

# PAR-KNIME: conjunto de plugins para extraer reglas de asociación cuantitativas en KNIME

## PAR-KNIME: A set of plugins to extract quantitative association rules in KNIME

**Katherine González Pupo<sup>1</sup>, Orenia Lapeira Mena<sup>2</sup>, Diana Martín Rodríguez<sup>3</sup>**

<sup>1</sup>Escola Politécnica da Universidade de São Paulo

<sup>2,3</sup>Universidad Tecnológica de la Habana, José Antonio Echeverría, Cujae, La Habana, Cuba

Correo electrónico: [kpupo@larc.usp.br](mailto:kpupo@larc.usp.br)

Este documento posee una licencia Creative Commons Reconocimiento/No Comercial 4.0 Internacional 

Recibido: 17 de octubre de 2017    Aprobado: 22 de enero de 2018

### Resumen

En la actualidad se han desarrollado diferentes herramientas de minería de datos para extraer conocimiento interesante en grandes volúmenes de datos. Una de las herramientas más populares es KNIME, porque ha demostrado ser muy útil para los usuarios al guiar fácilmente el proceso de descubrimiento de conocimiento en los datos a través de un flujo de trabajo. Sin embargo, esta herramienta no cuenta con funcionalidades que le permitan extraer reglas de asociación cuantitativas de alta calidad. El descubrimiento de reglas de asociación es una de las técnicas de minería de datos más utilizada para extraer conocimiento interesante a partir de bases de datos. En este trabajo se propone PAR-KNIME, un conjunto de plugins que representan 7 algoritmos evolutivos recomendados en la literatura para extraer reglas de asociación cuantitativas. Además, se incluye un nuevo nodo para evaluar las reglas obtenidas por los algoritmos, calculando diferentes medidas de calidad de las reglas. El uso de PAR-KNIME y los resultados obtenidos por los algoritmos se muestran a través de dos ejemplos ilustrativos.

Palabras claves: minería de datos, reglas de asociación cuantitativas, Algoritmos Evolutivos, KNIME

### Abstract

At present it have been developed different data mining tools to extract interesting knowledge in large volumes of data. One of the most popular tools is KNIME, because it has proved very useful for users to easily guide the process of knowledge discovery in data through a workflow. However, this tool does not have features that allow to extract quantitative association rules with high quality. The discovery of association rules is one of the most mining techniques used to extract interesting knowledge from database. We propose PAR-KNIME, a set of plugins that represent seven evolutionary algorithms proposed in the literature to extract quantitative association rules. Moreover, a new node is included to evaluate rules algorithms obtained by calculating different quality measures rules. We develop some examples to show the results obtained by PAR-KNIME.

Key words: data mining, quantitive association rules, evolutive algorithms, KNIME

## INTRODUCCIÓN

La técnica de Reglas de Asociación es una de las más utilizadas para extraer conocimiento interesante a partir de una base de datos [1,2]. Forma parte de las técnicas descriptivas, las cuales son utilizadas para explorar las características de los datos y no para predecirlo [1]. Expresan relaciones no explícitas entre diferentes atributos de una misma base de datos. Se representan de la forma  $X \rightarrow Y$  donde  $X$  es el antecedente o predecesor de la regla y  $Y$  el consecuente o sucesor. Lo cual significa que si se cumple  $X$  entonces se cumple  $Y$  con un alta 8 probabilidad [1,2]. Los conjuntos  $X$  y  $Y$ , son disjuntos, o sea, los ítems (atributos) de  $X$  son diferentes a los de  $Y$ .

Por tal motivo, en este trabajo se presenta PAR-KNIME, un conjunto de plugins que contiene 7 algoritmos evolutivos propuestos en la literatura (ALATAS [3], GAR [4], GENAR [5], EARMGA [6], MODENAR [7], MOPNAR [8], QAR-CIP-NSGAI [9]) que permiten extraer RACs de alta calidad en la herramienta KNIME [10].

Por otra parte, obtener reglas con alta calidad es el objetivo de todo algoritmo, es por eso que para conseguir un conocimiento interesante es importante medir cuán buenas pueden ser las reglas extraídas. Para ello, diversos autores [11, 12, 13] han propuesto medidas que han sido publicadas en la literatura. Por tal motivo en este trabajo se incluye un nodo que contiene varias de estas medidas, las cuales son de las más utilizadas actualmente para evaluar la calidad de las reglas.

También se muestra cómo pueden ser utilizados los algoritmos de PAR-KNIME para extraer RACs en diversas bases de datos. El código fuente de PAR-KNIME se encuentra disponible online en <http://sourceforge.net/projects/parknime/>.

Obtener reglas con alta calidad es el objetivo de todo algoritmo, es por eso que para conseguir un conocimiento interesante es importante medir cuán buenas pueden ser las reglas extraídas. Para ello diversos autores han propuesto medidas que han sido publicadas en la literatura. Por tal motivo en este trabajo se incluye un nodo que contiene varias de estas medidas, las cuales son de las más utilizadas actualmente para evaluar la calidad de las reglas.

La organización del trabajo es como sigue: La sección 2 contiene información preliminar sobre la herramienta de minería de datos KNIME y sobre las reglas de asociación cuantitativas. En la sección 3 se presenta una breve explicación sobre los algoritmos evolutivos que forman parte de PAR-KNIME, se describe la arquitectura y los nodos de nuestra propuesta y se muestra su uso a través de dos ejemplos ilustrativos. Finalmente, en la sección 4 se comentan las conclusiones alcanzadas con este trabajo.

## PRELIMINARES

En esta sección, primero se presentan las características generales de la herramienta KNIME. Luego se introducen las definiciones básicas de RACs y algunas de sus medidas de calidad.

### KNIME

Konstanz Information Miner (KNIME) [10] fue creada por la Universidad de Constanza de Alemania. Actualmente es una de las herramientas más utilizadas para realizar MD. Tiene como objetivo desarrollar procesos de descubrimiento de conocimiento en BD en un entorno visual. Permite la integración con varias tecnologías como Weka, R y python/jython, la mayoría de las cuales están disponibles mediante la instalación de las respectivas extensiones [10].

Se le considera una herramienta gráfica, ya que permite construir flujos de trabajo. Estos flujos ayudan en el proceso de descubrimiento de conocimiento y se componen de nodos y flechas que se pueden combinar entre sí. Los nodos contienen funcionalidades tales como, algoritmos de MD, formas de conexión a la BD, entre otros. Las flechas indican el orden de ejecución y el flujo de la información. La figura 1 muestra un ejemplo del ambiente de trabajo de la herramienta donde en la parte central se muestra el flujo de trabajo realizado, en la parte izquierda se encuentran todos los proyectos guardados además de los nodos disponibles y en la parte derecha se presenta una descripción del nodo que se ha seleccionado.

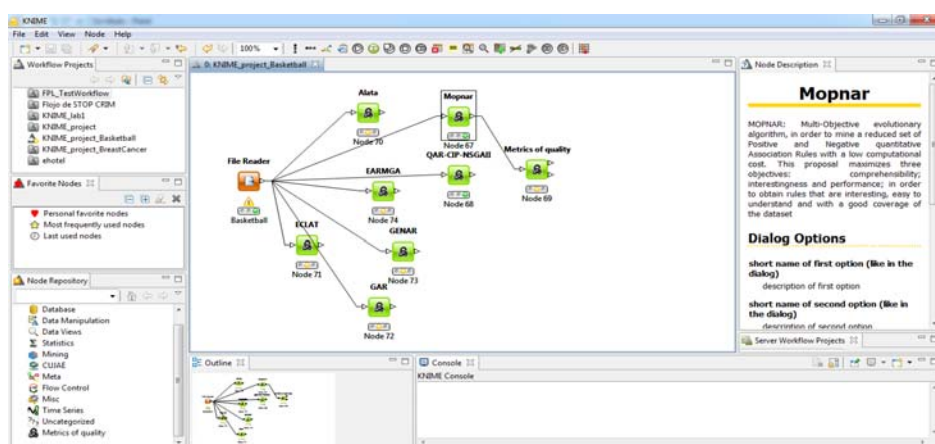


Fig. 1. Ambiente de trabajo en KNIME

Una de las mayores ventajas de KNIME es su extensibilidad, pues permite adicionar nuevas unidades de procesamiento, visualización y tratamiento de datos. KNIME puede ser extendida a través de plugins, lo cual favorece al enriquecimiento de la herramienta con la introducción de nuevas funcionalidades. Como se mencionó anteriormente una de las técnicas más utilizadas para extraer conocimiento de las bases de datos son los algoritmos de reglas de asociación. En estos momentos KNIME solo dispone del algoritmo a priori [14], un algoritmo clásico de extracción de reglas de asociación, basado en ítems frecuentes que presenta varias limitaciones para extraer RACs.

### Reglas de Asociación Cuantitativas

Las reglas de asociación obtenidas en BDs con valores numéricos son denominadas RAC [14], donde cada ítem es un par *atributo-intervalo*. Por ejemplo, una RAC es Salario [3000,3500] -> NumCoches.

Muchos de los estudios clásicos presentan dificultades para descubrir RACs debido a que los atributos cuantitativos contienen muchos valores distintos. Para evitar este problema un método normalmente utilizado es dividir el dominio de los atributos numéricos en intervalos, lo cual es un problema crítico en la extracción de RACs porque la información no está clasificada. Además, dichos intervalos pueden tener una gran influencia sobre las reglas obtenidas. Por esta razón, varios investigadores han propuesto métodos para aprender los intervalos de las RACs [3, 4, 7, 15].

Para determinar la calidad de la regla  $X \rightarrow Y$ , se han propuesto diversas métricas. Entre estas, el soporte y la confianza son las más utilizadas por la mayoría de los algoritmos, donde el soporte representa la probabilidad de ocurrencia de  $X$  y  $Y$  en la BD, y la confianza la probabilidad de ocurrencia de la regla si se cumple el antecedente. Implica también la fortaleza de la regla [5]. Ambas se definen como:

$$\text{Soporte } (X \rightarrow Y) = \text{SOP}(XY) \tag{1}$$

$$\text{Confianza } (X \rightarrow Y) = \text{SOP}(XY) / \text{SOP}(X) \tag{2}$$

donde SOP (I) es la probabilidad de que un conjunto de ítems I aparezca en la BD.

La mayoría de los algoritmos de reglas de asociación intentan obtener reglas con un mínimo de confianza y soporte definidos. Sin embargo, existen algunos inconvenientes al trabajar con estas métricas para determinar la calidad de las reglas [12]. La confianza no detecta independencia estadística o dependencia negativa entre los ítems de las reglas, ya que no tiene en cuenta el soporte del consecuente. En cuanto a los ítems con soporte muy alto, estos pueden generar reglas de baja calidad ya que cualquier conjunto de ítems puede ser un buen predictor de ellos. A partir de estos problemas se han propuesto en la literatura diversas medidas como alternativas para

determinar la calidad de las reglas, como por ejemplo: Lift [13], Factor de Certeza (FC) [16], NetConf [17], yulesQ [18], Conviction [19], entre otras.

## **PAR-KNIME**

En esta sección se presenta una breve descripción de los algoritmos evolutivos incluidos en PAR-KNIME. Además, se presenta la arquitectura propuesta para estos plugins. Finalmente, se muestra un ejemplo ilustrativo del uso de los nodos propuestos en la herramienta KNIME.

### **Algoritmos evolutivos para extraer RACs incluidos en KNIME**

En esta sección se describen los algoritmos evolutivos que permiten extraer RACs, incluidos en la herramienta KNIME.

En los últimos años diversos investigadores han propuestos método evolutivos [8, 9, 15] para extraer RACs. La mayor motivación de utilizar algoritmos evolutivos para resolver este problema es porque son considerados como una de las técnicas de búsquedas más exitosas para problemas complejos, además han demostrado ser muy buenos en el aprendizaje y la extracción de conocimientos.

Uno de los primeros algoritmos propuestos para extraer RACs fue GENAR (GENetic Association Rules) [5], presentado en el año 2001. En este algoritmo un cromosoma codifica una regla de asociación, que contiene los intervalos máximos y mínimos de cada atributo numérico. Sin embargo, cada regla contiene el total de atributos de la BD y solo el último atributo formará parte del consecuente, reduciendo considerablemente el conjunto de reglas interesantes que puede encontrar el método. La función objetivo solo considera el número de ejemplos incluidos en la regla y penaliza las reglas que cubran ejemplos de la BD que también han sido cubiertos por otras reglas.

Otro método propuesto por los mismos autores de GENAR es GAR (Genetic Association Rules) [4]. Este método extiende de GENAR para encontrar conjuntos de ítems frecuentes en BDs numéricas evitando discretizar los valores de los atributos. Cuando el algoritmo termina el proceso evolutivo ejecuta otro procedimiento para generar las reglas a partir de los ítems frecuentes que fueron extraídos.

En el año 2009 fue presentado EARMGA (Evolutionary Association Rules Mining with Genetic Algorithm) [6], un algoritmo que permite identificar reglas sin necesidad de definir un valor mínimo de soporte. Las reglas más interesantes se determinan de acuerdo con el grado de interés definido por la función objetivo, el cual está basado en el soporte de la regla del antecedente y del consecuente.

Otro algoritmo propuesto fue ALATAS [3] que permite buscar simultáneamente los intervalos de los atributos cuantitativos y descubrir reglas positivas y negativas asociadas a esos intervalos.

En el año 2008 fue propuesto un método multiobjetivo llamado MODENAR Multi-objective differential evolution algorithm for mining numeric association rules) [7]. Este método utiliza un algoritmo evolutivo multiobjetivo (AEMO) diferencial para extraer RACs precisas y comprensibles sin necesidad de definir un mínimo de soporte y confianza. Este método optimiza cuatro objetivos para mejorar la calidad de las reglas: soporte, confianza, comprensibilidad y amplitud del dominio de los intervalos.

En el año 2013 fue propuesto QAR-CIP-NSGAI (Quantitative Association Rules which maximizes the Comprehensibility, Interestingness and Performance) [9]. Este algoritmo intenta optimizar la comprensibilidad, el interés y el rendimiento de una regla, para lograr un equilibrio entre la exactitud y la interpretabilidad [9], el cual sigue el enfoque Non-dominated Sorting Genetic Algorithm II (NSGA por sus siglas en inglés). Este modelo es un método basado en población. Este tipo de método guía la búsqueda en diferentes direcciones para generar poblaciones de soluciones no dominadas [20]. Se utiliza este modelo para poder lograr un aprendizaje evolutivo de los intervalos de los atributos y una selección de condición para cada regla [21]. El modelo introduce una población externa y un proceso de reinicialización. El objetivo de estos dos componentes es poder almacenar todas las reglas no dominadas encontradas y mejorar la diversidad en las reglas obtenidas [9].

En el año 2014 se propuso otro algoritmo llamado MOPNAR (A New Multi-Objective Evolutionary Algorithm for Mining a Reduced Set of Interesting Positive and Negative Quantitative Association Rules) [8]. Este método permite extraer un conjunto reducido de RACs positivas y negativas con un buen equilibrio entre el número de reglas, el soporte y el cubrimiento de la BD [8]. Dicha propuesta extiende el AEMO basado en descomposición (MOEA/D-DE) [22]. Esto permitirá realizar un aprendizaje evolutivo de los intervalos de los atributos y una selección de las condiciones para cada regla. MOPNAR maximiza tres objetivos: rendimiento, interés y comprensibilidad. Además, este método introduce un proceso de reinicialización y una población externa al modelo evolutivo para promover diversidad en la población, almacenar todas las reglas no dominadas encontradas y mejorar el cubrimiento de la BD.

La tabla 1 resume los algoritmos que contiene PAR-KNIME, identificando los algoritmos genéticos monoobjetivos y los AEMOs que han sido diseñados para extraer RACs.

**Tabla 1. Algoritmos que contiene PAR-KNIME**

Algoritmos	Ref.	AG-mono-objetivo	AEMO
ALATAS	[3]	X	
GAR	[4]	X	
GENAR	[5]	X	
EARMGA	[6]	X	
MODENAR	[7]		X
MOPNAR	[8]		X
QAR-CIP-NSGAI	[9]		X

### Diseño de plugins para la extracción de RACs en KNIME

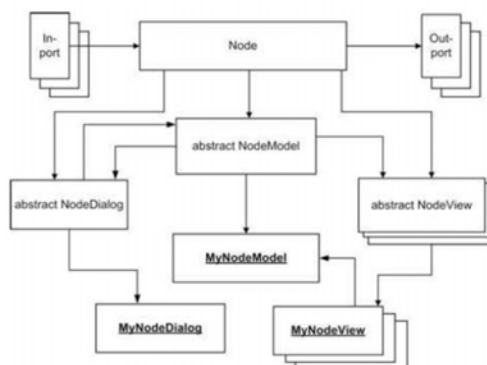
En esta sección, primero se describen las clases de KNIME que le permiten ser fácilmente extensibles, así como una breve descripción de la arquitectura de PAR-KNIME. Luego, se presentan las características de los nodos de esta propuesta.

### Extensibilidad de KNIME. Arquitectura de PAR-KNIME

KNIME se puede extender fácilmente porque está basada en la Plataforma de Cliente Enriquecido de Eclipse (Eclipse RCP, por sus siglas en inglés) [23]. Esta plataforma permite que las aplicaciones sean extensibles porque presenta una arquitectura modular basada en módulos o plugins, utilizando el contenedor de plugins de Eclipse Equinox (para más información acceder al sitio <http://www.eclipse.org/equinox/>) como implementación del conjunto de especificaciones Open Service Gateway Initiative (OSGi, por sus siglas en inglés) [24]. El uso de las especificaciones OSGi permite el desarrollo de plugins y servicios, a partir de los cuales se construyen aplicaciones modulares y extensibles.

KNIME se caracteriza por ser fácilmente extendida, de manera que los usuarios pueden añadir nuevas funcionalidades de acuerdo con sus necesidades. Puede ser extendida a través de estos plugins, la mayoría de los cuales son nuevos nodos, aunque las extensiones pueden ser a cualquier parte de la arquitectura. Para lograr añadir nuevos plugins basados en la plataforma Eclipse, la herramienta propone una estructura de clases determinada.

La figura 2 muestra un diagrama que describe dicha estructura, donde se puede apreciar que la misma se basa en una clase Node, que presenta todas las funcionalidades que van a permitir la implementación de la clase NodeModel, un posible NodeDialog y una o más instancias de la clase NodeView. En adición a esto, cada nodo contiene la cantidad de puertos de entrada y salida de datos, por donde va a transitar la información.



**Fig. 2. Diagrama de clases de un nodo en KNIME**

Descripción de las clases:

- *MyNodeModel*: se encarga de las principales computaciones. Se deben redefinir tres métodos fundamentales: *Configure*, *Execute* y *Restarting*; los cuales permiten, respectivamente, recibir la información de las tablas de entrada, ejecutarlos para crear los datos de salida o modelos y reinicializar los valores por defecto del nodo para que pueda ser ejecutado nuevamente.
- *MyNodeDialog*: es utilizada para especificar la interfaz visual que le va a permitir al usuario hacer ajustes a cada nodo. Posee un set estándar de objetos llamado *DefaultDialogComponent* que permite crear una interfaz gráfica que solo necesita algunos ajustes. Esto es posible porque Knime provee diversas clases que se encargan de crear componentes visuales. Gracias a esto el trabajo se torna más fácil y a su vez estandarizado.
- *MyNodeView* puede ser sobrescrita varias veces para tener diferentes vistas de los datos.

Para el diseño de esta propuesta, se tuvo en cuenta la estructura de clases propias de la herramienta KNIME. En la figura 3 se muestra una vista de la arquitectura, la cual fue proyectada en dos capas: Capa lógica del negocio que contiene paquetes referentes a los algoritmos propuestos y la capa de Acceso a Datos que refleja las clases e interfaces que brinda el entorno de desarrollo.

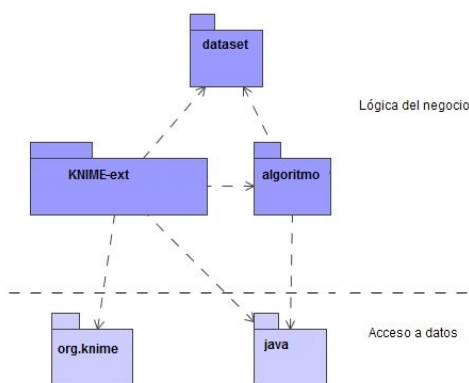


Fig. 3. Estructuración en capas de la propuesta

Descripción de los paquetes que se presentan en esta arquitectura:

- algoritmo.cujae.edu.cu: representa las clases que proporciona KNIME para adicionar nuevos plugins;
- algoritmo: contiene las clases lógicas del algoritmo y los métodos que permiten obtener las reglas de asociación;
- dataset: representa las clases que se encargan del trabajo con las excepciones y con los atributos o variables de la BD;
- org.Knime: Provee toda la estructura de clases necesarias para que el nodo funcione y pueda ser adicionado a KNIME;
- java: Representa a las clases e interfaces que brinda la plataforma de Java para poder implementar los algoritmos sobre esta.

### Características de los nodos de PAR-KNIME

En KNIME, un nodo está compuesto por cero, uno o varios puertos de entrada y salida de datos, además por un conjunto de estados, que indica en cuál estado se encuentra el nodo de todos por los que puede transitar (No configurado, Configurado y Ejecutado) y además posee un área de texto que el usuario puede utilizar para nombrar y así identificar al nodo en el flujo de trabajo. En la **¡Error! No se encuentra el origen de la referencia.** se muestra un ejemplo de un nodo y de todas las partes que lo componen.

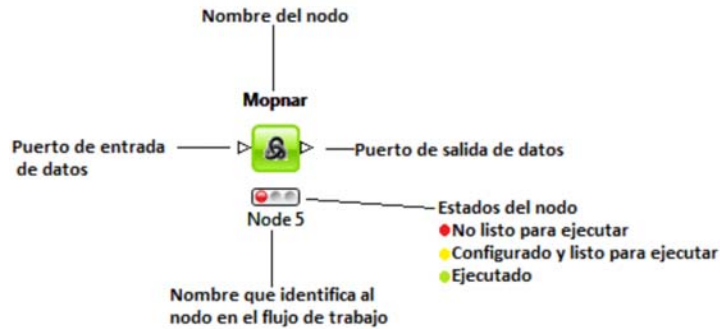


Fig. 4. Principales componentes de un nodo en KNIME

En PAR-KNIME, los nodos que representan a los algoritmos de MD tienen características similares, por tanto, todos cuentan con un puerto de entrada de datos que permite obtener los datos de la BD y un puerto de salida que posibilita mostrar las reglas de asociación obtenidas por el algoritmo.

También PAR-KNIME cuenta con un nodo que permite evaluar la calidad de las reglas obtenidas, llamado MetricsOfQuality. Este nodo permite calcular varias de las métricas de calidad que han sido publicadas en la literatura, tales como: Confianza, Lift, NetConf, yulesQ, Conviction, Factor de Certeza.

MetricsOfQuality tiene como entrada un conjunto de reglas de asociación y la información del soporte del antecedente, del consecuente y el soporte de cada regla. El nodo presenta dos salidas. La primera, representa una tabla con las reglas y sus medidas de calidad, y la segunda el promedio de cada una de las métricas teniendo en cuenta los valores de todas las reglas. La figura 5 muestra un flujo de trabajo de la herramienta, donde se utiliza este nodo y describe las salidas que ofrece el mismo al usuario.

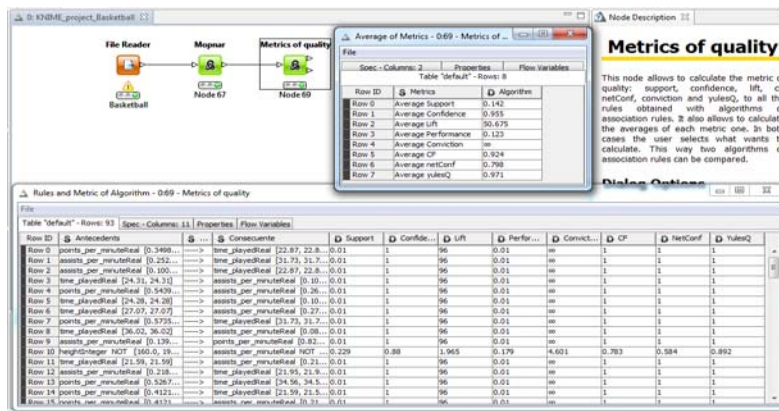


Fig. 5. Ejemplo de flujo de trabajo y salida de los resultados obtenidos

La utilidad de este nodo está dirigida a poder calcular diversas medidas de calidad de las reglas para de esta forma poder obtener resultados más interesantes y que aporten un mayor conocimiento.

## RESULTADOS

En esta sección se muestra cómo se puede utilizar PAR-KNIME. Para ello se eligieron MOPNAR y QAR-CIP-NSGAI, dos de los algoritmos incluidos en KNIME. Para este ejemplo experimental, se seleccionaron las BDs basketball y House16H. La tabla 2 muestra las principales características de las BDs seleccionadas, donde se

indica la cantidad de registros y la cantidad de atributos que poseen, los cuales pueden ser reales, enteros o numéricos. Ambas BDs están disponible en el repositorio de datos de KEEL a través del enlace: <http://sci2s.ugr.es/keel/datasets.php>.

**Tabla. 2 Características de las bases de datos**

Nombre	Atributos(R/E/N)	Cantidad de Registros
Basketball	5 (0/0/5)	96
House16H <sup>1</sup>	17(10/7/0)	22784

Para poder ejecutar los algoritmos MOPNAR y QAR-CIP-NSGAI, es necesario definir los valores de configuración inicial de estos. Los parámetros definidos son los siguientes:

**MOPNAR:**

- Cantidad de objetivos: 3.
- Cantidad de evaluaciones: 50 000.
- Parámetro de control H: 13.
- Tamaño del vector en la vecindad: 10.
- Probabilidad de que la solución padre sea seleccionada de la vecindad: 0,9.
- Máximo número de soluciones reemplazado por cada hijo: 2.
- Probabilidad de cruzamiento: 0,7.
- Probabilidad de mutación: 0,1.
- Factor de amplitud: 2.
- Porcentaje de actualización: 5.

**QAR-CIP-NSGA-II:**

- Cantidad de objetivos: 3.
- Cantidad de evaluaciones: 50 000.
- Tamaño de la población: 100.
- Tamaño del cruzamiento por torneo: 2.
- Probabilidad de cruzamiento: 0,7.
- Probabilidad de mutación: 0,1.
- Factor de amplitud: 2.
- Porcentaje de actualización: 5

Con estos valores, se ha tratado de facilitar las comparaciones, seleccionando parámetros estándares comunes que funcionan bien en la mayoría de los casos.

Para la realización de este experimento se diseñó un flujo de trabajo que contiene un nodo de pre-procesamiento que permite cambiar el nombre de los atributos de la BD con el fin de que estos sean más comprensibles para los usuarios. Una vez se ha transitado por esta fase se ejecuta el algoritmo seleccionado (en este ejemplo MOPNAR o QAR-CIP-NSGA-II) y finalmente se ejecuta el nodo *MetricsOfQuality* para visualizar los valores de las medidas de calidad calculadas. La figura 6 muestra el diseño de este flujo, utilizando el nodo MOPNAR como algoritmo para extraer RACs, el cual pudiera ser sustituido por cualquier otro de los nodos propuestos.

La tabla 3 muestra los resultados medios obtenidos por los algoritmos MOPNAR y QAR-CIP-NSGAI en KNIME, donde #R representa la cantidad de reglas obtenidas; *MedSop*, *MedConf*, *MedLift*, *MedConv*, *MedCF*, *MednetConf*, *MedyulesQ*, representan los valores medio de soporte, confianza, lift, conviction, Factor de Certeza, netConf y yulesQ respectivamente.

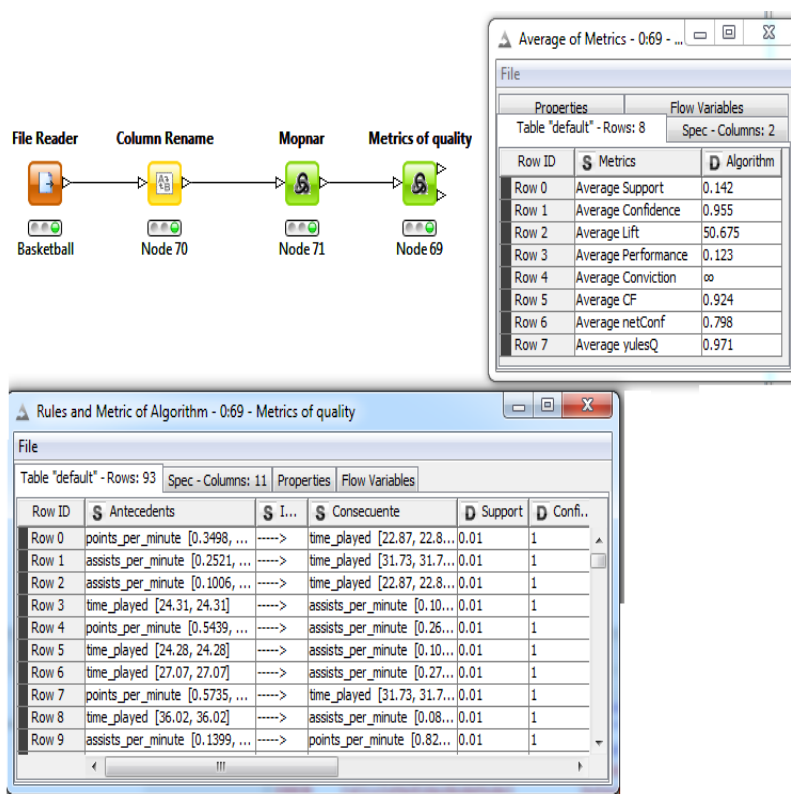
<sup>1</sup> House16H fue diseñada [<http://www.census.gov>] (Acceso Lookup [<http://www.census.gov/cdrom/lookup>]: Resumen del archivo 1). La Oficina del Censo de los Estados Unidos de América proporcionó una base de datos sobre la cual.



Como se puede observar, ambos métodos obtienen RACs con valores elevados para las medidas de interés, cercanos al máximo valor que estas métricas pueden alcanzar.

**Tabla 3. Resultados obtenidos por MOPNAR y QAR-CIP-NSGAI en KNIME**

BD	Algoritmo	#R	MedSop	MedConf	MedLift	MedConv	MedCF	MednetConf	MedyulesQ
Basketball	MOPNAR	93	0,142	0,955	50,675	∞	0,924	0,798	0,971
	QAR-CIP-NSGAI	100	0,058	0,582	2,087	∞	0,4	0,275	0,522
House16H	MOPNAR	91	0,32	0,88	6,08	∞	0,84	0,65	0,97
	QAR-CIP-NSGAI	288,20	0,18	0,93	2549,17	∞	0,91	0,71	0,84



**Figura 6. Diseño de flujo de trabajo para el nodo MOPNAR**

A continuación, se muestran algunos ejemplos de reglas que fueron obtenidas por MOPNAR en la BD Basketball:

R1: Si los puntos por minutos son igual a 0,3 498 entonces el tiempo jugado es 22,87 min.

R2: Si la altura se encuentra entre 160 y 202 cm entonces los puntos por minutos son 0,477.

La base de datos Basketball contiene información sobre jugadores de Basketball. El conocimiento obtenido con MOPNAR muestra que existe una fuerte relación entre los puntos por minuto de un jugador y el tiempo que

este ha jugado. También muestra que la altura de un jugador influye en los puntos por minutos. La R1 indica que si los puntos por minutos son 0,35 entonces el tiempo que ha jugado el basquetbolista ha sido 22,87. La R2 indica que si la altura de un jugador se encuentra entre 160 y 202 cm entonces ha realizado aproximadamente medio punto por minuto.

Como se ha demostrado, resulta muy sencillo utilizar PAR-KNIME. El uso de estos nodos le ofrece nuevas funcionalidades a la herramienta KNIME y le permite al usuario obtener un conjunto de RACs de alta calidad.

## CONCLUSIONES

En este trabajo se presenta PAR-KNIME, un conjunto de plugins para extraer RACs en la herramienta KNIME. Esta propuesta permite la utilización de 7 nuevos algoritmos evolutivos para la extracción de RACs en KNIME. Además, se incluye un nuevo nodo para evaluar las reglas obtenidas por los algoritmos, calculando diferentes medidas de calidad de las reglas. Finalmente, se presenta un ejemplo de cómo utilizar PAR-KNIME para mostrar las ventajas que este brinda y sus facilidades de uso.

## REFERENCIAS

1. Han J, Kamber M. Data Mining: Concepts and Techniques. 3th edition. 2011, Morgan Kaufmann.
2. Hernández Orallo J, Ramírez Quintana MJ, Ferri Ramírez C. Introducción a la Minería de datos, Pearson Educación.
3. Alatas B, Akin E. An efficient genetic Algorithm for automated mining of both positive and negative quantitative association rules. Applied Soft Computing. 2006, 10(3):230-237.
4. Mata J, Álvarez JL, Riquelme JC. An evolutionary algorithm to discover numeric association rules. ACM Symposium on Applied Computing. March 2002. Madrid.
5. Mata J, Álvarez JL, Riquelme JC. Mining numeric association rules with genetic algorithms. 5th International Conference on Artificial Neural Networks and Genetic Algorithms, April 2001, Taipei, Taiwan.
6. Yan X, Zhang C, Zhang Z. Genetic algorithm-based strategy for identifying association rules without specifying actual minimum support. Expert Systems with Applications. 2009, 36(2):3066-3076.
7. Alatas B, Akin E, Karcı A. MODENAR: Multi-objective differential evolution algorithm for mining numeric association rules. Applied Soft Computing, 2008, 8(1), 646pp.
8. Martín D, Rosete A, Alcalá Fernández J, Herrera F. A new multiobjective evolutionary algorithm for mining a reduced set of interesting positive and negative quantitative association rules. IEEE Transactions Evolutionary Computation, 2004, 18(1):54-69.
9. Martín D, Rosete A, Alcalá Fernández J, Herrera F. QAR-CIP-NSGA-II: A New Multi-Objective Evolutionary Algorithm to Mine Quantitative Association Rules. Information Science, 2014, 258: 1-28.
10. Knime. [cited 10/10/2015]; Available from: <http://www.knime.org>.
11. Yan X, Zhang C, Zhang Z. Genetic algorithm-based strategy for identifying association rules without specifying actual minimum support. Expert Systems with Applications, 2009.
12. Berzal F, Balncó I, Vila M. Measuring the accuracy and interest of association rules: A new framework. Intelligent Data Analysis. 2002.
13. Ramaswamy S, Silberschatz A. On the discovery of interesting patterns in association rules. 1998, pp368-379.
14. Srikant R, Agrawal R. Mining quantitative association rules in large relational tables. In ACM SIGMOD International Conference on Management of data (SIGMOD96), 1996. Montreal, Canadá.
15. Alcalá Fernández A, Flugy Pape N, Bonarini A, Herrera F. Analysis of the effectiveness of the genetic algorithms based on extraction of association rules. Fund. Inform, 2010, 98(1):1-14.
16. Buchanan BG, Shortliffe EH. A model of inexact reasoning in medicine. Mathematical Biosciences, 1975.
17. Ahn K, Kim J. Efficient mining of frequent itemsets and a measure of interest for association rule mining". Journal of Information & Knowledge Management, 2004, 3(3):245-257.
18. Tan PN, Kumar V, Srivastava J. Selecting the right interestingness measure for association patterns. Proceedings of the 8th International Conference on Knowledge Discovery and Data Mining (KDD 2002), 2002, Edmonton, Canadá, pp 31-42.

19. Brin S, Motwani R, Ullman JD, Tsur S. Dynamic itemset counting and implication rules for market basket data. ACM SIGMOD Record, 1997, 26(2):255-264.
20. Srinivas N, Deb K. Multiobjective optimization using nondominated sorting in genetic algorithms. Evolutionary Computation, 1995, (2):221-248.
21. Deb K, Pratap A, Agarwal S. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 2002. pp 182–197.
22. LI H, Zhang Q. Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II. IEEE Transactions on Evolutionary Computation, 2009, pp 284-302.
23. Vogel, L. Eclipse 4 RCP - Tutorial: Building Eclipse RCP applications based on Eclipse 4. 2013 [cited 2014; Available from: <http://www.vogella.com/tutorials/EclipseRCP/article.html>.
24. Alliance O. OSGi Alliance / Specifications / HomePage. 2013 [cited 2013; Available from: <http://www.osgi.org/Specifications/HomePage>.
25. Michael R, Berthold NC, et al. Wiswedel. KNIME: the konstanz information miner. June 5-7, 2006.