

# Algoritmo Evolutivo de Extracción de Reglas de Asociación aplicado a un Problema de Marketing\*

M.J. Del Jesus, P. González<sup>1</sup>, F. Herrera<sup>2</sup>, M. Mesonero<sup>3</sup>

**Resumen**—En el campo empresarial un problema abierto es el estudio de la influencia que las variables de planificación de un certamen ferial tienen sobre el nivel de consecución de los objetivos planteados para el mismo. En este trabajo se aborda este problema de Minería de Datos con una propuesta evolutiva que permite extraer un conjunto de reglas de asociación difusas y/o nítidas en forma normal disyuntiva. El proceso propuesto incluye un Algoritmo Genético en un esquema iterativo que extrae reglas mientras queden ejemplos sin describir y las reglas obtenidas superen un nivel de confianza mínimo especificado por el usuario. Además incluye un proceso de optimización que permite mejorar cada una de las reglas obtenidas tras el Algoritmo Genético. La aplicación de la propuesta a este problema real permite obtener un conjunto de reglas comprensibles, precisas y que determinan conocimiento de interés sobre el problema.

**Palabras clave**-- Minería de Datos, Reglas de Asociación, Algoritmos Genéticos

## I. INTRODUCCIÓN

La Minería de Datos consiste en la extracción automática de conocimiento de alto nivel de un conjunto de datos reales [7]. Forma parte de un proceso más amplio, el descubrimiento de conocimiento, en el que se incluyen distintas etapas: comprensión del problema, comprensión de los datos, pre-procesamiento (o preparación) de los datos, minería de datos y post-procesamiento (evaluación e interpretación de los modelos).

En Minería de Datos se pueden distinguir distintos tipos de problemas en función del objetivo del proceso de extracción de conocimiento [7][9]:

- *Clasificación*, cuyo objetivo es predecir el valor para un atributo objetivo especificado por el usuario en base a valores de otros atributos predictivos.
- *Modelado de dependencias*, que se puede considerar una generalización de la tarea de clasificación ya que intenta predecir el valor de varios atributos.
- *Agrupamiento*, una forma de aprendizaje no supervisado en la que el algoritmo de Minería de Datos debe determinar las clases dividiendo el conjunto de ejemplos en grupos.
- *Descubrimiento de reglas de asociación*, en la que se obtiene conocimiento interesante para los usuarios en forma de reglas de asociación que reflejan relaciones entre los atributos presentes en los datos.

Los Algoritmos Genéticos (AGs) [13][11] tienen un carácter de búsqueda global que hace que sean especialmente adecuados para resolver problemas presentes en las distintas etapas del proceso de descubrimiento de conocimiento en general y de Minería de Datos en particular [9]. En concreto, en procesos de extracción de reglas tratan de forma adecuada las interacciones entre atributos porque evalúan una regla como un todo mediante la función de adaptación en lugar de evaluar el impacto de añadir/eliminar una condición de una regla, como ocurre en los procesos de búsqueda local incluidos en la mayoría de los algoritmos de inducción de reglas y árboles de decisión.

En este trabajo se presenta un algoritmo evolutivo cuyo objetivo es la obtención de un conjunto de reglas de asociación para las que se ha fijado el atributo que debe aparecer en el consecuente. Se consideran reglas de asociación porque el objetivo en el proceso de extracción no es obtener un conjunto de reglas con alta capacidad predictiva sino un conjunto de reglas que aporten conocimiento interesante sobre un problema con relaciones desconocidas entre las variables [8]. El algoritmo tiene capacidad para extraer reglas difusas o nítidas en función de que las variables del problema sean continuas o nominales. Esta propuesta es especialmente adecuada para el

<sup>1</sup> Departamento de Informática. E.P.S. Universidad de Jaén.  
e-mail: [mjjesus@ujaen.es](mailto:mjjesus@ujaen.es), [pglez@ujaen.es](mailto:pglez@ujaen.es)

<sup>2</sup> Departamento de Ciencias de la Computación e Inteligencia Artificial. E.T.S.I. Informática. Universidad de Granada.  
e-mail: [herrera@decsai.ugr.es](mailto:herrera@decsai.ugr.es)

<sup>3</sup> Departamento de Organización y Marketing. Facultad Ciencias Empresariales. Mondragon Unibertsitatea.  
e-mail: [mmesoner@eteo.mondragon.edu](mailto:mmesoner@eteo.mondragon.edu)

\* Este trabajo está financiado por el Ministerio de Ciencia y Tecnología bajo los proyectos TIC-2002-04037-C05-01 y 04.

problema de marketing al que nos enfrentamos en el que el objetivo es extraer reglas que relacionen las variables de decisión con una variable que establece la eficacia final del stand. Estas reglas pueden permitir extraer conclusiones y planificar de mejor manera la organización y funcionamiento de los certámenes feriales.

Para ello el artículo se organiza de la siguiente forma. En la Sección II se describe el problema real de extracción de conocimiento en certámenes feriales. En la Sección III se hace una breve revisión de la utilización de los AGs en la tarea de descubrimiento de reglas y, en la Sección IV se expone la propuesta evolutiva. La experimentación y el análisis de los resultados obtenidos se describen en la Sección V. Por último, en la Sección VI se describen las conclusiones del estudio realizado.

## II. EXTRACCIÓN DE CONOCIMIENTO EN CERTÁMENES FERIALES

En este trabajo se aplican herramientas evolutivas de extracción de conocimiento a un problema de marketing estudiado en el Departamento de Organización y Marketing de la Universidad de Mondragón: La extracción de información útil sobre certámenes feriales.

Las empresas consideran los certámenes feriales un instrumento que facilita la consecución de objetivos comerciales tales como el contacto con los clientes actuales, la captación de nuevos clientes potenciales, la realización de pedidos o la mejora de la imagen corporativa entre otros [16]. Uno de los principales inconvenientes de este tipo de certámenes es la elevada inversión que suponen en términos tanto económicos como de tiempo. A esta inversión a veces se une una falta de planificación que enfatiza la sensación de que las ferias no son más que un “gasto” que las compañías han de afrontar por motivos varios (tradicición, exigencia clientes, no dar la sensación de que las cosas van mal, etc.) [15]. Es conveniente, por tanto, la extracción automática de información sobre las variables implicadas que permita obtener datos desconocidos determinantes en parte de la eficacia de los stands de un certamen.

En la Bienal de Máquina-Herramienta celebrada en Bilbao en Marzo de 2002, se recogió información sobre 104 variables de 228 expositores. De este conjunto de variables 4 de ellas son continuas y el resto son variables nominales, resultado de una discretización experta. Además, para cada uno de los expositores, en base a distintos criterios de marketing, se determinó la eficacia global de dicho stand en Eficacia Alta, Media o Baja en función del nivel de consecución de los objetivos planteados para el certamen. Sobre este conjunto de variables, los expertos realizaron un proceso de selección basado en el conocimiento para determinar un

subconjunto de 18 variables de las que tenían interés en extraer relaciones de asociación.

El objetivo del proceso de extracción de conocimiento para este problema es determinar la aportación que las distintas variables de planificación ferial ejercen sobre los resultados obtenidos por el expositor.

## III. ALGORITMOS GENÉTICOS PARA EXTRACCIÓN DE REGLAS

Se han desarrollado múltiples propuestas evolutivas para la extracción de reglas de distintos tipos, de clasificación, asociación o dependencias funcionales. Para el problema al que nos enfrentamos, el objetivo es generar reglas en las que en el consecuente aparezca un único atributo (y establecido a priori), por que en esta sección haremos referencia a las propuestas evolutivas para la extracción de reglas de clasificación y asociación.

Cualquier propuesta de AG de extracción de reglas debe determinar el esquema de representación utilizado para codificar cada una de las soluciones, los operadores genéticos y la función de adaptación.

### A. Esquema de representación

Los AGs siguen dos enfoques respecto a la forma de codificar reglas dentro de una población de individuos [4]:

- el enfoque “*Cromosoma = Regla*”, y
- el enfoque “*Cromosoma = Base de Reglas*” también denominado enfoque *Pittsburgh*.

En el primero cada individuo codifica una única regla y en el segundo, un conjunto de reglas.

Dentro del enfoque “*Cromosoma = Regla*” existen dos propuestas genéricas:

- el enfoque *Michigan* en el que cada individuo codifica una única regla pero la solución final será la población final o un subconjunto de la misma por lo que es necesario evaluar el comportamiento del conjunto de reglas al completo y la aportación de la regla individual al mismo, y
- el enfoque *IRL (Iterative Rule Learning)* en el que cada cromosoma representa una regla, pero la solución del AG es el mejor individuo y la solución global está formada por los mejores individuos de una serie de ejecuciones sucesivas.

La elección del esquema de representación depende, entre otros aspectos, de la tarea a realizar por parte del algoritmo de Minería de Datos y, por tanto, del tipo de regla a descubrir.

Si el objetivo es determinar un conjunto de reglas de clasificación, se debe evaluar el

comportamiento del conjunto de reglas al completo más que la calidad de una regla individual por lo que el esquema de representación más adecuado es el enfoque “*Cromosoma = Base de Reglas*” que considera la interacción entre las reglas. GABIL [6] y GIL [14] son ejemplos de AGs para clasificación que utilizan este esquema de representación. Este enfoque implica el uso de individuos con una longitud superior (y a menudo variable) lo que provoca un incremento del costo computacional del algoritmo y la modificación de los operadores genéticos. Esto hace que dentro del campo de la clasificación se hayan diseñado AGs con el enfoque “*Cromosoma = Regla*” como COGIN [12] y REGAL [10] que utilizan individuos con una sintaxis más reducida simplificando el diseño de los operadores genéticos. Este enfoque tiene dos inconvenientes: La dificultad del cálculo del valor de la función de adaptación, ya que cada regla se evalúa individualmente y es difícil determinar la calidad del conjunto de reglas al completo; y por otra parte, puesto que el objetivo es obtener un conjunto de reglas, no se puede permitir que el AG converja hacia un único individuo y se necesita alguna técnica de nichos [2] que fomente la existencia de individuos distintos dentro de la población.

En procesos de descubrimiento de reglas de asociación es más adecuado el enfoque “*Cromosoma = Regla*” ya que el objetivo es encontrar un conjunto reducido de reglas en las que la calidad de cada regla se evalúa de forma independiente al del resto. Este es el enfoque de codificación que se utiliza en esta propuesta evolutiva.

### B. Operadores genéticos

Además de los operadores genéticos clásicos - adaptados al esquema de representación utilizado-, existen distintas propuestas de operadores genéticos diseñados específicamente para AGs cuyo objetivo es el descubrimiento de reglas.

Para AGs con el esquema de representación “*Cromosoma = Regla*” es necesario evitar la convergencia de la población a un individuo particular. Para ello en REGAL se presenta un operador de selección denominado sufragio universal que elige los individuos que deben participar en la recombinación mediante una elección por sufragio de los ejemplos de entrenamiento. Cada ejemplo vota por la regla que determina la zona del espacio de búsqueda en la que está incluido de forma proporcional a la medida de adaptación de la regla. Es una forma de nichos que potencia la evolución de diferentes reglas que cubren diferentes zonas del espacio de búsqueda.

Una de las operaciones más adecuadas en el proceso de descubrimiento de reglas es la operación de generalización-especialización. En la bibliografía

especializada se existen propuestas en este sentido integradas en operadores de cruce, de mutación o bien operadores diseñados específicamente para ello.

### C. Función de adaptación

En el proceso de descubrimiento de reglas se intenta conseguir reglas con capacidad predictiva alta, comprensibles e interesantes. Esto se puede alcanzar en el contexto evolutivo mediante una combinación lineal con pesos de estas tres medidas [17], mediante una matriz de confusión o bien, mediante un AG con enfoque multiobjetivo [5] que orienta el proceso evolutivo hacia la búsqueda de un conjunto de soluciones no-dominadas en los distintos objetivos marcados.

## IV. ALGORITMO GENÉTICO PARA EXTRACCIÓN DE REGLAS

En este apartado se describe una propuesta evolutiva para la extracción de reglas de asociación. Se trata de un AG Iterativo para la extracción de reglas de asociación en forma normal disyuntiva (DNF) que representan en el antecedente una conjunción de variables y para cada variable una disyunción de valores.

El algoritmo permite generar reglas difusas y/o nítidas, para problemas con variables continuas y/o nominales. Las reglas DNF, difusas o nítidas, son un tipo de regla que permite representar el conocimiento de forma sencilla y comprensible. La disyunción de valores para una variable facilita la extracción de reglas genéricas en problemas con variables nominales, y permite flexibilizar la discretización en variables en las que se haya realizado este tipo de pre-procesamiento o cambiar la granularidad en el caso de variables que se hayan tratado como variables lingüísticas [19].

El objetivo de este proceso evolutivo es extraer para cada valor de la variable objetivo un número variable de reglas diferentes que expresen información sobre la mayoría de los ejemplos del conjunto de partida. Para ello el AG se incluye en un proceso iterativo que permite la obtención de varias reglas mientras las reglas generadas alcancen un nivel mínimo de confianza y describan información sobre zonas del espacio de búsqueda en las que aún quedan ejemplos no descritos por las reglas generadas en las iteraciones anteriores.

El mecanismo de repetición debe potenciar la generación de reglas distintas (en el sentido de que expresen información sobre grupos distintos de ejemplos). En esta propuesta esto se consigue penalizando -una vez obtenida una regla- el conjunto

de ejemplos representados por la misma para la generación de futuras reglas.

Se incorpora una etapa de post-procesamiento al AG para optimizar la regla obtenida con el objetivo de incrementar la completitud manteniendo el grado de confianza de la misma.

El esquema completo del algoritmo es el siguiente:

```

INICIO
  Cto_Reglas ← ∅
  REPETIR
    Ejecutar el AG obteniendo la regla R
    Búsqueda Local (R)
    Cto_Reglas ← Cto_Reglas + R
    Modificar el conjunto de ejemplos
  MIENTRAS confianza(R) >= confianza_min y
    R cubra ejemplos nuevos
FIN

```

A continuación se describen con más detalle los elementos que conforman esta propuesta.

### 1) Esquema de representación

El AG tiene como objetivo descubrir reglas de asociación en las que la variable del consecuente se define a priori, es decir, el objetivo está prefijado. Para ello se codifica cada solución candidata mediante el enfoque “*Cromosoma = Regla*” representando en el cromosoma sólo el antecedente y asociando todos los individuos de la población con el mismo valor de la variable objetivo.

Esta forma de codificar la variable objetivo, utilizada en algoritmos de extracción de reglas de clasificación [14], obliga a realizar múltiples ejecuciones del algoritmo evolutivo para descubrir reglas de las distintas clases. En la bibliografía especializada se ha considerado la variable objetivo también de otras formas:

- codificándolo dentro del genoma del individuo [6] de forma que también pueda evolucionar, o
- eligiendo para cada regla el valor de la variable objetivo de forma determinística: la clase que tenga más representantes en el conjunto de ejemplos que verifican el antecedente de la regla [10] o la clase que maximiza la función de adaptación del individuo [17].

En problemas como el que presentamos, el enfoque adoptado es adecuado porque se necesita un conjunto de reglas para todos y cada uno de los valores de la variable objetivo, y las dos alternativas anteriores no aseguran la extracción de información sobre todas las clases.

Como se ha mencionado se sigue el esquema de representación “*Cromosoma = regla*” y en particular, el enfoque IRL que incluye el AG en un

esquema iterativo para obtener en cada iteración la regla representada por el mejor individuo.

El antecedente de una regla DNF es una condición compuesta por una conjunción entre atributos y varias disyunciones sobre los valores del atributo. Para representar este tipo de antecedente, se ha optado por un esquema de codificación binario, en el que por cada atributo se almacena un bit para cada uno de los valores que puede tomar, de forma que si el bit correspondiente tiene valor 0 indica que no pertenece a la condición y si tiene valor 1 que sí pertenece [6]. Si en un individuo todos los bits correspondientes a un atributo tienen valor 1 esto indica que dicha variable no es relevante para la información aportada en la regla (cualquier valor de la variable verifica la condición de la regla), por lo que esta variable se ignora. De esta forma, se almacena para cada variable una lista de enteros (tantos como valores posibles puede tomar la variable) que pueden tomar el valor 0 ó 1. En la última posición se guarda (a efectos de eficiencia) si la variable interviene o no en la regla (no interviene si están todos los valores, o no está ninguno).

### 2) Función de adaptación

La función de evaluación en nuestra propuesta tiene la misma expresión que la utilizada en el AG propuesto en [1], salvo que considera sólo dos factores, la confianza y completitud de la regla según la siguiente expresión:

$$fitness(c) = \frac{\omega_1 \cdot Completitud(c) + \omega_2 \cdot Confianza(c)}{\omega_1 + \omega_2}$$

*Confianza.* Determina la precisión de la regla ya que refleja el grado con el que los ejemplos pertenecientes a la zona del espacio delimitado por el antecedente verifican la información indicada en el consecuente de la regla. Para el cálculo de este factor utilizamos la expresión de precisión aportada por Quinlan en [18] si las reglas son nítidas y una modificación de la misma, utilizada frecuentemente en la generación de reglas de clasificación [3], si son difusas: la suma del grado de pertenencia de los ejemplos de la clase a la zona determinada por el antecedente dividido entre la suma del grado de pertenencia de todos los ejemplos (independientemente de la clase a la que pertenezcan) a la misma zona. Para calcular estos grados de pertenencia se utilizan funciones de pertenencia triangulares y la t-norma mínimo. En el caso de reglas no difusas, los grados de pertenencia son los correspondientes a conjuntos clásicos, 0 ó 1.

*Completitud.* Es una medida del grado de cobertura que la regla ofrece a los ejemplos de la clase. Se calcula como el cociente entre el número de nuevos ejemplos de la clase que cubre la regla y el número

de ejemplos de la clase que quedaban por cubrir. Esta forma de medir la completitud toma sentido, al utilizar el AG dentro de un proceso iterativo, para potenciar la obtención de reglas distintas en cada ejecución del AG. A partir de la segunda iteración se penaliza indirectamente aquellas reglas que cubren ejemplos pertenecientes a las zonas delimitadas por reglas obtenidas previamente: el factor de completitud solo considera ejemplos no descritos por otras reglas ya obtenidas. No se utiliza ninguna función de distancia ya que se penalizan diferencias a nivel fenotípico.

El objetivo global de la función de evaluación es orientar la búsqueda hacia reglas que maximicen la precisión, minimizando el número de ejemplos negativos y no cubiertos.

### 3) Operadores Genéticos

El AG utiliza un modelo de reproducción de estado estacionario. La recombinación se realiza a través del operador de cruce multipunto y un operador de mutación uniforme sesgado con el que la mitad de las mutaciones realizadas tienen el efecto de eliminar la variable correspondiente, para incrementar la generalidad de las reglas.

### 4) Etapa de post-procesamiento del algoritmo genético: algoritmo de búsqueda local

La etapa de post-procesamiento, que mejora la regla obtenida mediante un proceso de ascensión de colinas, modifica la regla mientras se incrementa el grado de completitud. Para ello, en cada iteración se determina la variable tal que, al eliminarla aumenta en mayor grado la completitud de la regla resultante, obteniendo así reglas más generales. Finalmente, la regla optimizada sustituirá a la original sólo si supera la confianza mínima.

## V. EXPERIMENTACIÓN

La experimentación se ha realizado sobre los datos obtenidos en la Bienal de Máquina Herramienta celebrada en Bilbao en Marzo de 2002 para los que el objetivo del algoritmo de Minería de Datos es descubrir reglas que relacionen las variables reflejadas con la eficacia (alta, media, o baja) de los expositores en función del nivel de consecución de los objetivos planteados para el certamen. Hay que destacar que la selección de variables realizada por el experto reduce el conjunto total a un subconjunto con 18 variables nominales, descritas en la Tabla 1.

V	DESCRIPCIÓN	VALORES
-	Grupo en el que se clasifica al expositor.	1. Eficacia baja 2. Eficacia media 3. Eficacia alta

0	Zona geográfica en la que se sitúa la empresa	1. Zona Norte 2. Zona Centro 3. Zona Levante
1	Sector al que pertenece el expositor	1. Arranque +Deformación 2. Resto
2	Utilidad de las ferias	1. Poca utilidad 2. Mucha utilidad
3	Nº ferias en las que se participa anualmente como expositor	1. Más de 11 2. Menos de 11
4	Existencia de objetivos para la BIEMH por escrito	1. Si 2. No
5	Realización de promoción previa a la feria	1. Si 2. No
6	Listados para informar de la presencia en la feria	1. Si 2. No
7	Seguimiento de la campaña de promoción previa	1. Si 2. No
8	Importancia concedida a las operaciones realizadas después de la feria	1. Poco importante 2. Importancia media 3. Mucha importancia
9	Importancia concedida al nº de contactos realizados con clientes actuales	1. Poco importante 2. Importancia media 3. Mucha importancia
10	Importancia concedida a la calidad de los contactos realizados	1. Poco importante 2. Importancia media 3. Mucha importancia
11	Importancia concedida al nº de visitantes en el stand	1. Poco importante 2. Importancia media 3. Mucha importancia
12	Realización de llamadas telefónicas	1. No se utiliza 2. Si se utiliza
13	Tipo de demostraciones de maquinaria	1. Funcionamiento permanente de la máquina 2. Funcionamiento esporádico 3. Exhibición de maquinaria
14	Presencia de novedades en el stand	1. Auténtica novedad 2. Perfeccionamiento del producto 3. Exhibición de catálogo 4. Otros
15	Presencia de azafatas	0. No      1. Si
16	Stand en entrada de pabellón	0. No      1. Si
17	Cercanía escaleras	0. No      1. Si

Tabla 1: Descripción de las variables del problema

La experimentación se ha realizado de la siguiente forma:

- Se ha ejecutado el algoritmo propuesto y se han comparado los resultados de esta propuesta con los obtenidos con un AG con codificación entera que extrae reglas de asociación (difusas o nítidas) sin disyunción interna para cada una de las variables. Este algoritmo es una extensión del descrito en [1] que, bajo un esquema de codificación entera, incluye las nuevas características descritas en la Sección IV.
- Para ambos algoritmos se han utilizado poblaciones de 100 individuos, y un número máximo de 5000 evaluaciones.
- Se han ejecutado 5 veces ambos algoritmos para cada una de las clases del atributo predictivo (eficacia baja, media y alta de los stands).

- Los pesos asociados a los objetivos de la función de adaptación son los siguientes: para el AG con codificación entera, los pesos son 0.5 y 0.4 para la completitud y confianza respectivamente, mientras que son 0.4 y 0.5 para el AG con codificación binaria. En el AG que evoluciona reglas DNF se aumenta el peso relativo de la confianza porque el tipo de representación tiende a generar reglas más generales y es conveniente potenciar la confianza de las mismas.

Los resultados de las mejores ejecuciones se muestran en las Tablas 2 y 3. En ambas se muestra para cada clase el conjunto de reglas obtenidas, el nivel de completitud y confianza, y la descripción de cada una de las reglas. En el caso de la Tabla 2, que corresponde al AG con codificación entera, el antecedente de las reglas es una conjunción de condiciones simples para las variables que intervienen. Las reglas que se muestran en la Tabla 3, son una conjunción de condiciones compuestas para algunas variables, de forma que un valor 1,2 por ejemplo en una casilla debe interpretarse como que la variable en cuestión toma el valor 1 o el valor 2.

Como se puede observar el AG con codificación entera genera conjuntos de reglas que cubren casi la totalidad de los ejemplos con un nivel de confianza por encima de 0.9. Las reglas obtenidas para cada una de las clases son conjuntos reglas comprensibles por el reducido número de reglas y de condiciones en el antecedente de las mismas.

El AG con codificación binaria (reglas DNF) extrae conjuntos formados por menos reglas que la versión de codificación entera (reglas simples), y más generales, con nivel de completitud elevado, por encima del 85 en algún caso. El nivel de confianza, para alguna de las reglas es más bajo, pero se mantiene por encima del nivel mínimo especificado por el usuario, 0.6. Hay que destacar, a pesar de esto, que la mayoría de las reglas tienen un nivel de confianza por encima de 0.9.

Por la propia definición del problema, los valores de la variable objetivo más interesantes corresponden a las clases 1 y 3 que identifican los stands con eficacia baja y alta. Son también, en base a los resultados obtenidos, las clases de las que es más difícil obtener información general, ya que las propuestas evolutivas necesitan un mayor número de reglas para describir todos los ejemplos. En esta situación, el AG de extracción de reglas de asociación DNF permite describir de forma adecuada los stands de eficacia baja con solo cuatro reglas, al igual que ocurre con los de eficacia media.

Estos dos conjuntos de reglas describen información general y bastante precisa sobre el problema.

Para este problema el algoritmo evolutivo de extracción de reglas de asociación tipo DNF obtiene conjuntos de reglas que describen conocimiento comprensible y suficientemente general sobre distintas variables planificadoras en la gestión y organización de certámenes feriales.

## VI. CONCLUSIONES Y TRABAJO FUTURO

Los AGs realizan una búsqueda global e independiente del dominio, aspecto que los convierte en una herramienta robusta, escalable y aplicable en distintas etapas del proceso de extracción de conocimiento.

En este trabajo se presenta una propuesta evolutiva aplicada al proceso de extracción de reglas de asociación. El algoritmo permite generar un conjunto de reglas de asociación difusas y/o nítidas en forma normal disyuntiva. Para ello incluye un AG en un esquema iterativo que extrae reglas mientras queden ejemplos sin cubrir y las reglas obtenidas superen un determinado nivel de confianza especificado por el usuario.

La aplicación del algoritmo a un problema real ha determinado un conjunto sencillo de reglas con estructura también sencilla. Es un algoritmo que permite extraer conocimiento comprensible en forma de pocas reglas nítidas o difusas con pocas variables.

La experimentación muestra que el mismo algoritmo pero adaptado para la extracción de reglas no DNF (sin disyunción interna de valores para una variable) obtiene buenos resultados aunque la solución final determinada tiene un mayor número de reglas bastante más específicas.

Como líneas de trabajo futuro, se está trabajando en el desarrollo de un nuevo criterio objetivo de interés para reglas DNF, y en el diseño de algoritmos evolutivos multiobjetivo que permitan evolucionar hacia un conjunto de reglas que se encuentren en la frontera del pareto.

## REFERENCIAS

- [1] Aguilera, J.J., del Jesus, M.J., González, P., Herrera, F., Navío, M., Sáinz, J. "Extracción Evolutiva de Reglas de Asociación en un Servicio de Urgencias Psiquiátricas". Actas del II Congreso español sobre Metaheurísticas, Algoritmos evolutivos y bioinspirados. pp 548-555. 2003.
- [2] Beasley, D., Bull, D.R., Martin R.R. "A sequential niche technique for multimodal function optimization". Evolutionary Computation. Volumen 1. pp. 101-125. 1993.
- [3] Cerdón, O., del Jesus, M.J., Herrera, F., "Genetic Learning of Fuzzy Rule-based Classification Systems Co-operating with Fuzzy Reasoning

Methods". *International Journal of Intelligent Systems*. Volumen 13. Número 10/11. pp. 1025-1053. 1998.

[4] Cordón, O., Herrera, F. Hoffmann, F., Magdalena, L. *Genetic Fuzzy Systems. Evolutionary tuning and learning of fuzzy knowledge bases*. World Scientific. 2001.

[5] Deb, K. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons. 2001.

[6] De Jong, K.A., Spears, W.M., Gordon, D.F. "Using genetic algorithms for concept learning", *Machine Learning*. Volumen 13. pp. 161-188. 1993.

[7] Fayad, U.M., Piatetsky-Shapiro, G., Smyth, P., "From data mining to knowledge discovery: an overview". En: Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R.: *Advances in Knowledge Discovery & Data Mining*. AAAI/MIT. pp. 1-34. 1996.

[8] Freitas, A.A. "Understanding the crucial differences between classification and discovery of association rules. A position paper". *ACM SIGKDD Explorations*. Volumen 2. Número 1. pp. 65-69. 2000.

[9] Freitas, A.A., *Data Mining and Knowledge Discovery with Evolutionary Algorithms*, Springer. 2002.

[10] Giordana, A., Neri, F. "Search-intensive concept induction". *Evolutionary Computation*. Volumen 3. Número 4. pp. 375-416. 1995.

[11] Goldberg, D.E. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley. 1989

[12] Greene, D.P., Smith, S.F. "Competition-based induction of decision models from examples". *Machine Learning*. Volumen 13. pp. 229-257. 1993.

[13] Holland, J.H. *Adaptation in Natural and Artificial Systems*. University of Michigan Press. 1975.

[14] Janikow, C.Z. "A knowledge-intensive genetic algorithm for supervised learning". *Machine Learning*. Volumen 13. pp. 189-228. 1993.

[15] Miller, S. *Saque el máximo provecho de las ferias*. Traducido por Carlos Martínez Rueda, Ediciones Urano. 2003.

[16] Navarro, F. *Estrategias de Marketing Ferial*, ESIC Editorial. 2001

[17] Noda, E., Freitas, A.A., Lopes, H.S. "Discovering Interesting Prediction Rules with a Genetic Algorithm", in *Proceedings of the Congress on Evolutionary Computation* Vol. 2. pp. 1322—1329. 1999.

[18] Quinlan, J.R., *Generating production rules Machine Learning*. Morgan Kaufmann. 1987.

[19] Zadeh, L.A. "The concept of a linguistic variable and its applications to approximate reasoning, Parts I, II, III". *Information Sciences* Vol. 8-9. pp. 199-249, 301-357, 43-80, 1975.

CLASE	Compleitud	Confianza	v0	v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	v11	v12	v13	v14	v15	v16	v17
1	3,158	100,000		1					2					3	2					
	5,263	100,000		1				1			1									
	6,316	100,000	1	1							2							0		
	7,368	100,000		2	2		1					3	3					0		
	7,368	100,000				1			2		3									
	9,474	100,000	2							2									1	
	5,263	100,000		1	1			1		3										
	6,316	100,000		2	2								1					0		
	2,105	100,000		2				1	2			3					1	1	0	
	6,316	100,000			2				2		3					1		1		
	16,842	100,000				1	1											0		
55,789	69,737	1				1														
2	10,309	100,000	1			2	2							3		1			0	
	32,990	100,000			1	2						2								
	6,186	100,000											2	2						0
	12,371	100,000		1	1	2							1							
	74,227	77,419			1	2														
	2,062	100,000	3		2		2		1		3									
	15,464	100,000	2	1															1	
3	8,333	100,000			2		2				2		3							
	5,556	100,000				1							3							0
	2,778	100,000	2	2	1	2	1													
	5,556	100,000	1		2								2							1
	11,111	100,000		1	2							3		2	2	1		0		
	8,333	100,000			2										2	3		1		
	5,556	100,000	3																1	
	5,556	100,000			2	1	2	1											0	
	5,556	100,000	1				2					3		3				2		
	5,556	100,000	3		2		1													
	2,778	100,000									2								1	0
	8,333	100,000			2		2					2								1

Tabla 2: Resultados obtenidos con el Algoritmo Genético con codificación entera.

Clase	Compleitud	Confianza	v0	v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	v11	v12	v13	v14	v15	v16	v17
1	54,737	75,362										1,3					4			
	17,895	94,444	1,3						2		1,3	1,2				1,2	2,4			0
	14,737	93,333								1					1		4			
	6,316	100,000			2					1			3				1		1	
2	77,320	74,257				2	2													
	85,567	64,844			1															
	13,402	92,857					1				2,3	1,3	1		1				0	
	28,866	100,000						1				2		3						
3	11,111	100,000	1,3			2			2	1			2,3		1	2,3		1		
	11,111	100,000			2	2			1					2,3			1,3	1	1	
	5,556	100,000	3							1										
	8,333	100,000	3		2							1,2								
	5,556	100,000	1,3	1												3		1		0
	2,778	100,000	1,3							1	1			1,3						1
	5,556	100,000	3								1,3		2,3	1,2			1,4			
	8,333	100,000	1,3	1				1		2				1,2		2	1,3			
	5,556	100,000		2	2	2			1		3						2			0
	2,778	100,000	3						1							1				
5,556	100,000	1			1		1						3	2,3		3				

Tabla 3: Resultados obtenidos con el Algoritmo Genético con codificación binaria.