación sucede al negar los bits de la sección genética escogida. En las figuras 7 y 8, la mutación sucede al desplazar circularmente los *locus* de la sección genética escogida. Suele considerarse que la mutación explora nuevas soluciones tentativas, mientras que el cruzamiento procura explotar las soluciones "medio-buenas" encontradas en generaciones anteriores. En teoría, este último procedimiento le permitiría al algoritmo genético refinar dichas soluciones, pero, en realidad, no es muy efectivo en ello. Al respecto, pueden agregarse algunas técnicas que exploren las *cercanías* de dichas soluciones (Hart *et al.*, 2005), pero en este trabajo no se emplearon.

Figura 5. Individuo original y mutado, mediante inversión de *locus* genéticos



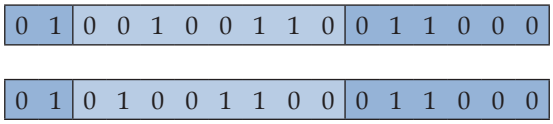
Fuente: elaboración propia.

Figura 6. Individuo original y mutado, mediante negación de *locus* binarios



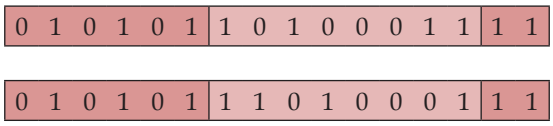
Fuente: elaboración propia.

Figura 7. Mutación mediante rotación por la izquierda de los *locus* genéticos



Fuente: elaboración propia.

Figura 8. Mutación mediante rotación por la derecha de los *locus* binarios



Fuente: elaboración propia.

Clusterización de datos

En el campo del análisis de datos para reconocer o identificar patrones hay 2 grandes paradigmas para el aprendizaje de máquina: el supervisado y el no supervisado. Con el primero se logra clasificar datos nuevos según posea las características de cierto grupo conocido, mientras que con el segundo se hace posible identificar los grupos de características semejantes entre los datos observados; a estos últimos grupos se les denomina *clústeres* (Bishop, 2006).

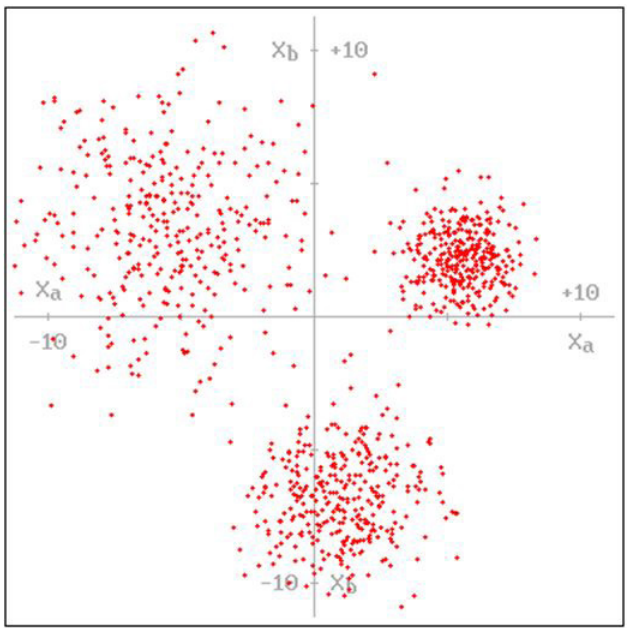
Existen varias técnicas para clusterizar datos. En esta oportunidad se emplea un mapa auto-organizado de Kohonen (SOM, por su sigla en inglés), cuyo propósito es visualizar los clústeres mientras se reduce la n-dimensionalidad de los datos a 2D, mediante las distancias entre las neuronas cognitivas del SOM. Cada neurona cognitiva del SOM codifica un prototipo, o modelo, a través del que se representan las características comunes en un subgrupo de datos y se conserva su

topología original. El mecanismo de aprendizaje del SOM no es de interés en este artículo, pues las neuronas del SOM no hacen parte del agente evolutivo artificial en cuestión. Para este último, dicha visualización es el problema que debe resolver al emplear el algoritmo genético que lo constituye.

Visualización de datos 2D

Para ilustrar la manera como el SOM representa la topología de los datos que observa, en la figura 9 pueden detallarse los datos bi-dimensionales (2D) con los cuales se entrenó un SOM conformado por [8 x 8] neuronas cognitivas.

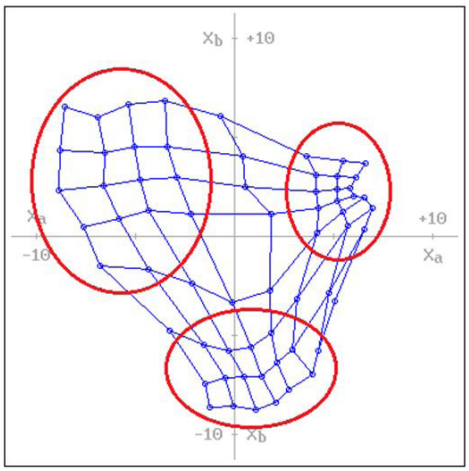
Figura 9. Datos bi-dimensionales (2D) con los cuales se entrenó el SOM de la figura 8



Fuente: elaboración propia.

A simple vista, puede notarse que dichos datos se agrupan en tres clústeres, uno de ellos más homogéneo que los otros dos. En la figura 10 puede detallarse el resultado de la clusterización realizada por las [8 x 8] neuronas del SOM. Allí, las distancias entre los nodos neuronales son las que representan, por cercanía, la semejanza existente entre los datos en ciertas regiones del espacio 2D.

Figura 10. Identificación de los clústeres existentes en los datos de la figura 7



Fuente: elaboración propia.

Una visualización adicional de los datos clusterizados es conocida como matriz unificada de distancias (matriz-U). Esta calcula el promedio de distancia euclídea entre cada nodo neuronal y sus vecinos en el SOM. Al conocer todos los promedios, cada uno se divide entre el mayor de ellos. El resultado obtenido para el ejemplo en esta sección puede detallarse en la figura 11.

Figura 11. Matriz-U con los tres clústeres existentes en los datos de la figura 7

0,287675	0,237023	0,174129	0,184503	0,246192	0,323648	0,859457	
0,372827	0,259182	0,218791	0,227141	0,323603	0,755756		0,666783
0,762953	0,502861	0,463866	0,408469	0,748371		0,635568	0,417232
	0,849132	0,776351		0,865686	0,520506	0,374011	0,380581
0,69223	0,698727	0,658723		0,859867	0,469433	0,367211	0,340686
0,559559	0,519186	0,519163	0,706645		0,576998	0,376377	0,38227
0,518039	0,53249	0,563629	0,60308		0,728886	0,402957	0,372676
0,588696	0,645006	0,641309	0,611247			0,541549	0,35044

Fuente: elaboración propia.

En la matriz-U, la coloración obtenida para cada nodo neuronal es relativa a las distancias aprendidas por el SOM, dado cierto conjunto de datos. Así, entre más cercano sea un nodo a sus vecinos se le verá más blanco, mientras que entre más lejano sea un nodo a sus vecinos se le verá

más negro. De esta manera los clústeres resaltan como una especie de *región iluminada*, separada de otras mediante *zanjas oscurecidas*.

Sin embargo, esta visualización puede resultar engañosa, pues, para dos SOM del mismo tamaño, pero entrenados con datos diferentes, sus matrices-U pueden tener la misma *iluminación* en cierta región, pero es posible que las distancias reales entre sus respectivos nodos no sean comparables entre los dos SOM. Esto se ilustrará cuando se presenten los resultados de este trabajo, pues se trata precisamente del aporte logrado.

Adicionalmente, puede pensarse que los nodos en la figura 8 corresponden con los nodos en la figura 9, pero no. Como el SOM se ajusta libremente a los datos de entrenamiento, en este caso la fila superior de nodos, en la figura 8, es la primera columna de nodos, al lado izquierdo, en la figura 9. Así, el clúster de la esquina superior derecha, en la figura 8, es el clúster de la esquina superior izquierda, en la figura 9. Por lo tanto, hay que ser muy cuidadoso al contrastar las dos visualizaciones.

Desarrollo metodológico

El ejemplo antes mencionado es un caso simple en el que cada dato solo tiene dos valores: {Xa; Xb}. La figura 7 es suficiente para apreciar su topología. Sin embargo, normalmente los datos que se analizan con técnicas de clusterización suelen contener más de dos valores por dato. Para ilustrar esto, el lector puede pensar en la gran cantidad de hipervínculos que explora al buscar conocimientos, productos o servicios en la *web*. Así se obtienen los datos n-dimensionales que acarrean implícitamente las tendencias, o preferencias, de estudiantes, clientes, pacientes, etc. Al clusterizar los datos se hacen evidentes dichas preferencias, información con la que se pueden personalizar recomendaciones de posible interés para los usuarios.

Dejando establecido que, para datos de 4 o más dimensiones (nD), la malla de cuadrángulos de la figura 8 no se puede graficar de forma tan simple como se ilustró en la sección anterior, y que la matriz-U no muestra de forma precisa las distancias reales entre las neuronas del SOM, el problema que enfrenta el agente evolutivo artificial es encontrar las coordenadas 2D que permitan ver dichas distancias, como apoyo a la visualización de la matriz-U. En la actualidad, esto se intenta hacer con otro método iterativo (Sammon, 1969). En este artículo se presenta una alternativa basada en algoritmos genéticos –por su sencillez y popularidad, sin la intención de compararla con la de Sammon–.

Geometría del problema

Con el fin de diseñar el genotipo de los individuos y la función de idoneidad genética en el algoritmo genético para el agente evolutivo artificial, se consideró la posibilidad de encontrar al tiempo todas las coordenadas (x ; y) de la malla de cuadrángulos, y exigirle al agente que

las distancias euclídeas 2D, entre ellos, cumplan el requerimiento de tener la misma distancia euclídea nD que hay entre las neuronas del SOM. Al realizar tal agente, se lo dejó *pensar* por varios días y no encontró una solución ni poco satisfactoria. Por lo tanto, se decidió usar la estrategia de dividir el problema en partes, que, en este caso, implica graficar cada cuadrángulo por aparte.

$$d_{pq} = \sqrt{\sum_{k=1}^{n} (p_k - q_k)^2}$$

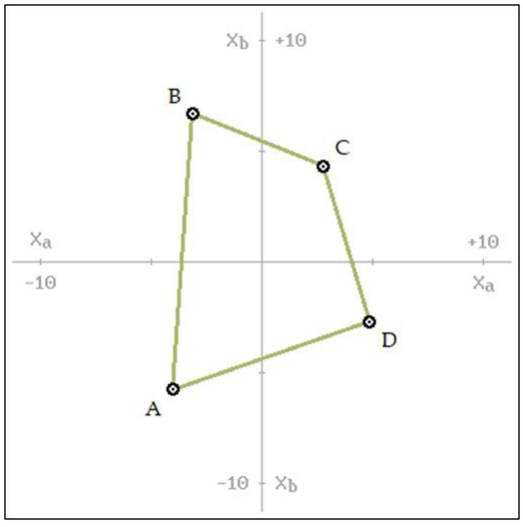
Ecuación 2. Distancia euclídea entre dos puntos nD, P: (p_1 … p_n) y Q: (q_1 … q_n).

$$d_{pq} = \sqrt{(x_p - x_q)^2 + (y_p - y_q)^2}$$

Ecuación 3. Distancia euclídea entre dos coordenadas 2D, P: (x_p ; y_p) y Q: (x_q ; y_q).

Cada cuadrángulo se construye al unir cuatro puntos en el espacio 2D con segmentos de recta, tal como se muestra en la figura 12. Los puntos: {A, B, C, D} son sus vértices, mientras que las rectas: {AB, BC, CD, DA} son sus aristas. Las coordenadas de cada vértice se expresan con las siguientes parejas ordenadas: {(x_a ; y_a), (x_b ; y_b), (x_c ; y_c), (x_d ; y_d)}. Con la ecuación 3 se calculan las medidas de arista entre sus respectivos vértices.

Figura 12. Ejemplo de cuadrángulo, con sus vértices en cada cuadrante del plano 2D



Fuente: elaboración propia.

Las variables conocidas del problema son las cuatro medidas requeridas de arista, cuya expresión algebraica es la distancia euclídea 2D, mientras que las incógnitas son los cuatro puntos de los vértices, para los cuales no hay ninguna restricción. De esta manera, se tiene un sistema de cuatro ecuaciones con ocho incógnitas y no se puede aplicar ningún método convencional de resolución algebraica, por lo tanto, un algoritmo genético es una alternativa con la cual se puede resolver este problema.

Genotipo de los individuos

Como el genotipo debe representar una posible solución al problema, entonces se establecieron cuatro cromosomas de 32 *locus* binarios cada uno. Cada uno de ellos representa los valores: (x ; y) de la coordenada 2D, de cada vértice. Así, cada cromosoma tiene dos genes, cada uno de 16 *locus* binarios. En total, se tienen entonces 8 genes: $\{(x_a ; y_a), (x_b ; y_b), (x_c ; y_c), (x_d ; y_d)\}$. Los alelos de cada gene están definidos por los pesos numéricos: $\{2^4, 2^3, 2^2, 2^1, 2^0, 2^{-1}, 2^{-2}, 2^{-3}, 2^{-4}, 2^{-5}, 2^{-6}, 2^{-7}, 2^{-8}, 2^{-9}, 2^{-10}, 2^{-11}\}$, para calcular la conversión de binario a decimal en la respectiva manifestación fenotípica.

Fenotipo de los individuos

La expresión del genotipo en forma de solución *tentativa* emitida por medio del agente evolutivo, se evidencia con el trazado del cuadrángulo en una porción de la pantalla del PC destinada para ello. Vale resaltar que, en parte, el fenotipo no está directamente codificado en el genotipo, sino que resulta posterior a la decodificación de los ochos genes, pues, en gran medida, corresponde a las aristas del cuadrángulo resultante.

Idoneidad para las aristas

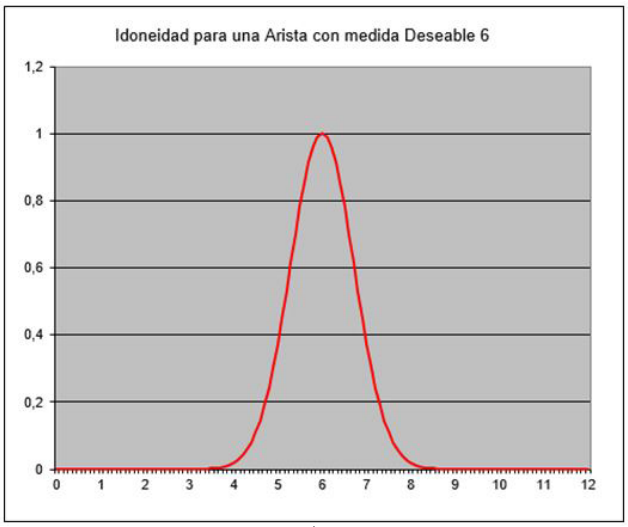
El ecosistema debe evaluar el genotipo con el cual cada individuo representa un cuadrángulo y, para ello, considera las distancias deseables para sus aristas. La ecuación 4 evalúa la medida de arista (λ_i) en contraste con su medida deseable (Λ_i). Esta función es una campana gaussiana cuya constante de anchura depende de la medida deseable.

$$F(\lambda_i) = \begin{cases} e^{-\left[\frac{(\lambda_i-\Lambda_i)^2}{(0{,}167442*\Lambda_i)^2}\right]} ; \Lambda_i \geq 0{,}1 \\ e^{-\left[\frac{(\lambda_i-\Lambda_i)^2}{2{,}803682E-4}\right]} ; \Lambda_i < 0{,}1 \end{cases}$$

Ecuación 4. Idoneidad genética para las aristas (λi), donde i: {AB, BC, CD, DA}

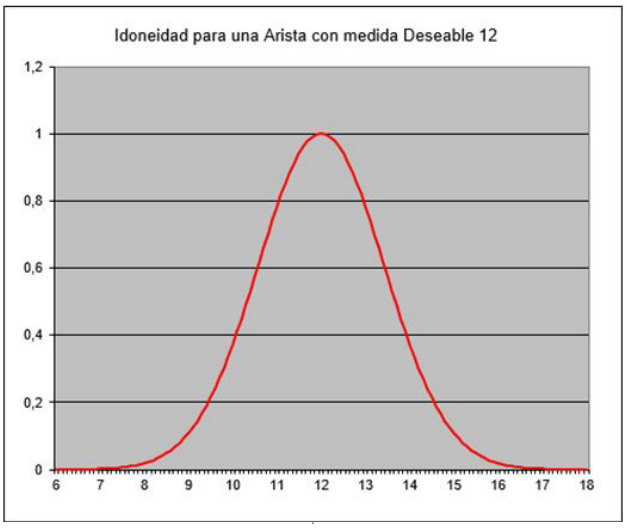
El propósito de ajustar la anchura de la campana gaussiana al valor deseable es para que el valor de idoneidad resulte 0,7 cuando λ es $\pm 0{,}1*\Lambda$, y siempre que $\Lambda \geq 0{,}1$, pues cuando no se cumple esta última condición, la función gaussiana mantiene una anchura fija.

Figura 13. Función de idoneidad genética para una medida deseable Λ_i=6



Fuente: elaboración propia.

Figura 14. Función de idoneidad genética para una medida deseable Λ=12



Fuente: elaboración propia.

En la figura 13, se ejemplifica la función de idoneidad dada Λ_i = 6. En este caso puede notarse que para λ_i = 5,4 y para λ_i = 6,6 el valor de idoneidad es 0,7. En la figura 14 se ejemplifica dicha función para Λ_i = 12. En esta también se nota que para λ_i = 10,8 y para λ_i = 13,2 el valor de idoneidad es 0,7. Con los valores de idoneidad de las cuatro aristas: $F(\lambda_{AB})$, $F(\lambda_{BC})$, $F(\lambda_{CD})$ y $F(\lambda_{DA})$, se calcula un promedio. Este resultado se penaliza multiplicando por un puntaje [0,0 ; 1,0], obtenido si cumple las restricciones para los vértices.

Idoneidad para los vértices

El ecosistema también debe evaluar la ubicación de los vértices, con respecto a las aristas, con el fin de evitar ciertas deformidades, como las que presentaría el cuadrángulo de la figura 12 si el vértice D estuviese al lado izquierdo de la recta AB, o si el vértice B estuviese al lado derecho de la recta CD, etc. Para ello se emplean las ecuaciones 5, 6, 7 y 8, con las que se representan las rectas: {AB, BC, CD, DA}.

$$y_{AB}(x) = \left(\frac{y_a - y_b}{x_a - x_b}\right)x + \left[y_a - \left(\frac{y_a - y_b}{x_a - x_b}\right)x_a\right]$$

Ecuación 5. Ecuación de la recta AB

$$y_{BC}(x) = \left(\frac{y_b - y_c}{x_b - x_c}\right)x + \left[y_b - \left(\frac{y_b - y_c}{x_b - x_c}\right)x_b\right]$$

Ecuación 6. Ecuación de la recta BC

$$y_{CD}(x) = \left(\frac{y_c - y_d}{x_c - x_d}\right)x + \left[y_c - \left(\frac{y_c - y_d}{x_c - x_d}\right)x_c\right]$$

Ecuación 7. Ecuación de la recta CD

$$y_{DA}(x) = \left(\frac{y_d - y_a}{x_d - x_a}\right)x + \left[y_d - \left(\frac{y_d - y_a}{x_d - x_a}\right)x_d\right]$$

Ecuación 8. Ecuación de la recta DA

En las ecuaciones 9, 10, 11, 12, 13 y 14 se establecen las restricciones para los dos vértices por los cuales no pasa cada recta. Nótese que las restricciones con respecto a las rectas AB y CD dependen del signo de su respectiva pendiente. Con el cumplimiento de cada restricción se gana 0,125 en el puntaje que penaliza la idoneidad promedio de las aristas.

$$y_{BC}(x_a) > y_a$$

$$y_{BC}(x_d) > y_d$$

Ecuación 9. Restricciones para A: (x_a ; y_a) y D: (x_d ; y_d), con respecto a la recta BC

$$y_{DA}(x_b) < y_b$$

$$y_{DA}(x_c) < y_c$$

Ecuación 10. Restricciones para B: (x_b ; y_b) y C: (x_c ; y_c), con respecto a la recta DA

$$y_{AB}(x_c) > y_c \;\; ; \;\; \left(\frac{y_a - y_b}{x_a - x_b}\right) > 0$$

$$y_{AB}(x_c) < y_c \;\; ; \;\; \left(\frac{y_a - y_b}{x_a - x_b}\right) < 0$$

Ecuación 11. Restricción para C: (x_c ; y_c), con respecto a la recta AB

$$y_{AB}(x_d) > y_d \;\; ; \;\; \left(\frac{y_a - y_b}{x_a - x_b}\right) > 0$$

$$y_{AB}(x_d) < y_d \;\; ; \;\; \left(\frac{y_a - y_b}{x_a - x_b}\right) < 0$$

Ecuación 12. Restricción para D: (x_d ; y_d), con respecto a la recta AB

$$y_{CD}(x_a) < y_a \;\; ; \;\; \left(\frac{y_c - y_d}{x_c - x_d}\right) > 0$$

$$y_{CD}(x_a) > y_a \;\; ; \;\; \left(\frac{y_c - y_d}{x_c - x_d}\right) < 0$$

Ecuación 13. Restricción para A: (x_a ; y_a), con respecto a la recta CD

$$y_{CD}(x_b) < y_b \;\; ; \;\; \left(\frac{y_c - y_d}{x_c - x_d}\right) > 0$$

$$y_{CD}(x_b) > y_b \;\; ; \;\; \left(\frac{y_c - y_d}{x_c - x_d}\right) < 0$$

Ecuación 14. Restricción para B: (x_b ; y_b), con respecto a la recta CD

Idoneidad para la diagonal AC

Pese a que en la generación inicial los individuos procuran codificar su propio cuadrángulo, con cada vértice en alguno de los cuadrantes cartesianos, semejante a como se ilustra en la figura 12, la evolución del algoritmo permitiría que el cuadrángulo gire libremente. Esto no es deseable porque se necesita que los vértices del cuadrángulo correspondan con sus respectivos nodos en la matriz unificada de distancias. Por lo tanto, la pendiente de la diagonal AC (φ) siempre debe ser positiva y puede establecérsele cierto valor deseable (Φ). Así, la función de la ecuación 15 permite establecerle un valor de pendiente a la diagonal AC, y el valor obtenido de idoneidad se puede promediar junto con las idoneidades de las aristas, comentadas antes.

$$F(\varphi) = \begin{cases} e^{-\left[\frac{(\varphi-\Phi)^2}{(0{,}167442*\Phi)^2}\right]} ; \ \Phi \geq 0{,}1 \\ e^{-\left[\frac{(\varphi-\Phi)^2}{2{,}803682E-4}\right]} \ ; \ \Phi < 0{,}1 \end{cases}$$

Ecuación 15. Idoneidad genética para la pendiente de la diagonal (φ)

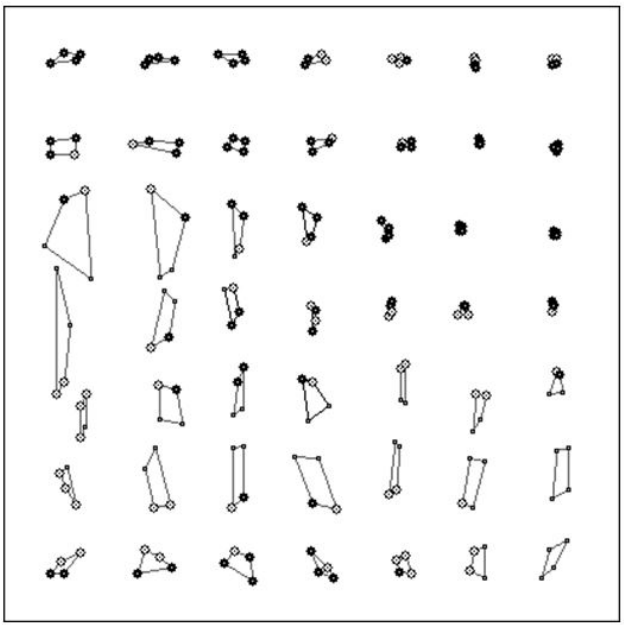
Implementación del agente

Para la implementación del agente acá reportado, primero se diseñó y se ajustó en hojas de cálculo de Microsoft® Excel 2010, y luego se programó en lenguaje *Assembler*, empleando una máquina con Pentium 4, a 2,4 GHz. La población del algoritmo genético se estableció en 200 individuos y, tras ejecutar el programa con el DOSBox, versión 0.74, se lograron obtener los [7 x 7] cuadrángulos de un SOM de [8 x 8] neuronas cognitivas, en un promedio de 40 minutos.

Resultados

Con el fin de contrastar dos SOM, cada uno de ellos se entrenó con ciertos datos tetradimensionales (4D), diferentes. En la figura 15 se visualiza la clusterización del primer SOM, correspondiente con la matriz-U detallada en la figura 16.

Figura 15. Cuadrángulos de apoyo a la matriz-U detallada en la figura 16



Fuente: elaboración propia.

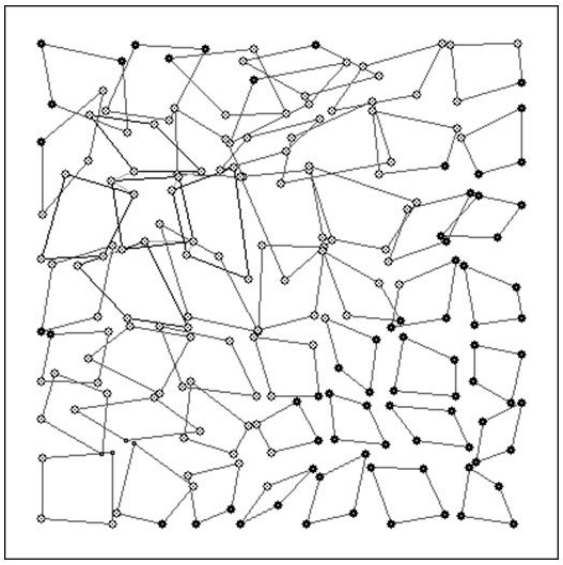
Figura 16. Matriz-U resaltando el clúster existente en el primer conjunto de datos 4D

0,207188	0,139143	0,205586	0,189694	0,132898	0,095578	0,065847	0,050862
0,237368	0,254357	0,226281	0,167775	0,136106	0,080643	0,056135	0,051005
0,373291	0,553042	0,405886	0,257616	0,170732	0,079821	0,042706	0,038347
	0,698194	0,358682	0,232985	0,133502	0,07426	0,057027	0,034298
0,73031	0,401682	0,339067	0,298832	0,185184	0,206326	0,160711	0,112076
0,246475	0,344292	0,404468	0,410169	0,367625	0,330474	0,318301	0,303751
0,223964	0,33118	0,347189	0,360439	0,318215	0,273593	0,349812	0,387512
0,181598	0,31978	0,361133	0,30108	0,173828	0,207527	0,272075	0,297117

Fuente: elaboración propia.

La figura 17 muestra la clusterización del segundo SOM, correspondiente con la matriz-U detallada en la figura 18.

Figura 17. Cuadrángulos de apoyo a la matriz-U detallada en la figura 18



Fuente: elaboración propia.

Figura 18. Matriz-U con un supuesto clúster en el segundo conjunto de datos 4D

0,877431	0,914576	0,961122	0,946842	0,968357	0,946265	0,847436	0,658384
0,880031	0,884314	0,83162	0,761406	0,855575	0,897804	0,76622	0,656125
0,952262	0,854438	0,816035	0,881371	0,920858	0,912167	0,627277	0,570334
0,895253	0,845726	0,923193		0,824827	0,695308	0,711836	0,53114
0,712548	0,748722	0,897424	0,844859	0,713431	0,635512	0,607511	0,533718
0,615765	0,762906	0,840702	0,653072	0,557775	0,609473	0,553342	0,580679
0,722579	0,844515	0,708155	0,544624	0,591874	0,636526	0,543828	0,510003
0,809386	0,777914	0,549774	0,516579	0,568069	0,659644	0,636118	0,572824

Fuente: elaboración propia.

La figura 18 pone en evidencia lo engañosa que puede ser la matriz-U, pues, al detallar las distancias reales en la figura 17, la *iluminación* en la figura 18 no tiene sentido. Este segundo conjunto de datos 4D parecen ser muy heterogéneos entre sí, pues no exhiben clústeres en la figura 17, y puede decirse que la región *iluminada* en la figura 18 no representa semejanza alguna entre los datos observados.

Conclusiones

El aporte específico de este trabajo es la visualización de las distancias reales en un mapa auto-organizado de Kohonen (SOM) que, aunque no se presenta en forma de malla –como suele hacerse convencionalmente–, sí garantiza plena correspondencia entre las distancias n-dimensionales (nD), propias del SOM, y las distancias bi-dimensionales (2D) en la visualización resultante. Además, se garantiza también que los nodos en la visualización del SOM corresponden tal cual con los nodos en la matriz-U. En cuanto a la complejidad temporal, hipotéticamente no importa la cantidad de dimensiones en los datos originales: este algoritmo genético siempre tardará algún lapso, pero este no crecerá en el caso de que aumente la cantidad de dimensiones en los datos de entrenamiento del SOM, pues durante la evolución de dicho algoritmo solo se calculan distancias euclídeas 2D.

Al reflexionar un poco con respecto al algoritmo genético, se hace evidente que la evolución artificial de una población de individuos artificiales en un ecosistema artificial puede verse como el proceso mediante el cual un agente *piensa*, con utilidad racional, y deja abierta la posibilidad de que nuestra propia intelectualidad corresponda a la evolución de diversas ideas, en un tipo de *ecosistema mental*. Sin embargo, vale la pena mencionar que al haber vislumbrado cierto paralelo entre el conjunto neuronas en el cerebro y el conjunto de individuos en el algoritmo genético de un agente artificial, no se pretende modelar lo racional con genes. Al menos no directamente.

Como trabajo futuro puede considerarse ensamblar los cuadrángulos que entrega la técnica acá reportada y contrastarla con la técnica de Sammon (1969).

Referencias

Bentley, P. y Corne, D. (Eds.). (2002). *Creative Evolutionary Systems*. Massachusetts: Morgan Kaufmann.

Bishop, C. (2006). *Pattern Recognition and Machine Learning*. Berlín: Springer.

Floreano, D. y Mattiussi, C. (2008). *Bio-inspired Artificial Intelligence: Theories, Methods and Technologies*. Cambridge: MIT Press.

Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Nueva York: Addison-Wesley.

Hart, W., Krasnogor, N. y Smith, J. (Eds.). (2005). *Recent Advances in Memetic Algorithms*. Berlín: Springer.

Holland, J. (1975). *Adaptation in Natural and Artificial Systems*. Michigan: University of Michigan Press.

Jacob, B. (1995). *Composing with Genetic Algorithms*. Conferencia presentada en the International Computer Music Conference, Banff, Canada.

Matić, D. (2010). A Genetic Algorithm for Composing Music. *Yugoslav Journal of Operations Research*, *20*(1), 157-177.

Russell, S. y Norvig, P. (1995). *Artificial Intelligence: A Modern Approach*. Nueva Jersey: Prentice-Hall.

Sammon, J. (1969). A Nonlinear Mapping for Data Structure Analysis. *IEEE. Transactions on Computers*, *18*(5), 401-409.