Un nuevo algoritmo Evolutivo Multi-Objetivo para la extracción de Reglas de Asociación Cuantitativas Positivas y Negativas

D. Martín[1] *, A. Rosete[1], J. Alcalá-Fdez[2], and F. Herrera[2]

[1] Dpto. de Inteligencia Artificial e Infraestructura de Sistemas,
Instituto Superior Politécnico J.A Echeverría, Cujae,
19390 La Habana, Cuba
{dmartin,rosete}@ceis.cujae.edu.cu
[2] Dpto. de Ciencias de la Computación e Inteligencia Artificial,
CITIC-UGR, Universidad de Granada,
18071 Granada, España
{jalcala,herrera}@decsai.ugr.es

Resumen Los métodos de extracción de reglas de asociación cuantitativas han sido normalmente enfocados para generar reglas de asociación positivas sin tener en cuenta las reglas de asociación negativas, las cuales pueden ser interesantes para el usuario al indicar la ausencia de un conjunto de ítems ante la presencia de otros. Además, muchos de estos métodos utilizan un único objetivo para medir la calidad de las reglas, sin embargo, recientemente algunos investigadores han planteado la extracción de reglas de asociación como un problema multi-objetivo para poder tener en cuenta varios objetivos en el proceso de extracción. En este trabajo nosotros proponemos MOPNAR, un nuevo algoritmo evolutivo multi-objetivo para extraer reglas de asociación cuantitativas positivas y negativas de alta calidad, maximizando tres objetivos: rendimiento, interés y comprensibilidad. Este algoritmo extiende el modelo evolutivo MOEA/D-DE para realizar un aprendizaje evolutivo de los intervalos de los atributos y una selección de condiciones para cada regla. Además, esta propuesta introduce en el modelo evolutivo un proceso de especialización y una población externa para almacenar todas las reglas no dominadas encontradas, provocar diversidad y mejorar el cubrimiento de la base de datos. Los resultados obtenidos sobre siete bases de datos reales muestran la efectividad del método propuesto.

Keywords: Minería de Datos, Reglas de Asociación Cuantitativas, Reglas de Asociación Negativas, Algoritmo Evolutivo Multi-Objetivo, MOEA/D-DE

* Este trabajo está soportado en parte por el proyecto nacional TIN2011-28488, por el programa CEI del MICINN bajo el proyecto PYR-2012-13 CEI BioTIC GENIL (CEB09-0010), y por el proyecto regional P10-TIC-6858.

1. Introducción

La Minería de Datos (MD) es el proceso utilizado para el descubrimiento de conocimiento de alto nivel en el mundo real a través de la obtención de información de conjuntos de datos grandes y complejos. Una de las técnicas de MD que más ha sido utilizada para extraer conocimiento interesante a partir de Bases de Datos (BD) es el descubrimiento de reglas de asociación [20]. Estas reglas representan e identifican dependencias entre los atributos de una BD y son definidas como $X \rightarrow Y$, donde X y Y son conjuntos de ítems (parejas atributo-valor) y cumplen que $X \cap Y = \oslash$.

Muchos trabajos previos para la extracción de reglas de asociación han sido enfocados sobre BDs con valores binarios, sin embargo las BDs en las aplicaciones del mundo real están compuestos normalmente por valores cuantitativos. Debido a ello, varios estudios han sido presentados para extraer reglas de asociación cuantitativas (RACs) a partir de BDs con valores cuantitativos [32]. Muchos de estos métodos se centran solo en extraer reglas positivas sin poner atención a reglas negativas que pueden ser interesantes al expresar, por ejemplo, la ausencia de Y ante la presencia de X ($X \rightarrow \neg Y$). Las reglas de asociación negativas pueden incluir ítems negativos en el antecedente, en el consecuente o en ambos. Algunos investigadores han propuesto métodos para extraer RACs positivas y negativas (RACPNs) [3].

En los últimos años muchos autores han propuesto algoritmos evolutivos (AE) para extraer RACs [6,26] porque son considerados una de las técnicas de búsqueda más exitosas para problemas complejos y han demostrado ser muy buenos en el aprendizaje y la extracción de conocimiento. Estos métodos suelen considerar un único objetivo para medir la calidad de las reglas. Sin embargo, recientemente varios autores han abordado el proceso de extraer reglas de asociación como un problema multi-objetivo para optimizar varias medidas a la vez [4,19]. Los algoritmos evolutivos multi-objetivos (AEMOs) son un mecanismo interesante para tratar problemas de naturaleza multi-objetivo por lo que algunos AEMOs han sido aplicados para extraer RACs, donde cada solución del frente de Pareto representa una RAC [4,19]. Este enfoque elimina algunas de las limitaciones de los algoritmos mono-objetivo y nos permite optimizar varias medidas con el fin de extraer reglas de alta calidad.

En este trabajo proponemos MOPNAR, un nuevo AEMO que extiende el AEMO basado en descomposición MOEA/D-DE [24] para extraer un conjunto reducido de RACPNs de alta calidad con un buen equilibrio entre el número de reglas y el cubrimiento de la BD. Nuestra propuesta realiza un aprendizaje evolutivo de los intervalos de los atributos y una selección de las condiciones para cada regla, maximizando tres objetivos: rendimiento, interés y comprensibilidad. Además, MOPNAR introduce un proceso de reinicialización y una población externa (PE) al modelo evolutivo, con el fin de promover diversidad en la población, almacenar todas las reglas no dominadas encontradas y mejorar el cubrimiento de la BD.

Para evaluar la efectividad del método propuesto, hemos presentado un estudio experimental utilizando siete BDs del mundo real, con un número de variables

que van desde 5 a 91 y un número de registros que van desde 96 a 22784. En este estudio hemos comparado nuestro método con un algoritmo clásico para extraer reglas de asociación, Apriori [32], con un algoritmo evolutivo mono-objetivo (GAR [26]), dos AEMOs (MODENAR [4] y MOEA_Ghosh [19]) para obtener RACs, y con Alatasetal, un algoritmo evolutivo para extraer RACPNs propuesto por Alatas y otros en [3].

La organización del trabajo es como sigue. En la siguiente sección se presenta un breve estudio de los AEMOs existentes para propósito general y se introducen definiciones básicas de las RACPNs y de algunas de sus medidas de calidad. En la Sección 3 se detalla el algoritmo evolutivo propuesto para obtener RACPNs. En la Sección 4 se muestran los resultados obtenidos sobre cuatro BDs reales. En la Sección 5 se presentan las conclusiones.

2. Preliminares

En esta sección presentamos un breve estudio sobre los AEMOs. Además, introducimos las definiciones básicas de las RACPNs y de algunas de sus medidas de calidad.

2.1. Algoritmos Evolutivos Multi-Objetivos

En los últimos años han habido avances significativos en el desarrollo de EAs para problemas de optimización multi-objetivos. Los AEMOs tratan simultáneamente con un conjunto de posibles soluciones (llamado población), permitiéndoles encontrar varios miembros del conjunto optimal de Pareto en una sola ejecución del algoritmo. Además, estos algoritmos no son demasiado susceptibles por la forma o la continuidad del frente de Pareto (por ejemplo, pueden tratar fácilmente con frentes de Pareto discontinuos y cóncavos) [34].

El primer indicio respecto a la posibilidad de usar un AEs para resolver un problema multi-objetivo aparece en tesis doctoral de 1967 [29], aunque en una tesis no se desarrolló un AEMO actual porque el problema multi-objetivo fue formulado como un problema de un solo objetivo y resuelto con un algoritmo genético. David Schaffer es considerado el primer investigador en haber diseñado un AEMO a mediados de los años ochenta [30]. El enfoque de Schaffer, llamado *Evaluated Genetic Algorithm* (VEGA) está basado en un algoritmo genético simple con un mecanismo de selección modificado. Sin embargo, VEGA presenta una serie de problemas donde el más importante es su incapacidad para retener soluciones con buena calidad, tal vez por encima de la media, pero no excepcional para alguno de los objetivos.

Después de VEGA, fue diseñada la primera generación de AEMOs caracterizada por su simplicidad, donde la principal lección aprendida fue que los AEMOs debían combinar un buen mecanismo para seleccionar los individuos no dominados (no necesariamente basado en el concepto de Pareto óptimo) y un buen mecanismo para mantener la diversidad. Los AEMOs más representativos de esta generación son: NSGA [33], NPGA [21] y MOGA [17].

Tabla 1. Clasificación de los AEMOs

Referencia	*AEMOs*	1^{ra}*Gen.*	2^{da}*Gen.*
[17]	MOGA	√	
[21]	NPGA	√	
[33]	NSGA	√	
[14]	AEMOs híbridos		√
[36]	AEMOs basados en indicadores		√
[23]	AEMOs meméticos		√
[9]	Micro-GA		√
[35,24]	MOEA/D y MOEA/D-DE		√
[13]	AEMOs basados en coevolución		√
[17]	AEMOs basados en referencia		√
[15]	NPGA 2		√
[12]	NSGA-II		√
[22]	PAES		√
[11,10]	PESA y PESA-II		√
[38,37]	SPEA y SPEA2		√

La segunda generación de los AEMOs comenzó cuando el elitismo llegó a ser en un mecanismo estándar. De hecho, el uso del elitismo es un requisito teórico para garantizar la convergencia de los AEMOs. Muchos AEMOs han sido propuestos en la segunda generación, la cual todavía estamos viviendo, pero muy pocos han sido considerados como un referente. SPEA2 [37] y el NSGA-II [12] pueden ser los AEMOs más representativos de esta generación. También resultan de interés otros como el PAES [22], los AEMOs basados en descomposición (MOEA/D y MOEA/D-DE) [35] [24], AEMOs basados en referencia [17], AEMOs basados en indicadores [36], AEMOs híbridos [14], AEMOs meméticos [23] y AEMOs basados en coevolución [13]. La Tabla 1 muestra un resumen de los AEMOs más representativos de ambas generaciones.

Por último, destacar que en la actualidad ha crecido el interés por los AEMOs basados en descomposición (MOEA/D [35] y MOEA/D-DE [24]), los cuales ganaron la competición del CEC2009. Ellos explícitamente descomponen el problema de optimización multi-objetivo en N subproblemas de optimización escalares y los optimizan de manera simultánea. Estos enfoques han mostrado algunas ventajas sobre otros AEMOs porque presentan menor complejidad computacional y un mejor funcionamiento en los problemas continuos de 3-objetivos. Esto ha suscitado un creciente interés por estos enfoques dentro de la comunidad de los AEMOs.

2.2. Reglas de Asociación Positivas y Negativas

Las reglas de asociación son usadas para representar dependencias entre los ítems en una BD [20]. Como hemos mencionado anteriormente, ellas son expresiones del tipo $X \rightarrow Y$, donde X y Y son conjuntos de ítems (parejas atributo-valor) y cumplen que $X \cap Y = \oslash$. Esto significa que si todos los ítems de X están en un registro de la BD, entonces todos los ítems de Y están también en la registro con una alta probabilidad, y X y Y no tienen ningún ítem en común [1]. Las reglas de asociación obtenidas en BDs con valores numéricos son denominadas RAC [32], donde cada ítem es un par *atributo-intervalo*. Por ejemplo, una RAC

positiva es $Peso \in [10, 19] \rightarrow Talla \in [75, 109]$. Muchos de los estudios clásicos presentan dificultades para descubrir RACs debido a que los atributos cuantitativos contienen muchos valores distintos. Por esta razón, varios investigadores han propuesto métodos para aprender los intervalos de las RACs [3,4,6,26].

Muchos de estos métodos solo consideran ítems positivos en la extracción de reglas de asociación sin tener en cuenta los negativos que también son interesantes puesto que ofrecen un nuevo conocimiento para apoyar la toma de decisiones. Asimismo, las reglas de asociación negativas pueden incluir ítems negativos dentro del antecedente($\neg X \rightarrow Y$), del consecuente ($X \rightarrow \neg Y$), o de ambos ($\neg X \rightarrow \neg Y$). Destacar que, las reglas de asociación negativas deben incluir al menos un ítem negativo. La Fig 1 muestra la representación del ítem negativo $Edad \in \neg[5, 25]$.

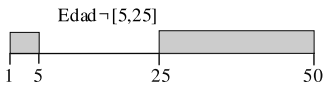

Figura 1. Ejemplo de ítem negativo

Las medidas el soporte y la confianza son las más utilizadas para evaluar las RACs, se basan en el soporte de un conjunto de ítem I, el cual se define como:

$$SOP(I) =\mid \{e \in D \mid I \in e\} \mid / \mid D \mid \quad (1)$$

donde el numerador es el número de registros de la BD cubiertos por I, y $\mid D \mid$ es el número de registros de la BD.

Luego, el soporte y la confianza de una regla $X \rightarrow Y$ se define como:

$$Soporte(X \rightarrow Y) = SOP(XY) \quad (2)$$

$$Confianza(X \rightarrow Y) = SOP(XY)/SOP(X) \quad (3)$$

Las técnicas clásicas para extraer reglas de asociación intentan obtener reglas con valores de soporte y confianza mayores que un mínimo de soporte (minSop) y una mínima confianza (minConf). Sin embargo, varios autores han señalado algunos inconvenientes de este marco de trabajo que conduce a encontrar reglas de baja calidad [8]. La confianza no detecta independencia estadística o dependencia negativa entre los ítems debido a que no tiene en cuenta el soporte del consecuente. Por ejemplo, una confianza de 0.9 en una regla $A \rightarrow B$ no es mejor que una confianza de 0.7 en la regla $C \rightarrow D$ si el soporte de B es 0.95 y el soporte de D es 0.1, ya que la primera de las reglas se corresponde con una dependencia negativa (observando A se reduce la probabilidad de B), mientras que en la segunda la probabilidad de D incrementa significativamente cuando observamos C. Asimismo, los ítems con soporte muy alto pueden generar reglas de baja calidad porque cualquier conjunto de ítems parece ser un buen predictor de ellos.

Por esta razón, en la literatura se han propuesto otras medidas de calidad para la selección y ranking de ejemplos de acuerdo con su interés potencial para el usuario [18]. A continuación describimos brevemente las medidas que han sido utilizadas en este trabajo.

La medida Lift [28] representa el ratio entre la confianza de la regla y la confianza esperada de la regla. Esta medida toma valores en el intervalo $[0,\infty)$ donde valores menores de 1 representan dependencia negativa, 1 representa independencia y mayores que 1 representan dependencia positiva. Lift de una regla $X \rightarrow Y$ se define como:

$$Lift(X \rightarrow Y) = SOP(XY)/(SOP(X)SOP(Y)) \tag{4}$$

El Factor de Certeza (FC) [31] mide la variación de la probabilidad de que Y esté en una registro cuando se consideran solo los registros donde está X. FC toma valores en el intervalo [-1,1] donde valores positivos y negativos representan dependencia positiva y negativa respectivamente y 0 representa independencia. Esta medida para una regla $X \rightarrow Y$ se define de tres maneras dependiendo de si la confianza es menor, mayor o igual que $SOP(Y)$:

Si $Confianza(X \rightarrow Y) > SOP(Y)$

$$(Confianza(X \rightarrow Y) - SOP(Y))/(1 - SOP(Y)) \tag{5}$$

Si $Confianza(X \rightarrow Y) < SOP(Y)$

$$(Confianza(X \rightarrow Y) - SOP(Y))/SOP(Y) \tag{6}$$

Sino es 0

Netconf [2] evalúa una regla basándose en el soporte de la regla, del antecedente y del consecuente. Netconf obtiene valores en el intervalo [-1,1] donde valores positivos y negativos representan dependencia positiva y negativa respectivamente y 0 representa independencia. Netconf de una regla $X \rightarrow Y$ se define como:

$$Netconf(X \rightarrow Y) = (SOP(XY) - SOP(X)SOP(Y))/(SOP(X)(1 - SOP(X))) \tag{7}$$

3. MOPNAR: Algoritmo Evolutivo Multi-Objetivo para extraer Reglas de Asociación Cuantitativas Positivas y Negativas

En esta sección se describe nuestra propuesta, MOPNAR, para obtener un conjunto de RACPNs de alta calidad con un buen equilibrio entre el número de reglas y el cubrimiento de la BD. En las siguientes subsecciones describiremos en detalle cada uno de los componentes de esta propuesta: modelo evolutivo, objetivos, población inicial y operadores genéticos.

3.1. Modelo Evolutivo Multi-Objetivo MOEA/D-DE

Nosotros extendemos el modelo evolutivo basado en descomposición MOEA/D-DE [24] para extraer RACPNs. Esta propuesta descompone el problema de optimización multi-objetivo en N subproblemas de optimización escalares con el propósito de cada uno de estos subproblemas optimice una agregación distinta de todos los objetivos. Con el propósito de almacenar todas las reglas no dominadas generadas hemos introducido una PE a este modelo evolutivo. La PE se actualiza en cada generación con los hijos generados para cada solución, manteniendo las reglas no dominadas de la población. Destacar que el tamaño de la PE no está limitado, lo que nos permite devolver un conjunto de reglas no dominadas independientemente del tamaño de la población y reducir el tamaño de la población, siguiendo un enfoque independiente de la BD.

Además, para provocar diversidad en la población y mejorar el cubrimiento de las BDs hemos introducido un proceso de reinicialización. Este proceso se aplica cuando el número de nuevas soluciones en la población es menor que $\alpha\,\%$ ($\alpha\,\%$ definido por el usuario, normalmente al $5\,\%$). En este caso la población se reinicia basándose en los registros que no hayan sido cubiertos por las reglas de la PE y se actualiza la PE con la nueva población (ver subsección 3.5).

Con estas modificaciones, el modelo evolutivo sería el siguiente. Este modelo primero genera un vector de pesos para cada subproblema, que son utilizados para calcular el valor del enfoque de descomposición de cada subproblema. Luego se selecciona un conjunto de vecinos para cada vector de pesos, conteniendo T vectores de pesos más cercanos. Después el algoritmo genera una población inicial, inicializa el punto de referencia para cada objetivo, con los mejores valores encontrados hasta el momento e inicializa la PE con las reglas no dominadas de la población inicial. A continuación se generan dos hijos aplicando los operadores de cruce, mutación y reparación. Estos se aplican sobre una solución de la población y sobre otra solución seleccionada aleatoriamente con una probabilidad δ entre su vecindad y la población (δ es definida por el usuario). Estos hijos se utilizan para actualizar los puntos de referencias y sustituir algunas soluciones de la población actual que tengan los peores valores del enfoque de descomposición. Téngase en cuenta que el máximo número de soluciones que se pueden sustituir por una solución hija es limitado y debe ser mucho menor que T. Estos pasos se repiten para cada solución de la población, se actualiza la PE y si es necesario se aplica el proceso de reinicialización. Todo este proceso se repite hasta que se cumpla la condición de parada (ver [24] para más información).

En [24] los autores presentaron tres enfoques de descomposición donde recomendaron el propuesto por Tchebycheff [27], el cual es el utilizado en este trabajo.

3.2. Objetivos

Nuestra propuesta maximiza tres objetivos: rendimiento, interés y comprensibilidad. Rendimiento es el producto entre el soporte y el FC (ver Sección 2.2), lo cual nos permite obtener un conjunto de reglas con un buen equilibrio entre

reglas locales y generales. Destacar que solamente nos interesan las reglas con dependencia fuerte [8] entre los ítems porque representan dependencias positivas entre ellos y evitan el problema del soporte (ver Sección 2.2). Además, teniendo en cuenta que las reglas de asociación negativas nos permiten representar dependencias negativas, solo obtendremos aquellas reglas con FC > 0.

El Interés intenta medir como de interesante es una regla, permitiendonos extraer solo aquellas reglas que sean interesante para el usuario. En este trabajo hemos utilizado la medida de interés lift (ver Sección 2.2), la cual nos permite detectar la dependencia negativa, positiva o independencia entre los ítems y su rango de valores no está limitado, permitiendo representar mejor las diferencias entre las reglas y reducir el número de empates.

La comprensibilidad pretende medir lo fácil de comprender que puede ser una regla [16]. Las reglas cuando involucran muchos atributos pueden resultar difíciles de comprender. En este trabajo hemos usado la medida de comprensibilidad de una regla $X \rightarrow Y$ según el número de atributos que contiene. Esta se define como sigue, donde $Atrib_{X \rightarrow Y}$ es el número de atributos involucrados en la regla.

$$Comprensibilidad(X \rightarrow Y) = 1/Atrib_{X \rightarrow Y} \tag{8}$$

3.3. Esquema de Codificación y Población Inicial

Un cromosoma es un vector de n genes que representa los atributos e intervalos de una regla, donde n es el número de atributos de la BD. Nuestra propuesta utiliza una codificación posicional en la que el i-ésimo gen codifica el i-ésimo atributo. Cada gen consta de cuatro partes: ac indica si un gen es considerado en la regla; pn indica si el intervalo es positivo o negativo; li y ls representan el límite inferior y superior del intervalo del atributo respectivamente. Tenga en cuenta que si el atributo es nominal, li y ls serán iguales, representando solo un valor del atributo nominal. Por tanto un cromosoma C_T se codifica de la siguiente manera:

$$C_T = Gen_1 Gen_2 \dots Gen_n, \;\; i = 1, \dots, n$$
$$Gen_i = (ac_i, pn_i, li_i, ls_i)$$

Para evitar que el aumento de los intervalos cubra la totalidad del dominio hemos definido *amplitud*, el cual representa el tamaño máximo que el intervalo de un atributo puede alcanzar. La *amplitud* de un atributo i se define como:

$$Amplitud_i = (Max_i - Min_i)/\gamma \tag{9}$$

donde γ es un valor dado por el experto que determina el equilibrio entre la generalización y la especificidad de las reglas, y Min_i y Max_i son los valores de mínimo y máximo del dominio del atributo i, respectivamente. Para los intervalos negativos la amplitud representa el tamaño mínimo que el intervalo de un atributo puede alcanzar.

La población inicial estará compuesta por un conjunto de reglas que contienen un solo atributo en el consecuente y presentan un buen cubrimiento de la BD.

Para crear la población inicial, primero se selecciona aleatoriamente los atributos que formarán parte del antecedente y del consecuente de la regla. Después se selecciona si el intervalo será positivo o negativo. Luego se generan los intervalos que tendrán un tamaño igual al 50 % de la amplitud de cada atributo y centrados en un registro seleccionado. Finalmente se marcan los registros que cubre la regla en la BD. Este proceso se repite para todos los registros que no han sido marcados hasta que se complete la población inicial. Si todos los registros se han marcado y la población inicial no se ha completado, se vuelven a desmarcar todos los registros y el proceso se repite hasta que la población inicial sea completada.

3.4. Operadores Genéticos

Esta propuesta utiliza los operadores genéticos de cruce, mutación y reparación. Primero, el operador de cruce genera dos hijos intercambiando aleatoriamente los genes de los padres (exploración). Después, el operador de mutación selecciona aleatoriamente un gen del cromosoma. De este gen, selecciona al azar uno de los límites del intervalo y aumenta o disminuye su valor de manera aleatoria, y modifica los valores de ac y pn aleatoriamente. Por último, el operador de reparación corrige las reglas que tengan más de un atributo en el consecuente o no tengan antecedente o consecuente. Si el consecuente contiene más de un atributo, uno de ellos se selecciona aleatoriamente para ser el consecuente y el resto pasan al antecedente. Si no hay ningún atributo en el antecedente y/o consecuente, se seleccionan al azar entre los atributos que no han sido considerados en la regla. Además para obtener reglas más simples este operador decrementa el tamaño de los intervalos mientras el número de registros cubiertos sea igual que el cubierto por los intervalos originales. Destacar que, si el intervalo es negado, este se incrementa reduciendo el dominio que cubre.

3.5. Proceso de Reinicialización

Para alejarse de los óptimos locales este algoritmo utiliza un proceso de reinicialización. Este proceso elimina los registros que hayan sido cubiertos por las reglas de la PE y aplica nuevamente el proceso de inicialización (ver subsección 3.3). Además, la PE se actualiza con la nueva población siguiendo el criterio de no-dominancia. Este proceso de reinicialización se aplica cuando el número de nuevas soluciones en una población es menor que α % del tamaño de la población actual.

4. Estudio Experimental

Varios experimentos han sido realizados para analizar el funcionamiento de nuestra propuesta. En las siguientes subsecciones, primero, describimos las BDs que se utilizan en estos experimentos; segundo, introducimos una breve descripción de los métodos considerados para la comparación y presentamos la configuración de los mismos, y finalmente comparamos nuestro método con un algoritmo

Tabla 2. BDs consideradas en el estudio experimental

Nombres	*#Atrib(R/E/N)*	*#Reg*
Basketball (bas)	5(3/2/0)	96
House_16H (hh) [3]	17 (10/7/0)	22784
Magic (mag)	11 (10/0/1)	19020
Movement Libras (mov)	91 (90/0/1)	360
Segment (seg)	20 (19/1/0)	2310
Stock Price (sto)	10 (10/0/0)	950
Texture (tex)	41 (40/1/0)	5500

Disponible en `http://sci2s.ugr.es/keel/datasets.php`

clásico de extracción de reglas de asociación, un AE para extraer RACPNs, un AE mono-objetivo y dos AEMOs para obtener RACs.

4.1. Bases de Datos

En estos experimentos hemos seleccionado 7 BDs de la vida real con distintos tamaños para analizar la efectividad de nuestra propuesta. La Tabla 2 resume las principales características de las 7 BDs y muestra el enlace al repositorio de datos KEEL-dataset [5] del cual podemos descargarlas. Para cada BD se muestra el número de registros ("#Reg") y el número de atributos reales, enteros o nominales que contiene ("#Atrib(R/E/N)").

4.2. Métodos considerados para la comparación y sus parámetros

En este estudio comparamos el método propuesto con otros cinco métodos, los cuales están disponibles en KEEL [7]. A continuación presentamos una breve descripción de estos métodos.

1. *AE para extraer RACPNs (Alatasetal)* [3]: Este método diseña un algoritmo genético para simultáneamente buscar los intervalos de los atributos cuantitativos y descubrir las RACPNs asociadas a esos intervalos. Los cromosomas representan reglas de asociación, en los cuales cada gen tiene 4 partes. La primera parte representa si forma parte del antecedente o del consecuente de la regla, la segunda si el intervalo es positivo o negativo, la tercera y cuarta representan el límite inferior y superior del intervalo del atributo respectivamente.
2. *Apriori* [32]: En cada iteración, este método genera conjuntos de ítems candidatos a partir de los conjuntos de ítems que han sido considerados frecuentes en la iteración anterior. Un conjunto de ítems es considerado frecuente cuando tiene un soporte mayor a un umbral establecido por el usuario (minSop). Una vez extraídos todos los conjuntos de ítems frecuentes, Apriori genera a partir de ellos todas las reglas con una confianza mayor a un umbral definido por el usuario (minConf).

[3] Esta BD fue diseñada sobre la base de los datos proporcionada por la Oficina del Censo de EE.UU. [http://www.census.gov] (Acceso Lookup [http://www.census.gov/cdrom/lookup]: Resumen del archivo 1).

Tabla 3. Parámetros considerados en la comparación de los métodos

Algoritmos	*Parámetros*
Apriori	*minSop = 0,1, minConf = 0,8*
Alatasetal	N_{eval}=50000, nCromoInicialAleat=12, r = 3, TamTorneo = 10, P_{sel}= 0,25, P_{cru} = 0,7, P_{mut_min} = 0,05, P_{mut_max} = 0,9, $Peso_{sop}$ = 5, $Peso_{conf}$ = 20, $Peso_{amplRule}$ = 0,05, $Peso_{amplInterv}$ = 0,02, $Peso_{cubrimiento}$ = 0,01
GAR	*TamPop = 100, nItems = 100,* N_{eval} *= 50000,* P_{sel}= 0,25, P_{cru} = 0,7, P_{mut} = 0,1, ω = 0,4, Ψ= 0,7, μ= 0,5, *minSop = 0,1, minConf = 0,8*
MODENAR	*TamPop = 100,* N_{eval}*=50000, Umbral= 60, CR = 0,3,* $Peso_{sop}$ *= 0,8,* $Peso_{conf}$ *= 0,2,* $Peso_{comp}$ *= 0,1,* $Peso_{amplInterv}$ *= 0,4*
MOEA_Ghosh	*TamPop = 100,* N_{eval}*=50000, PuntoCruce=2,* P_{cru}*=0.8,* P_{mut}= 0.02
MOPNAR	N_{eval}=50000, H=13, m=3, TamPop=N^{m-1}_{H+m-1}, T=10, δ=0,9, η_r=2, γ=2, P_{mut}= 0,1, α = 5 %

3. *Genetic Association Rules (GAR)* [26]: Este método extiende el AE GENAR [25] para encontrar conjuntos de ítems frecuentes en BDs numéricas sin tener que discretizar los valores de los atributos. Cada cromosoma representa un conjunto de ítems, donde cada gen representa el máximo y el mínimo de los valores de los atributos que pertenecen al conjunto de ítems. Cuando el algoritmo termina el proceso evolutivo, se ejecuta otro procedimiento para generar las reglas a partir de los conjuntos de ítems frecuentes extraídos.
4. *Multi-objective differential evolution algorithm for mining numeric association rules (MODENAR)* [4]: Este algoritmo utiliza un AEMO diferencial basado en Alatasetal [3] para extraer RACs precisas y comprensibles sin que el usuario especifique minSop y minConf. MODENAR utiliza una codificación similar a Alatasetal pero sin incluir la parte del gen donde se representa si el intervalo es positivo o negativo. Además, este método optimiza cuatro objetivos para mejorar la calidad de las reglas: soporte, confianza, comprensibilidad y amplitud del dominio de los intervalos.
5. *Multi-objective rule mining using genetic algorithms (MOEA_Ghosh)* [19]: Este algoritmo utiliza un algoritmo genético basado en el frente de Pareto para extraer reglas de asociación útiles e interesantes para cualquier BD. Cada cromosoma representa una regla de asociación. Este método optimiza tres medidas: comprensibilidad, interés y presición. Este método emplea una población externa para almacenar las soluciones no dominadas encontradas.

La Tabla 3 muestra los parámetros de los métodos analizados. Con estos valores para nuestra propuesta, hemos tratado de facilitar las comparaciones, seleccionando parámetros estándar comunes que funcionan bien en la mayoría de los casos, y para el resto de los algoritmos los hemos seleccionado de acuerdo a las recomendaciones de los autores de cada propuesta. Destacar que Apriori y GAR necesitan un minSop y minConf para extraer RACs, seleccionando para ellos valores estándar que funcionan bien en la mayoría de los casos para todas las BDs. Además los resultados obtenidos para los AEMO se refieren al conjunto de reglas no dominadas obtenido.

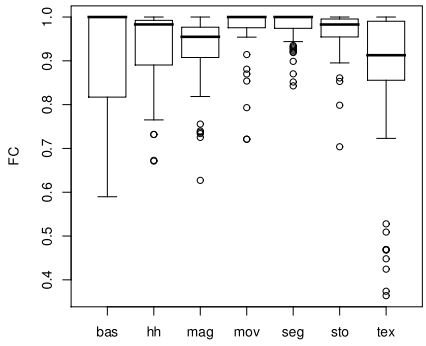

Figura 2. Boxplot de la medida FC para MOPNAR en todas las BDs

4.3. Análisis comparativo con otros métodos

Esta sección analiza el funcionamiento de nuestra propuesta en comparación con los algoritmos descritos en la subsección 4.2. La Tabla 4 muestra los resultados medios de 5 ejecuciones de los algoritmos analizados, donde $\#R$ representa el número medio de reglas, Med_{Sop}, Med_{Conf}, Med_{Lift}, Med_{FC}, $Med_{Netconf}$ representan los valores medios de soporte, confianza, lift, FC y netconf, respectivamente, Av_{Amp} el número medio de atributos involucrados en las reglas y $\%Reg$ es el tanto por ciento de registros de la BD cubiertos por las reglas generadas. Hay que señalar que el método Apriori no puede ejecutarse en Movement Libras, Segment y Texture porque tienen muchos atributos.

Como podemos observar el método propuesto obtiene mejores valores para casi todas las medidas de interés que el resto de los algoritmos en todas las BDs. Destacar que algunos de los métodos obtienen reglas con valores altos para las medidas de soporte y confianza (incluso en varias BDs mejores que nuestra propuesta) pero ellos presentan peores valores para el resto de las medidas. Esto se debe a que obtienen reglas con conjuntos de ítems con soportes altos en el consecuente o reglas que representan dependencia negativa. Además, el número medio de reglas obtenidas es reducido (menor que 100 en todas las BDs) y con un bajo número de atributos lo que facilita su interpretabilidad desde el punto de vista del usuario. Por otra parte, el cubrimiento de la BD es muy alto (cercano a 100 % en todas las BDs) aportando un conocimiento sobre la totalidad de la BD. Destacar que en la mayoría de las BDs el resto de los algoritmos obtienen menos reglas que MOPNAR pero con valores más bajos de cubrimiento.

La Fig. 2 es un boxplot que muestra los valores del FC para todas las reglas obtenidas en una de las 5 ejecuciones realizadas por nuestra propuesta para todas las BDs, seleccionada aleatoriamente. Podemos observar como todas las reglas obtenidas representan dependencias positivas con valores cercanos al máximo valor que puede alcanzar esta medida (ver sección 2.2). Destacar que más del 75 % de las reglas tienen valores mayores que 0.80.

Tabla 4. Resultados obtenidos por los algoritmos analizados

BD	*Algoritmos*	*#R*	Med_{Sop}	Med_{Conf}	Med_{Lift}	Med_{CF}	$Med_{Netconf}$	Med_{Amp}	*%Reg*
Basketball	Apriori	4	0,15	0,87	4,88	0,84	**0,81**	2,75	33,34
	Alatasetal	9,20	**0,98**	**1**	1	-0,01	-0,01	3,18	**100**
	GAR	2	0,77	0,89	1,02	0,11	0,11	**2**	97,09
	MODENAR	61,40	0,32	0,79	2,37	0,33	0,15	2,30	97,5
	MOEA_Ghosh	20,80	0,89	0,98	1,58	0,14	0,03	3,51	**100**
	MOPNAR	**91,60**	0,16	0,95	**47,73**	**0,92**	0,78	2,33	99,79
House16H	Apriori	**1749917**	0,22	0,97	2,19	0,83	0,45	8,65	**100**
	Alatasetal	90,67	0,19	**0,99**	1,03	0,58	0,03	8,76	98,09
	GAR	105,60	**0,76**	0,90	1,03	0,20	0,17	**2,01**	99,99
	MODENAR	64,40	0,69	**0,99**	1,15	0,72	0,19	7	81
	MOEA_Ghosh	19,80	0,60	0,79	**269,17**	0,36	0,11	7,75	98,87
	MOPNAR	84,80	0,34	0,89	6,18	**0,86**	**0,70**	2,91	99,37
Magic	Apriori	**9785**	0,19	0,96	2,73	0,87	0,52	5,53	99,96
	Alatasetal	12	0,46	**1**	1,27	**0,89**	0,35	4,75	89,90
	GAR	57,80	0,67	0,91	1,10	0,48	0,35	**2,08**	96,84
	MODENAR	71,60	0,36	0,92	**284,56**	0,52	0,07	3,80	72,78
	MOEA_Ghosh	29,40	**0,72**	0,90	1,46	0,52	0,17	5,26	98,33
	MOPNAR	99,40	0,38	0,89	8,21	0,86	**0,66**	2,60	**99,98**
Movement Libras	Apriori	-	-	-	-	-	-	-	-
	Alatasetal	0	-	-	-	-	-	-	-
	GAR	2,60	**0,42**	0,94	3,73	0,90	**0,90**	**2**	53,28
	MODENAR	23,40	0,01	0,15	32,31	0,04	0,15	61,61	3,17
	MOEA_Ghosh	10,80	0,01	0,22	**79,47**	0,22	0,22	80,08	0,28
	MOPNAR	**59,80**	0,31	**0,97**	19,46	**0,95**	0,84	2,63	**99,50**
Segment	Apriori	-	-	-	-	-	-	-	-
	Alatasetal	47,50	**0,53**	0,94	1,03	0,26	0,02	4,14	**100**
	GAR	18,80	0,36	0,89	2,49	0,58	0,47	**2**	97,97
	MODENAR	58,80	0,33	0,97	1,72	0,93	0,58	10,60	56,49
	MOEA_Ghosh	28,20	0,36	0,86	**108,60**	0,73	0,50	12,63	72,95
	MOPNAR	**83,20**	0,29	**0,98**	16,75	**0,97**	**0,85**	2,74	99,91
Stock	Apriori	**855**	0,13	0,91	4,77	0,88	0,76	4,16	99,48
	Alatasetal	14,40	0,08	**1**	**96,54**	0,92	0,72	2,73	21,04
	GAR	2	0,56	0,87	1,35	0,62	0,62	**2**	73,30
	MODENAR	63,80	0,48	0,92	1,75	0,61	0,30	3	81,86
	MOEA_Ghosh	19,80	**0,61**	0,91	42,56	0,53	0,36	5,28	96,40
	MOPNAR	83	0,22	0,94	15,07	**0,93**	**0,86**	2,92	**99,64**
Texture	Apriori	-	-	-	-	-	-	-	-
	Alatasetal	0	-	-	-	-	-	-	-
	GAR	39,40	**0,71**	**0,94**	1,24	0,71	0,66	**2,05**	98,02
	MODENAR	29	0,08	0,51	17,53	0,28	0,42	27,50	43,05
	MOEA_Ghosh	54,60	0,01	0,19	**1004,24**	0,19	0,01	36,42	0,04
	MOPNAR	**95,60**	0,30	**0,94**	11,18	**0,92**	**0,84**	3,08	**99,78**

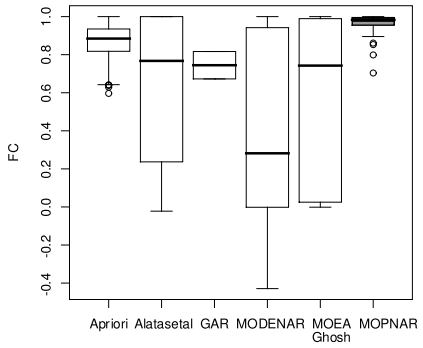

Figura 3. Boxplot de la medida FC para los algoritmos analizados en la BD Stock

La Fig. 3 representa un boxplot de los valores del FC para todas las reglas obtenidas en una de las 5 ejecuciones realizadas en la BD stock por los algoritmos analizados, la ejecución fue seleccionada aleatoriamente. Observe que MOPNAR presenta mejores valores de FC que el resto de los algoritmos analizados y todas sus reglas obtienen valores cercanos al máximo valor que esta medida puede alcanzar. Destacar que algunas reglas obtenidas por Alatasetal y MODENAR representan independencia o dependencia negativa según los valores de esta medida.

5. Conclusiones

En este trabajo hemos propuesto un nuevo AEMO para extraer RACPNs llamado MOPNAR, que nos permite obtener un conjunto reducido de RACPNs fáciles de comprender, interesantes y con un buen cubrimiento de la BD, maximizando tres objetivos: rendimiento, interés y comprensibilidad. Para hacer esto, esta propuesta extiende el AEMO basado en descomposición MOEA/D-DE para realizar un aprendizaje evolutivo de los intervalos de los atributos y una selección de condiciones para cada regla. Además, introduce al modelo evolutivo una PE y un proceso de reinicialización para almacenar todas las reglas no dominadas encontradas, promover su diversidad y mejorar el cubrimiento de la BD.

Si observamos los resultados obtenidos podemos ver como el método propuesto obtiene un conjunto reducido de RACPNs, con un buen equilibrio entre el número de reglas, el soporte y el cubrimiento, presentando altos valores de cubrimiento en todas las BDs. Además, estas reglas tienen pocos atributos, lo que permite una mejor comprensión del usuario y presentan altos valores de las medidas de interés en todas las BDs.

Referencias

1. R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *SIGMOD*, pages 207–216, Washington D.C., 1993.

2. K.-I. Ahn and J.-Y. Kim. Efficient mining of frequent itemsets and a measure of interest for association rule mining. *JIKM*, 3(3):245–257, 2004.
3. B. Alatas and E. Akin. An efficient genetic algorithm for automated mining of both positive and negative quantitative association rules. *Soft. Comput.*, 10(3):230–237, 2006.
4. B. Alatas and E. Akin. MODENAR: Multi-objective diferential evolution algorithm for mining numeric association rules. *Appl. Soft Comput.*, 8(1):646–646, 2008.
5. J. Alcala-Fdez, A. Fernandez, J. Luego, J. Derrac, S. Garcia, L. Sanchez, and F. Herrera. Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *J. Mult-Valued Log. S*, 17(2-3):255–287, 2011.
6. J. Alcala-Fdez, N. Flugy-Pape, A. Bonarini, and F. Herrera. Analysis of the effectiveness of the genetic algorithms based on extraction of association rules. *Fund. Inform.*, 98(1):1–14, 2010.
7. J. Alcala-Fdez, L. Sanchez, S. Garcia, M. del Jesus, S. Ventura, J. O. J. Garrell, C. Romero, J. Bacardit, V. Rivas, J. Fernandez, and F. Herrera. Keel: A software tool to assess evolutionary algorithms to data mining problems. *Soft. Comput.*, 13(3):307–318, 2009.
8. F. Berzal, I. Blanco, D. Sanchez, and M. Vila. Measuring the accuracy and interest of association rules: A new framework. *Intell. Data Anal.*, 6(3):221–235, 2002.
9. C. Coello and G. Toscano. A micro-genetic algorithm for multiobjective optimization. In *First International Conference on Evolutionary Multi-Criterion Optimization, LNCS 1993*, pages 126–140, London, UK, 2001.
10. D. Corne, N. Jerram, J. Knowles, and M. Oates. Pesa-ii: Region based selection in evolutionary multiobjective optimization. In *Genetic and Evolutionary Computation Conf.*, pages 283–290, San Francisco, CA, 2001.
11. D. Corne, J. Knowles, and M. Oates. The pareto envelopebased selection algorithm for multiobjective optimization. In *Parallel Problem Solving from Nature VI Conf., LNCS 1917*, pages 839–848, Paris, France, 2000.
12. K. Deb, S. Agrawal, A. Pratab, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE T Evolut. Comput.*, 6(2):182–197, 2002.
13. K. Deb, M. Mohan, and S. Mishra. Evaluating the epsilon-domination based multiobjective evolutionary algorithm for a quick computation of pareto-optimal solutions. *Evol. Comput.*, 13(4):501–525, 2005.
14. A. Elhossini, S. Areibi, and R. Dony. Strength pareto particle swarm optimization and hybrid ea-pso for multi-objective optimization. *Evol. Comput*, 18(1):127–156, 2010.
15. M. Erickson, A. Mayer, and J. Horn. The niched pareto genetic algorithm 2 applied to the design of groundwater remediation systems. In *First International Conference on Evolutionary MultiCriterion Optimization, LNCS 1993*, pages 681–695, London, UK, 2001.
16. M. Fidelis, H. Lopes, and A. Freitas. Discovering comprehensible classification rules with a genetic algorithm. In *Proceedings of the 2000 Congress on Evolutionary Computation*, pages 805–810, CA, USA, 2000.
17. C. Fonseca and P. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In *5th International Conference on Genetic Algorithms*, pages 416–423, San Mateo, CA, 1993.
18. L. Geng and H. Hamilton. Interestingness measures for data mining: A survey. *ACM Comput. Surv.*, 38(3), 2006.

19. A. Ghosh and B. Nath. Multi-objective rule mining using genetic algorithms. *Inform. Sciences*, 163(1-3):123–133, 2004.
20. J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, second edition edition, 2006.
21. J. Horn, N. Nafpliotis, and D. Goldberg. A niched pareto genetic algorithm for multiobjective optimization. In *First IEEE Conf. on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, pages 82–87, Piscataway, NJ, 1994.
22. J. Knowles and D. Corne. Approximating the nondominated front using the pareto archived evolution strategy. *Evolutionary Computation*, 8(2):149–172, 2000.
23. A. Lara, G. Sanchez, C. Coello, and O. Schutze. Hcs: a new local search strategy for memetic multiobjective evolutionary algorithms. *IEEE T Evolut. Comput.*, 14(1):112–132, 2010.
24. H. Li and Q. Zhang. Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II. *IEEE T Evolut. Comput.*, 13(2):284–302, 2009.
25. J. Mata, J. Alvarez, and J. Riquelme. Mining numeric association rules with genetic algorithms. In *5th International Conference on Artificial Neural Networks and Genetic Algorithms*, Taipei, Taiwan, April 2001.
26. J. Mata, J. Alvarez, and J. Riquelme. An evolutionary algorithm to discover numeric association rules. In *ACM Symposium on Applied Computing*, Madrid, Spain, March 2002.
27. K. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer, Norwell, MA, 1999.
28. S. Ramaswamy, S. Mahajan, and A. Silberschatz. On the discovery of interesting patterns in association rules. In *24rd International Conference on Very Large Data Bases*, pages 368–379, San Francisco, CA, USA, 1998.
29. R. Rosenberg. Simulation of genetic populations with biochemical properties. Master's thesis, Univ. Michigan, Ann Harbor, Michigan, 1967.
30. J. Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. In *First International Conference on Genetic Algorithms*, pages 93–100, Pittsburgh, USA, 1985.
31. E. Shortliffe and B. Buchanan. A model of inexact reasoning in medicine. *Math. Biosci.*, 23:351–379, 1975.
32. R. Srikant and R. Agrawal. Mining quantitative association rules in large relational tables. In *ACM SIGMOD International Conference on Management of data (SIGMOD96)*, pages 1–12, Montreal, Quebec, Canada, 1996.
33. N. Srinivas and K. Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3):221–248, 1994.
34. S. Srinivasan and S. Ramakrishnan. Evolutionary multi objective optimization for rule mining: a review. *Artificial Intelligence Review*, 36(3):205–248, 2011.
35. Q. Zhang and H. Li. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE T Evolut. Comput.*, 11(6):712–731, 2007.
36. E. Zitzler and S. Kunzli. Indicator-based selection in multiobjective search. In *Parallel Problem Solving from Nature, PPSN VIII in LNCS*, volume 3242, pages 832–842, 2004.
37. E. Zitzler, M. Laumanns, and L. Thiele. Spea2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In *Evolutionary Methods for Design, Optimization and Control: Applications to Industrial and Societal Problems*, pages 95–100, Barcelona, Spain, 2001.
38. E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE Transactions Evolutionary Computation*, 3(4):257–271, 1999.