Universidad
Rey Juan Carlos

TESIS DOCTORAL

Minería de reglas de asociación cuantitativas mediante adaptación de la meta-heurística Optimización por Mallas Variables (VMO)

Autor:

Iván Fredy Jaramillo Chuqui

Directores:

Dr. Javier Garzás

Dr. Andrés Redchuk

Programa de Doctorado en Tecnologías de la Información y las Comunicaciones

Escuela Internacional de Doctorado

2022

Agradecimientos

A todos los que la presente vieren y entendieren.

Inicio de las Leyes Orgánicas. Juan Carlos I

Estos párrafos relacionados al agradecimiento podrían ser interminables, son en realidad tantas personas que se han cruzado en el camino durante este periodo de tiempo que me ha llevado este trabajo de tesis. Personas cercanas y lejanas que de forma directa e indirecta contribuyen al desarrollo de un trabajo, son varias las experiencias en escenarios diversos que han sido útiles, porque me ha permitido conocer y compartir pensamientos similares y opuestos. Todo esto ha enriquecido las ideas, los procesos y resultados en esta investigación.

En primer lugar y como no podría ser de otra manera, este trabajo le debe mucho a Javier Moguerza, quien ha hecho posible tanto el inicio como la continuidad de este trabajo de investigación, a mi director de tesis Javier Garzás, que con su conocimiento y experiencia ha sabido guiarme, sobre todo durante la última fase de este trabajo. También expreso un reconocimiento a mi codirector de tesis, Andrés Redchuk, por la confianza que ha depositado en mí desde el primer momento del comienzo de este proyecto. Gracias por su paciencia y sus enseñanzas, las cuales me han ayudado y servido para desenvolverme en este proceso, tanto a nivel científico como personal.
También quiero dar las gracias a David Ríos, quien estuvo con nosotros al inicio de este camino con el proyecto de maestría en Ingeniería de la Decisión, eslabón principal para llevar a cabo este curso de doctorado. Un sincero reconocimiento a cada uno de los profesores que de una u otra forma me han sabido escuchar, especialmente a Amilkar, que aún siendo un profesor con muchas ocupaciones y no tener alguna obligación con mi trabajo, siempre me ha escuchado y sugerido ideas para mejorar y encaminar la investigación. Además, me gustaría destacar y agradecer el trabajo y apoyo de Roberto y

Carlos; ha sido una inolvidable experiencia aprender y compartir con ellos, tanto en lo personal como en el trabajo profesional.
A todos los compañeros que trabajan y han trabajado durante todos estos años en los estudios doctorales, como parte del grupo de docentes patrocinados por nuestra noble institución, la Universidad Técnica Estatal de Quevedo.
Al Doctor Eduardo Díaz en calidad de rector de la UTEQ, por haber depositado su confianza y haberme favorecido con el apoyo de la institución para todo este proceso. Además, quiero agradecer y dedicar esta tesis a mi familia. A mi esposa, Yadira, mis hijos Marco y Andrés, quienes han tenido que soportar mi ausencia tanto material como inmaterial en el hogar; a mis padres, Angela y Vicente, que siempre me han apoyado y me han sabido dar lo que he necesitado en todo momento. Gracias sobre todo por inculcarme valores y educación, sin los cuales no hubiera llegado hasta aquí. A mis hermanos, Milton, Luis, Paúl y Diana, gracias por su demostración de cariño y calidez que me han regalado cuando estamos juntos.

Resumen

La presente tesis se enmarca en el contexto de las técnicas de minería de datos, específicamente la minería de reglas de asociación cuantitativa, campo de estudio que se enfoca en la extracción de asociaciones entre los datos y que son representado en forma de reglas. En este trabajo se estudia una propuesta para el descubrimiento de reglas de asociación de calidad, el problema se aplica a conjuntos de datos que contienen variables tanto numéricas como categóricas, además que sus tiempos de respuesta sean aceptables, para esto, una forma muy posible de alcanzar es mediante la adaptación de un nuevo algoritmo.

El trabajo realizado durante la presente tesis se centra en la adaptación del algoritmo VMO (Optimización por Mallas Variables) y su posterior evaluación de la calidad. Como primer eslabón para alcanzar los objetivos planteados en esta tesis es muy necesario acudir a la literatura existente. Trabajos referentes a esta temática han comenzado hace aproximadamente treinta años, y es notable el surgimiento de diferentes líneas de investigación en el campo de la minería de reglas de asociación.

Una de las líneas que ha sido relevante y ha despertado el interés para este trabajo, son las brechas que aún existen entre la aplicación de algoritmos evolutivos y la optimización de las reglas de asociación. Un enfoque que ya tiene considerable literatura, pero que dado la continua evolución de las tecnologías y modelos de optimización, aún son objeto de investigación y experimentaciones con el propósito de alcanzar técnicas de mayor calidad.

Variedades de algoritmos evolutivos han sido utilizados para resolver problemas de descubrimiento de reglas de asociación, cada uno de ellos con diferentes enfoques sobre la problemática. Los problemas de nuestro entorno pueden tener diferentes formas de representación, así pueden ser aplicados modelos de optimización cuando es posible la definición de funciones de calidad. Es así que, un algoritmo puede ser adaptado para la resolución de problemas que cumplan con las condiciones de optimización. En este trabajo la adaptación del algoritmo VMO se ha alcanzado con éxito, para ello ha sido necesario trabajar sobre un ambiente de desarrollo denominado QUANTMINER. Las

funciones básicas que se encuentran en este entorno reduce significativamente las implementaciones que pueden considerarse irrelevantes a la problemática, de tal manera que permite focalizar el esfuerzo en los componentes esenciales de la solución, pero además de estos beneficios, las herramientas traen implementaciones de otros algoritmos que sirven como referentes en una evaluación comparativa.

La técnica propuesta en este trabajo se ha denominado QARM_VMO (Minería de Reglas de Asociación Cuantitativas mediante algoritmo VMO) basado en el algoritmo VMO para optimización continua. El algoritmo es capaz de trabajar con datos representados en forma tabular que incluyen variables tanto numéricas como categóricas. Un esquema de regla debe ser definido por el usuario para comenzar el proceso de exploración de reglas, en el esquema se especifican las variables tanto del antecedente como del consecuente de la regla. Adicional a esto se requieren de parámetros tales como los umbrales mínimos de soporte y confianza, también se puede personalizar valores referentes a la población y al número de evaluaciones.

Los resultados obtenidos por QARM_VMO han sido satisfactorios, en las pruebas experimentales el algoritmo demostró alta precisión en la identificación de reglas de asociación importantes, dentro de conjuntos de datos tanto sintéticos como reales.

Las pruebas comparativas de rendimiento fueron efectuados con diez conjuntos de datos reales y seis algoritmos para extracción de reglas, este grupo se compone del tradicional *APRIORI*, dos algoritmos evolutivos multi-objetivo *MODENAR-A* y *QAR_CIP_NSGAII-A*, dos algoritmos mono objetivo *GAR* y *GENAR*, también se agrega la versión genética de QUANTMINER. Se consiguió integrar los resultados de las pruebas en la herramienta R project mediante el uso de la librería RKeel, que tienen las implementaciones de los algoritmos en el segmento de reglas de asociación, mientras que para QUANTMINER y QARM_VMO los datos fueron generados desde Java y pasados a R project.

Entre los algoritmos evolutivos mono-objetivo, QARM_VMO alcanzó el valor más alto de calidad obtenido por la función de ajuste, la misma que fue implementada como una función adicional en R-project para el cálculo en las otras técnicas probadas. En comparación con el algoritmo *GAR*, para el peor de los casos QARM_VMO está 1.31 unidades por encima, en el conjunto de datos *basketball*, y en el mejor de los casos está 146.95 unidades por encima en el conjunto de datos *abalone*. En relación al algoritmo *GENAR*, este algoritmo obtuvo valores muy bajos para el función de ajuste, cediendo ventaja considerable a QARM_VMO. En lo referente a la versión genética de QUANTMINER, este se encuentra ligeramente por debajo de QARM_VMO, en la mayor parte de casos con una diferencia menor a uno,

y en dos conjuntos de datos de los diez, QUANTMINER resulta ligeramente superior.
Para la función de calidad implementada en la técnica propuesta y normalizada en el resto de técnicas probadas. Por una parte, en comparación con las versiones genéticas mono-objetivo, QARM_VMO demuestra superioridad en comparación a *GAR* y *GENAR*. En lo referente a QUANTMINER, aunque es ligeramente superior en los resultados, no tiene una diferencia significativa. Por otro lado con respecto al algoritmo clásico *APRIORI* se evidencia que QARM_VMO es superior, y finalmente en comparación con dos técnicas multi-objetivo, se pudo evidenciar que es superior a *MODENAR*, y no hay diferencia significativa con *QAR_CIP_NSGAII-A*.

Índice

Índice de figuras

Índice de Tablas

Capítulo 1

Introducción

1.1. Introducción

Con la evolución de la computadora, como efecto del resultado de grandes investigaciones en la tecnología digital se logran reducciones de componentes, los tamaños considerablemente diminutos de las partes del hardware de computadora, junto a las altas velocidades de procesamiento, almacenamiento y transmisión de los datos, han dado lugar a que programas y algoritmos que consumen altos recursos computacionales, entreguen resultados accesibles [1].

El análisis de grandes volúmenes de datos es posible gracias a la potencia de las tecnologías de hardware y software; varios investigadores soportados en estas tecnologías han realizado valiosos aportes a la ciencia a través de la formulación de modelos matemáticos y estadísticos, que han sido implementados mediante algoritmos en un programa de computadora. Investigadores relacionados con la ciencia de la computación en todo el mundo, han expuesto las bondades que ofrecen las tecnologías computacionales en diversas aplicaciones, reduciendo así, las barreras del espacio y tiempo en el consumo de recursos. Campos como el desarrollo de algoritmos evolutivos bio-inspirados, comportamientos poblacionales y de ciertos minerales que anteriormente por las limitadas capacidades de procesamiento del computador, las investigaciones no trascendieron hasta su implementación [2].

Documentos científicos comienzan a emplear términos relacionados con el análisis de datos, tales como: minería de datos, inteligencia de negocios y análisis inteligente de datos. El termino descubrimiento del conocimiento (KDD por si siglas en inglés)[3] se posesiona en el lenguaje de los adminis-

tradores, gerentes y científicos de la información. Varias áreas de la ciencia se benefician de la minería de datos, y obtienen sorprendentes resultados al lograr el tratamiento de conjuntos de datos inmensos [4].

El desarrollo de los algoritmos evolutivos es una base muy importante, para que las técnicas de minería de datos funcionen adecuadamente con diversidad de tipos de datos y de dimensiones finitas e infinitas [5]. Es así. que algoritmos de exploración total han sido considerados menos eficientes en poblaciones de gran tamaño, en donde resultan útiles y prácticos los evolutivos, que aunque entreguen soluciones aproximadas, estos lo hacen en tiempos y calidad aceptables.

1.2. Planteamiento del problema

Una de las mayores contribuciones a la ciencia de los datos ha sido la *Minería de Reglas de Asociación* (ARM)[3], desde su origen hasta la actualidad ha tenido amplia aplicación en el descubrimiento de patrones de compra a partir de datos transaccionales. Son varios los problemas que han sido modelados con esta técnica, tales como: gestión de relaciones con el cliente (CRM), recomendación de productos, promoción y ventas asociados a productos [6, 7], pero ARM no se limita solo a problemas de marketing y ventas. Éste ha sido adaptado en varias disciplinas, así como: la salud, educación e ingeniería entre otras. En general, se puede decir que su aplicación es posible para cualquier área de las que se disponga de datos, a partir de los cuales se puedan generar reglas que permitan descubrir conocimiento [8, 9, 10].

Se conoce que los conjuntos de datos no están compuestos solo por atributos cualitativos, sino que también están los cuantitativos, que poseen características diferentes tales como el rango amplio en el que se definen los valores. Estos no ocurren con una frecuencia significativa en el conjunto, lo que hace difícil la identificación de patrones con técnicas similares a las utilizadas con atributos categóricos, es así que la Minería de Reglas de Asociación Cuantitativa (QARM)[11] es una subdivisión que abarca a los atributos numéricos en el proceso de búsqueda de patrones por medio de reglas en los datos. Éste enfoque del problema no es nuevo, de hecho ya existen investigaciones que han sido desarrollados desde los mismos principios cualitativos, pero que han reflejado resultados limitados en cuanto a la calidad y elevado consumo de recursos de cómputo [12], dando lugar a un campo abierto para la definición de líneas prometedoras de investigación en este ámbito. Una derivación parti-

cular que se considera importante para esta investigación, es la combinación de atributos tanto categóricos como numéricos, en especial aquellas que tienen como consecuente un atributo de clase, a este tipo de reglas se llaman *Reglas de Asociación Clasificatorias (CARs)* [13].

Muchos de los algoritmos de minería de reglas de asociación numéricos siguen un estilo de desarrollo tradicional, definido en dos sub-procesos como se describe a continuación [14, 15]:

- Encontrar todos los conjuntos de elementos frecuentes de una gran base de datos que satisfagan un valor de soporte definido por el usuario y
- Generar reglas a partir de esos conjuntos de elementos frecuentes que satisfacen un valor de confianza definido por el usuario.

Aunque la exploración de todos los elementos frecuentes son eficientes en conjuntos de datos pequeños y de tipo cualitativo, estos incurren en los siguientes problemas mayores [16, 17]:

- Los usuarios requieren especificaciones tanto de los atributos como de la definición de cada parte de la regla de asociación.
- Conjuntos de datos numéricos con variables continuas tienen que ser ajustados a valores discretos mediante procesos de discretización, esto podría generar una pérdida de reglas interesantes.
- Reglas de asociación obtenidas en su mayoría vienen compuestas solo para atributos categóricos o solo numéricas, esto ocasiona limitaciones para los usuarios.

1.3. Motivación del trabajo de investigación

Descubrir reglas de asociación de calidad en conjuntos de datos numéricos y que sus tiempos de respuesta sean aceptables es el objetivo de investigar la adaptación de un nuevo algoritmo; y que además sea competitivo con los ya existentes. Fabricantes de software de base de datos, minería de datos, estadísticos y otras aplicaciones de análisis de datos están a la expectativa de nuevos descubrimientos en la ciencia de los datos, no disponer de alternativas a una técnica de análisis puede llegar a reducir muy buenas opciones de rendimiento con determinados problemas, afectando a la calidad de toma de decisiones. Como se discute en el apartado 1.2, las tareas de minería de reglas

de asociación tradicionales siguen dos procesos. En el primero se identifican patrones frecuentes a partir del conjunto de datos y el segundo se generan las reglas bajo criterio de la medida de confianza mínima, este enfoque da lugar a los siguientes desafíos:

- La exploración de reglas sin un direccionamiento producen resultados inesperados, agregando costo computacional extras en búsqueda de patrones no requeridos por los usuarios.

- La minería de reglas de asociación es un problema de complejidad NP-difícil, porque la búsqueda de patrones frecuentes bajo una cierta condición se da en un espacio exponencial de 2^n, lo que conlleva a que en grandes bases de datos, solo la fase de generación de conjuntos de ítems frecuentes (Frequent Itemset) tome gran parte de tiempo de cómputo [18].

- Los algoritmos que se caracterizan por realizar una exploración total de las poblaciones son eficientes en conjuntos de datos pequeños; no obstante para reducir costos computacionales en los problemas que incluye gran volumen de datos se han aplicado algoritmos evolutivos, de hecho se han ido adaptando variedades de técnicas metaheurísticas con el fin de encontrar resultados competitivos.

- El advenimiento de la era de "bigdata", donde las bases de datos contienen datos de diferentes tipos y de gran volumen, un ejemplo de bases de datos voluminosas son las redes sociales, en donde prácticamente los lenguajes de consultas tradicionales basadas en SQL no son factibles, sino más bien prevalecen las técnicas de minería de datos, lo que lleva a una evaluación minuciosa de variedades de algoritmos poblacionales.

Una variedad de metaheurística basada en poblaciones es VMO (Variable Mesh Optimization) que presenta un buen nivel de escalabilidad, con resultados bastante competitivos en problemas de optimización continua complejos [19]. Los resultados reportados en [20] fueron prometedores para problemas de tamaño (dimensiones) pequeño (D $<$50), y concluyendo que el algoritmo fue más competitivo a medida que aumentaba la dimensión de las funciones de prueba. Los resultados alentadores en problemas de optimización que ha reportado el algoritmo VMO, además de las anteriormente mencionadas, ha sido uno de las principales razones que ha llevado a experimentar la aplicación de esta metaheurística, para problemas de Minería de Reglas de Asociación.

1.4. Hipótesis, Objetivos y Contribuciones

Dada la motivación expuesta en la sección anterior, la presente tesis plantea la siguiente hipótesis:

Las variaciones y modificaciones introducidas en los procesos de exploración del algoritmo de Optimización por Mallas Variable (VMO), que obtiene resultados competitivos con otros modelos del estado del arte, es probable que la obtención de reglas de asociación numéricas tengan mayor impacto en la calidad del conjunto de reglas seleccionadas.

Esta hipótesis da lugar al planteamiento del objetivo general:

El objetivo de esta investigación es desarrollar un modelo de adaptación para la metaheurística Optimización por Mallas Variable (VMO), a problemas de minería de reglas de asociación, en donde se tiene conjuntos de datos con variables continuas, que con los algoritmos tradicionales se incurre en limitaciones de tiempo y de calidad. Se pretende que el tiempo de computación consumido por el algoritmo modificado sea competitivo y su calidad de selección sea mejor al que precisan otros modelos del estado del arte.

Se puede encontrar con tres tipos de problemas

I. La integración de atributos tanto numéricos como categóricos en un solo esquema de regla, requiere de alguna separación de procesos para la parte cualitativa y cuantitativa.

II. Encontrar un método para adaptar una población basada en un esquema de reglas de asociación, a las estructuras de datos que utiliza el algoritmo VMO, y manipular los operadores de contracción y expansión.

III. La nueva propuesta para búsqueda de reglas de asociación de calidad en un problema determinado, requiere un análisis comparativo con otros ya existentes para demostrar su eficiencia, en principio se comparará con el algoritmo tradicional y otros similares basados en poblaciones.

Para alcanzar este objetivo es importante la revisión de la literatura concerniente en particular a la técnica de minería de reglas de asociación, las investigaciones más recientes en la aplicación de la computación evolutiva para estas tareas. El estudio en detalle del algoritmo VMO y de un framework(ambiente de trabajo) que permita reciclar las funciones básicas del proceso.

Las contribuciones de esta propuesta de investigación son las siguientes:

- Se contribuye a la literatura y estado del arte en lo referente a la minería de reglas de asociación, a la aplicabilidad de variantes de algoritmos evolutivos en problemas de optimización de reglas.
- Una técnica para búsqueda de reglas de asociación basados en un esquema, y que además incluyen atributos tanto numéricos como categóricos es incluido en el estudio.
- Se define un modelo para adaptar una metaheurística basada en población para resolver problemas sobre minería de reglas de asociación.
- Se introduce un estudio comparativo entre el algoritmo propuesto y otros de su clase.

1.5. Metodología

Durante la realización de esta tesis se toma como referencia la metodología experimental, debido a que es la que mejor se ajusta a la naturaleza de los problemas abordados, siguiendo las siguientes etapas:

- Estudiar las metodologías existentes para desarrollar minería de reglas de asociación.
- Estudiar las investigaciones sobre aplicación de algoritmos evolutivos para minerías de reglas de asociación.
- Investigar y analizar implementaciones del algoritmo VMO para casos de problemas combinatorios y continuos.
- Diseñar un modelo de adaptación del algoritmo VMO para resolver problemas de extracción de reglas de asociación cuantitativas.
- Establecer las herramientas de programación y análisis apropiados, para la implementación de los algoritmos y su evaluación.
- Definir casos de prueba para la evaluación de técnicas, y analizar mediante resultados tabulares y gráficos.
- Extracción de conclusiones y planteamiento de las posibles líneas futuras.

1.6. Organización de la tesis

La estructura de esta tesis comprende un primer capítulo introductorio, que resalta la importancia de la investigación y los aspectos más relevantes, que han motivado para llevar a cabo este estudio. Es importante la visión global del problema, como enfocarlo y cuáles serán las mayores contribuciones a la ciencia. El capítulo 2 está orientado a la investigación de la literatura, y del estado del arte sobre los temas que están estrechamente relacionados con el núcleo de la tesis. Estos temas que asocian la adaptación del algoritmo VMO a problemas de minería de reglas de asociación, conlleva a una revisión de los conceptos de minería de datos, y sus variedades de técnicas para clasificación supervisada y no supervisada, enfocando mayor interés en la minería de reglas de asociación, y los algoritmos que se han empleado para resolver estos tipos de tareas.

El capítulo 3 describe la adaptación del algoritmo VMO al problema de minería de reglas de asociación. Comienza detallando la estructura interna de la versión para funciones continuas, se muestra un pseudocódigo que describe las órdenes y funciones principales en forma secuencial del algoritmo, y esto se amplía con el detalle de los operadores de expansión y contracción. La representación lógica de las reglas de asociación también es agregada en este capítulo, y se expone el problema de optimización asociado al proceso. Finalmente se detalla el algoritmo propuesto junto con las funciones de preprocesado.

El capítulo 4 presenta la evaluación de la propuesta, para ello varios algoritmos, conjuntos de datos reales y sintéticos son declarados. Pruebas de sensibilidad para los principales parámetros son realizados y se discuten los resultados alcanzados con ayuda de tablas y gráficos comparativos. Por último en el capítulo 5 se reflejan las conclusiones y contribuciones más relevantes ofrecidas por esta tesis.

Capítulo 2

Revisión de la literatura

2.1. Introducción

Este capítulo presenta una revisión de la literatura que soporta la motivación y construyen las bases teóricas de esta tesis. Junto con la intersección de técnicas de minería de datos y los algoritmos evolutivos, se discuten diferentes áreas dentro de este contexto, como la minería de patrones frecuentes, extracción de reglas de asociación y algoritmos evolutivos.

Este capítulo se centra en los antecedentes teóricos y técnicos de los campos mencionados anteriormente y describe los conceptos subyacentes con ejemplos cuando es necesario. La sección 2.2 cubre los antecedentes esenciales de las técnicas de minería de datos. Esta sección revisa los trabajos típicos relacionados en la minería de patrones frecuentes, y reglas de asociación utilizando enfoques convencionales y métodos evolutivos. Los conceptos básicos del algoritmo genético, las medidas de interés y los antecedentes esenciales de los enfoques evolutivos, se tratan en la sección 2.3. Finalmente, los antecedentes técnicos y trabajos relacionados de una población inicial en algoritmo genético se discuten en la sección 2.4.

2.2. Descubrimiento del conocimiento

2.2.1. Conceptos generales

De manera general se han empleado variedad de términos para la tarea cuyo objetivo principal es encontrar conocimiento en los datos, así se ha denominado Minería de Datos(MD), extracción de conocimiento, análisis in-

teligente de datos, descubrimiento de información y otros nombres más; el término minería de datos ha sido más ampliamente utilizado por estadísticos, analistas de datos y sistemas de información gerenciales; por otro lado KDD fue utilizado por primera vez en 1989 [21], para enfatizar que el conocimiento es el producto final de investigaciones en los datos, y ha sido ampliamente utilizado en los campos de inteligencia artificial y máquinas de aprendizaje [3].

El Grupo Gartner[22] define a la minería de datos como el proceso de descubrir correlaciones significativas, patrones y tendencias a partir de grandes cantidades de datos que pueden estar almacenados en repositorios, usando tecnologías de reconocimiento de patrones, así como técnicas matemáticas y estadísticas. Otras definiciones, así como en [23] puntualiza en el análisis de grandes volúmenes de datos, para encontrar relaciones poco comunes y nuevas formas de resumirlos, de tal forma, que sean comprensivos y útiles para la persona interesada. Como mecanismo de análisis asistido se puede decir que es un término usado para relacionar los datos con las herramientas informáticas y de inteligencia artificial existentes, para encontrar patrones y relaciones dentro de los datos, permitiendo la creación de modelos, es decir representaciones abstractas de la realidad; pero es el descubrimiento del conocimiento que se encarga de la preparación de los datos y la interpretación de los resultados obtenidos, los cuales dan un significado a estos patrones encontrados mediante procedimientos que van más allá de técnicas tradicionalmente conocidas, como los lenguajes de consultas estructurados (Structured Query Language o SQL) [24]. Considerando la definición de sistema de información [25] para analizar los procesos que transforman los datos en información mediante el usos de la informática, se conoce que para obtener un producto es necesario la transformación de la materia prima, así los datos (materia prima) e información (producto), se aproxima a una primera representación de la obtención de conocimiento, donde el grado de conocimiento que un proceso de ordenador puede entregar en un momento depende de la calidad de los datos a ser procesados. Por consiguiente, si los datos son limpiados, transformados y se aplican algoritmos de análisis inteligente de datos, entonces la información obtenida del proceso aporta mayor conocimiento; y se tiene que el valor real de los datos reside en la información que se puede extraer de ellos; información que ayude a tomar decisiones o mejorar la comprensión de los fenómenos que nos rodean.

La minería de datos usa diferentes disciplinas del conocimiento, así como la estadística, matemáticas, inteligencia artificial y técnicas de aprendizaje

basados en exploración de datos; se puede establecer como área de las ciencias de la computación utilizada para dar soporte a la toma de decisiones y aplicada en investigaciones científicas, fisiológica, sociológicas, militares y sistemas estratégicos de decisiones. Recientemente el término "bigdata" empieza a aparecer en los artículos de investigación y ciencia de los datos relacionado más ampliamente con la capacidad de manejar grandes volúmenes de datos [26]; varios estudios de revisiones históricas en relación a este término se remonta hace unos 80 años estrechamente relacionado con lo antes conocido como *explosión de la información* [27]. Desde 1997 muchas definiciones han sido atribuidas a este término, de todos los conceptos tres de ellas son las más populares y han sido ampliamente citadas y aceptadas. La primera es llamada la interpretación de Gartner o 3Vs y este fue una contribución de Douglas Laney [28], es así que el concepto ha ido creciendo con estas tres dimensiones *Velocidad*, *Volumen* y *Variedad* (3Vs). IBM agregó otro atributo al concepto; la *Veracidad* para representar el grado de incertidumbre en los datos como parte del problema en "bigdata", lo que entonces ya se denominó la definición de 4Vs adoptado por IBM. Para sacar máximo provecho en los negocios Microsoft agregó 3Vs más a la definición de Douglas Laney para representar en el problema *Variabilidad*, *Veracidad* y *Visibilidad*.

2.2.2. Evolución de la Minería de Datos

Se dice que los acontecimientos que causan grandes tragedias también son oportunidades para el desarrollo de la ciencia, y las tecnologías de la computación no son una excepción, pues luego de la segunda guerra mundial, el progreso en las tecnologías de comunicación y de transmisión de datos empieza a ser notable; primero en el campo militar y luego en las universidades norteamericanas. La evolución de las tecnologías digitales con la miniaturización mediante la aparición de los transistores y los circuitos integrados, contribuyen progresivamente al desarrollo de software de gestión de datos. Es así que en la década de los años sesenta, cuando la computadora empieza a verse con fines comerciales, surge el concepto de base de datos como una forma de organización lógica de datos en un archivo[29].

En la década de 1970, Edgar Codd [30] hace una contribución potencial al concepto de sistema manejador de bases de datos, con el propósito de facilitar el manejo de las bases de datos que hasta el momento era muy difícil, introduce a través de una investigación "A Relational Model of Data for Large Shared Data Banks" el concepto de modelo relacional, entonces ya se habla de tablas y registros, lo que sería el comienzo de los DBMS relacionales.

Los modelos de bases de datos relacionales evolucionaron básicamente en tres dimensiones (Integridad, Consistencia y Seguridad) garantizando organización y ordenamiento en los datos. Ya en la década de 1980 muchos fabricantes de software empiezan a comercializar sistemas de gestión de bases de datos (DBMS), el lenguaje SQL (Structure Query Language) se estandariza como una herramienta para interactuar con el motor de bases de datos. Otro acontecimiento destacado en la década de los años ochenta es el surgimiento de la inteligencia artificial. Con la aparición de este concepto muchos beneficios se obtuvieron, tales como tecnología de las máquinas de aprendizaje y algoritmos, que junto con las tecnologías de las bases de datos, capacidades de almacenamiento y computación paralela emerge la industria de minería de datos; primero como investigaciones académicas en donde sobresale por ser una colección de algoritmos de dominio público, y finalmente como productos robustos con fines comerciales.

La industria de MD ha madurado de forma acelerada en las últimas décadas, el fortalecimiento y la innovación de los métodos han sido los factores principales del cambio. Este crecimiento ha tomado varias formas, asi:

Los veinte años pasados de la economía ha sido una transición, dentro de una era en que la información ha ido ganando posiciones como elemento valioso en las organizaciones, de esta forma, han llegado a ser las bases para tomar decisiones en muchas industrias.

Las organizaciones están continuamente recolectando grandes volúmenes de información sobre sus clientes, productos, procesos, ventas y muchas acciones adicionales. Son cada vez más, los métodos de captación de datos que se utilizan para contenidos de diferentes tipos. Esta información histórica es la que puede ser minada, para desarrollar modelos predictivos que sirvan de guía para futuras tomas de decisiones.

La continua evolución en el campo de máquinas de aprendizaje por parte de las comunidades académicas, las que introducen nuevos conceptos, nuevos algoritmos y estructuras de sistemas para optimizar el uso de hardware y software. Estos han sido aplicados a problemas del mundo real, con el fin de perfeccionar métodos y técnicas para obtener productos finales para las industrias. La experiencia en la gestión de los datos, ha hecho que madure la importancia de la minería de datos como una fase previa para descubrir conocimiento en grandes volúmenes. El desarrollo de una metodología siste-

mática que capta datos por diferentes medios, los transforma en información y simultáneamente con la existencia de otros procesos, convierte esa información en conocimiento. La década entre 1970 y 1980 ha sido fundamental para el desarrollo de las concepciones así como: la minería de datos, la inteligencia artificial y teorías de los sistemas expertos, que han sido muy bien utilizados por los académicos e investigadores en la construcción de técnicas, para alcanzar excelentes modelos descriptivos y predictivos para diversidad de casos del mundo real.

Los sistemas expertos son un ejemplo claro del uso de técnicas de minería de datos en un sistema informático. Así, se conciben como sistemas que emulan decisiones humanas tomadas anteriormente bajo las mismas condiciones, casos reales como: los de diagnósticos médicos, reparación de máquinas y recomendaciones técnicas han venido siendo empleados por la sociedad como medio de ayuda a la toma de decisiones.

Actualmente, la minería de datos podría correr el riesgo de pensar que consiste una "caja mágica" en el que se ingresa ciertos parámetros de entrada y luego se obtiene una solución de negocio, esta forma de interpretarlo está muy lejos de la realidad, pues para que un modelo de minería de datos sea exitoso se requiere una correcta interpretación, en principio del problema que se quiere resolver, luego de los datos que se han de ingresar, y por último una interpretación de los resultados ajustado a lo que se quiere encontrar.

En los últimos cinco años emerge un nuevo concepto, denominado "big data". Es el término que los científicos de datos están haciendo uso actualmente, para conceptualizar en una sola palabra el volumen datos, tipos de datos variados que se pueden encontrar en las bases de datos, y la velocidad que estos requieren para ser procesados (3Vs). El uso de este término ha sido objeto de debates, unos a favor, otros en contra, acerca de: su relevancia en la ciencia de datos [31] y la frontera en innovación, competitividad y productividad debido a que los datos están embebidos en las actividades diarias.

Datos generados en cantidades exorbitantes por máquinas y humanos, cada segundo en el mundo están siendo almacenados e inferidos por miles de servidores y programas alrededor del planeta[32]. En cierto grado Mayer y Cukier[33] afirman que "big data" podría revolucionar nuestra forma de pensar, trabajar y vivir, al determinar una estrecha dependencia tecnológica para tomar decisiones en un futuro muy cercano, pues los datos hablarían por sí mismos, y literalmente se dejaría que los datos hablen por nosotros.

2.2.3. Taxonomía de las técnicas de minería de datos

El objetivo básicamente consiste en analizar los datos para extraer conocimiento. Este conocimiento puede estar en forma de relaciones, patrones o reglas inferidas desde los datos, o bien en forma de una descripción más concisa (resumen). Estas relaciones o resumenes constituyen el modelo proveniente del análisis de datos. Existen diferentes formas de representar los modelos, y cada una de ellas determina el tipo de técnica que puede usarse para inferencia.

En varias técnicas de MD, una de las fases importantes en el proceso de obtención del modelo es el *Aprendizaje*. A simple vista el término es bastante entendible, se puede dar una definición bastante comprensible de aprendizaje, pero está claro que interesa reflejar el concepto de *aprendizaje artificial*, entonces las definiciones se orientan a técnicas de inferencia de datos, y de procesamiento combinadas con inteligencia artificial, así se tiene la definición:

Desde el punto de vista general, según Mitchell[34] es mejorar el comportamiento a partir de la experiencia (Aprendizaje=inteligencia), o también si se quiere dar una visión más estática se puede decir que es la identificación de patrones, de regularidades existentes en la evidencia. Considerando un punto de vista externo se define como la predicción de observaciones futuras con plausibilidad.

La clasificación de las técnicas y métodos de MD son muy variadas, y dependen del enfoque de cada autor, debido a que se derivan en una taxonomía de acuerdo al tipo de problema que están tratando, así existen clasificaciones considerando el tipo de algoritmo que utilizan, al modelo descriptivo o predictivo, al tipo de aprendizaje supervisado o no supervisado, y también existen otras más al detalle.

En la práctica, los modelos pueden ser de dos tipos: predictivos y descriptivos. Un modelo predictivo es una función matemática, que es capaz de aprender y definir un patrón entre un conjunto de datos de entrada, que por lo general pertenecen a un historial de comportamiento y una respuesta o la variable objetivo [35].

Dentro del campo de reconocimiento de patrones, hay dos enfoques distintos del problema de la clasificación. Por una parte, la clasificación no supervisada, también conocida como “conglomerados” enfoca la clasificación

como el descubrimiento de las *clases* del problema. Las observaciones únicamente vienen descritos por un vector de características, sin que se conozca la *clase* a la que pertenece cada uno de ellos. Así, el objetivo de la clasificación no supervisada es la determinación de los grupos a las que ha de pertenecer cada elemento, por su afinidad en ciertas características[37].
En la Figura 2.1 se indica una taxonomía desde el punto de vista descriptivo y predictivo, que es el más común.

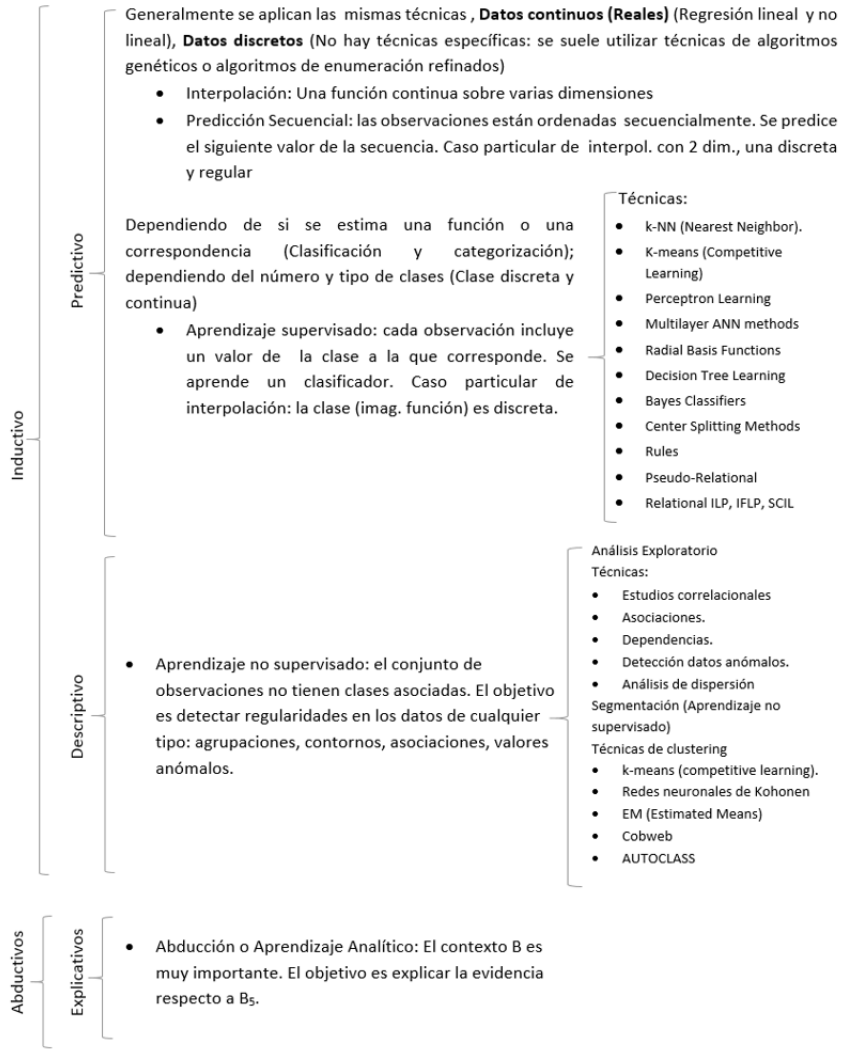

Figura 2.1: Técnicas de minería de datos.

Por otro lado, se encuentra la clasificación supervisada o dirigido. Se trata de construir una función o relación basada en un conjunto de datos de en-

trenamiento, y utiliza esa función para asignar nuevos datos no clasificados. Para esto, el modelo es desarrollado a partir del conjunto de datos de entrenamiento, donde los valores de entrada y salida son previamente conocidos. El modelo generaliza la relación entre las variables entrantes y salientes, y lo usa para predecir el conjunto de datos donde únicamente es conocida la variable de entrada. En los modelos supervisados se requieren un número suficiente de datos (clasificados) para que aprenda correctamente de los datos [36].

Ejemplos de modelos predictivos con aprendizaje supervisado son: redes neuronales, máquinas de vector soporte (SVM) y árboles de decisión. El modelo predictivo también puede usar aprendizaje no supervisado, cuando únicamente se ha proporcionado al modelo un conjunto de datos como entrada, entonces el objetivo es encontrar patrones en los datos basado en la relación que existe entre ellos; conglomerados es la técnica más comúnmente usada que aplica aprendizaje no supervisado.

Dentro del problema de clasificación pueden producirse dos situaciones muy distintas que determinan su complejidad, por una parte, está el escenario en que las clases que definen el problema sean separables, es decir, cuando aquellos elementos con características comunes pertenecen a una misma clase, cuando esto ocurre, se dice que estamos en un "problema degenerado" o "sin ruido". Por otra parte, están los problemas en que las clases no son separables, esto ocurre cuando dos o más elementos que aunque tengan las mismas características pertenecen a clases distintas, es muy habitual en problemas reales y se denominan problemas "con ruido". El ruido en clasificación [38] puede ser tratado desde diversas perspectivas en la fase de preprocesado, sin embargo, para cantidades significativas es necesario verificar las fuentes de esas anomalías.

2.2.4. Etapas de un proceso de minería de datos

Los pasos a seguir para un proceso de minería de datos tienen un patrón común, independiente de la técnica con la que se vaya a trabajar; o a su vez, dentro de un solo proyecto que obedezca a la misma metodología se puede combinar varias técnicas de minería de datos, con el propósito de hallar varios resultados y luego ser analizados en una discusión.
Son varias las metodologías que se han propuesto para el desarrollo de proyectos de minería de datos, tales como: SEMMA (Sample, Explore, Modify, Model, Assess) [39],

DMAIC (Definir, Medir, Analizar, Mejorar, Controlar) [40] y CRISP-DM (Cross Industry Standard Process for Data Mining) [41], sin embargo, el modelo que predomina en los ambientes académicos e industriales es CRISP-DM.

CRISP-DM es un modelo de proceso de minería de datos, que describe los enfoques comunes que utilizan los expertos en minería de datos; se divide en cuatro niveles de abstracción organizados de forma jerárquica, con tareas que van desde el nivel más general hasta los casos más específicos, y organiza el desarrollo de un proyecto de minería de datos en una serie de seis fases (ver Figura 2.2).

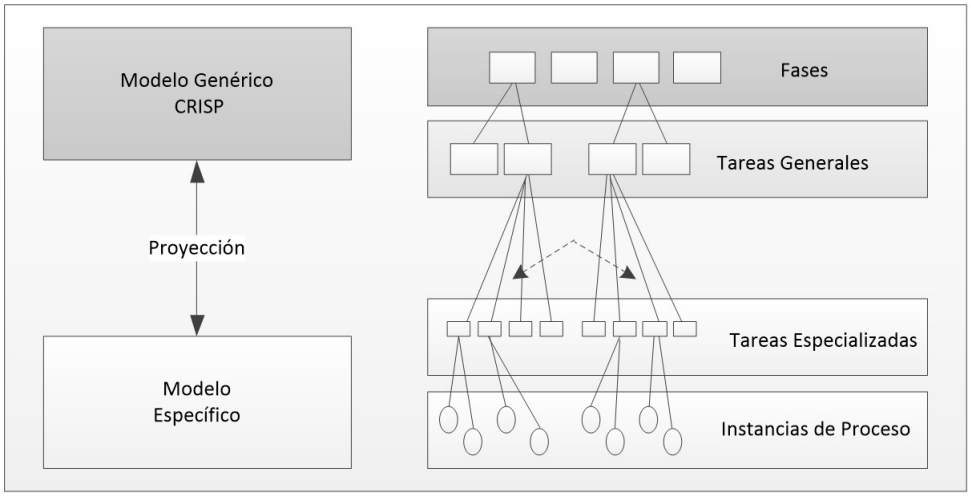

Figura 2.2: Esquema de cuatro niveles CRISP-DM

Cada fase es estructurada en varias tareas generales de segundo nivel. Estas se proyectan a tareas específicas, donde finalmente se describen las acciones que deben ser desarrolladas para situaciones específicas, pero en ningún momento se propone como realizarlas.

A continuación, en la Figura 2.3 se representa el modelo de seis fases y sus interacciones, note que, los datos es el centro de los procesos en cada fase, los datos son objeto de transformaciones durante el ciclo, algo similar a que cuando ingresa la materia prima a un proceso de construcción de un producto terminado, y entonces la fase precedente se encarga de transformar para convertir en la materia prima de la siguiente fase.

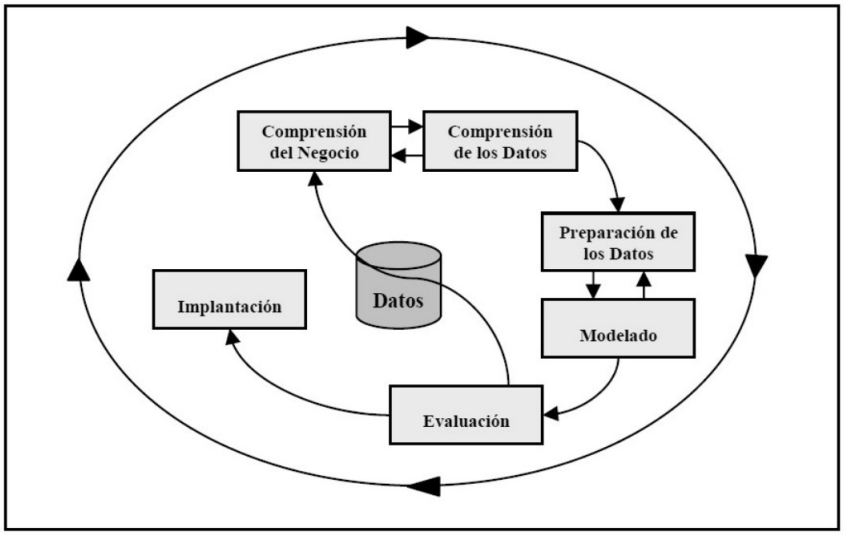

Figura 2.3: Modelo de proceso CRISP-DM

1. **Comprensión del negocio o el problema** es probablemente la más importante, aglutina las tareas de comprensión de los objetivos y requisitos del proyecto desde una perspectiva empresarial o institucional, con el fin de convertirlos en objetivos técnicos y en un plan de proyecto.

2. **Comprensión de datos** comprende la recolección inicial de datos, con el objetivo de establecer un primer contacto con el problema, familiarizándose con ellos, identificando su calidad y estableciendo las relaciones más evidentes que permitan definir las primeras hipótesis.

3. **Preparación de Datos** incluye las tareas generales de selección de datos a los que se va a aplicar una determinada técnica de modelado, limpieza de datos, generación de variables adicionales, integración de diferentes orígenes de datos y cambios de formato.

4. **Modelado**, selecciona las técnicas más apropiadas para el proyecto de minería de datos específico.

5. **Evaluación**, esta tarea involucra la evaluación del modelo en relación a los objetivos del negocio, y busca determinar, si hay alguna razón de negocio para la cual el modelo sea deficiente, o si es aconsejable probar el modelo en un problema real, si el tiempo y restricciones lo permiten

6. **Implementación**, una vez que el modelo ha sido construido y validado se transforma el conocimiento obtenido en acciones dentro del proceso de negocio, sea que el analista recomiende acciones basadas en la

observación del modelo y sus resultados para diferentes conjuntos de datos o como parte del proceso, como por ejemplo, en análisis de riesgo crediticio, detección de fraudes, etc.

En el trabajo [12], los autores concluyen que SEMMA funciona perfectamente cuando se tiene un sistema SAS, el cual es muy popular en empresas grandes. Po otra parte CRISP-DM no sólo se ajusta un poco más a los parámetros de la KDD(Knowledge Discovery in Database)[3], sino también a los procesos que una empresa realiza con los datos. También se puede percibir cierta similitud en el proceso de CRISP-DM, con otros asociados a desarrollo de proyectos software como RUP (Proceso Unificado Racional)[42], en donde las fases de CRISP-DM aparentan ser similares con respecto al ciclo de vida.

Las metodologías citadas tienen un enfoque empresarial e industrial, en el caso de SEMMA, su estándar se vincula con el software que la empresa fabrica, CRISP-DM tiene gran similitud al modelo KDD, el cual tiene un enfoque académico e investigativo.

La Figura 2.4 ilustra un esquema representativo de la jerarquía del conocimiento, y a su vez detalla la relación entre volumen y valor de los datos en cada nivel.

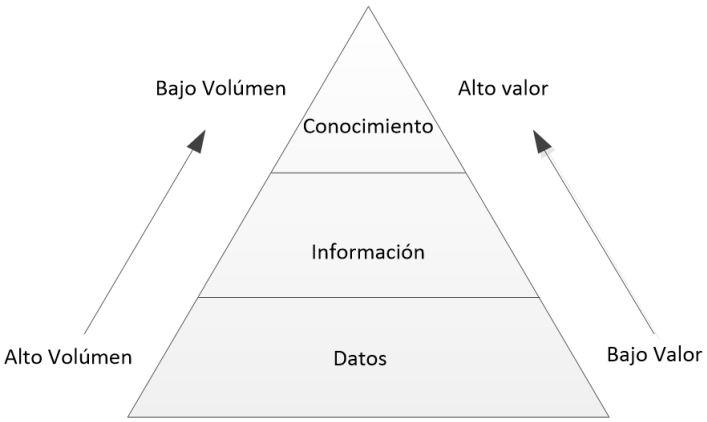

Figura 2.4: Esquema jerárquico del conocimiento

Las implementaciones para el descubrimiento de conocimiento procesan automáticamente grandes cantidades de datos, para ser utilizado por el usuario según su conveniencia. En general, KDD es el proceso no trivial de identificar patrones válidos, novedosos, útiles y comprensibles a partir de los datos.

El proceso de KDD (ver Figura 2.5) consiste en usar métodos de minería de datos (algoritmos) para extraer (identificar) lo que se considera como conocimiento, para ello es necesario las especificaciones de parámetros sobre las fuentes de datos, además de tareas de preprocesamiento y postprocesamiento. Se estima que la extracción de patrones (minería) de los datos ocupa solo el 15 % - 20 % del esfuerzo total del proceso de KDD.

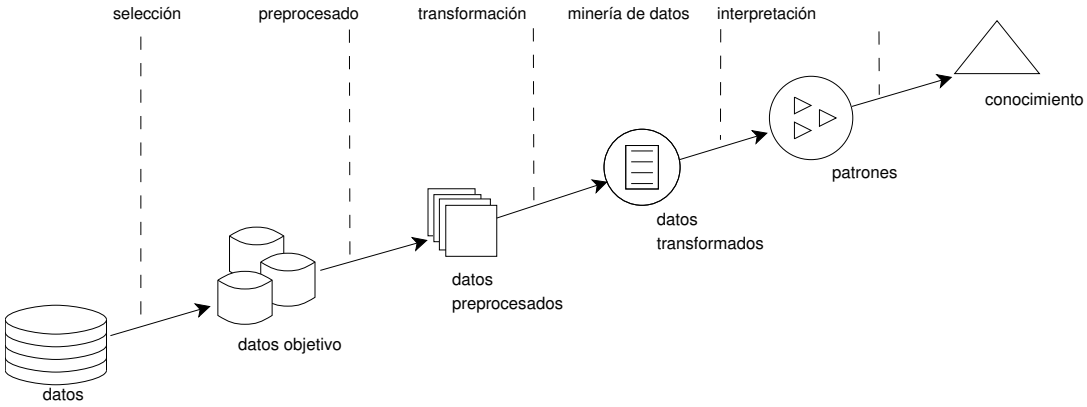

Figura 2.5: Etapas para extracción de conocimiento (KDD)

El proceso de descubrimiento de conocimiento en bases de datos involucra varios pasos:

1. Determinar las fuentes de información que pueden ser útiles y dónde conseguirlas.

2. Diseñar el esquema de un almacén de datos (Data Warehouse), que consiste en unificar de manera operativa toda la información recogida.

3. Implantación del almacén de datos, que permita la navegación y visualización previa de sus datos para discernir qué aspectos interesantes pueden ser estudiados. Es la etapa que puede llegar a consumir el mayor tiempo.

4. Selección, limpieza y transformación de los datos que se van a analizar. La limpieza y preprocesamiento de datos se logra diseñando una estrategia adecuada para manejar ruido, valores incompletos, secuencias de tiempo, casos extremos (si es necesario), etc.

5. Seleccionar y aplicar el método de minería de datos apropiado, incluye la selección de la tarea de descubrimiento a realizar (por ejemplo: clasificación, agrupamiento, regresión, etc.), la selección de los algoritmos

a utilizar, la transformación de los datos al formato requerido por el algoritmo específico de minería de datos y llevar a cabo el proceso de minería de datos. Además se buscan patrones que puedan expresarse como un modelo o simplemente que expresen dependencias de los datos, el modelo encontrado depende de su función (clasificación) y de su forma de representarlo (árboles de decisión, reglas, etc.), se tiene que especificar un criterio de preferencia para seleccionar un modelo dentro de un conjunto posible de modelos, se tiene que especificar la estrategia de búsqueda a utilizar (normalmente está predeterminada en el algoritmo de minería)

6. Evaluación, interpretación, transformación y representación de los patrones extraídos. Es una etapa enfocada en los resultados, y posiblemente regresar a los pasos anteriores. Esto puede involucrar repetir el proceso, quizás con otros datos, otros algoritmos, otras metas y otras estrategias. Este es un paso crucial en donde se requiere tener conocimiento del dominio. La interpretación puede beneficiarse de procesos de visualización, y sirve también para borrar patrones redundantes o irrelevantes.

7. Difusión y uso del nuevo conocimiento, para incorporar los hallazgos al sistema (normalmente para mejorarlo), lo cual incluye la resolución de conflictos potenciales en relación a lo existente. El conocimiento se obtiene para realizar acciones, ya sea incorporándolo dentro de un sistema de desempeño o simplemente para almacenarlo y reportarlo a las personas interesadas.

En este sentido, KDD implica un proceso interactivo e iterativo involucrando la aplicación de varios algoritmos de minería de datos.

2.2.5. Aplicaciones de la minería de datos

Constantemente se desarrollan investigaciones que buscan aplicar las técnicas de minería de datos, o el proceso KDD para resolver un determinado problema concerniente a diversas áreas de la ciencia, tales como: sociales, psicológicos, tecnológicos, económicos, académicos, etc.
El éxito que han reflejado estas herramientas en las tomas de decisiones a nivel empresarial y tecnológico, da lugar a que organizaciones y personas de diversas regiones del mundo tomen conciencia de la importancia de los datos.

En la salud ha incrementado significativamente la dependencia de los datos. Los investigadores de la salud, médicos y farmacéuticas responden efi-

cientemente a los problemas de salud actuales, cuando ellos tienen acceso a variedades de sistemas de información clínicos con grandes bases de datos [43]. Las bases de datos de los sistemas de información clínicos tienen grandes volúmenes de registros sobre historias clínicas, diagnósticos médicos, información de monitoreo, etc. Datos que resultan bastante útiles para los sistemas de soporte a las decisiones clínicos (CDSS)[44].
Otras aportaciones de la minería de datos en el campo de la medicina es mejorar la toma de decisiones en pronósticos y diagnóstico para oncología, patologías relacionadas con el hígado, neuropsicología y ginecología [45, 46, 47, 48]. Investigaciones aplicadas a enfermedades coronarias del corazón(CHD)[49, 50, 51] son aportes significativos, pues estas patologías constituyen un problema de salud bastante serio, que causa muchas muertes en todo el mundo. Las investigaciones demuestran la efectividad de las técnicas predictivas, cuando se tienen conjuntos de datos muy variados, siendo en muchas ocasiones comparados con variedades de algoritmos para ajustar la calidad de los resultados. Varios métodos de aprendizaje automático se han aplicado a temas relacionados al cáncer de mama, en este sentido se han construido modelos y por consiguiente se ha evaluado el rendimiento de los mismos. Estos métodos han atraído mucho la atención, con la esperanza de que puedan proporcionar resultados precisos, sin embargo, existen aspectos críticos sobre estas técnicas que exigen de procesos experimentales muy bien diseñados[52, 53, 54]. También, se ha utilizado técnicas de predicción en fertilizaciones invitro, su aplicación ha permitido identificar el número de embriones a ser implantados en el útero de la mujer, además de una selección con más alta viabilidad reproductiva[55, 56].

La educación, uno de los pilares fundamentales para el desarrollo de las naciones, a través de la inserción de profesionales a la sociedad con características responsables, comprometidas con la investigación, el medio ambiente y factores de crecimiento con emprendimiento. La inteligencia de datos no podría estar apartado de este sector, y su contribución ha sido notable tanto en el ámbito de la enseñanza como en los procesos de aprendizaje. Así se han creado de modelos capaces de predecir con alta precisión el rendimiento estudiantil, la deserción, tasas de retención, eficiencia terminal y otros indicadores esenciales para la toma de decisiones, tanto del docente como de las unidades de control [57, 58]. Aunque existen procedimientos estandarizados de minería de datos, hay sectores con características particulares que requieren mayor personalización en los procedimientos. Así, proveer de mecanismos basados en técnicas de inteligencia de datos, enfocados a resolver problemas de interés específico, también constituyen aportes significativos para este campo[59, 60].

Desde un enfoque de la mercadotecnia, el éxito de un producto o servicio es que éste resulte atractivo para los consumidores, pues además de la venta, que de hecho representan ingresos para las compañías, está la fidelización de los consumidores. La introducción de estrategias basadas en técnicas inteligentes son indispensables para alcanzar objetivos a largo plazo con los clientes. Un factor importante es la relación con el consumidor, lo que ha conducido a los negocios agreguen elementos analíticos a los modelos de gestión con el cliente CRM (Customer Relationship Management)[61, 62].
Los sistemas de recomendación se utilizan ampliamente en los sistemas de venta en línea para sugerir productos a los usuarios. Estas recomendaciones se generan utilizando en particular dos técnicas: filtrado basado en contenido y filtrado colaborativo. Nuevas propuestas combinadas y/o agregadas con los conceptos de la mercadotecnia construyen sistemas de recomendación en marketing, un sistema que sirve al marketing y utiliza técnicas y métodos de la economía digital [63, 64, 65].

La lista de aplicaciones de la minería de datos es grande, tan inmensa como los campos de estudio que existan en el universo. Las limitaciones estarían sujetas a la no disponibilidad de la materia prima (datos), lo que impediría crear experimentaciones basados en datos reales, aunque las simulaciones son alternativas contempladas y que pueden ser útiles en los análisis a priori. La importancia del estudio de cada componente, proceso o tarea de la minería de datos, cuyos resultados representen un aporte significativo constituye un impacto en todos los campos de estudios. El eje transversal de su aplicabilidad motiva el desarrollo de investigaciones afines, que incluyen cualquier aporte representativo de la ciencia de datos.

2.3. Minería de Reglas de Asociación

El problema de Minería de Reglas de Asociación (ARM por sus siglas en inglés) fue introducido en 1993 por [66], en cuya investigación se recogen los resultados más relevantes sobre asociaciones entre grupos de ítems, provenientes de transacciones de compras efectuadas en un supermercado durante un lapso de tiempo, además se presenta un algoritmo eficiente para éste propósito (Algoritmo Apriori).
El objetivo principal de ARM es la extracción de conjuntos frecuentes de ítems (frequent itemset), mediante correlación y asociación entre diferentes conjuntos de ítems en bases de datos transaccionales, relacionales u otros repositorios de información. El problema de ARM se puede dividir en dos

subproblemas, el primero de ellos es el descubrimiento de los elementos frecuentes, y el segundo consiste en la generación de reglas de asociación más significativas que cumplan con una métrica determinada.

Hoy en día, la acumulación de grandes cantidades de datos a causa de operaciones diarias es un factor común en las organizaciones. Por ejemplo, diariamente se recopilan enormes cantidades de datos referentes a la compra de los clientes, tanto de transacciones que provienen de tiendas físicas como de los sitios para ventas en línea. La Tabla 2.1 ilustra un ejemplo típico de compras, comúnmente conocidos como transacciones de la canasta básica. Cada fila de esta tabla corresponde a una transacción, que contiene un identificador único etiquetado como IDT, y un conjunto de artículos comprados por un cliente determinado. El personal de mercadeo y ventas son algunos interesados en analizar los datos para conocer el comportamiento de compra de los clientes. Esta valiosa información se puede utilizar para respaldar una variedad de aplicaciones relacionadas con el negocio, como promociones de mercadotecnia, gestión de inventarios y manejo de relaciones con los clientes.

Este apartado presenta una metodología conocida como análisis de asociación. Un campo útil para descubrir relaciones interesantes en grandes conjuntos de datos que pueden estar ocultos. Los elementos descubiertos se representan en forma de reglas de asociación o conjuntos de artículos frecuentes.

IDT	Artículos
1	{Pan, Leche}
2	{Pan, Pañales, Cerveza, Huevos}
3	{Leche,Pañales, Cerveza,Cola}
4	{Pan, Leche, Pañales, Cerveza}
5	{Pan, Leche, Pañales, Cola}

Tabla 2.1: Ejemplo de transacciones en comisariato

Por ejemplo la siguiente regla puede ser extraída de los datos mostrados en la Tabla 2.1

$$\{\text{Pañales}\} \rightarrow \{\text{Cerveza}\}$$

La regla sugiere que existe una fuerte relación entre la venta de pañales y cerveza, porque muchos clientes que compran pañales también compran cerveza. Se puede utilizar este tipo de reglas para ayudar a identificar nuevas oportunidades de venta cruzada de productos a los clientes.

Además del ejemplo sobre la canasta básica, también el análisis de asociación es aplicable a otros problemas, como la educación, bioinformática, el diagnóstico médico, la minería web y el análisis de datos científicos. En el análisis de los datos educativos, por ejemplo, los patrones de asociación pueden revelar conexiones interesantes entre los procesos de enseñanza, aprendizaje y recursos. Tal información puede ayudar a los directivos académicos a desarrollar una mejor comprensión, sobre cómo, los diferentes métodos de enseñanza-aprendizaje interactúan entre sí.
Aunque las técnicas presentadas aquí son generalmente aplicables a una variedad más amplia de conjuntos de datos, con fines ilustrativos, el análisis se centrará principalmente en los datos de la canasta básica. Hay dos cuestiones clave que deben abordarse al aplicar el análisis de asociación a los datos de la canasta básica. Primero, descubrir patrones a partir de un gran conjunto de datos de transacciones puede resultar computacionalmente costoso. En segundo lugar, algunos de los patrones descubiertos son potencialmente innecesarios, porque pueden ocurrir simplemente por casualidad.

El resto de esta sección está organizado en torno a estos dos temas. La primera parte está dedicada a explicar los conceptos básicos del análisis de asociación y los algoritmos utilizados para extraer eficientemente dichos patrones. La segunda parte trata el tema de la evaluación de los patrones descubiertos para evitar la generación de resultados innecesarios.

2.3.1. Definición del problema de ARM

Este apartado revisa la terminología básica utilizada en el análisis de asociación y presenta una descripción formal de la tarea.

Representación binaria, los datos de la canasta básica se pueden representar en un formato binario como se muestra en la Tabla 2.2, donde cada fila corresponde a una transacción y cada columna es un artículo. Un artículo puede tratarse como una variable binaria, cuyo valor es uno si el artículo está presente en una transacción y cero en caso contrario, tal determinación es debido a que la presencia de un artículo en una transacción a menudo se considera más importante que su ausencia, así un artículo es una variable binaria asimétrica.

IDT	Pan	Leche	Huevos	Panales	Cola	Cerveza
1	1	1	0	0	0	0
2	1	0	1	1	0	1
3	0	1	0	1	1	1
4	1	1	0	1	0	1
5	1	1	0	1	1	0

Tabla 2.2: Representación binaria del ejemplo transacciones de la canasta básica

itemset y soporte-contado, sea $I = \{I_1, I_2, I_3, \ldots, I_m\}$ un conjunto de m atributos binarios presentes en los datos que corresponden en este caso a la canasta básica, y $T = \{t_1, t_2, t_3, \ldots, t_n\}$ el conjunto de todas las transacciones. Cada transacción t_i contiene un subconjunto de artículos seleccionados desde I. En análisis de asociación, una colección de cero o más artículos es denominado un "itemset"(en minería de datos es una convención del término compuesto de la palabra en inglés "item-set"), si un itemset contiene k artículos, entonces es llamado un "k-itemset". Por ejemplo {Pan,Leche,Huevos} corresponde a *3-itemset*, asimismo se puede encontrar con itemset nulos o vacíos (aquellos que no tienen ningún ítem en el conjunto).
La amplitud de una transacción se define como el número de artículos presentes en la transacción. Así una transacción t_j se dice que contiene un itemset A, si A es subconjunto de t_j. Por ejemplo la tercera transacción mostrada en la Tabla 2.2 contiene el itemset {Cola,Cerveza}, pero no {Pan, Huevos}. Una propiedad interesante de los itemset es soporte-contado, este se refiere al número de transacciones que contienen un itemset específico. Matemáticamente el soporte-contado($\sigma(A)$) para un itemset A puede ser representado como se indica a continuación:

$$\sigma(A) =\mid \{t_i \mid A \subseteq t_i, t_i \in T\}$$

donde $\mid \cdot \mid$ denota el número de elementos en un conjunto. En el ejemplo de datos mostrado en la Tabla 2.2, el soporte-contado para {Leche,Pañales,Cerveza} es igual a dos, porque el itemset respectivo está presente solo en dos transacciones de las cinco.

Regla de Asociación, una regla de asociación es una implicación de la forma $A \rightarrow B$, donde $A \subset I$,$B \subset I$, $A \neq \emptyset$, $B \neq \emptyset$ y $A \cap B = \emptyset$. La regla $A \rightarrow B$, aparece en T con soporte *sop*, donde *sop* es el porcentaje de transacciones que contienen $A \cup B$ (es decir probabilidad $P(A \cup B)$); de igual forma la regla $A \rightarrow B$ tiene confianza $conf$ en el conjunto de transacciones

T, donde $conf$ es el porcentaje de transacciones en T conteniendo A y que también contienen B (se toma como la probabilidad condicional $P(B \mid A)$).

$$soporte, sop(A \rightarrow B) = \frac{\sigma(A \cup B)}{N} = P(A \cup B) \tag{2.1}$$

$$confianza, conf(A \rightarrow B) = \frac{\sigma(A \cup B)}{\sigma(A)} = P(B \mid A) \tag{2.2}$$

El ejemplo a continuación detalla el cálculo de estas dos primeras métricas, considerando los datos mostrados en la Tabla 2.2.
Dado la regla {Pan,Leche} $\rightarrow$ {Pañales}, el soporte contado para el itemset {Pan,Leche,Pañales} es igual a 2, mientras que el soporte de la regla, obtenido por la Ecuación 2.1 es 2/5=0.4. Asimismo la confianza de la regla se obtiene mediante la Ecuación 2.2, que divide el soporte-contado de {Pan,Leche,Pañales} para el soporte-contado de{Pan,Leche}, dando un resultado 2/3=0.67.

Aspectos sobre las métricas de soporte y confianza, soporte es una medida importante porque una regla que tiene un soporte muy bajo puede ocurrir simplemente por casualidad. También es probable que una regla con soporte bajo, no sea interesante desde una perspectiva de decisiones estratégicas, porque puede ser indeseable promocionar artículos que los clientes rara vez compran juntos. Por estas razones, el soporte se utiliza a menudo para eliminar reglas poco interesantes. Sin embargo, los criterios de importancia de esta métrica pueden verse afectados cuando las reglas de asociación son cuantitativas.
La confianza, por otro lado, mide la confiabilidad de la inferencia que es probable que B esté presente en transacciones que contienen A.
La confianza también se obtiene a partir de una regla, así dado $A \rightarrow B$ que define a una regla, cuanto mayor es el valor alcanzado, más proporciona una estimación de la probabilidad condicional de B dado A. Los resultados del análisis de asociación deben interpretarse con precaución. La inferencia hecha por una regla de asociación no implica necesariamente causalidad. En cambio, sugiere una fuerte relación de co-ocurrencia entre elementos en el antecedente y consecuente de la regla. La causalidad, por otro lado, requiere conocimiento sobre los atributos causales y de efecto en los datos, y típicamente involucra relaciones que ocurren a lo largo del tiempo (por ejemplo, el agotamiento del ozono conduce al calentamiento global).

Problema de ARM en bases de datos transaccionales, dado un conjunto de transacciones T encontrar todas las reglas cuyo $soporte \geq minsop$

y $confianza \geq minconf$, donde $minsop$ y $minconf$ son umbrales definidos según el interés tanto para el soporte como para la confianza. Suponiendo que se considera un procedimiento por fuerza bruta, este involucraría el cómputo del soporte y confianza para cada regla posible, lo cual daría lugar a un procedimiento demasiado costoso, debido a una explosión demasiada amplia de posibles reglas a partir de un conjunto de datos. Concretamente, el número total de posibles reglas extraídas desde un conjunto de datos que contiene d ítems es mostrado en la Ecuación 2.3, los detalles sobre su derivación están en [68].

$$R = 3^d - 2^{d+1} + 1 \tag{2.3}$$

Considere los datos especificados en la Tabla 2.1 para ilustrar mediante un cálculo real. En este caso se tiene que $d = 6$, por tanto $3^d - 6^7 + 1 = 602$ reglas. La introducción de umbrales mínimos para el soporte y la confianza, consigue reducciones significativas en el número de reglas a mostrar. Además se puede introducir estrategias que permitan evitar la exploración de ítemsets compuestos por los mismos ítems, cuando se calcula el soporte. Es así que varias de las técnicas para extracción de reglas de asociación trabajan eficientemente.

Algoritmos tradicionales por lo general dividen al problema en dos subtareas.

1. Generación de itemsets frecuentes, consiste en la búsqueda de todos los itemsets frecuentes que cumplan con el umbral de soporte mínimo establecido.

2. Generación de reglas, consiste en la estructuración de la regla y la selección de todas aquellas que cumplan con el umbral de confianza mínimo establecido.

A continuación se describen cada una de las etapas indicadas.

Generación de itemsets frecuentes

Los recursos computacionales consumidos en esta etapa se puede decir que son más altos que los requeridos para la extracción de reglas. El problema de la exploración de itemsets frecuentes, se reduce a la selección de aquellos que cumplan con la condición de soporte mínimo. Un subconjunto de un itemset frecuente, también debe ser frecuente, es decir si {Pan,Leche} es un itemset frecuente, entonces {Pan} y {Leche} también son itemsets frecuentes. El proceso puede extenderse desde una cardinalidad de 1 hasta k (k-itemsets).

La reducción del número de candidatos es una de las tareas relevantes en esta etapa, así el principio apriori determina que si un itemset es frecuente, entonces todos sus subconjuntos deberían ser frecuentes. El principio apriori se mantiene debido a la siguiente propiedad de la métrica de soporte.

$$\bigvee A, B : (A \subseteq B) \rightarrow sop(A) \geq sop(B)$$

el soporte de un itemset nunca va a exceder el soporte de alguno de sus subconjuntos, esta es conocida como la propiedad de anti-monotonía del soporte.

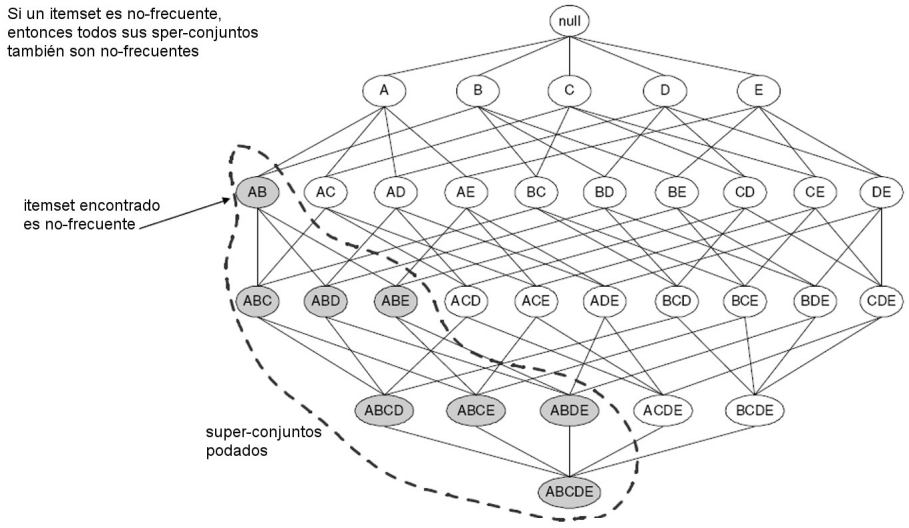

Figura 2.6: Representación de la generación de itemsets

En la Figura 2.6, si el itemset de primer nivel (AB) es encontrado no-frecuente, entonces todos los itemset potenciales que contienen (AB) pueden ser ignorados (32 → 24).

A continuación se muestra el pseudo-código del proceso de generación de itemsets frecuentes en el algoritmo *apriori*. C_k denota el conjunto de candidatos de k-itemset y L_k el conjunto de k-itemset frecuentes:

- El algoritmo comienza determinando el soporte de cada ítem del conjunto de datos, de esta forma se inicializa L_1 con todos los 1-itemset (Líneas 2 y 3).

- A continuación genera iterativamente nuevos candidatos k-itemset usando los (k-1)-itemset encontrados en la iteración previa (Línea 4).

- Para contar el soporte de los candidatos, el algoritmo necesita realizar una pasada adicional sobre el conjunto de datos (Líneas 7 a 8). La función de subconjunto se utiliza para determinar todos los conjuntos de elementos candidatos en C_k que están contenidos en cada transacción t.

- Después de contar sus soportes, el algoritmo elimina todos los conjuntos de elementos candidatos cuyo recuento de soporte es inferior a *minsop* (Línea 9).

- Finalmente el algoritmo finaliza, cuando ya no hay nuevos itemset frecuentes generados.

```
1 apriori_itemsetFrecuentes(datos){
2      L_k: itemsets frecuentes de tamaño k
3      L_1=items frecuentes con k=1
4      C_k: itemsets candidatos de tamaño k
5      for(k = 1; L_k! = φ; k++) do begin
6          C_{k+1}= candidatos generados desde L_k
7          for each (transaction t in datos) do
8           incrementa la cuenta de C_{k+1} contenidos en t
9           L_{k+1}=candidatos en C_{k+1} con minsop
10         end
11      return ∪_k L_k }
```

El procedimiento que corresponde con la generación frecuente de conjuntos de elementos del algoritmo *apriori* tiene dos características importantes. Primero, es un algoritmo de nivel; es decir, atraviesa la malla del conjunto de elementos un nivel a la vez, desde los 1-itemset frecuentes hasta el tamaño máximo de itemsets frecuentes. En segundo lugar, emplea una estrategia de generación y prueba para encontrar itemsets frecuentes. En cada iteración, se generan nuevos itemsets candidatos a partir de los itemsets frecuentes encontrados en la iteración anterior, tal como se describe en la Figura 2.7. El soporte para cada candidato es contabilizado y se compara con el umbral de *minsop*. El número total de iteraciones que necesita el algoritmo es $k_{max}+1$, donde k_{max} es el tamaño máximo de los itemsets frecuentes.

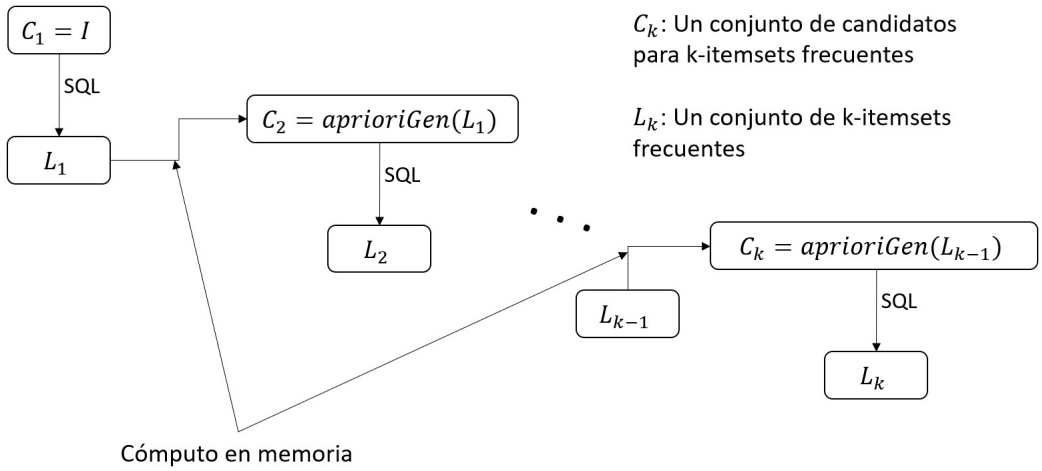

Figura 2.7: Idea básica del algoritmo apriori

En la Figura 2.8 se ilustra un ejemplo basado en el pequeño conjunto de datos que se ha venido referenciando, nótese el tamaño del conjunto cuando no se considera la poda, existe un número significativo de itemsets que son irrelevantes de acuerdo al soporte mínimo definido para el ejemplo.

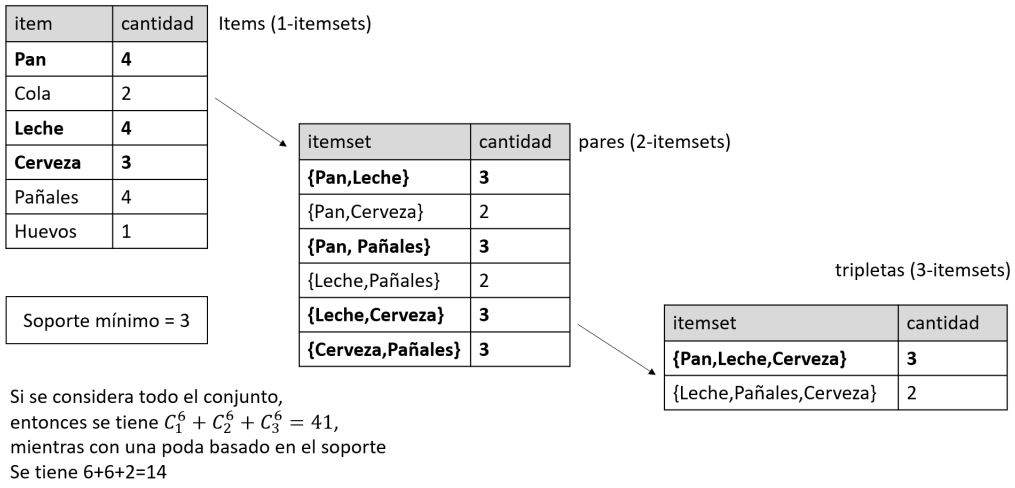

Figura 2.8: Ilustración de la generación de candidatos itemsets

Generación de reglas

A continuación se describe como extraer eficientemente reglas dado un itemset frecuente L, donde $l \subseteq L$, tal que $l \rightarrow L - l$ satisface los requerimien-

tos de confianza mínimos. Es decir si $\{A, B, C, D\}$ es un itemset frecuente, entonces las reglas candidatas son las siguientes:

$$\{A, B, C\} \rightarrow \{D\} \quad \{A, B, D\} \rightarrow \{C\} \quad \{A, C, D\} \rightarrow \{B\} \quad \{B, C, D\} \rightarrow \{A\}$$

$$\{A, B\} \rightarrow \{C, D\} \quad \{A, C\} \rightarrow \{B, D\} \quad \{A, D\} \rightarrow \{B, C\} \quad \{B, C\} \rightarrow \{A, D\}$$

$$\{A, B, C\} \rightarrow \{D\} \quad \{A, B, D\} \rightarrow \{C\} \quad \{A, C, D\} \rightarrow \{B\} \quad \{B, C, D\} \rightarrow \{A\}$$

$$\{B, D\} \rightarrow \{A, C\} \quad \{C, D\} \rightarrow \{A, B\}$$

Si $\mid L \mid = k$ entonces hay $2^k - 2$ reglas de asociación candidatas (ignorando los conjuntos vacíos). Para alcanzar una generación de reglas de manera eficiente, considerando que la confianza como tal no tiene la propiedad de anti-monotonía, sin embargo, la confianza de las reglas generadas a partir de un mismo itemset, si tienen esta propiedad, pero la confianza no aumenta a medida que el número de ítems de la regla crecen, así que:
$L = \{A, B, C, D\}$:
$conf(\{A, B, C\} \rightarrow \{D\}) \geq conf(\{A, B\} \rightarrow \{C, D\}) \geq conf(\{A\} \rightarrow \{B, C, D\})$
La Figura 2.9 representa una vista de la poda sobre un ejemplo de regla con baja confianza, nótese que es posible la reducción varias reglas derivadas en la malla marcada.

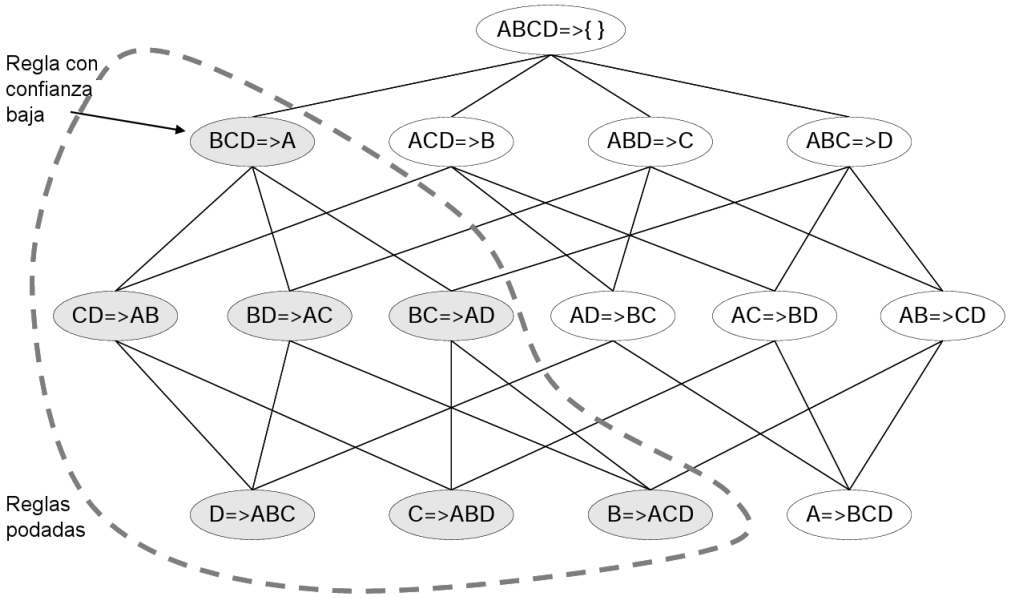

Figura 2.9: Ilustración de la generación de reglas por apriori

Una regla candidata es generado por la combinación de dos reglas que comparten el mismo prefijo en el consecuente de la regla.
Así la combinación entre $\{C, D\} \rightarrow \{A, B\}, \{B, D\} \rightarrow \{A, C\}$ podría dar lugar a la generación de la regla candidata $\{D\} \rightarrow \{A, B, C\}$. Asimismo se puede podar la regla $\{D\} \rightarrow \{A, B, C\}$, si allí existiera un subconjunto (ejemplo $\{A, D\} \rightarrow \{B, C\}$) que no tenga la medida de confianza alta.

Con el algoritmo Apriori la base de datos debe analizarse para todos los niveles. Si hay m conjuntos de elementos frecuentes, entonces necesita escanear la base de datos m veces. Se generan demasiados candidatos que necesitan ser probados. Existen extensiones que se han aplicado al clásico Apriori, así por ejemplo:

- Técnica hashing (DHP) [69], introduce la aplicación de un algoritmo basado en "hash"para la generación de los itemsets candidatos, la reducción resulta significativa a partir de 2-itemsets, de esta manera afronta el problema de los cuellos de botella presentes en el método previo.

- Contador de itemsets dinámico (DIC) [70], este cuenta dinámicamente los itemsets candidatos a medida que el algoritmo progresa y reduce los candidatos generados. En otras palabras, el algoritmo DIC no espera a que se complete el análisis de la base de datos para crear los candidatos, sino que analiza los subconjuntos de un itemset y determina si todos son frecuentes. Cuando DIC determina que todos los subconjuntos de un itemset son o se estiman que son frecuentes, agrega el itemset a la lista de candidatos y comienza a contar el soporte para este. El proceso reduce la cantidad de escaneo a la base de datos.

- Partición[71], el algoritmo de particionamiento escanea la base de datos solo dos veces, por lo que reduce sustancialmente el acceso a la base de datos. En el primer escaneo, divide toda la base de datos en secciones verticales más pequeñas que son lo suficientemente pequeñas como para caber en la memoria. Luego continúa con las operaciones de búsqueda de itemsets frecuentes.

- Muestreo, este algoritmo encuentra los itemsets frecuentes de muestras seleccionadas al azar. Verifica estos resultados en toda la base de datos y crea las reglas de asociación completas para estos itemsets frecuentes verificados, sin embargo, existe la probabilidad de que algunos de los itemsets frecuentes no aparezcan en las muestras seleccionadas; por lo tanto, nunca se encuentra que sean frecuentes y no se crearían reglas de asociación para esos itemsets. Para minimizar este problema, una

alternativa es utilizar un umbral de soporte muy bajo, pero esto crearía muchos candidatos como penalización [72].

AprioriTID tiene la misma función de generación de candidatos que Apriori. La característica interesante es que no usa la base de datos para contar el soporte después de la primera pasada. Se utiliza una codificación de los conjuntos de elementos candidatos utilizados en la pasada anterior. En pasadas posteriores, el tamaño de la codificación puede llegar a ser mucho más pequeño que la base de datos, lo que ahorra esfuerzo de lectura. El algoritmo Apriori funciona mejor que AprioriTID en los pasos iniciales, pero en los últimos pasos AprioriTID tiene un mejor rendimiento que Apriori. Por esta razón, se puede usar otro algoritmo llamado Apriori Hybrid en el que se usa Apriori en los pasos iniciales pero se cambia a AprioriTid en los últimos pasos [14].

2.3.2. Minería de Reglas de Asociación, tipos

Tradicionalmente, los algoritmos de minería de reglas de asociación apuntan a la extracción de patrones frecuentes (itemsets), es decir, patrones que se caracterizan por ser de alta frecuencia en una base de datos transaccional. Sin embargo, estos algoritmos ignoran muchos itemsets importantes, con poco soporte (es decir, poco frecuentes). Estos conjuntos de elementos poco frecuentes, a pesar de su bajo soporte, pueden producir reglas de asociación negativas (NAR) potencialmente importantes con medidas de confianza altas, que no son observables entre elementos de datos frecuentes[73, 74, 75]. Por lo tanto, el descubrimiento de posibles reglas de asociación negativas son importantes para construir un sistema de apoyo a las decisiones confiable. Así un primer enfoque puede ser visto como:

- Extracción de reglas de asociación positivas y negativas desde los conjuntos de datos
- Extracción de reglas de asociación negativas desde los itemsets frecuentes
- Extracción de reglas de asociación positivas desde itemsets no frecuentes

Según el número de dimensiones de datos involucradas en la regla, podemos distinguir dos tipos de dimensiones de reglas de asociación:

- **Regla de asociación unidimensional**, una regla de asociación es unidimensional, si los elementos o atributos de una regla de asociación

hacen referencia solo a una dimensión. Por ejemplo, una regla unidimensional podría reescribirse de la siguiente manera

$$\{producto = \text{“}Pan''\} \rightarrow \{producto = \text{“}Leche''\}$$

- **Regla de asociación multidimensional**, si una regla hace referencia a más de una dimensión, como las dimensiones edad, ocupación y compra, se trata de una regla de asociación multidimensional. Así, la siguiente regla es un ejemplo de regla multidimensional:

$$\{edad \in [20, 25], ocupacion = \text{“}estudiante''\} \rightarrow \{compra = \text{“}Cerveza''\}$$

Con base en los tipos de valores contenidos en la regla, se puede distinguir dos tipos de reglas de asociación:

- **Regla de asociación booleana**: se define como regla de asociación booleana, si implica asociaciones entre la presencia o ausencia de elementos. Por ejemplo, la siguiente es una regla de asociación booleana, relacionado al análisis de la canasta de compras:

$$\{compra(producto = \text{“}Pan'')\} \rightarrow \{compra(producto = \text{“}Leche'')\}$$

- **Regla de asociación cuantitativa**: se define como regla de asociación cuantitativa, si describe asociaciones entre elementos o atributos cuantitativos. En estas reglas, los valores cuantitativos de los elementos o atributos se dividen en intervalos. Por ejemplo, la siguiente regla es una regla de asociación cuantitativa:

$$\{edad \in [20, 25], ingresos \in [1500, 2500]\} \rightarrow \{compra = \text{“}Auto''\}$$

Según los tipos de reglas que se van a extraer, se puede distinguir las reglas de correlación definidas de la siguiente manera: En general se puede generar una gran cantidad de reglas, muchas de las cuales son redundantes o no indican una relación de correlación entre los itemsets. En consecuencia, las asociaciones descubiertas se pueden analizar más a fondo para descubrir correlaciones estadísticas, lo que conduce a reglas de correlación.

2.3.3. Minería de Reglas de Asociación Cuantitativas

La minería de reglas de asociación numéricas (QARM por sus siglas en ingles)[11], es una subdivisión de esta técnica, orientado al conjuntos de datos con atributos numéricos.

Hoy en día es muy común encontrarse con variables numéricas, ya sean estas, continuas o discretas; así mismo, no es motivo de preocupación para un análisis, pues en los últimos años se han venido desarrollando variedades de herramientas y técnicas, que procesan eficientemente este tipo de atributos, y a la vez han reflejado que no siempre, los métodos utilizados en variables categóricas, son los más adecuados para este tipo de datos.

Las técnicas tradicionales tienden a resultados poco interesantes, con altos costos computacionales[12]; esto lo convierte en un campo abierto para desarrollar y derivar líneas de investigación prometedoras. Adicional a esto, los problemas relacionados a minar reglas de asociación, que combinen tipos de variables cuantitativas y cualitativas han sido inclusive menos estudiados.

En general esta problemática tiende a presentar diversos inconvenientes, primero, porque los atributos numéricos están definidos en un amplio rango de valores, y un valor determinado no se presenta con una frecuencia significativa, lo que hace difícil encontrar un patrón, y por otra parte, requiere mas atención cuando se tiene una combinación entre categórico y numérico.

Adhikary y Roy [11] definen QARM como una forma de minería de reglas de asociación, donde su aplicación está enfocado a datos que contienen tanto atributos categóricos como numéricos; y en su estudio acerca de las tendencias sobre QARM hace una clasificación, junto con un análisis crítico de los métodos y técnicas computacionales empleados. En la figura 2.10 se ilustra la clasificación establecida, cada método tiene su técnica y características muy bien definidas, las ventajas y desventajas que presentan en escenarios diferentes.

Una de las formas clásicas de tratar el problema es la discretización, que consiste en dividir el dominio de la variable en intervalos; este procedimiento ha sido evaluado en varios trabajos [76, 77, 78], y principalmente destacan la alta sensibilidad en la estimación del soporte y la confianza, cuando la amplitud de los intervalos son demasiados pequeños o extremadamente grandes.

También existen investigaciones que utilizan algunas métricas estadísticas en la caracterización de los atributos numéricos. Asimismo están por otra parte las técnicas de agrupamiento que emplean métodos no supervisados y supervisados para identificar zonas densas en la distribución. Adicional se encuentran los métodos difusos, muy útiles cuando no existe precisión en la definición de un intervalo, y finalmente mediante métodos de optimización.

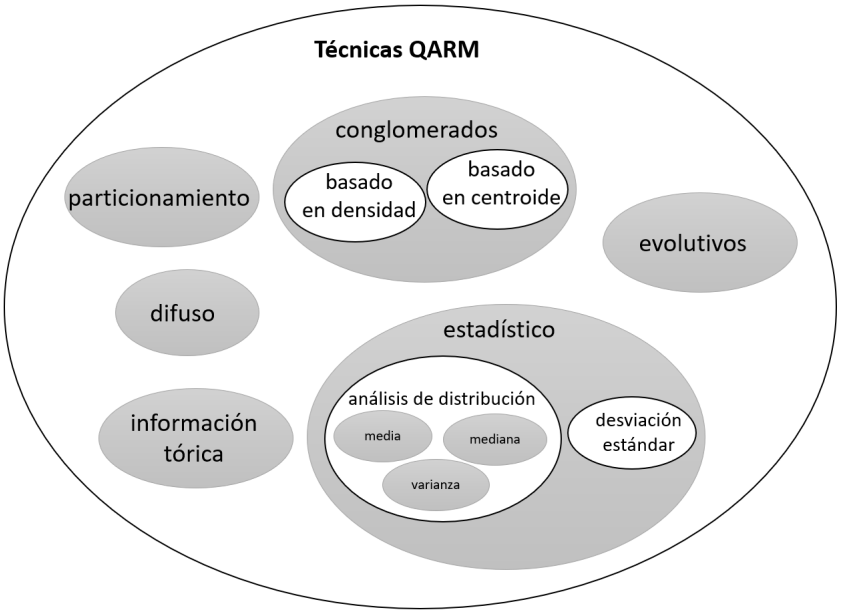

Figura 2.10: Clasificación de técnicas QARM

Estos últimos son basados en la adaptación de algoritmos evolutivos para tratamiento de altos volúmenes y variedades de datos con velocidades moderadas.

Es importante diferenciar, entre minería de reglas de asociación y minería de reglas de clasificación (CARs), sean estas cualitativas o cuantitativas [13]. Primero que todo, las dos técnicas son muy útiles en minería de datos, pero aunque provengan del mismo concepto general son distintos en el objetivo para el cual se aplican. En el primer caso, por lo general se enfatiza en la búsqueda de reglas que cumplan criterios de soporte y confianza mínimos entre la combinación de todos los atributos, mientras que en CARs, el objetivo se centra en hallar reglas que concluyan en un atributo de clase predeterminado.

Desde la aparición del enfoque de minería de reglas de asociación cuantitativas, muchas técnicas han sido estudiadas por numerosos investigadores para dar solución a problemas derivados de los conjuntos de datos numéricos. ARM desde su origen se planteó como un problema de tipo combinatorio, en donde intervienen elementos discretos. Sin embargo, a medida que su importancia y utilidad fueron creciendo, éste se ha ido adaptando a conjuntos de datos diversos. Es así, que se han aplicado varias técnicas de procesamiento, que unidos a las capacidades de cómputo actuales, se obtienen algoritmos

bastante competitivos.

En general, lo que dificulta el tratamiento de un atributo cuyo dominio es muy amplio, como el caso de una variable numérica, es la necesidad de convertir estos datos a un tipo manejable sin alterar su información. Pero, el problema está en encontrar los intervalos más óptimos para la formulación de reglas de calidad; de esta forma tenemos varias técnicas, tales como: *particionamiento, agrupamiento, estadísticos, técnicas difusas y algoritmo evolutivos*; a continuacón se describen algunos trabajos importantes de cada uno.

Discretización de las variables numéricas

La discretización de la variable numérica es el primer método aplicado por Srikant R. et al. [76], para tratar este problema, en sus primeros trabajos sobre ARM, ya incluyen una técnica para lidiar con variables numéricas. Ellos dividen el dominio del atributo numérico en intervalos, y para encontrar un número de particiones apropiado, primero hacen varios test con las propiedades de los atributos, y en base a esas propiedades se decide el número de intervalos. Esta técnica trae consigo el problema de la sensibilidad del soporte y confianza , debido a que la presencia de muchos intervalos ocasiona valores de soporte bajos para cada intervalo, dando lugar a la generación de muy pocas reglas. Por otra parte, si el tamaño individual de los intervalos se incrementa, la perdida de información persiste en términos de confianza.

Chan K., et al.[79] hacen una contribución importante en esta área. En APACS2 (nombre de la técnica) además de evitar la intervención del usuario en la definición de los umbrales de soporte y confianza. Se emplea un método denominado diferencia ajustada, para la identificación de asociaciones tanto positivas y negativas entre atributos. Experimentaciones realizadas han demostrado que APACS2 es capaz de obtener reglas de asociación cuantitativas interesantes y significativas a diferencia de otros algoritmos de la época.

Song C. et al. [80] proponen un nuevo algoritmo denominado *divide y conquistaras* junto con una técnica de optimización que mejora drásticamente la velocidad y mantienen las reglas. En su estudio experimental, el cual tiene un enfoque comparativo con la versión genética de QUANTMINER [81], ellos demuestran que su técnica es uno a dos veces en magnitud más rápido, descubriendo de manera significativa más reglas que son útiles para varias tareas de predicción. Sin embargo, tienen limitaciones con respecto a variables de tipo categóricas

La selección adecuada de un discretizador depende de su aplicación post-discretización. Para el propósito de clasificación, el discretizador supervisado puede ser una buena opción, pero el efecto a lo mejor no es satisfactorio para otros tipos de tareas como las de simplemente buscar asociaciones. Con este antecedente el enfoque propuesto por Adhikary D., et al.[82]consiste en una técnica de discretización no supervisada, usando un enfoque que no requiere de umbrales para la generación de los intervalos. Éste utiliza la desviación estándar como instrumento de medida para el análisis y establecimiento del método. Experimentaciones realizadas demuestran que la técnica es mejor que los discretizadores existentes según varias métricas de rendimiento, además supera varias técnicas comúnmente utilizadas en lo que respecta a generación de intervalos razonables.

Método estadístico

El análisis estadístico de las variables, es una estrategia que también se ha empleado para encontrar reglas de asociación cuantitativas. En ésta, el consecuente de una regla puede obedecer a una distribución de las variables numéricas del antecedente. Aumann y Lindell [83] introducen una nueva definición de reglas de asociación cuantitativas, basado en la teoría de inferencia estadística. Su concepción refleja la intuición de que el objetivo de las reglas de asociación es encontrar fenómenos extraordinarios, y por lo tanto, interesantes en las bases de datos. Luego de ejecutar varias experimentaciones, con la introducción de distribuciones significativas al perfil del atributo, en dos casos específicamente se obtuvo resultados interesantes en reglas, sin embargo, presentan la dificultad para tratar con reglas de mas de una dimensión y que combinen atributos categóricos.

Otro enfoque considera la extracción de reglas de asociación cuantitativas aplicando concepto de la conversión de atributos a tipo binario. Kang G., et al. [84] describe que básicamente su método se reduce a dos pasos. Primero, una etapa de preprocesado en que atributos numéricos son convertidos a tipo booleano, y un segundo paso de postprocesamiento que reconvierte las reglas de asociación binarias en cuantitativas. Basándose en el concepto de partición del dominio, proponen tres técnicas de particionamiento binario, (bi-partición basado en media, bi-partición basado en la mediana y minimización de la desviación estandar), quedando éste último por encima de los otros métodos.

Ténica de agrupamiento

Agrupamiento es ampliamente utilizado para implementar algoritmos eficientes QARM. Chien B.-C. et al. [85] proponen un algoritmo eficiente de agrupamiento jerárquico, basado en variación de la densidad para resolver el problema de particionamiento de intervalos. Su principal contribución es proveer de un método automático, para ello implementan DRminer y DBS-Miner. Estos dos algoritmos utilizan el concepto de densidad para asociar las características de atributos, y para la exploración de zonas densas. DBSminer es una consecuencia de DRMInerm e incluyen exploración en conjuntos de datos de alta dimensionalidad, y su exploración se reduce a la conexión con sus vecindades, evitando pasar por todo el espacio poblacional.

MQAR(Mining Quantitative Association Rules based on dense grid) es un algoritmo que usa una estructura tipo árbol DGFP-tree para el agrupamiento de subespacios densos[86]. Este resuelve algunos problemas relacionados a subespacios densos, además del conflicto existente entre parámetros (soporte mínimo y confianza mńimo).

Método difuso

Existen situaciones en las que valores de atributos no son expresados con precisión, sino mas bien obedecen a un margen de definición. Yang J., et al. [87] utiliza la técnica de agrupación de FCM (Fuzzy C-means clustering) para asignar conjuntos de datos cuantitativos a conjuntos de datos difusos. Utiliza métodos de búsqueda bidireccional, de alta a baja y de baja a alta dimensión cuando busca conjuntos difusos de elementos frecuentes. De esta forma se logra reducir el tiempo de búsqueda y mejorar la eficiencia de la minería de datos.

Optimización de intervalos

La aplicación de métodos evolutivos no es una excepción en el tratamiento de esta problemática. Así pues, la búsqueda de intervalos óptimos lo convierte en un problema de optimización, por consiguiente, varios son los algoritmos que han adaptado sus estrategias hacia un problema de optimización, basado en reglas de asociación cuantitativas.

Los algoritmos genéticos han sido los primeros métodos evolutivos aplicados en esta problemática. En el 2002 Mata J. et al [270][271] usaron un algoritmo evolutivo, para encontrar los intervalos de cada atributo que con-

forma un itemset frecuente. La función de evaluación utilizada calcula la amplitud de estos intervalos. Así, ellos demostraron obtener resultados satisfactorios en itemsets frecuentes que no tienen solapamiento.

QUANTMINER [81] consigue reglas altas en confianza. Su procedimiento comienza con un conjunto de plantillas de regla, y entonces busca dinámicamente, los mejores intervalos para los atributos numéricos presentes en la plantilla. Este algoritmo requiere cierta iteración con el usuario para construir el esquema de experimentación, de esta forma evita la extracción de cientos de reglas innecesarias. Existen ciertas desventajas con respecto a la orientación a intervalos pequeños, exponiéndose a la pérdida de reglas significativas en intervalos de mayor amplitud.

Un algoritmo recientemente desarrollado ha sido aplicado a QARM. Afshari M., et al. [88] introduce una nueva meta-heurística eficiente que usa el algoritmo de optimización *cuckoo* para el ocultamiento de reglas de asociación consideradas sensibles. En este estudio, también se han introducido tres funciones que pueden encontrar la solución con mínimos efectos secundarios. Además, se ha definido un operador de inmigración que mejora la capacidad del algoritmo propuesto para escapar de los óptimos locales. Pruebas experimentales en conjuntos de datos reales y sintéticos han dado resultados satisfactorios en criterios *Hiding failure(HF),Lost Rules (LR), Ghost Rules (GR), Number of Iterations (NI).*

La introducción de metaheurísticas basadas en población en los últimos años para resolver estos problemas han dado resultados satisfactorios, pues se ha conseguido, reducir los costos computacionales asociados a los algoritmos genéticos, debido a su rápida convergencia hacia las soluciones. Beiranvand V., et al.[89] desarrollan un algoritmo denominado MOPAR, basado en PSO (Particular Swarm Optimization) para hallar reglas de asociación numéricas usando un enfoque multiobjetivo. Los resultados mostraron que MOPAR extrae reglas de asociación numéricas confiables (con valores de confianza cercanos al 95 %), comprensibles e interesantes.

2.3.4. Métricas de Reglas de Asociación

Medir es importante para evaluar la calidad de una técnica o método, en este caso es importante estudiar las medidas aplicadas a la calidad de reglas de asociación en general, varias investigaciones se han enfocado al respecto y existen diversas métricas que poner en consideración.

La extracción de reglas de asociación valiosas, y previamente desconocidas en una base de datos puede producir una gran cantidad de soluciones diferentes, dando lugar a un difícil proceso de análisis de cada regla. Además, un gran porcentaje de este conjunto de soluciones puede resultar poco interesante e inútil. Siendo responsabilidad del usuario, la selección mediante un proceso de cuantificación de las reglas de asociación. Para resolver este problema, diferentes autores han propuesto varias medidas de calidad (ver 2.3) basadas en el análisis de las propiedades estadísticas de los datos.

Nombre	Ecuacion	rango factible
Support	p_{ab}	$[0,1]$
Coverage	p_a	$[0,1]$
Prevalence	p_b	$[0,1]$
Confidence	$\frac{p_{ab}}{p_a}$	$[0,1]$
Lift	$\frac{p_{ab}}{p_a \times p_b}$	$[0,1]$
Cosine	$\frac{p_{ab}}{\sqrt{p_a \times p_b}}$	$[0,1]$
Leverage	$p_{ab} - (p_a \times p_b)$	$[-0{,}25, 0{,}25]$
Conviction	$\frac{p_a \times p_{\bar{b}}}{p_{a,\bar{b}}}$	$[\frac{1}{n}, \frac{n}{4}]$
Gain	$\frac{p_{ab}}{p_a} - p_b$	$[-1, 1 - \frac{1}{n}]$
Certainty Factor(CF)	$\begin{cases} \frac{\frac{p_{ab}}{p_a} - p_b}{p_{\bar{b}}} & \text{if } (\frac{p_{ab}}{p_a} - p_b) \geq 0 \\ \frac{\frac{p_{ab}}{p_a} - p_b}{p_b} & \text{otro caso} \end{cases}$	$[-1, 1]$
Recall	$\frac{p_{ab}}{p_a}$	$[0,1]$
Laplace	$\frac{p_{ab} \times n+1}{p_a \times n+2}$	$[\frac{1}{n_a+2}, \frac{n_a+1}{n_a+2}]$
NetConf	$\frac{p_{ab} - (p_a \times p_b)}{p_a \times (1-p_a)}$	$[-1, 1]$
Yule'sQ	$\frac{p_{ab} \times p_{\overline{ab}} - p_{a\bar{b}} \times p_{\bar{a}b}}{p_{ab} \times p_{\overline{ab}} + p_{a\bar{b}} \times p_{\bar{a}b}}$	$[-1, 1]$

Tabla 2.3: Métricas de calidad para reglas de asociación

El principal concepto en la minería de reglas de asociación es el "Patrón" P, que es definido como un subconjunto de todo el conjunto de ítems $I = \{I_1, I_2, I_3, \ldots, I_m\}$ en un conjunto de datos D, ejemplo $P = \{I_j, \ldots, I_k\} \subseteq I, 1 \leq j, k \leq m$. Una regla de asociación es una implicación de la forma $A \rightarrow B$ que es formado desde P, de tal forma que el antecedente A es definido como $A \subset P$, mientras que el consecuente se denota como $B = P/A$. Cualquier medida de calidad en este tema esta basado en el número de transacciones del conjunto de datos satisfecho por la regla (denotado como n_{ab}), el antecedente (n_a) y el consecuente (n_b). Todos estos valores pueden también ser representados como frecuencias relativas, considerando el número n de transacciones en el conjunto de datos.

$p_{ab} = n_{ab}/n; p_{a\bar{b}} = n_{a\bar{b}}/n; p_{\bar{a}b} = n_{\bar{a}b}/n; p_{\overline{ab}} = n_{\overline{ab}}/n$

$p_a = p_{ab} + p_{a\bar{b}} = 1 - p_a; p_b = p_{ab} + p_{\bar{a}b} = 1 - p_{\bar{b}}$

$p_{\bar{a}} = p_{\bar{a}b} + p_{\overline{ab}} = 1 - p_a;$ b $p_{\bar{b}} = p_{a\bar{b}} + p_{\overline{ab}} = 1 - p_b$

Support(Soporte) [90]

Es una de las medidas de calidad más conocidas en éste ámbito, en secciones anteriores ya se hizo algunas definiciones y referencias con respecto a la métrica.
Se define como la frecuencia relativa de ocurrencia de una regla de asociación $A \rightarrow B$, así $soporte(A \rightarrow B) \equiv p_{ab}$. Una característica relevante de esta métrica es la *simetría*, pues está dado que $sop(A \rightarrow B) = sop(B \rightarrow A)$. Los valores mínimo y máximo para esta medida de calidad son 0 y 1, respectivamente. Sin embargo, una regla de asociación se considera dudosa si se obtiene uno de estos dos valores. Así, en situaciones en las que $p_{ab} = 1$, la regla puede considerarse irrelevante, debido a que aparece en cualquier transacción, por lo que no proporcionaría ningún conocimiento nuevo sobre las propiedades de los datos. Por el contrario, si $p_{ab} = 0$, entonces la regla no representa ninguna transacción, por lo que se considera engañosa.
Adicional a esto, otra propiedad reconocible del soporte de la regla es que su valor siempre es mayor a cero, cuando se cumple $p_a + p_b > 1$, y el valor máximo esta siempre dado por el valor mínimo de los valores p_a y p_b, asi $p_{ab} \leq Min(p_a, p_b)$.

Coverage y Prevalence [91]

Las medidas de calidad se definen como el soporte en base al antecedente o respecto del consecuente, respectivamente. "Coverage", definido como p_a, determina el porcentaje de transacciones en las que aparece el antecedente A. Por el contrario, la medida de calidad de "Prevalence", es decir, p_b, determina el porcentaje de transacciones donde aparece el consecuente B. Es de destacar que, de manera similar al soporte tradicional, estas dos medidas de calidad operan en el rango [0,1].

Confidence(Confianza) [90]

Es una medida de calidad que aparece en la mayoría de los problemas de minería de reglas de asociación. Esta medida de la calidad se define como el porcentaje de transacciones en un conjunto de datos que contiene A y, al mismo tiempo, también B.
De manera formal, esta medida de calidad se puede expresar como $conf(A \rightarrow B) = p_{ab}/p_b$, o como una estimación de la probabilidad condicional $P(B \mid A)$, su rango de operación está entre [0,1] y no es simétrico ($conf(A \rightarrow B) \neq conf(B \rightarrow A)$),sin embargo, es posible alguna simetría cuando $p_a = p_b$ independientemente del valor p_{ab}.

Lift [92]

También denominado "interés", es una de las muchas medidas alternativas de calidad propuestas por diferentes autores. Esta medida de calidad calcula la relación entre la Confianza de la regla y su "Prevalence", como se muestra en la Tabla 2.3.
Se describe como $Lift(A \rightarrow B) = p_{ab}/(p_a \times p_b)$, o también $Lift(A \rightarrow B) = conf(A \rightarrow B)/p_b$. Esta medida calcula una relación entre la probabilidad conjunta de dos variables de observación (antecedente A y consecuente B) con respecto a sus probabilidades bajo el supuesto de independencia, note que la métrica podría producir resultados no validos cuando $p_a = 0$ o $p_b = 0$. En otras circunstancias su rango de operación está entre $[0, n]$ y es simétrico. El valor mínimo se consigue cuando $p_{ab} = 0$ y $p_a \neq 0$ y/o $p_b \neq 0$, mientras que el valor máximo se alcanza cuando $p_{ab} = p_a = p_b = 1/n$ (con solo una transacción satisfecha dentro del conjunto de datos). Esta métrica también puede definirse como una medida de correlación, calculando el grado de dependencia entre el antecedente y el consecuente de una regla. Los valores de "Lift" inferiores a 1 determinan una dependencia negativa (dependencia positiva para valores superiores a 1), mientras que un valor de 1 significa independencia.

Cosine [91]

Es una medida de calidad derivada de "Lift", formalmente es definido como $cosine(A \rightarrow B) = \sqrt{Lift \times sop} = p_{ab}/\sqrt{p_a \times p_b}$, similar a "Lift" con la diferencia de la raíz cuadrada de los productos entre p_a y p_b en el denominador de la ecuación. Esta raíz cuadrada significa que "Cosine" sólo es influenciado por p_a, p_b y p_{ab}, pero no por el número total de transacciones N. El intervalo en que la medida opera está entre [0,1]. Cabe destacar que la medida de calidad del "Cosine" incluye las siguientes propiedades: (a) tiene en cuenta tanto el interés como la importancia de una regla de asociación ya que contiene dos medidas de calidad importantes en este sentido, es decir, "Soporte" y "Lift", (b) Es equivalente a la media geométrica de la medida de "Confianza" y (c) Similar a "Lift", métrica "Cosine" tiene la propiedad de simetría ($cosine(A \rightarrow B) = cosine(B \rightarrow A)$).

Leverage [93]

Fue propuesto por *Piatetsky-Shapiro*, como medida de calidad bastante similar a "Lift". "Leverage" determina qué tan diferente es la co-ocurrencia del antecedente A y el consecuente B de una regla desde el concepto de in-

dependencia. También conocida como "Novelty", formalmente se define como $Leverage(A \rightarrow B) = p_{ab} - (p_a \times p_b)$, y toma valores entre $[-0{,}25, 0{,}25]$, devolviendo un valor de cero en casos donde el antecedente y consecuente son independientes. Finalmente, esta medida también incluye la simetría como una propiedad relevante ($Leverage(A \rightarrow B) = Leverage(B \rightarrow A)$)

Conviction [92]

Fue presentado como una propuesta para representar el grado de implicación de una regla, y los valores cercanos a la unidad indican reglas interesantes. Formalmente se define como $Conviction(A \rightarrow B) = (p_a \times p_{\bar{b}})/p_{a\bar{b}}$ también puede ser definido así $Conviction(A \rightarrow B) = (p_a - (p_a \times p_b))/(p_a - p_{ab})$. "Conviction" toma valores en el rango entre $[1/n, n/4]$, el umbral más bajo es alcanzado, cuando $p_a = p_{\bar{b}} = p_{a\bar{b}}$ note que $p_a = 0$ produce una indeterminación, asimismo el umbral superior se consigue cuando $p_a = p_{\bar{b}} = 0{,}5$ y $p_{a\bar{b}} = 1/n$.

Gain [94]

También llamado "Centered Confidence (CC)", formalmente es definido como $gain(A \rightarrow B) = conf(A \rightarrow B) - p_b$. Esta medida produce valores en el rango de $[-1, 1]$, siendo imposible que tome valores igual a la unidad, note también que el máximo valor factible es alcanzado cuando p_b es mínimo y $conf(A \rightarrow B)$ es máximo, el valor mínimo factible de p_b es $p_b = 1/n$, dado que los valores 0 no producen valores máximos de confianza, es decir, $conf(A \rightarrow B) = 0$. Asimismo el valor máximo para "gain" es igual a $gain(A \rightarrow B) = 1 - 1/n$, cuando $p_{ab} = p_a$ y $p_b = 1/n$.

Otras medidas de calidad

Medidas de calidad diversas han sido propuestas por investigadores alrededor del mundo, casi todos los trabajos existentes en el campo de la minería de reglas de asociación incluyen, al menos, una de las medidas de calidad antes mencionadas, hay muchas otras métricas adicionales que podrían usarse en diferentes escenarios. Así "Certainty factor(CF)"[95] se define como la ganancia normalizada en el intervalo $[-1, 1]$, "Recall"[90] se denota como el porcentaje de transacciones en un conjunto de datos conteniendo B, y al mismo tiempo también A, la métrica de "Laplace"[90], el "coeficiente de correlación de Pearson"[96], "Information Gain (IG)"[96],"sebag"[96],"Least Contradiction (LC)"[90],"example" y "Counter Example Rate (ECR)"[90],

"netconf"[91] descrito como una medida para estimar la fuerza de la regla de asociación, "Yule'sQ"[91].

Selección de métricas

La extracción de reglas de asociación mediante el uso de algoritmos evolutivos implica la construcción de funciones de ajuste. El problema de optimización puede ser de objetivo simple o múltiple, y el principio de funcionamiento está en la convergencia a la solución mediante optimización de la o las funciones objetivo. Una tarea elemental en este proceso es la construcción de estas funciones, debiendo tener muy claro los conceptos que están relacionados con las métricas asociadas, para este caso a la calidad de reglas de asociación. El estudio realizado en [97] contribuye significativamente en esta tarea, pues presenta resultados relacionados a métricas que están relacionadas entre sí, y que podrían ser usadas al mismo tiempo dentro de la definición de alguna función de ajuste. Además ponen a las medidas de soporte y confianza como elementales para cualquier buena métrica de calidad, pues éstas son ampliamente utilizadas en este tipo de problema. Sin embargo, es esencial que se usen junto a otras medidas de calidad para conseguir cualquier información adicional. Una investigación realizada para la selección de las medidas de mayor importancia en minería de reglas de asociación determinó que "Confianza","Soporte", "Gain" y "Leverange" son las más correlacionadas en cada componente principal [98]. Así, estas podrían ser muy buenas candidatas para ser optimizadas mediante alguna técnica de objetivos múltiples.

La elección de métricas que reflejen el requerimiento humano real sigue siendo un tema abierto. Un enfoque prometedor es utilizar el metaaprendizaje para seleccionar o combinar automáticamente las medidas adecuadas. Otra posibilidad es desarrollar una interfaz de usuario interactiva basada en la interpretación visual de los datos utilizando una medida seleccionada para ayudar al proceso de selección. Los experimentos extensos que comparan los resultados de las medidas de interés con el requerimiento humano real podrían usarse como otro método de análisis. Dado que las interacciones del usuario son indispensables en la determinación del interés de las reglas, es deseable desarrollar nuevas teorías, métodos y herramientas para facilitar la comprensión del usuario.

2.3.5. Extracción de Reglas de Asociación mediante algoritmos evolutivos

El proceso de descubrimiento de Reglas de Asociación (RA) a partir de conjuntos de datos es considerado un problema de complejidad *NP-difícil*. Así, si se tiene n elementos en un conjunto de datos, el número de itemsets es 2^n, el número máximo de RAs que pueden ser extraídas de cada itemset es $2^k - 2$, donde k es la longitud del itemset. Asimismo la complejidad de tiempo de los algoritmos basados en Apriori es $O(2^n) + O(2^k)$. Como resultado la complejidad para tiempo en los algoritmos de descubrimiento de reglas de asociación es $O(k \times 2^n)$[99]. Esto demuestra que el tiempo de ejecución crece exponencialmente con un aumento en el número de elementos [89].
Los algoritmos ARM tradicionales requieren una cantidad considerable de tiempo de cálculo. Además, dependen de la preparación de los datos, antes de aplicar el algoritmo, lo que provoca una pérdida de información. Además, un límite definido entre los intervalos en los atributos numéricos y la distinción del grado de pertenencia del intervalo en conjuntos difusos son otros dos inconvenientes de los métodos ARM convencionales.

Los algoritmos de Computación Evolutiva (EC) son una estrategia eficiente y de vanguardia para encontrar soluciones casi óptimas. Los algoritmos EC codifican el problema en términos de solución(es) a desarrollar para mejorar su calidad. Una característica clave de los enfoques de EC es que se pueden establecer condiciones de terminación estrictas para limitar el tiempo de cálculo mientras se puede obtener una solución casi óptima. Además, el uso de algoritmos EC permite el descubrimiento de reglas de asociación sin el paso previo correspondiente a la generación de itemsets. Esto conduce a una reducción en el cálculo del tiempo [100].

En las última década, varios investigadores han presentado el descubrimiento de las RA basadas en metaheurísticas para abordar las limitaciones de los enfoques tradicionales. En algunos trabajos de revisión que cubren un área pequeña de ARM evolutivo. Un análisis comparativo de tres métodos ARM evolutivos mostró la eficacia de GA(Genetic Algorithm) para minar AR cuantitativos (QAR) [17]. Asimismo en otro trabajo[101] se estudió la aplicación de tres enfoques metaheurísticos (incluidos GA, PSO y ACO(Ant Colony Optimization)) para la minería de itemsets frecuentes y la minería de itemsets con alta utilidad (HUIM). Algunos de los métodos ARM que se basan en algoritmos evolutivos multiobjetivo se pueden encontrar en [102].
En 2014, los autores categorizaron los algoritmos en tres tipos: ARM ca-

tegórico, ARM difuso y ARM numérico. Ventura y Luna [103] describieron algunos de los métodos construidos en base a la programación genética guiada por gramática (G3P) para la extracción de AR. Se publicó una revisión de la optimización multiobjetivo en ARM [104], que analizó ARM en términos de diferentes aspectos como representaciones cromosómicas, operadores genéticos y funciones de aptitud. Asimismo otros autores[105] presentaron una revisión más reciente en comparación a este último enfoque. [106] revisaron los enfoques evolutivos de ARM publicados en 2011. Ghafari y Tjortjis [107] proporcionaron un estudio comparativo de ciertos algoritmos evolutivos de ARM.

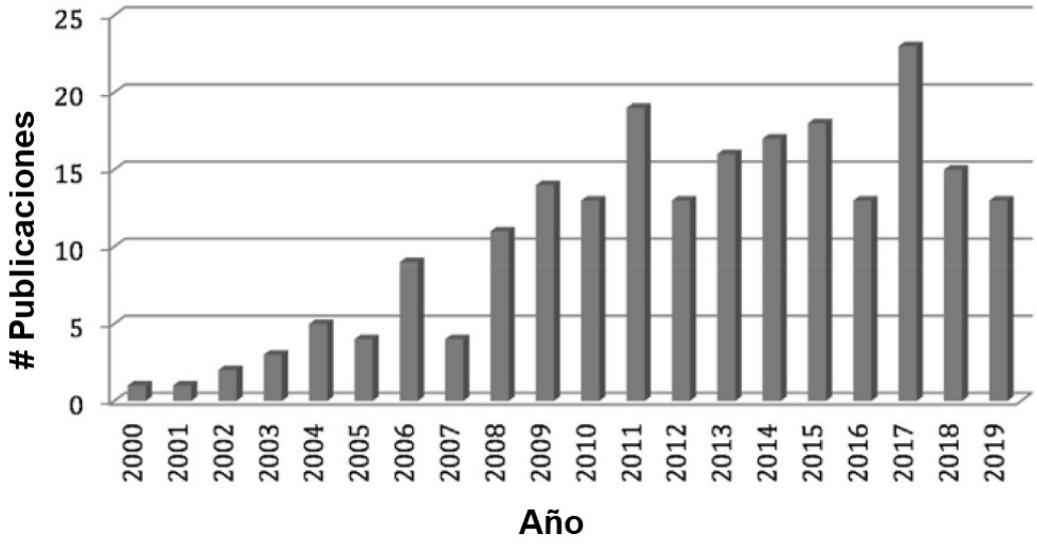

Figura 2.11: Número de publicaciones sobre ARM entre años 2000 y 2019[108]

De una variedad de artículos recopilados en el estudio llevado a cabo por [108], se seleccionaron 214 artículos. Dentro de estos 214 artículos se propusieron un total de 219 algoritmos ARM. La Figura 2.11 muestra la distribución de los artículos recopilados publicados entre 2000 y 2019. Tal como se presenta, el mayor número de artículos se propuso en 2017 (23 de 214, 10 %). Esta figura demuestra que el ARM evolutivo se ha convertido en un tema popular en los últimos años. Adicional, proponen una clasificación que divide los algoritmos ARM en cuatro grupos.

1. Técnicas basadas en Computación Evolutiva, que incluyen (enfoques basados en algoritmos genéticos (GA) y Evolución diferencial (DE))

2. Inteligencia basada en enjambres

3. Optimización basada en Honey Bee (HBO)

4. Hibridación aplicado a la minería de reglas de asociación

Debido a la naturaleza de los espacios de gran dimensión, ARM es difícil de resolver. Por lo tanto, los métodos heurísticos tradicionales no pueden proporcionar soluciones sofisticadas, lo que ha dado lugar a una mayor popularidad de enfoques de optimización innovadores no exactos conocidos como algoritmos EC. Estos enfoques utilizan un proceso heurístico iterativo para buscar en el espacio del problema y producir una solución suficientemente buena [109]. Basados en población y de solución única son las dos categorías principales de algoritmos metaheurísticos. Un enfoque basado en la población parte de un conjunto de soluciones aleatorias iniciales. El algoritmo intenta alcanzar una solución óptima a través de operadores probabilísticos inspirados en la naturaleza, que acercan progresivamente a la población hacia mejores soluciones. Algunos ejemplos son GA, PSO y ACO. En los casos de solución única, que también toman denominaciones como métodos de trayectoria, un algoritmo parte de una solución aleatoria inicial. Luego, genera iterativamente una nueva solución utilizando la anterior hasta que se alcanza un número específico de iteraciones. El recocido simulado (SA) y la búsqueda tabú son dos ejemplos clásicos.

Otra clasificación para los algoritmos EC son los inspirados en la naturaleza y no inspirados en la naturaleza [110]. Los algoritmos inspirados en la naturaleza se clasifican en cuatro grupos: bio-inspirados, inspirados en la física, inspirados en la geografía e inspirados en la cultura social. Los algoritmos bio-inspirados están influenciados por la ciencia biológica. Los algoritmos basados en inteligencia de enjambre y basados en la evolución son dos clases de algoritmos bio-inspirados; el origen de estos enfoques es el comportamiento biológico de los objetos naturales. Los enfoques de inteligencia de enjambre simulan los comportamientos colectivos de enjambres sociales de aves o insectos que viven en una colonia. Los algoritmos basados en la evolución se inspiran en los principios darwinianos de la capacidad de la naturaleza para desarrollar seres vivos que se adapten bien a su entorno. Los algoritmos inspirados en la física que buscan el espacio del problema se derivan de reglas físico-químicas. Dos ejemplos son SA y el algoritmo de búsqueda gravitacional (GSA). Los algoritmos inspirados en la geografía generan soluciones aleatorias en el espacio de búsqueda geográfica; búsqueda Tabú entra en esta categoría. Los algoritmos inspirados en la cultura social se basan en los sistemas sociales y culturales de optimización. El algoritmo memético es un buen ejemplo de este tipo de algoritmo, que utiliza la búsqueda heurística local para imitar el proceso de mutación. Además de las metaheurísticas inspiradas

en la naturaleza, algunos algoritmos que no forman parte de este conjunto están basados en fuentes, que no están relacionadas con la naturaleza, como la sociedad humana. Por ejemplo, el algoritmo de campeonato de liga [111] simula un entorno de campeonato con equipos artificiales que juegan en una liga artificial durante varias iteraciones.

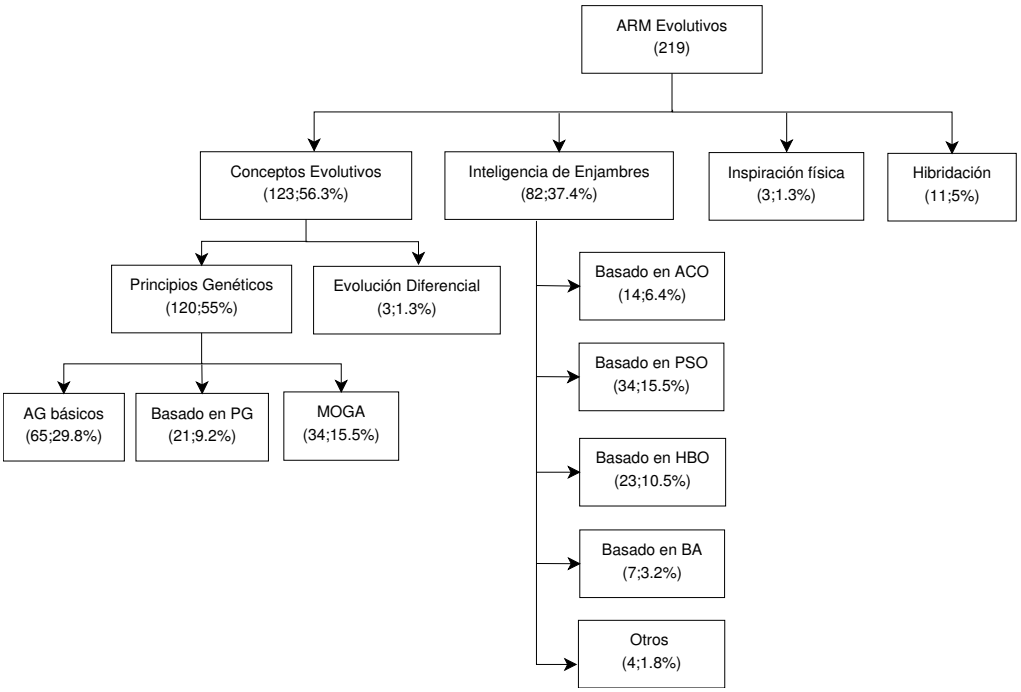

Figura 2.12: Taxonomía de algoritmos ARMs Evolutivos, tomado de [108]

De manera general, en el campo ARM, se han utilizado muchos algoritmos de EC para abordar los desafíos de los algoritmos ARM tradicionales en términos de reducción del tiempo y el número de patrones extraídos de conjuntos de datos a gran escala. En la Figura 2.12 se presenta una clasificación de los algoritmos ARM evolutivos. De una muestra de 219 métodos ARM basados en el estudio realizado por [108], la mayoría utiliza GA para extraer RA (54 %; 120 de 219). Debido a la gran cantidad de métodos basados en la genética, se dividen en tres grupos: GA básica, Programación genética (PG) y GA multiobjetivo (MOGA).

A continuación se describe las dos divisiones más importantes, aquellos basados en la computación evolutiva y los que se relacionan con la inteligencia de enjambres.

Algoritmos Genéticos para ARM

Los algoritmos evolutivos son métodos de búsqueda estocásticos basados en ideas evolutivas sobre la selección natural y la genética. Estos algoritmos usan operadores biológicos así como cruce, mutación y selección natural. La adaptabilidad y auto-organización son dos características principales de estos algoritmos. En algoritmos basados en la evolución, la población entera es remplazado por una nueva generación usando operadores naturales como cruce y mutación.

Algoritmos Genéticos es el método más popular y ampliamente utilizado, de manera general consta de cinco fases: inicialización, evaluación, reproducción, cruce y mutación. La representación más común de un cromosoma en este tipo de problema es mediante una secuencia binaria de longitud fija, la población inicial es generada de forma aleatoria al comienzo de la ejecución, cada cromosoma es evaluada mediante una función de aptitud para seleccionar los individuos que darán lugar a la descendencia.

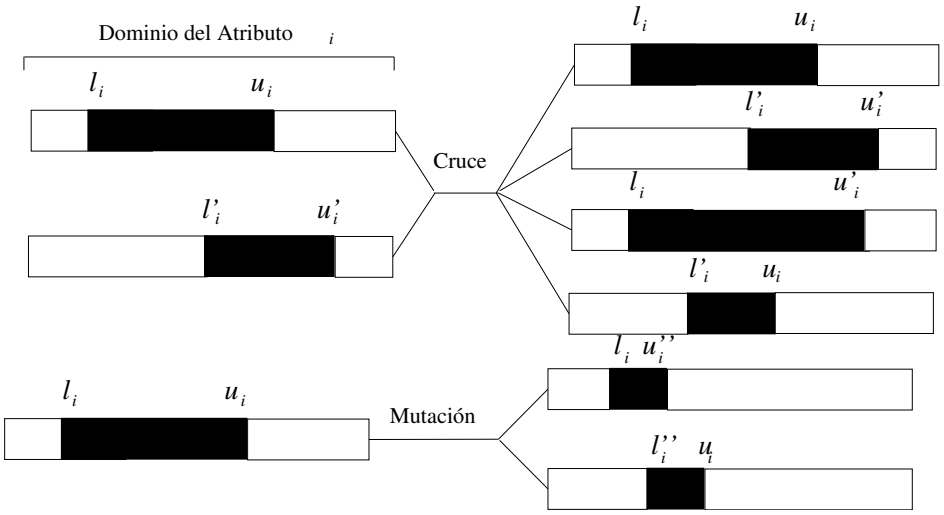

Figura 2.13: Operadores cruce y mutación

La mutación y el cruce (Figura 2.13) son operadores que dan lugar a un nuevo individuo, mejorando su calidad, de forma alterna pueden existir diversas representaciones de los individuos en esta clase de algoritmos poblacionales. El GA aplicado a ARM fue formalmente descrito en un trabajo realizado por Wang y Bridges [112]. Los trabajos iniciales sobre aplicación de GA para ARM estuvieron asociado a una fase postprocesamiento de los algoritmos ARM convencionales, donde variantes del algoritmo Apriori se usaban

para identificar las Reglas de Asociación y los GA se aplicaban en una etapa de optimización [113][114]. Varios esfuerzos en todos los procesos que componen la estructura del GA se han venido realizando, así por ejemplo [113] diseñaron mediante aprendizaje supervisado una estrategia para identificar puntos de corte más precisos en la definición de intervalos sobre atributos continuos. Asimismo se introdujeron mejoras en relación a los operadores de cruce[116] y mutación[115], además alternativamente se han experimentado con tasas de mutación adaptativas (con cálculo de la probabilidad de mutación en cada iteración), sin embargo, los tiempos de respuesta no fueron favorables. ARMGA[117] es un algoritmo bien conocido para minería de QARs, su único defecto es la generación de soluciones no admisibles, sin embargo, múltiples mejoras se han realizado, como la aplicación de múltiples semillas para la generación de una población inicial efectiva [118]. Adicional a esto, ARMGA también has sido extendido para extraer QARs positivas y negativas [119],[267].

ARM multinivel es una estrategia eficaz para descubrir relaciones entre atributos. La mayoría de los algoritmos ARM descubren las AR en un solo nivel de abstracción (NDA) de los ítems. En algunas aplicaciones, se requiere que las asociaciones se extraigan en múltiples NDA. Por ejemplo, un tomador de decisiones no necesita información sobre el pan de trigo y el pan blanco, que son dos tipos de pan. Por lo tanto, la regla $pan \rightarrow leche$ puede ser más útil para el tomador de decisiones que la regla $pan : trigo \rightarrow lacteos : leche$. Extraer AR en diferentes NDA es más informativo que un solo nivel.
La taxonomía de categorizaciones es esencial para la minería de AR multinivel en los que los conceptos de nivel primitivo se generalizan a los de alto nivel. En cada nivel de abstracción, existe su propio umbral de soporte mínimo. Los elementos del nivel superior tienen mayor soporte y viceversa. Por lo tanto, para encontrar asociaciones en niveles de abstracción bajos, se debe reducir el soporte mínimo. Diferentes umbrales de soportes y/o confianzas mínimos pueden ser configurados en cada nivel[120].

Wang y Bridges[112] publicaron por primera vez el problema de aplicar conjuntos difusos sobre ARM basado en GA para el ajuste de las Funciones Miembro (FM) basándose en la similitud de los AR difusas. Entiéndase que ARM-difuso es el proceso de utilizar el conjunto difuso y los conceptos de FMs en QARM. ARM-difuso hace una transición suave entre un miembro y un no miembro de un conjunto, haciendo que el resultado sea más interpretativo que el ARM preciso [121][122]. En ARM difuso, los valores de atributos numéricos se representan en términos de términos lingüísticos. Las FM se

utilizan primero para transformar cada valor numérico en términos lingüísticos. Luego, la cardinalidad escalar de cada término lingüístico se calcula en todas las transacciones. Finalmente, el proceso de minería basado en recuentos difusos se realiza para encontrar AR difusos (FAR).
Uno de los principales desafíos en ARM-difuso es encontrar un conjunto de FM adecuados. Propuestas presentadas por [123],[124], describen un método de agrupamiento basado en GA para encontrar un número dado de FM, las FM con los puntos de centroide óptimos se utilizan para extraer AR. Otro método basado en agrupamiento fue propuesto por [125][126]. Una base de datos se agrupa a través de k-medias para reducir el tiempo de ejecución y el escaneo de la base de datos. Sin embargo, este método no es eficiente porque la evaluación de cromosomas en grupos requiere una cantidad de tiempo significativa. Además en [127], propusieron una solución para extraer términos lingüísticos y sus FM para datos cuantitativos. Cada individuo consta de dos tipos de genes: genes de control para determinar si las FM son activos y genes paramétricos que codifican FM para un elemento. Otro aporte interesante se define en [128], consiste de un ARM basado en GA para encontrar reglas de asociación difusas(FAR) en problemas de clasificación. El número y la forma de las FM en cada atributo y el soporte difuso mínimo se determinan automáticamente mediante un cromosoma binario. El valor de aptitud es la maximización de la tasa de precisión de clasificación y la minimización del número de reglas difusas. En [129], las FM se optimizaron de modo que se minimizaran el soporte y la confianza. Las FM se seleccionan maximizando el número de conjuntos de elementos grandes y el promedio del intervalo de confianza. En el método propuesto en [130], se utilizó representación lingüística de 2 tuplas para reducir el espacio de búsqueda. En [131] se aplicó una representación lingüística de 3 tuplas para derivar FM, y luego se adaptó un algoritmo basado en el crecimiento de FP utilizando las FM. Por otra parte en [132] se presentó una nueva representación cromosómica para las FM considerando la estructura de las FM y su relación. Se propusieron dos heurísticas para reducir la disposición inapropiada de FM y reducir el espacio de búsqueda.

Algoritmos Meméticos (AMs) son un tipo de metaheurísticas que combinan un conjunto de técnicas de búsqueda local dentro de un ambiente evolutivo [133]. Un AM robusto consiste de una lista de algoritmos de búsqueda locales que tienen varios mecanismos de trabajo para permitir la exploración de manera complementaria. Así, el equilibrio entre búsqueda global y local representa el funcionamiento satisfactorio de un AM [134]. Por lo general, los métodos de EC, como los GA, requieren tiempos de cálculo

prolongados, la evaluación de los cromosomas es un desafío. En este sentido se han presentado diferentes mecanismos que reducen la complejidad temporal de la evaluación cromosómica. Un mecanismo que combina de grandes conjuntos de 1-itemset y FM para evaluar los cromosomas es una propuesta de [275]. De hecho, calcular conjuntos grandes de 1-itemset requiere mucho tiempo, una mejora al enfoque fue introducida por [274] a través de k-medias para juntar una población en grupos, lo que aceleró el proceso de evaluación. Más extensiones fueron propuestas por [135], mediante el uso de un método de dos fases: el soporte mínimo y las FM se encuentran en la primera fase, mientras que las reglas difusas se extraen en el segundo paso [136]. Estrategias de divide y vencerás [121][275] para manejar el problema del tiempo de ejecución en el que hay una población para el soporte mínimo de cada ítem y FM. Las mejores FM de todas las poblaciones se utilizan para extraer FAR. Los autores [137][138] también diseñaron una paralelización de [274] utilizando arquitectura maestro/esclavo en la que los procesos asociados al GA se realizan en procesadores maestros, mientras que las tareas de evaluación de aptitud se distribuyen en procesadores esclavos. Una versión producto de la combinación entre la arquitectura paralela maestro/esclavo propuesta en [137] y la técnica ARM basada en clústeres [125][126][121] para reducir el tiempo de ejecución de evaluación de los cromosomas. La arquitectura de CPU/GPU es otra estrategia para manejar el problema de descubrir RA en el contexto de "bigdata". Todas las reglas se generan utilizando una CPU de varios núcleos a partir de varios subespacios independientes, mientras que las reglas se evalúan en unidades de procesamiento de gráficos (GPU) [139].

En el contexto de ARM cuantitativo, el algoritmo GENetic ARs (GENAR) [271] es el primer esfuerzo que incluye intervalos mínimos y máximos para cada atributo cuantitativo. La longitud de cada regla es igual al número de atributos, y el consecuente de la regla define como el último atributo. La función de ajuste se diseñó utilizando el número de registros cubiertos. El algoritmo (GAR) [270], que es una extensión del anterior, extrae itemsets frecuentes sin discretización previa de los atributos. En los algoritmos GAR y GENAR, cada individuo está representado por tres grupos de genes. En cada grupo, el primer gen representa el atributo, mientras que los genes restantes indican los límites mínimo y máximo del intervalo. Una limitación es que el tamaño de los individuos puede ser diferente entre poblaciones. Esto conduce a la degradación del rendimiento de los operadores de cruce y mutación. El algoritmo GAR se amplió en [140] para datos con variables tanto continuas como nominales.

Recientemente, algunos trabajos han combinado GA con técnicas de aprendizaje automático como las redes neuronales [141] y mapas auto-organizados (en inglés, self-organizing map, SOM) [142] para realizar eficazmente ARM. Un SOM es un tipo de red neuronal artificial, que es entrenada usando aprendizaje no supervisado para producir una representación discreta del espacio de las muestras de entrada, llamado mapa. En [142], se utilizó primero un SOM para generar patrones frecuentes agrupados precisos, y luego se utilizó GAs para generar RA positivas y negativas. Un método ARM temporal híbrido fue presentado en [143], este método mejoró la precisión de las predicciones de tráfico, como se indica a continuación. Primero, un algoritmo de agrupamiento, DBSCAN, se aplica para encontrar el ambiente de tráfico. Luego, las RA temporales se extraen utilizando un enfoque de minería de reglas basado en GA. Por último, se utiliza un mecanismo de clasificación para predecir el nivel de congestión del tráfico.

Programación Genética para ARM

La programación genética (PG) es una clase de GA para optimizar programas de computadora. PG representa un problema como árbol, mientras que GAs usan una estructura de cadena [144]. A diferencia de GA, el tamaño de los individuos no se fija en PG [145]. La programación de redes genéticas (del inglés Genetic Network Programming - GNP) es otra mejora de GA, donde el genoma es una estructura de red que se utiliza para manejar la baja eficiencia de búsqueda de PG [146].
Una estructura básica de GNP consta de tres tipos de nodos: nodo de inicio, nodo de decisión y nodo de procesamiento. Un nodo de inicio indica la primera posición a ejecutar, un nodo de decisión sirve como funciones de selección y determina el siguiente nodo, y nodo de procesamiento representa ciertas acciones funcionales.

En ARM basado en GNP, los ítems y sus valores corresponden a las funciones de los nodos de decisión. Las reglas de asociación se representan utilizando las conexiones entre estos nodos. Hay un nodo de procesamiento para cada nodo de decisión. Un atributo se mueve a otro nodo de decisión si se satisface, de lo contrario, se traslada a otro nodo de procesamiento. El ARM basado en el GNP ha atraído la atención recientemente [147][148]. Estas contribuciones se centran principalmente en las RA de clase. El uso de GNP para ARM fue propuesto por primera vez por [148], para descubrir RA de clase, aquí los atributos de la base de datos corresponden a los nodos de decisión y las RA están representados por las conexiones de los nodos.

G3P es una extensión de PG que usa la gramática para aplicar restricciones en árboles GP [149]. En G3P, cada individuo es un árbol de derivación que se genera usando gramática de contexto libre, donde las restricciones gramaticales se generan aplicando un conjunto de reglas de producción. La profundidad máxima del árbol se determina para evitar árboles muy profundos.

En general, PG para ARM se ha aplicado en menor grado que GAs. RA para clasificación y AR binarios han sido el nicho de estudio para esta técnica, debido a una menor complejidad tanto en implementación como en representación directa de GAs. Además, PG se puede utilizar como algoritmo de búsqueda y también como algoritmo de clasificación. Debido a la representación flexible, los métodos basados en PG pueden manejar conjuntos de datos a gran escala. La notable superioridad de G3P en relación a GNP es evidente en cuanto al grado de aplicación en los últimos años, un aspecto relevante es la estrategia de codificación G3P, que permite iniciar el proceso evolutivo con individuos factibles, y en cada iteración se eliminan aquellos que tienen calidad inferior, para mantener a los individuos con reglas de asociación relevantes.

Algoritmos Genéticos Multiobjetivo para ARM

La optimización multiobjetivo intenta lograr una compensación entre diferentes funciones de rendimiento en conflicto utilizando un conjunto de soluciones no dominadas [150]. Por ejemplo, lo más común, las AR pueden evaluarse tanto en las medidas de su soporte y confianza. Según las preferencias del usuario, se selecciona una única solución del conjunto. Este esquema se utiliza principalmente en clasificación y agrupamiento. Por otro lado, en ARM, todas las soluciones no dominadas son consideradas como el conjunto final.
MOGA (Multi-Objective Genetic Algorithm) se ha vuelto cada vez más popular en el área de ARM y varios métodos de ARM han utilizado múltiples medidas en el proceso de evaluación. La investigación sobre MOGA para ARM comenzó en 2004, en la que se utilizó MOGA para el descubrimiento de FMs [151][152]. Funciones asociadas al tamaño de los itemsets y al tiempo necesario para determinar conjuntos difusos se definieron como dos objetivos en [153], se utilizó un GA basada en óptimo de Pareto para abordar el conflicto de objetivos. Por otra parte el soporte, la confianza y el número de conjuntos difusos se incluyeron en [152] como tres objetivos. Un MOGA multinivel

es propuesto por [154], para minería de FMs y FARs, el algoritmo primero codifica las FMs de cada clase de elemento(categoría) en un cromosoma de acuerdo con la taxonomía dada. Luego se consideran dos funciones objetivos. El primero es la cantidad de conocimiento extraído en los diferentes niveles y el segundo es la idoneidad de las FM. Estas dos funciones objetivo fueron adoptados posteriormente por SPEA2[155][156][157] para minería de FMs y FARs. Medidas adicionales asociadas a la fuerza de asociación, el interés y la comprensibilidad fueron considerados por [158] para FARs. NMEEF-SD [159], es un algoritmo multiobjetivo no dominado para extracción de reglas difusas para el descubrimiento de subgrupos, basado en el modelo conocido NSGA-II, pero está orientado hacia el descubrimiento de subgrupos interpretables y de alta calidad. De manera análoga en [160][161] aplicaron NSGA-II para la extracción de patrones difusos desde datos cuantitativo.

Muchos de los esfuerzos de MOGA basados en ARM estuvieron enfocados en la minería de RA a partir de conjuntos de datos cuantitativos (QARM). En [162], el método define la confianza, el interés y la comprensibilidad como objetivos diferentes para la optimización multiobjetivo que se amplifica con el enfoque de algoritmos genéticos, y las mejores reglas se obtienen a través del óptimo de Pareto. En otro trabajo de [163] se desarrollo un algoritmo evolutivo multiobjetivo para QARM, el propósito fue tratar el problema asociado al descubrimiento de genes más relevantes e identificar las relaciones reguladoras directas entre ellos, todo esto desde micro-arreglos de datos sin un paso previo de discretización. [149] utilizó G3P con capacidades para la extracción de AR tanto cualitativas como numéricas en un simple paso, esta propuesta combina las bondades de G3P con la filosofía de las técnicas NSGA-II y SPEA2.

Una propuesta nueva es introducida por [164], basado en la representación de los individuos, operadores genéticos nuevos y un esquema de aprendizaje basado en ventanas, para alcanzar una mejor escalabilidad de las técnicas de QARM basadas en GA. Esto permite manejar conjuntos de datos a gran escala sin pérdida de calidad en los resultados obtenidos. Específicamente, las técnicas propuestas se integran en el algoritmo evolutivo multiobjetivo denominado QARGA-M para evaluar su desempeño. Tanto la versión estándar como la mejorada de QARGA-M se han probado en varios conjuntos de datos que presentan un número diferente de atributos e instancias.

En un contexto más centrado en la afinidad de productos y requerimientos de consumidores, [165] propone un método para generar AR considerando la

ambigüedad de los requerimientos de los consumidores, lidia tanto con atributos categóricos como cuantitativos en el proceso. El trabajo [166] propone una metodología para encontrar las configuraciones más adecuadas en función del conjunto de objetivos a optimizar y medidas de distancia para jerarquizar las soluciones no dominadas. Para ello, se analizan varias medidas de calidad que seleccionan el mejor conjunto de ellas para optimizar. Además, se aplican estrategias diferentes para reemplazar la distancia de aglomeración (Crowding) utilizada por NSGA-II, esto ayuda a clasificar las soluciones en cada frente de Pareto, debido a que dicha distancia no es adecuada para manejar problemas de muchos objetivos. Las mejoras propuestas se han integrado en el algoritmo multiobjetivo denominado MOQAR.

El algoritmo (Rare-PEARs)[99] da una oportunidad a cada regla con diferente longitud y apariencia para ser creada. Por lo tanto, se pueden encontrar varias reglas interesantes, raras o interesantes y raras. Algunas de estas reglas pueden resultar poco interesantes (aquellas que contienen conjuntos de elementos frecuentes). Sin embargo, se trata de evitarlos con Rare-PEARs. Para lograr este objetivo, el método descompone el proceso de extracción de reglas de asociación subproblemas, y cada subproblema es manejado por un subproceso independiente durante la ejecución de Rare-PEARs. Cada subproceso comienza individualmente con una población inicial diferente. Luego explora el espacio de búsqueda de su subproblema correspondiente para encontrar reglas con intervalos semióptimos para cada uno de los atributos.

[283][280] extendieron NSGA-II para QARM, en este realiza el aprendizaje evolutivo de los intervalos de los atributos y una selección de condiciones para cada regla. El algoritmo NSGA-II-QAR [283] utiliza un cruce multipunto. Se utilizan dos mutaciones diferentes para dos partes de un cromosoma. Una estrategia similar fue aplicado en [279] para descubrimiento de QARs positivas y negativas.

Evolución Diferencial para ARM

Evolución Diferencial (DE)[167] es un algoritmo sencillo y robusto basado en poblaciones y ha surgido como una de las familias más competitivas y versátiles de enfoques de computación evolutiva para resolver numerosos problemas del mundo real [168]. En el algoritmo DE se utilizan operadores de origen genético como el cruce, la mutación y la selección. Aunque el proceso de evolución de DE es similar a GA, se basa en una operación de mutación en lugar de cruzamiento. DE se realiza sobre la base de las diferencias entre

pares de soluciones que guían la búsqueda adicional [169]. Las operaciones de mutación, cruce y selección se realizan utilizando las ecuaciones 2.4, 2.5 y 2.4, respectivamente [170].

$$v_i^G = x_{r1}^G + F.(x_{r2}^G - x_{r3}^G) \tag{2.4}$$

$$u_{i,j}^G = \begin{cases} x_{i,j}^G & \text{if } rand_{i,j}[0,1] \leq CR \text{ or } i = i_{rand} \\ v_{i,j}^G & \text{otherwise} \end{cases} \tag{2.5}$$

$$u_i^{G+1} = \begin{cases} u_i^G & \text{if } f(u_i^G) < f(x_i^G) \\ x_i^G & \text{otherwise} \end{cases} \tag{2.6}$$

En la Ecuación 2.4, G indica la generación actual y F es el factor de escala utilizado para determinar la longitud de exploración. Hay dos tipos de operación de cruce: cruce binario (uniforme) y cruce exponencial. La Ecuación 2.5 presenta el cruce binario en esta ecuación $rand_{i,j}[0,1]$ es un número aleatorio entre $[0,1]$ y CR es la tasa de cruce, la Ecuación 2.4 compara la calidad de la solución obtenida por el operador de cruce con la calidad de su vector objetivo correspondiente, para especificar la solución que sobrevivirá para la próxima generación. La elección de los valores F y CR juega un papel significativo en el éxito de DE y la velocidad de convergencia. Los mecanismos de adaptación automática de parámetros son más beneficiosos para el desempeño de DE que una selección fija.

El uso de un DE mult-iobjetivo basado en Pareto para extraer AR numéricas fue la primera contribución en el campo ARM [277]. Se consideraron las medidas de *soporte*, *confianza*, *comprensibilidad* y *amplitud* como cuatro objetivos para formular ARM como un problema de optimización multiobjetivo. Las primeras tres medidas son objetivos de maximización, mientras que el último criterio es el objetivo de minimización. Los AR se extraen directamente sin generar itemsets frecuentes y sin determinar los umbrales de soporte y confianza. En el trabajo [171], los autores presentaron un método ARM basado en DE para conjuntos de datos con atributos tanto numéricos como categóricos. El soporte y la confianza se combinan en una función de aptitud como un solo problema de optimización objetivo. En [172] se propuso una modificación de DE que es capaz de controlar estrictamente el riesgo de reglas difusas falsas a través de dos pruebas estadísticas sobre reglas.

2.3.6. Algoritmos basados en inteligencia de enjambres para ARM

Los algoritmos de inteligencia de enjambres se inspiran en el comportamiento colectivo de enjambres como pájaros, peces, abejas y colonias de

hormigas. Los miembros de la población de un enjambre actualizan sus posiciones para que se ajusten al entorno respectivo [173]. Esta capacidad de inteligencia de enjambre ha atraído un gran interés en las últimas dos décadas, y muchos algoritmos de optimización basados en inteligencia de enjambre han ganado una gran popularidad en ARM.

Optimización por colonia de hormigas para ARM

El enfoque ACO (del inglés Ant Colony Optimization) para resolver problemas computacionales se inspiró en el comportamiento de búsqueda de alimento de las hormigas reales. Las hormigas comienzan a explorar al azar el área que rodea el nido [174]. Después de encontrar fuentes de alimento, las hormigas llevan parte del alimento al nido después de evaluar la cantidad y calidad del alimento. Las hormigas construyen soluciones viajando en una gráfica. Durante el viaje de regreso, dejan feromonas químicas en su camino para otras hormigas en función de la cantidad y calidad de la comida [144]. Este proceso ayuda a las hormigas a descubrir la dirección más corta entre el nido y la buena fuente de alimento que fue previamente explorada por las otras hormigas.

Sistema de Colonia de Hormigas (ACS pos us siglas en inglés)[175] ha sido la variante más común de ACO. En el enfoque ACS, la transición de estados y las reglas de actualización de feromonas se mejoraron para obtener mayor efectividad.

ACS se ha aplicado con éxito para optimizar problemas NP-difíciles. Uno de los primeros trabajos de ACS para ARM fue introducido por [176]. En la primera fase, la técnica de K-medias y ACS se combinaron para agrupar una base de datos y, en el segundo paso, las AR se descubren a través de ACS. Adicional a esto, un enfoque para variables de dominio continuo se desarrollo en [177]. Las contribuciones [178][179] propusieron dos algoritmos basados en ACS para optimizar las FM en espacio discreto para FAR.
Estos métodos se ampliaron a un espacio de solución continuo en [180][181]. Sin embargo, este último no fija bordes y nodos en el proceso de búsqueda, a diferencia de otros métodos que utilizan un mapa de ruta fijo con los nodos y bordes discretizados. Mejoras significativas se realizaron sobre [179] utilizando un proceso de búsqueda jerárquica, las búsquedas aproximadas con escalas grandes se realizan inicialmente en las primeras generaciones, y luego las escalas se ajustan a espacios reducidos para búsquedas precisas.

La programación de hormigas es una nueva variante de ACO, que mejora la capacidad de la programación automática para crear programas informáticos. En este enfoque se utiliza una estructura de árbol para representar a un individuo [182]. Extensiones enfocados en programación de hormigas multiobjetivo[183], también son aportes interesantes sobre el tema, en este contexto para limitar el espacio de búsqueda y asegurar la generación de individuos válidos, se aplicó gramática libre de contexto. Este enfoque fue mejorado por [183] tras formular una función de aptitud basada en Pareto utilizando soporte y confianza; también presentaron una gramática de hormigas monoobjetivo para ARM utilizando una función de aptitud escalar. Se desarrollaron dos algoritmos basados en programación de hormigas para AR raras[184], el uno se basó en la función de aptitud de un solo objetivo y el otro multiobjetivo basado en soporte y lift.

En general, existe una mayor tendencia hacia el desarrollo de algoritmos con enfoques sobre ARM difuso y ARM binario, funciones tanto monoobjetivo y multiobjetivo han sido implementadas, aunque existe mayor preferencia sobre el primero. En resumen, ACO representa un problema en forma de grafo, que es más flexible que las otras técnicas metaheurísticas conocidas como GA y PSO. Sin embargo, la representación gráfica y la codificación del problema como nodos influyen en el rendimiento. De hecho, debido al mejor rendimiento de ACS en comparación con ACO, todos los métodos ARM utilizan ACS para hacer evolucionar el problema.

Optimización por inteligencia de enjambres para ARM

Optimización por enjambre de partículas (del inglés Particle Swarm Optimization (PSO))[185] es un algoritmo de optimización basado en la población que se inspiró en el comportamiento social de los animales, como el vuelo de las aves y la formación de bancos de peces. Eficientemente, la simplicidad y la rápida tasa de convergencia son las principales ventajas de PSO. Cada partícula representa a una solución del problema y tiene la velocidad que representa una dirección de vuelo hacia otras soluciones. en detalle, existen cuatro conceptos asociados a PSO:

1. $X_i(t) = (X_{i1}, X_{i2}, \cdots, X_{iD})$ sujeto a: $X_{i,n}(t) \in [l_n, u_n], 1 \leq n \leq N$
2. $pbest_i(t) = (p_{i1}, p_{i2}, \cdots, p_{iD})$
3. $gbest_i(t) = (g_{i1}, g_{i2}, \cdots, g_{iD})$
4. $v_i(t) = (v_{i1}, v_{i2}, \cdots, v_{iD})$

El término $X_i(t)$ es la posición de la i_{ma} partícula en la t_{ma} iteración $pbest_i(t)$ y $gbest_i(t)$ son la mejor solución local y global respectivamente, $v_i(t)$ es la velocidad de la partícula i en la t_{ma} iteración; durante cada iteración las propiedades de cada partícula de la población es actualizado en base a *pbest* y *gbest*.

Una técnica PSO con enfoque multiobjetivo fue introducida por [186] como una estrategia de búsqueda para minería de reglas de asociación para clasificación, la extensión de PSO utiliza una medida de similitud para la búsqueda de vecindarios con la idea de almacenar las mejores partículas globales encontradas de manera multiobjetivo. Trabajos basados en PSO no lineal dirigidos hacia la minería de FARs fueron realizados por [184], cada FM se genera en base a la mejor partícula de la población.

La continua introducción de mejoras sobre PSO han dado lugar a otras contribuciones tales como: [187][188][189], el primero de estos trabajos mencionados mejoró PSO utilizando un operador adicional en las formas de mutación de GA, este operador se utiliza después de la fase de inicialización de PSO, Los itemsets frecuentes se generan primero utilizando Apriori, y luego se aplica el algoritmo PSO mejorado para optimizarlos. En el siguiente trabajo, los autores calcularon parámetros de control en PSO basados en la estimación del estado de evolución y adaptaron el peso de inercia basándose sobre los valores de aptitud. El último trabajo mencionado al inicio de este párrafo, se introdujo una mejora en PSO para controlar la velocidad de un enjambre individual utilizando un método de control de coeficiente de aceleración adaptativo basado en la distancia.

Un trabajo reciente señala mejoras en el algoritmo de FP-growth utilizando PSO, en el que el mejor soporte es encontrado por PSO, para ser utilizado por FP-growth. Estos dos pasos son seguidos por la entropía de la información como el grado de interés para medir la efectividad de las AR.
En [191] se diseñó un mecanismo para buscar el valor óptimo de aptitud de cada partícula mediante PSO, el soporte mínimo y los umbrales de confianza fueron variables de operación en un espacio de búsqueda binaria. En [192] se presentó un método similar utilizando un PSO ponderado. Sheikhan y Rad [193] aplicaron PSO para descubrir FAR en la fase de selección de características de los sistemas de detección de intrusos. Utilizaron PSO para determinar un parámetro de ajuste de tamaño y umbrales de apoyo y confianza.

Trabajos adicionales basados en PSO multiobjetivo (MOPSO)[194] para ARM se realizaron aplicado al filtrado colaborativo para sistemas de recomendación, el soporte y la confianza fueron considerados objetivos, pudiéndose descubrir la asociación indirecta entre usuarios y artículos, incluso en transacciones muy reducidas. En otra línea se abordó ARM cuantitativo utilizando MOPSO basado en el óptimo de Pareto [89], aquí se define como objetivos la confianza, la comprensión y el interés. Otros trabajos basados en MOPSO para ARM cuantitativo se amplían en [195][196][197].

En el contexto de PSO binario (BPSO por sus siglas en inglés), un aporte relevante esta en [198]. Este algoritmo genera las reglas de asociación a partir de la base de datos transaccional mediante la formulación de un problema de optimización global combinatorio, sin especificar el soporte mínimo y la confianza mínima, la calidad de la regla se mide mediante una función de ajuste definida como el producto del soporte y la confianza. Trabajos análogos aplicados al ámbito de la minería de itemsets de alta utilidad (HUIM por sus siglas en inglés) fueron agregados en [199][200][201], dado que el mecanismo tradicional de PSO se utiliza para manejar el problema continuo, en este trabajo se adopta el PSO discreto para codificar las partículas como variables binarias. Trabajos basados en BPSO también abordaron problemas de minería de itemset frecuentes con el uso de mecanismos de recursividad [202][203].

ARM basado en PSO ha sido estudiado desde diferentes perspectivas, las contribuciones en este ámbito son numerosas tanto en AR temporales [204], AR con clases[205], AR positivas y negativas [207], en términos de bigdata el concepto de PSO paralelizado para extracción de reglas de asociación cuantitativas se estudió en [208].

Otros métodos de optimización para ARM

Técnicas basadas en comportamiento de las abejas cuando ellas buscan su alimento, el apareamiento y su reproducción. En la mayor parte estos algoritmos se inspiran en las abejas melíferas que imitan el comportamiento de búsqueda de alimento. En estos métodos, se utiliza una búsqueda exploratoria aleatoria para encontrar ubicaciones prometedoras; luego, estos algoritmos aplican la búsqueda de explotación en las ubicaciones para lograr una solución óptima global [209]. Tres algoritmos bien conocidos que utilizan el comportamiento de las abejas para explorar las fuentes de alimentos más ricas y accesibles son: la optimización por enjambre de abejas (BSO), colonia de abejas artificiales (ABC) y el algoritmo de abejas propiamiente dicho.

La estrategia BSO simula tres tipos de abejas experimentadas con diferentes patrones de vuelo: abejas recolectoras, observadoras y exploradoras. Cada abeja está asociada con una posición que representa una solución factible para un problema óptimo. En cada iteración, las abejas con la mejor y la peor aptitud se seleccionan como abejas recolectoras y exploradoras experimentadas, respectivamente. Las abejas espectadoras evalúan la información del néctar proporcionada por las abejas recolectoras experimentadas para ajustar su trayectoria en movimiento la próxima vez. La heterogeneidad en los patrones de vuelo de las abejas da como resultado un equilibrio entre la explotación y la exploración[210].

La curiosidad e innovación de investigadores en el campo de ARM es una motivación para aplicar diversas formas y técnicas evolutivas que surgen, así se han propuesto algunos algoritmos ARM basados en la metaheurística BSO. En uno de los primeros trabajos, se describe un BSO propuesto para el algoritmo ARM (BSO-ARM) utilizando la estrategia de eliminación y descomposición [211]. Ellos, también agregaron dos extensiones de BSO-ARM, en [212] se aplicaron tres heurísticas diferentes para explorar el espacio de búsqueda; mientras que otra mejora incluida en [203] aplica el uso de la propiedad recursiva de los itemsets frecuentes, donde se determinan y exploran las regiones de las abejas en lugar de realizar una búsqueda local.
Recientemente, se han explorado algunos métodos ARM a gran escala basados en GPU(Graphics Processing Unit). Las GPU suelen estar equipadas con una gran potencia informática y un gran ancho de banda de memoria [213], por lo que son adecuadas para la computación paralela para ARM a gran escala. El algoritmo BSO-ARM [211] se ha paralelizado utilizando arquitecturas de CPU/GPU [214][215][101].

ABC [216] se inspiró en el comportamiento de búsqueda inteligente del enjambre de abejas melíferas para optimizar los problemas numéricos. ABC puede resolver eficazmente problemas de optimización multimodal y multidimensional. Aparte de la naturaleza de la selección aleatoria, utiliza menos parámetros de control. Un algoritmo ABC multiobjetivo integrado basado en Pareto para el problema ARM fue desarrollado en el trabajo [217]. Utilizaron cuatro medidas que incluían la confianza, el soporte, la comprensibilidad y el interés como objetivos.

El algoritmo Bees [218] tiene funciones similares para la búsqueda local y los procesos de búsqueda global como el algoritmo ABC. Sin embargo, ABC usa un enfoque probabilístico durante la etapa de vecindad, mientras que el

algoritmo Bees usa la evaluación de la aptitud para impulsar la búsqueda. El algoritmo Bees se aplicó en [219] para encontrar funciones de pertenencia adecuadas para la minería de reglas de asociación temporal difusas.

Algoritmo murciélago (BA) [220][221] se inspira en el comportamiento de eco-localización de los micro-murciélagos para resolver problemas de optimización. Para equilibrar la exploración y la explotación durante el proceso de búsqueda, se aplica el zoom automático. Además, se utiliza un mecanismo de sintonización de frecuencia para aumentar la diversidad de la población [222]. En el BA, cada murciélago busca la solución óptima cambiando su frecuencia, velocidad y posición.
En un trabajo denominado Bat-ARM fue propuesto por primera vez el enfoque de BA para la minería de reglas de asociación. Se propuso un BA modificado para ARM por primera vez, llamado Bat-ARM [223][224]. Entonces, una estrategia maestro/esclavo fue aplicado para mejorar Bat-ARM dividiendo la población en diferentes sub-poblaciones. Sin embargo, un inconveniente es que la exploración del espacio de búsqueda se reduce en Bat-ARM debido a la falta de comunicación entre los murciélagos de la población. Por tanto en [225] se introdujeron tres estrategias para abordar este problema, llamadas anillo, maestro/esclavo e híbrido.

Otras técnicas de inteligencia de enjambre que han sido aplicados para ARM, incluyen el algoritmo de la luciérnaga (Firefly Algorithm (FA)), el algoritmo de búsqueda cuco (Cuckoo Search (CS)), la optimización de la migración animal (AMO) y el algoritmo de optimización de la búsqueda de los pingüinos (PeSOA). FA es un algoritmo inteligente de enjambre que anima los patrones y comportamientos destellantes de las luciérnagas y fue desarrollado por [226]. En FA, una luciérnaga menos brillante se mueve hacia una más brillante; de lo contrario, se mueve aleatoriamente. La función objetivo está asociada con las características de luz intermitente de la población de luciérnagas.

El algoritmo de búsqueda cuco [227] se basa en el comportamiento de los parásitos de cría obligados de ciertas especies de cuco en combinación con el comportamiento de vuelo de Levy de algunas aves y moscas de la fruta [228]. Una versión de búsqueda de cuco binaria modificada para ARM fue propuesto en [229], donde cada individuo codifica la regla de asociación correspondiente. Todos los atributos se codifican como cadenas binarias con tres bits de control adicionales que determinan la presencia/ausencia del atributo en la regla.

AMO [230] se inspira en el comportamiento migratorio de grupos de animales. Este algoritmo consta de dos procesos: el primer proceso simula cómo los animales se mueven desde su posición actual a una nueva posición, mientras que el segundo proceso simula cómo los animales abandonan el grupo y se vuelven a unir al grupo durante la migración. En [231] utilizaron esta estrategia para reducir el número de RA eliminando las reglas que estaban débilmente respaldadas por los datos. Durante el proceso de optimización, el algoritmo elimina las reglas que tienen un alto valor de aptitud. Esto se debe a que las reglas asociadas con un valor de aptitud pequeño deben migrarse, lo que se hace calculando la probabilidad de migración de las reglas.

PeSOA es una metaheurística propuesta en [232]. PeSOA se basa en los comportamientos mostrados por pingüino durante la caza. El trabajo [233] utilizó PeSOA en ARM para facilitar la exploración del espacio de la solución. Ellos Incorporaron una distancia superpuesta para generar solo reglas no redundantes con poca superposición entre las reglas minadas.

2.4. Software para minería de reglas de asociación

El software de minería de datos se refiere a los programas de computadora que permite a las empresas y otros usuarios extraer datos utilizables de un gran conjunto de datos sin procesar para encontrar correlaciones, patrones y anomalías. Los resultados del proceso de minería de datos ayudan a las empresas a predecir los resultados. Las técnicas claves utilizadas por el software de minería de datos para extraer datos incluyen análisis estadísticos, algoritmos específicos, aprendizaje automático, estadísticas de bases de datos e inteligencia artificial.

El objetivo principal de utilizar estos métodos es recuperar información útil de un gran conjunto de datos y transformarla en una estructura que sea fácil de entender y usar cuando sea necesario. En términos simples, las aplicaciones de minería de datos ayudan a las empresas a obtener conocimientos de grandes volúmenes de datos y a transformarlos en información procesable. Hay muchos sistemas de minería de datos y algunos de ellos ofrecen funcionalidades más avanzadas, los productos individuales también utilizan diferentes métodos para procesar información y validar resultados; por lo tanto, una elección de software de minería de datos dependerá de las preferencias y/o necesidades. La Figura 2.14 ilustra un conjunto de programas que

incluyen técnicas de minería de reglas de asociación, entre los más destacados se ha realizado una clasificación aproximada entre programas comerciales, de código abierto y entornos de desarrollo.

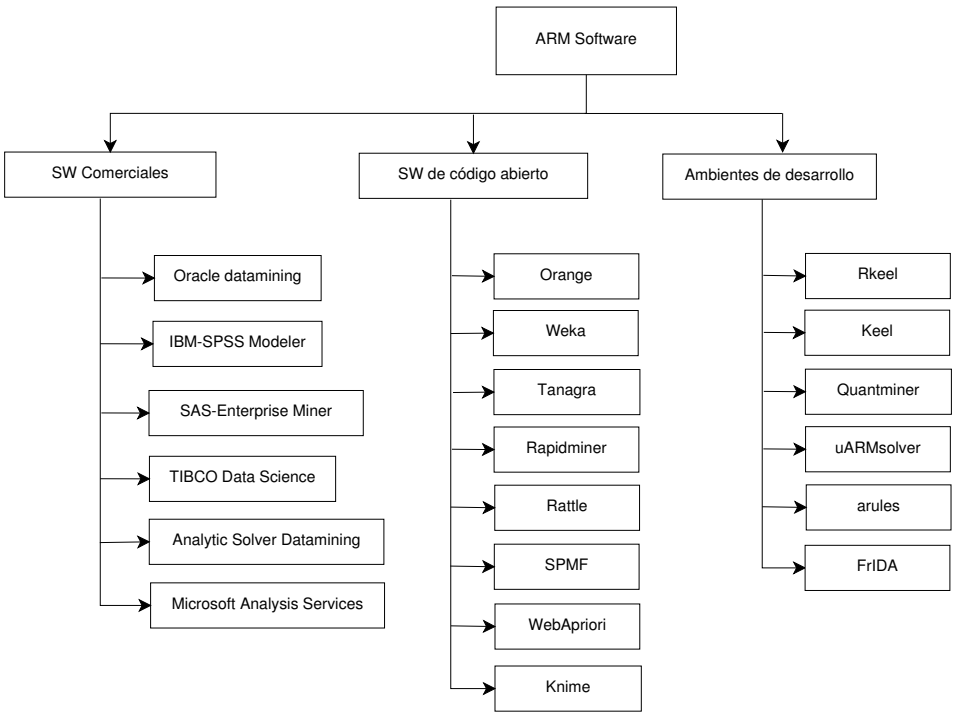

Figura 2.14: Programas que incluyen ARM

Software comercial para ARM

IBM SPSS Modeler 14.2[234], es un producto software de IBM, que incluye las funciones necesarias para ciencia de datos, tales como la preparación y el descubrimiento de datos, la analítica predictiva, la gestión e implementación de modelos y aprendizaje máquina para monetizar activos de datos.
En el área de minería de reglas de asociación incluye tres algoritmos diferentes para la generación de reglas de asociación, estos son: "Apriori", "Carma" y "Sequential". Los datos de entrada pueden tener el formato tanto de transacción como tabular, entre los tres métodos cubren datos tanto de tipo categórico como numéricos, y el último permite la generación de reglas secuenciales.

Oracle data miner (ODMr)[235], permite a los científicos de datos, analistas de datos y empresas trabajar directamente con los datos dentro de la base de datos, mediante un editor gráfico de flujo de trabajo tipo "arrastrar y

soltar". Es una extensión de Oracle SQL Developer, captura y documenta en flujos de trabajo analíticos gráficos, los pasos que toman los usuarios mientras exploran datos, y desarrollan metodologías de aprendizaje automático. Los flujos de trabajo ODMr son útiles para volver a ejecutar metodologías analíticas, y para compartir conocimientos con los miembros del equipo. ODMr genera scripts SQL y PL/SQL y ofrece una API de flujo de trabajo, para acelerar la implementación del modelo en toda la empresa.
ODMr emplea una implementación del algoritmo Apriori, basada en SQL para el cálculo de las reglas de asociación a partir de itemsets frecuentes. La generación del candidato, y los pasos de recuento de soporte se han implementado mediante consultas de SQL.

SAS Enterprise Miner[236], Está diseñado para capacidades analíticas avanzadas, como la creación de modelos descriptivos y predictivos con precisión para las empresas. Agiliza todos sus datos en un solo lugar con funciones de arrastrar y soltar, y al permitir mencionar cada proceso paso a paso. El software funciona en SEMMA, es decir, muestra, explora, modifica, modela y evalúa. En banca, se puede utilizar para la segmentación de clientes, la detección de morosidad en varios segmentos, la identificación de cohortes específicas para la orientación.
Minería de reglas de asociación está localizado en la categoría "Exploración" de la metodología SEMMA, y mediante el nodo "Asociación" se puede proceder al uso de las funciones para descubrimiento de reglas y secuencias. Los datos de entrada debería estar en formato de transacción, y los resultados pueden tener varias formas de visualización (matriz de reglas, tablas de reglas, gráfico de líneas, gráfico estadístico, gráfico de confianza, tabla descriptiva de reglas, mapa de vínculos).

TIBCO Data Science[237], Es un software que democratiza, colabora y pone en funcionamiento el aprendizaje automático para toda la organización, "TIBCO Data Science - Workbench (Statistica)", integra una plataforma de ciencia de datos y aprendizaje automático (ML). Construye canalizaciones para ciencia de datos de extremo a extremo: preparación de datos desde formularios, creación de modelos, implementación y monitoreo.

- En la fase de preparación de datos, conecta y combina fácilmente fuentes de datos dispares. Prepara datos para análisis con inteligencia incorporada y automatización para imputar valores perdidos, eliminar valores atípicos, crear funciones y otras características.

- Crea rápidamente modelos estadísticos y de aprendizaje automático con cientos de algoritmos integrados, con mecanismos para selección

del mejor según el enfoque de trabajo. Permite inserción de código Python y R, mediante nodos de código incrustados dentro de su canal de ciencia de datos para integrar a la perfección lenguajes y bibliotecas de código abierto.

- Simplifica la implementación y elimina la recodificación exportando modelos en una variedad de lenguajes, incluidos C, C++, PMML, Visual Basic, SAS, Java o C+ procedimientos almacenados.
- Garantiza la seguridad y el gobierno de datos mediante un repositorio de metadatos común, control de versiones, aprobaciones y capacidad de reversión.

Para trabajar con ARM, el software tiene implementado el algoritmo Apriori, El módulo reglas de asociación puede incluir variables categóricas simples, variables dicotómicas y/o variables de respuesta múltiple. El algoritmo determinará las reglas de asociación sin requerir que el usuario especifique el número de categorías distintas presentes en los datos, o cualquier conocimiento previo sobre el grado factorial máximo o complejidad de las asociaciones importantes. En cierto sentido, el algoritmo construirá tablas de tabulación cruzada sin la necesidad de especificar el número de dimensiones para las tablas o el número de categorías para cada dimensión. Por lo que, esta técnica es particularmente adecuada para la minería de datos y texto de grandes bases de datos.

Analytic Solver Datamining[238], es la nueva plataforma de análisis para Excel. Realiza minería de datos completa y de uso fácil, minería de texto y análisis predictivo. Puede hacer muestreo de datos desde bases de datos SQL, Power Pivot y Apache Spark, explora los datos visualmente, limpia y transforma datos, crea, evalúa y aplica una gama completa de modelos de predicción para series de tiempo y minería de datos, desde regresión múltiple y regresión logística hasta árboles de clasificación y regresión, redes neuronales y reglas de asociación. Analytic Solver ahora viene en dos versiones: Analytic Solver Desktop, que se ejecuta en Excel para Windows (solo) y resuelve modelos por completo en su PC, y Analytic Solver Cloud, que se ejecuta en Excel para Windows, Excel para Macintosh y Excel para la Web. y resuelve modelos "en la nube", utilizando el servidor RASON de Frontline en Microsoft Azure (o, si se desea, en el servidor RASON de la empresa).

La mayoría de los datos a los que se aplicarán los métodos de XLMiner están estructurados adecuadamente como DataFrames. No es así con los datos de entrada para las reglas de asociación. Para un análisis de reglas de asociación, XLMiner recepta en una forma que represente transacciones o eventos

individuales, donde cada transacción puede involucrar un número variable de valores. Durante el proceso se construye un nuevo estimador de reglas de asociación y se establecen dos parámetros, soporte mínimo y el método. El soporte mínimo es para especificar el número mínimo de transacciones en las que debe aparecer un conjunto de elementos en particular para que califique para su inclusión en una regla de asociación, el valor predeterminado es el 10 % del número total de filas. Mientras que el método puede ser de tipo Apriori o T_tree.

Micosoft Analysis Services[239], Analysis Services es un motor de datos analíticos que se utiliza en soporte de decisiones y análisis de negocios. Proporciona capacidades de modelo de datos semánticos de nivel empresarial para inteligencia empresarial(BI), análisis de datos y aplicaciones de informes como Power BI, Excel, Reporting Services y otras herramientas de visualización de datos. Analysis Services está disponible en diferentes plataformas:

1. Azure Analysis Services, creados como un recurso de Azure.

2. Power BI Premium, el motor de Analysis Services proporciona programabilidad, aplicación cliente y compatibilidad con herramientas para conjuntos de datos Power BI Premium.

3. SQL Server Analysis Services, instalado como una instancia de servidor VM o local, SQL Server Analysis Services admite modelos tabulares en todos los niveles de compatibilidad (según la versión), modelos multidimensionales, minería de datos y Power Pivot para SharePoint.

Software de código abierto para ARM

Orange Datamining[240], es un software de código abierto creado por la Facultad de Informática de la Universidad de Ljubljana. Construye visualización de datos y aprendizaje automático con flujos de trabajo de análisis de datos de forma visual, con una caja de herramientas amplia y diversa. Orange está enfocado tanto para principiantes como para científicos de datos expertos. Gracias a su interface, los usuarios pueden centrarse en el análisis de datos en lugar de una codificación laboriosa, lo que simplifica la construcción de complejas canalizaciones de análisis de datos.
El análisis de datos se realiza apilando componentes en flujos de trabajo. Cada componente, llamado widget, incorpora alguna tarea de recuperación, preprocesamiento, visualización, modelado o evaluación de datos. La combinación de diferentes widgets en un flujo de trabajo le permite crear esquemas de análisis de datos completos sobre la marcha. Con una gran biblioteca de widgets, no le faltarán opciones. Los widgets adicionales están disponibles a

través de complementos y permiten una investigación más enfocada y orientada a temas.
Orange incluye en su gama de funciones las reglas de asociación, éstas ayudan al usuario a descubrir de forma rápida y sencilla las relaciones y conexiones subyacentes entre instancias de datos. Actualmente, el complemento tiene dos widgets: uno para las reglas de asociación y el otro para los conjuntos de elementos frecuentes. Con los conjuntos de elementos frecuentes, primero se verifica la frecuencia de los elementos y conjuntos de elementos en una matriz de transacciones.

Weka[243], es un software que tiene una colección de algoritmos de aprendizaje automático para tareas de minería de datos. Contiene herramientas para la preparación de datos, clasificación, regresión, agrupamiento, minería de reglas de asociación y visualización[241]. Las características de Weka son:

- Está disponible libremente bajo la licencia pública general de GNU.
- Es muy portable porque está completamente implementado en Java y puede correr en casi cualquier plataforma.
- Contiene una extensa colección de técnicas para preprocesamiento de datos y modelado.
- Es fácil de utilizar por un principiante gracias a su interfaz gráfica de usuario.

WEKA proporciona la implementación del algoritmo Apriori, donde el usuario puede definir el soporte mínimo y un nivel de confianza aceptable al calcular las reglas.

Tanagra[242], es un software de minería de datos gratuito para fines académicos y de investigación. Propone varios métodos de minería de datos del área de análisis exploratorio de datos, aprendizaje estadístico, aprendizaje automático y bases de datos. Más detallado, Tanagra contiene algo de aprendizaje supervisado pero también otros paradigmas como agrupamiento, análisis factorial, estadística paramétrica y no paramétrica, regla de asociación, selección de características y construcción de algoritmos. Tanagra es un "proyecto de código abierto", así cada investigador puede acceder al código fuente y agregar sus propios algoritmos, en la medida en que esté de acuerdo y cumpla con la licencia de distribución de software. De manera general el proyecto apunta a cumplir con tres objetivos:

- El objetivo principal del proyecto Tanagra es brindar a los investigadores y estudiantes un software de minería de datos fácil de usar, que se ajuste a las normas actuales del desarrollo de software en este dominio (especialmente en el diseño de su GUI y la forma de usarlo), y permitir analizar datos reales o sintéticos.

- El segundo propósito de Tanagra es proponer a los investigadores una arquitectura que les permita agregar fácilmente sus propios métodos de minería de datos para comparar sus rendimientos. Tanagra actúa más como una plataforma experimental para concentrarse en lo esencial del trabajo, prescindiendo de ellos para hacer frente a la parte desagradable en la programación de este tipo de herramientas: la gestión de datos.

- Dirigido a desarrolladores novatos, consiste en difundir una posible metodología para construir este tipo de software. Se debería aprovechar el acceso gratuito al código fuente para ver cómo se construye este tipo de software, los problemas que deben evitarse, los pasos principales del proyecto, las herramientas y bibliotecas de código que deben utilizar. De esta forma, Tanagra puede considerarse como una herramienta pedagógica para el aprendizaje de técnicas de programación.

El software incluye la versión del algoritmo Apriori para la extracción de reglas de asociación, aunque la implementación no está muy bien definida, el tamaño máximo de la tabla es de 250.000 observaciones, y los cálculos pueden ser deseables en cuanto a tiempo y recursos de memoria.

Rapidminer[244], anteriormente denominado YALE (Yet Another Learning Environment) es un programa informático para el análisis y minería de datos. La versión inicial fue desarrollada por el departamento de inteligencia artificial de la Universidad de Dortmund en 2001. Se distribuye bajo licencia AGPL y está hospedado en SourceForge desde el 2004.
Ésta herramienta todo en uno, presenta cientos de algoritmos de preparación de datos y aprendizaje automático, para respaldar todos los proyectos de minería de datos. Diseñador de flujo de trabajo visual, que ofrece ciencia de datos y aprendizaje automático a todo el equipo de análisis, desde analistas hasta expertos. Ofrece una interfaz intuitiva visual de arrastrar y soltar, con una amplia biblioteca de más de 1500 algoritmos y funciones de aprendizaje automático para construir el mejor modelo para cualquier caso de uso.

El proceso de generación de reglas de asociación se construye mediante operadores visuales, primero se procede con un operador de recuperación de datos, los datos deben ser transformados al tipo requerido por el operador que

ejecuta el algoritmo Fp-Growth, finalmente mediante el operador de creación de reglas de asociación se recibe los itemsets frecuentes desde el operador previo, y se procede a la generación de reglas resultantes.

Rattle[245], es una interfaz gráfica de usuario popular para la minería de datos que utiliza R. Presenta resumenes estadísticos y visuales de datos, transforma los datos para que puedan modelarse fácilmente, crea modelos de aprendizaje automático supervisados y no supervisados a partir de los datos, presenta el rendimiento de los modelos gráficamente, y puntúa nuevos conjuntos de datos para su implementación en producción. Una característica clave, es que todas sus interacciones a través de la interfaz gráfica de usuario se capturan como un script R, que se puede ejecutar fácilmente en R independientemente de la interfaz Rattle. Rattle es software de código abierto gratuito, y el código fuente está disponible en el repositorio git de Bitbucket. Análisis de asociación en Rattle se construye mediante el algoritmo Apriori, el procedimiento
proc{*Apriori*} devuelve un conjunto de reglas de asociación, cada una de las cuales consta de un lado izquierdo, un lado derecho y una tupla de soporte y confianza.

SPMF[246], es una biblioteca de minería de datos de código abierto, especializada en minería de patrones, que ofrece implementaciones de varios algoritmos de minería de datos. Se ha utilizado en cerca de 1000 artículos de investigación para resolver problemas aplicados en una amplia gama de dominios. Sus implementaciones también se utilizan comúnmente como puntos de referencia en artículos de investigación, y también se ha integrado en varios programas de software de análisis de datos.
La versión 2.49 contiene 224 algoritmos, distribuidos entre minería de reglas de asociación, itemsets, secuencia de patrones, minería de reglas secuenciales, predicción de secuencia, minería de patrones periódicos, minería de patrones de alta utilidad, minería sobre series de tiempo, clasificación y agrupamientos.

WebApriori[247], consiste en una aplicación web para minería de reglas de asociación. La aplicación web se llama *WebApriori* y ofrece una interfaz web moderna y receptiva, y un servicio web para las comunidades científicas que trabajan en el campo de ARM. También es apropiado para fines educativos. *WebApriori* implementa un motor Apriori que puede descubrir de manera eficiente las asociaciones ocultas en los datos, y es capaz de procesar diferentes tipos de conjuntos de datos. Parte del proceso implica la eliminación de reglas de asociaciones redundantes. La comunicación asincrónica

entre las capas de front-end, back-end, servicio web y motor Apriori maneja de manera eficiente múltiples solicitudes de usuarios concurrentes.

Kanime[248], KNIME (o Konstanz Information Miner) fue desarrollado originalmente en el departamento de bioinformática y minería de datos de la Universidad de Constanza, Alemania, bajo la supervisión del profesor Michael Berthold. En la actualidad, la empresa KNIME.com GmbH, radicada en Zúrich, Suiza, continúa su desarrollo además de prestar servicios de formación y consultoría.
KNIME está desarrollado sobre la plataforma Eclipse y programado, esencialmente, en java. Está concebido como una herramienta gráfica y dispone de una serie de nodos (que encapsulan distintos tipos de algoritmos) y flechas (que representan flujos de datos) que se despliegan y combinan de manera gráfica e interactiva. Integra diversos componentes para aprendizaje automático y minería de datos a través de su concepto de fraccionamiento de datos (data pipelining) modular. La interfaz gráfica de usuario permite el montaje fácil y rápido de nodos para preprocesamiento de datos (ETL: extracción, transformación, carga), para el análisis de datos, modelado y visualización.
La herramienta incluye el módulo para aprendizaje de reglas basado en el algoritmo Apriori, implementado para este software como un programa de línea de comandos, que debe llamarse en una ventana de comandos o desde un script de shell. Si se prefiere ejecutar el programa de manera independiente al entorno Knime, se puede abrir una ventana de terminal/comando, cambiar al directorio donde se almacena el programa, y luego se escribe el comando de invocación.

Ambientes de desarrollo para ARM

Un entorno, o ambiente de software, es una plataforma para desarrollar aplicaciones de software. proporciona una base sobre la cual los desarrolladores de software pueden crear programas para una plataforma específica. Por ejemplo, un ambiente puede incluir clases y funciones predefinidas que se pueden usar para procesar entradas, administrar dispositivos de hardware e interactuar con el software del sistema. Esto agiliza el proceso de desarrollo, ya que los programadores no necesitan reinventar componentes básicos cada vez que desarrollan una nueva aplicación.
A continuación se describen algunos ambientes conocidos para ARM.

Keel y RKeel[264][263][262], KEEL (Knowledge Extraction based on Evolutionary Learning) es una suite de código abierto implementada en Ja-

va, diseñada para utilizar diversas tareas de descubrimiento de conocimiento. Esta herramienta permite diseñar experimentos basados en flujo de datos, con diferentes conjuntos de datos y algoritmos de inteligencia computacional, con el fin de evaluar el desempeño de los algoritmos. Contiene una amplia variedad de algoritmos de conocimiento clásico, técnicas de preprocesamiento, algoritmos de aprendizaje basados en inteligencia computacional, modelos híbridos, metodologías estadísticas para contrastar experimentos, etc. Con todas estas características, el usuario puede realizar un análisis completo de nuevas propuestas de inteligencia computacional para compararlas con las existentes.

Existe fusión en muchos casos entre herramientas de desarrollo, en especial aquellas que son de código abierto. En este sentido, un trabajo relevante es la fusión entre KEEL y R , esto proporciona una interfaz para usar las características de KEEL en el código fuente de R, creando el paquete RKEEL [264] y poniéndolo disponible a través del repositorio CRAN. Por lo tanto, con RKEEL se podría preprocesar los datos, usar algoritmos de KEEL, obtener los resultados y dibujar un gráfico con ellos, todo desde el código R. Además, el manejo con algoritmos KEEL en R hace que el proceso del experimento sea mucho más completo, lo que permite concatenar experimentos de otros paquetes o utilidades de R, lo cual es una gran ventaja.

QuantMiner[261], QuantMiner es una herramienta para la minería de reglas de asociación cuantitativa que toma en consideración atributos numéricos en el proceso de minería sin un agrupamiento/discretización previo de los datos. Utiliza dos innovadoras técnicas relacionadas con el uso de algoritmos evolutivos para la minería de reglas cuantitativas publicada en [81]. QuantMiner es software libre, y puede redistribuirse y/o modificarse según los términos de la Licencia Pública General GNU V3 publicada por la Free Software Foundation.

uARMSolver[249], es un entorno de desarrollo escrito completamente en C++ y se ejecuta en todas las plataformas. Permite a los usuarios preprocesar los datos en una base de datos transaccional, hacer discretización de datos, buscar reglas de asociación y visualización de las mejores reglas encontradas utilizando herramientas externas. A diferencia de los paquetes de software o entornos existentes, esto también admite tipos de atributos numéricos y de valor real además de los categóricos.

En uARMSolver la minería de las reglas de asociación se define como una optimización y se resuelve utilizando los algoritmos inspirados en la na-

turaleza que se pueden incorporar fácilmente. Debido a que los algoritmos normalmente descubren una gran cantidad de reglas de asociación, la herramienta permite una inclusión modular de asistentes visuales para extraer el conocimiento oculto en los datos y visualizarlos utilizando herramientas externas.

aRules[250], es un paquete para R, que proporciona la infraestructura para representar, manipular y analizar patrones y datos de transacciones utilizando conjuntos de elementos frecuentes y reglas de asociación. También proporciona una amplia gama de medidas de interés y algoritmos de minería que incluyen interfaces y el código de las eficientes implementaciones en C de Borgelt de los algoritmos de minería de asociación Apriori y Eclat.
Tanto apriori como eclat reciben como argumento un objeto de tipo "Transaction" con los datos de las transacciones, un argumento "parameter" que determina las características de los itemsets o reglas generadas (por ejemplo, el soporte mínimo) y un argumento control que determina el comportamiento del algoritmo (por ejemplo, ordenación de los resultados). En la función apriori() también se incluye el argumento "aparence" que impone restricciones sobre las reglas generadas, por ejemplo, crear solo reglas que contengan un determinado item. El resultado de ambas funciones es un objeto de tipo "association" que puede ser manipulado con toda una serie de funciones que ofrece el paquete.

FrIDA[251], Es un proyecto que se viene desarrollando por más de dos décadas, en el ámbito de minería de datos, inicialmente consistió en un conjunto de programas basados en línea de comando en C++, no muy amigable para usuarios inexpertos. Sin embargo, esto no impidió que se haya hecho popular, continuamente se agregaron interfaces individuales basados en Java para las diferentes divisiones de técnicas como: árboles de decisión, regresiones, reglas de asociación y conglomerados probabilísticos y difusos. Sistemáticamente se ha intentado combinar estas interfaces de usuario individuales en una sola caja de herramientas, para hacerlas más accesibles, y también para transmitir a un usuario, que puede haber estado usando uno de los programas, la gran variedad de métodos disponibles. Como consecuencia el programa FrIDA combina todas las interfaces de usuario individuales que se han desarrollado hasta ahora en una arquitectura uniforme y consistente.

Capítulo 3

Adaptación de VMO para optimización de intervalos

3.1. Introducción

En el capítulo anterior se presentó los diferentes enfoques para resolver y clasificar los problemas de minería de reglas de asociación, partiendo de las características de los conjuntos de datos. Aunque en algoritmos clásicos se han podido incorporar variables numéricas para ARM aplicando métodos de discretización. Estos han presentado limitaciones en cuanto a calidad y diversidad. En la actualidad y gracias a la evolución en computación, los investigadores han probado diversidad de algoritmos evolutivos y metaheurísticas exigentes en recursos desde una óptica de problema de optimización.

Dado que estas técnicas son mucho más apegados a la definición y evolución de una población, por un lado presentan soluciones en tiempos más cortos, y por otro lado pueden resultar en soluciones aproximadas. Esta posible tendencia a presentar resultados aproximados, unido a quedar atrapado en zonas locales, supone una desventaja y un gran coste en ajustar los operadores de exploración.

Partiendo de estas limitaciones, en este capítulo se propone la inserción de la lógica de Optimización por Mallas Variables definido en el algoritmo VMO, que permita encontrar los intervalos mejor puntuados para los atributos numéricos que componen la regla. Esta nueva definición en VMO permite, a su vez, conseguir una representación de la regla de asociación en la estructura principal del algoritmo. Siendo esta representación la base fundamental para construir una población de reglas acorde al concepto de nodos de la malla, y

consiguiendo que cada nodo sea el objeto de evaluación y convergencia hacia soluciones de calidad.

En concreto, se propone un conjunto de modificaciones al algoritmo original, que permitirán la codificación de reglas de asociación compuestos de atributos numéricos. El objetivo es que la definición de intervalos dentro de la regla, alcancen la mayor calidad posible en relación a una función de ajuste. Así se consigue un enfoque similar a la optimización de funciones continuas definidas en VMO y ajustado a las fases del entorno QUANTMINER.

El proceso metodológico incluye una primera etapa para el desarrollo de la versión QARM_VMO que incluye (análisis, diseño e implementación), una segunda etapa de experimentación que abarca (selección de conjuntos de datos, algoritmos de comparación, diseño de los experimentos, generación de visualizaciones), finalmente la etapa de resultados y contrastes involucra (pruebas de significancia, Discusión de los resultados). La Figura 3.1 muestra una abstracción del proceso general.

Figura 3.1: Etapas generales del proyecto.

3.2. El ambiente de desarrollo

Los estudios preliminares sobre el tema han sido fundamentales en la construcción de este proceso. La estructuración de una base teórica en relación al estado del arte ha permitido la identificación de herramientas, recursos y estrategias para implementar este algoritmo.

3.2.1. El entorno de QUANTMINER

QUANTMINER es un software gratuito, puede redistribuirse y/o modificarse según los términos de la Licencia Pública General GNU V3 publicada por la Fundación de Software Libre. En la Figura 3.2 se muestran una representación de los nombres de paquetes que dispone este entorno.

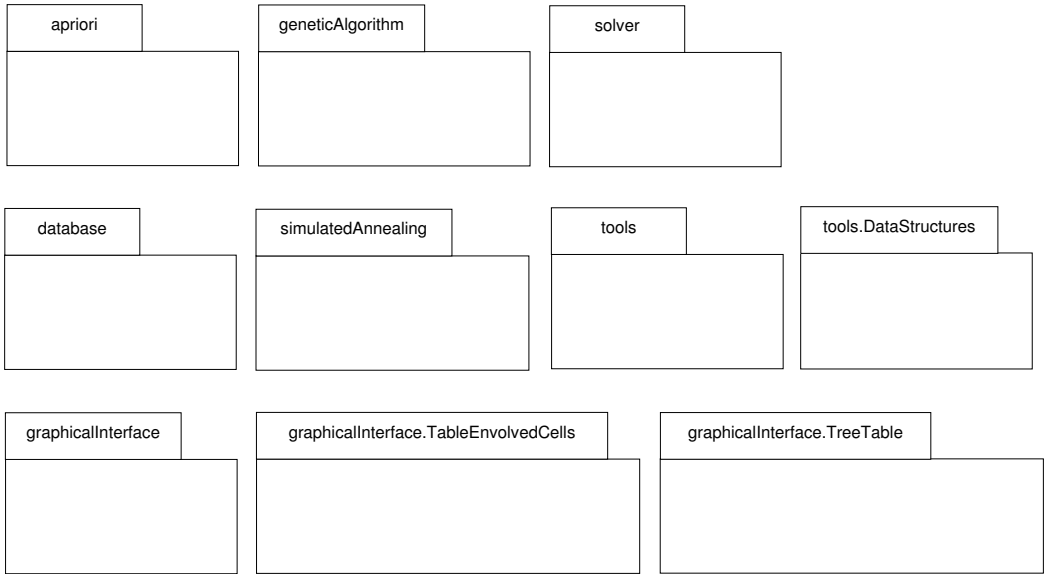

Figura 3.2: Representación de paquetes QUANTMINER.

- **Paquete *apriori***, QUANTMINER implementa el algoritmo apriori dentro del paquete del mismo nombre, este es utilizado para la extracción de reglas de asociación con atributos cualitativos y para el descubrimiento de conjuntos de ítems frecuentes (frequent itemsets) como parte del preprocesamiento de la minería de reglas de asociación. Contiene las clases que define la regla de asociación, tanto para las reglas puramente cuantitativas así como categóricas y combinadas.

- **Paquete *database***, se compone de cuatro clases que se encargan de la gestión de los datos, tales como: lectura y escritura de los archivos en formato .csv o .dbf en medios de almacenamiento físico; además organiza los conjuntos de datos en las estructuras de datos internos (Memoria RAM).

- **Paquete *geneticAlgorithm***, la implementación del algoritmo genético que se distribuye con esta versión de QUANTMINER, contiene tres clases que dividen el proceso en la definición de parámetros, control de optimización y el algoritmo.

- **Paquete solver**, el núcleo de QUANTMINER está definido en este paquete, tiene seis clases y se encarga principalmente de la generación, gestión y control de la población de reglas desde un esquema de prueba; además organiza y controla la experimentación para el algoritmo evolutivo.

- **Paquete *simulatedAnnealing***, implementación del algoritmo recocido simulado (SA), que también viene con la versión de QUANTMINER, similar al paquete geneticAlgorithm tiene también tres clases, pero su implementación se orienta a la lógica del SA.

- ***graphicalInterface***, ***graphicalInterface.TableEvolvedCells*** y ***graphicalInterface.TreeTable***, son paquetes que contienen clases para organización y control de interfaces gráficas propias del software, estas definen objetos que se asocian con las estructuras que almacena resultados y datos para su visualización o impresión.

- ***tools*** y ***tools.dataStructures***, constituido por clases que definen funciones, variables y constantes útiles para el software en general.

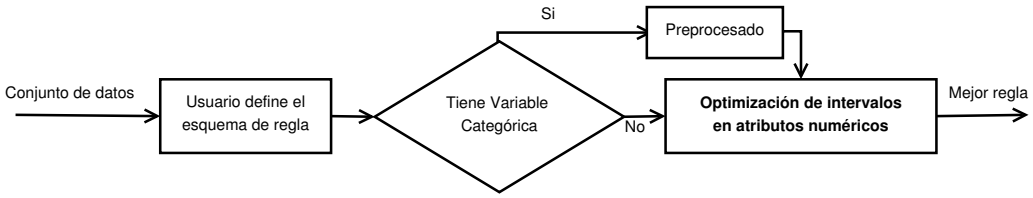

Figura 3.3: Representación general del proceso de extracción de reglas de asociación basado en el entorno QUANTMINER.

Desde un enfoque funcional QUANTMINER se divide en tres fases (Figura 3.3) . La primera de ellas está asociado a la definición de un esquema de regla por parte del usuario, aquí se utiliza un módulo para selección y carga de datos. Este esquema unido a otros parámetros se utilizan como base para la generación de otras plantillas, las que son derivadas desde la combinación de los atributos. En la segunda fase, el preprocesamiento se divide en una función para tratar la variables cualitativas (categóricas), y otra para la generación de una lista de plantillas de regla, que se utilizá en la siguiente etapa. La optimización es la tercera etapa, abarca el proceso para encontrar la mejor regla desde cada esquema definido en la fase dos. A continuación se describe cada fase de una forma más detallada.

- **Definición del esquema de la regla**.
  Esta etapa tiene como objetivo principal, la especificación de los atributos que compondrán tanto el Lado Izquierdo de la Regla (LHS), como el Lado Derecho de la Regla (RHS). Esto permite la generación de otros posibles esquemas, derivados de la combinación y selección de los atributos, de manera que se pueda obtener esquemas personalizados. Se incluye parámetros que especifican la colocación de atributos tanto en RHS como LHS, o a su vez, en los dos extremos de la regla. Para ello, se propone el uso de un archivo de configuración inicial, este es editado por el usuario, y a partir del cual se obtienen los parámetros junto con la codificación, para formar las plantillas derivadas. Los esquemas contienen dos o más atributos numéricos, y además pueden incluir atributos categóricos, si se quiere dar un enfoque hacia reglas de clasificación.

  Un archivo de extensión *.prf* es el perfil de configuración de una experimentación, que es administrado por el entorno de trabajo (framework) QUANTMINER. Este ambiente contiene los módulos para la creación, almacenamiento y recuperación de datos referentes a parámetros de la experimentación para los algoritmos que dispone. La Tabla 3.1 detalla la información concerniente al contenido del archivo de configuración.

Categoría	Descripción
Datos	Nombre del archivo de datos asociado al perfil creado, Información acerca del archivo de datos, tales como: atributos (Cantidad, tipo de dato, nombre, atributos seleccionados y no seleccionados)
Plantilla	Información acerca de pre-extracción de reglas (atributos definidos en esquema de regla, distinción entre atributo categórico y no categórico)
Algoritmo	Identificador de algoritmo seleccionado, soporte mínimo, confianza mínimo, en caso del genético registra (población, número de generaciones, porcentaje de cruce y mutación)

Tabla 3.1: Parámetros del archivo de perfil

- **Preprocesado**
  Una vez que el esquema de regla ha sido definido, existen dos procesos para preparar los datos que se utilizan como entrada al algoritmo. Por un lado está la resolución del esquema de regla general, este da lugar a una explosión combinatoria de atributos, y configuración de longitudes diversas. Así se genera la lista de plantillas de regla para ser probadas.

Por otra parte, si la plantilla contiene atributos categóricos, un proceso basado en el algoritmo *Apriori* es ejecutado para encontrar conjuntos de ítems frecuentes, y de esta manera formar plantillas de regla que contengan tipos categóricos y numéricos. Una lista de reglas son almacenadas por el proceso explicado en el párrafo anterior. Éstas reglas no tienen establecidos valores en los intervalos de los atributos numéricos, los umbrales inferior y superior en los intervalos de cada atributo son obtenidos mediante el algoritmo de optimización, cuando define la población inicial.

- **Optimización**
  Tres algoritmos vienen definidos en el entorno de QUANTMINER para búsqueda de reglas de asociación, esto son: *Apriori*, *Optimización con algoritmo genético* y *optimización con recocido simulado.*
  El proceso de optimización aplica a cada regla de la lista de prueba, así se genera una población inicial de individuos (Reglas de asociación con intervalos aleatorios) y el algoritmo de optimización trabaja con la Función Objetivo (FO) construida. El resultado por cada esquema definido es un conjunto de reglas(optimizada) similares en su estructura, y que cumplen con las especificaciones de soporte y confianza mínimos preestablecidos. Se selecciona la de mejor ajuste según la FO, y se agrega a la lista de soluciones. Cuando se ha cubierto todas las reglas de la lista de prueba, un conjunto solución con diversidad de reglas es presentado.

3.2.2. El algoritmo VMO para minería de reglas de asociación cuantitativas (QARM_VMO)

Comprende la etapa de desarrollo para la adaptación del algoritmo VMO, el objetivo es alcanzar una versión, que permita realizar minería de reglas de asociación numéricas. En general, el proceso se ajusta parcialmente a la construcción de un software, por tal razón se han establecido fases importantes del ciclo de vida del software (Análisis, Diseño e Implementación).

QARM_VMO de forma alternativa puede ser parte del banco de algoritmos del ambiente de desarrollo QUANTMINER, por lo que se reutiliza las funciones que puedan ser útiles de este entorno de trabajo. Dos enfoques de requerimientos se toman en cuenta, la arquitectura del algoritmo y lo referente a especificaciones sobre el problema de QARM, así para el primer caso se tiene:

Especificaciones generales de VMO

- VMO consiste de un conjunto de nodos que se distribuyen en una malla, éstos representan a la población.
- La versión de VMO que se dispone optimiza parámetros de una función n-dimensional (enfoque para problemas de optimización continua)
- El nodo es la estructura de almacenamiento que utiliza VMO (punto de la malla), éste contiene los parámetros de la función.
- Las acciones inteligentes que se ejecutan en los nodos actualizan los valores de los parámetros y se evalúan con la función.
- La condición de parada lo establece el número de evaluaciones configurado a priori.

Especificaciones generales del framework QUANTMINER (Reutilización)

- Definición de una plantilla de regla
- Generación de conjuntos ítems-frecuentes desde la plantilla de regla (reglas candidatas)
- Generación de la población de reglas (por cada regla candidato)
- Interfaces visuales

Especificaciones de QARM

- Estructuración de la regla
- Estructuración del nodo
- Función de evaluación (monoobjetivo)
- Aleatoriedad en la generación de reglas individuales

En resumen los recursos basados en las especificaciones del algoritmo se tiene lo siguiente:

1. Algoritmo VMO para optimización de parámetros en funciones multimodales
2. Framework QUANTMINER que incluye un algoritmo genético y recocido simulado.

3. Problemática de Minería de Reglas de Asociación
4. Entorno de programación Java (Netbeans 8.0)
5. Herramienta estadística de apoyo (R Project)

La versión del algoritmo VMO corresponde a la publicada en [19], está codificado en Java, a continuación en la Figura 3.4, se muestra una representación aproximada de la arquitectura mediante el uso de un diagrama de clases.

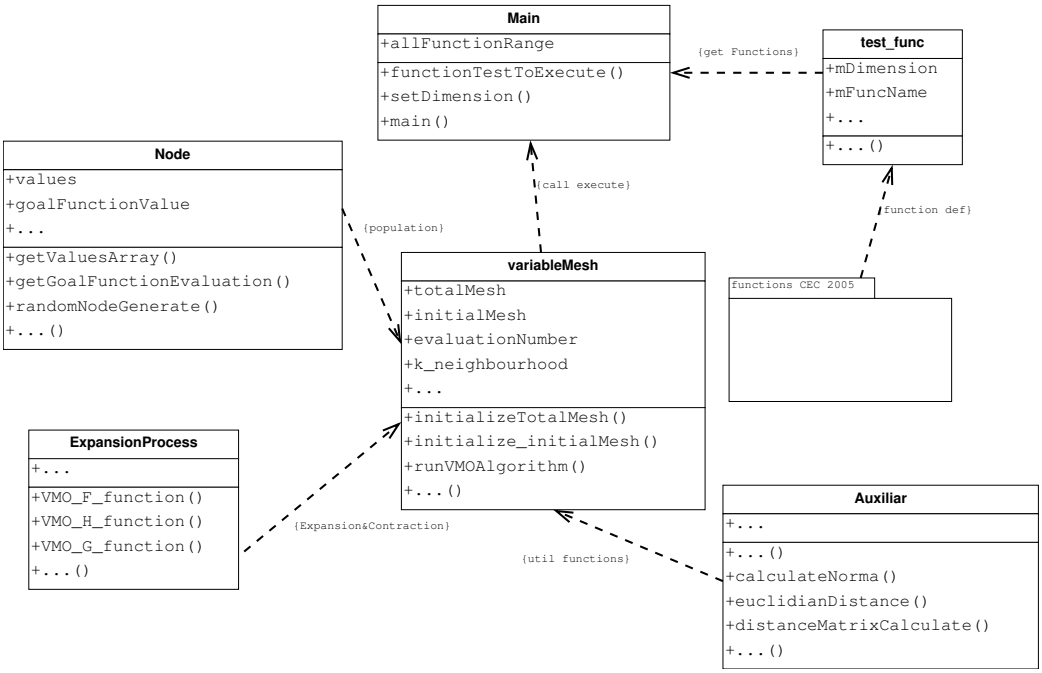

Figura 3.4: Vista resumida del diagrama de clases VMO.

A continuación se describe de manera general cada una de sus clases

- **Node**. Ésta clase permite la construcción de cada nodo que forma parte de la población, contiene los datos de una posible solución a la función que está siendo optimizada. Así define propiedades y métodos para la gestión de valores que toma cada variable que interviene, cálculo de la FO y operaciones auxiliares.
- **Main**. Esta función recibe una cadena de caracteres, contiene una expresión de las funciones que se quieren minimizar. Los operadores especiales para minimizar más de una función son "," y el "-". el primer operador define las funciones a cargar de forma individual. Ejemplo "f6,f12,f17,f25" y el otro operador define un intervalo de funciones.

Ejemplo “f6-f12”, donde se ordena cargar la funciones desde la “f6” hasta la “f12”. Si se quiere ejecutar todas las funciones se puede poner la palabra “All”. También se puede mezclar ambos operadores, por ejemplo “f6,f8-f12,f20-f25”.

- **variableMesh**. Esta clase es instanciada desde “main” para ejecutar el algoritmo. Contiene las estructuras para gestionar a la población de nodos basados en malla inicial y total. Los parámetros se inicializan aquí a través de la definición de propiedades y métodos, que permiten receptar valores especificados por el usuario y/o establecidos por defecto. En esta clase se encuentra implementado mediante una función el algoritmo VMO, también se encuentran varios atributos y operaciones adicionales que cumplen roles utilitarios en el proceso de ejecución.

- **ExpansionProcess**. Define las principales funciones de expansión propias del algoritmo VMO, para generar nuevos nodos a partir de los existentes en la malla inicial, con respecto a su extremo local en caso de tener, también para generar nodos a partir de los existentes en la malla inicial, respecto solamente al extremo global, y generación de los nodos faltantes a partir de otros más y menos externos de la malla inicial.

- **Auxiliar**. Contiene un conjunto de funciones para cálculo de algunas medidas utilizadas en el algoritmo principal.

- **test_func**. Implementa interfaces para la utilización de las funciones CEC’05 [252]. Define 25 funciones de prueba 5 unimodales y 20 multimodales

Partiendo del algoritmo VMO predefinido, se define la arquitectura y el acoplamiento de QARM_VMO dentro del entorno QUANTMINER. En la Figura 3.5 se muestra una representación gráfica del proceso.

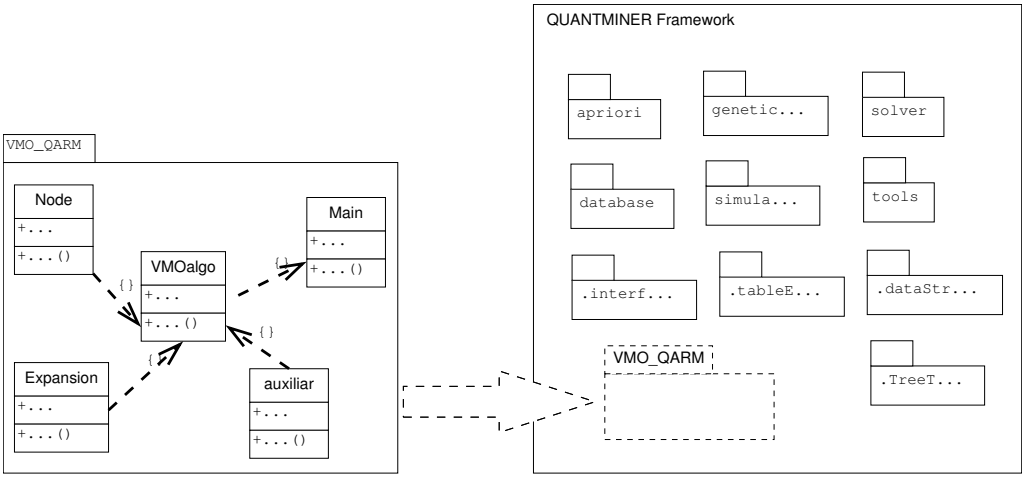

Figura 3.5: Representación de inserción VMO en QUANTMINER.

El algoritmo de Optimización por Mallas Variables (VMO) es introducido como una metaheurística poblacional en el año 2012, y se describe en el Algoritmo 1. En esta metaheurística la población esta representado por una malla de p nodos $S = n_1, n_2, ..., n_p$, cada nodo está compuesto de un arreglo multidimensional $n_i = (v_1^i, v_2^i, ..., v_m^i)$ y representan a una posible solución al problema. El método tiene dos mecanismos básicos de acción, *expansión* y *contracción*.

En las líneas 1 y 2, una población inicial de nodos es creado. Cada nodo tiene un vector de valores para las variables de la función a ser optimizada, y cada una de las variables que forman parte de cada nodo son inicializadas. En la línea 3, el nodo óptimo del conjunto global n_g es obtenido y el proceso de expansión se ejecuta. El proceso de expansión es definido en las líneas 5–17. En esta fase la población es explorada, y los mejores nodos son seleccionados en los siguientes tres pasos que se describen abajo.

Paso 1 Las líneas 5–10 definen esta fase. Aquí se determina el extremo local n_i^* alrededor de los k vecinos más cercanos de cada nodo n_i, y solo cuando n_i^* es mejor, entonces un nuevo nodo es generado entre n_i y n_i^*; esto es definido en la función $n_{new}(i) = f(n_i, n_i^*, Pr(n_i, n_i^*))$. La función f para la generación de nuevos nodos desde cada nodo que no corresponde a un extremo local y al mejor vecino es definido por la Ecuación 3.3.

$$P_r(n_i, n_i^*) = \frac{1}{1 + |fitness(n_i) - fitness(n_i^*)|} \tag{3.1}$$

La función f establece el procedimiento para la generación del nuevo nodo y podría estar sujeto a variaciones dentro del problema. $Pr(n_x, n_y)$

es un factor de relación entre el valor de calidad del nodo actual con su extremo, y es calculado con la Ecuación 3.1.

$$D_{euclidean}(n_1, n_2) = \sqrt{\sum_{j=1}^{M}(v_j^1 - v_j^2)^2} \tag{3.2}$$

Algorithm 1 VMO algorithm original [19]

Require: C,k,test_function
Ensure: best_node
1: Create initial mesh (N_0) from test_function
2: Evaluate the nodes of initial mesh
3: select best global node n_g
4: **repeat**
5: **foreach:** $n \in N_0$ **do**
6: find its closets k nodes using euclidean distance by Equation (3.2)
7: select the best node local n_i^*
8: **if** n_i^* is better than n_i **then**
9: calculate near factor Pr between n_i and n_i^* by Equation (3.1)
10: create new node $n_{new} = f(n_i, n_i^*, Pr)$
11: **foreach:** $n \in N_0$ **do**
12: calculate near factor Pr between n_i and n_g by Equation (3.1)
13: create new node $n_{new} = g(n_i, n_g, Pr)$
14: Create $n_{frontiers}$ frontier nodes using $n_{new} = h(n_i, w)$
15: Fill nodes in the total mesh from $n_{frontiers}$
16: sort the nodes by fitness quality in total mesh
17: Apply clearing operator
18: build mesh for next iteration of elitist way
19: **until** C iterations
20: best_node is obtained from mesh(initial index)
21: **return** best_node

El cálculo de los vecinos más cercanos a cada nodo de la malla se realiza por medio de la distancia euclidiana, definido en la Ecuación 3.2.

$$n_{new}(i) = \begin{cases} vm_i & \text{if } |vm_i - n_i^*| > \delta \text{ and } \mathcal{U}[0,1] \leq P_r(n_i, n_i^*) \\ n_i^* + \mathcal{U}[-\delta, \delta] & \text{if } |vm_i - n_i^*| \leq \delta \\ \mathcal{U}[vm_i, n_i^*] & \text{othercase} \end{cases} \tag{3.3}$$

donde vm_i representa al valor medio entre el nodo actual y su extremo local para el i-th dimensión, y es calculado por la Ecuación 3.4.

$$vm_i = \frac{n_i + n_i^*}{2} \tag{3.4}$$

Adicional a esto, $\mathcal{U}[x, y]$ denota un valor aleatorio (uniforme) en el intervalo$[x, y]$, y δ es un límite de distancia adaptativo y se calcula de acuerdo con la Ecuación 3.5.

$$\delta = \begin{cases} \frac{range(a_j,b_j)}{4} & \text{if } c < 15\,\%\ \text{C} \\ \frac{range(a_j,b_j)}{8} & \text{if } 15\,\% \leq c < 30\,\%\ \text{C} \\ \frac{range(a_j,b_j)}{16} & \text{if } 30\,\% \leq c < 60\,\%\ \text{C} \\ \frac{range(a_j,b_j)}{50} & \text{if } 60\,\% \leq c < 80\,\%\ \text{C} \\ \frac{range(a_j,b_j)}{100} & \text{if } c \geq 80\,\%\ \text{C} \end{cases} \tag{3.5}$$

donde C se refiere al máximo valor de evaluaciones para la función objetivo, dependiendo del valor porcentual calculado en relación al total de evaluaciones, se definen valores de distancia que representan partes del rango permitido. Además, $rango(a_i, b_i)$ se define como la amplitud del dominio (a_i, b_i) de cada componente.

Paso 2 Las líneas 11–13 tienen como objetivo acelerar la convergencia. Se encuentra el nodo con mejor calidad en la malla actual; de esta manera, el extremo global n_g es definido, y luego se crean nuevos nodos con la función $n_{new}(i) = g(n_i, n_g, Pr(n_i, n_g))$ uno por cada n_i en dirección de n_g. Para esto, el valor de P_r calculado por la Ecuación (3.1) es usado, sustituyendo n_i^* por n_g. Entonces la función de generación g es definido en la Ecuación 3.6.

$$n_{new}(i) = \begin{cases} vm_i & \text{if } \mathcal{U}[0,1] \leq P_r(n_i, n_g) \\ \mathcal{U}[vm_i, n_g^i] & \text{othercase} \end{cases} \tag{3.6}$$

donde el valor medio vm entre el nodo actual y el extremo global se calcula usando la Ecuación 3.1.

Paso 3 Las líneas 14 y 15 definen este paso. Se completa el número total de nodos que debe tener la malla, partiendo de los nodos limítrofes. Para este caso de estudio es necesario la detección de este tipo de nodos, por lo que se utiliza un criterio denominado *norma* para cada uno, como se define en la Ecuación 3.7. Los nuevos nodos son creados mediante la función $n_{new} = h(n_i, w)$, completando así el proceso de expansión. El número de nodos creados en cada uno de estos casos no tiene un valor exacto. Sin embargo, es controlado por el parámetro que representa el número de nodos especificado para expansión.

$$\|n\| = \sqrt{\sum_{i=1}^{n} (n_i)^2} \tag{3.7}$$

Los nodos con los valores de *norma* más altos se ubican en el límite de la malla inicial, y los que tienen valores más bajos están muy cerca del origen (puntos más internos). La función h permite la generación de nuevos nodos en la dirección de los límites usando la Ecuación 3.8 para los nodos más externos y usando la Ecuación 3.9 para los nodos internos.

$$n_{new}(i) = \begin{cases} n_i^* + w & \text{if } n_i^* > 0 \\ n_i^* - w & \text{if } n_i^* < 0 \end{cases} \tag{3.8}$$

$$n_{new}(i) = \begin{cases} |n_i^* + w| & \text{if } n_i^* > 0 \\ |n_i^* - w| & \text{if } n_i^* < 0 \end{cases} \tag{3.9}$$

La Ecuación 3.10 se utiliza en la tercera forma de exploración. Se parte de los nodos límites y calcula un movimiento decreciente controlado; por lo tanto, w_j^0 y w_j^1 son los desplazamientos inicial y final, respectivamente, de manera que $w_j^0 > w_j^1$. Los parámetros C y c se utilizan en el cálculo y representan el número máximo de iteraciones y el valor de iteración actual, respectivamente.

$$w_j = (w_j^0 - w_j^1) \cdot \frac{C - c}{C} + w_j^1 \tag{3.10}$$

El proceso de contracción se describe en las líneas 16–18 y selecciona los nodos de malla que serán parte de la próxima iteración. Tres tareas principales componen este proceso.

1. Ordena los nodos de la malla total según la calidad. Los nodos con mejor calidad tienen mayor probabilidad de formar parte de la nueva población.

2. Aplicar el operador de limpieza adaptativa, es una estrategia para orientar el proceso hacia exploraciones más generales, y también reducir su frecuencia para enfocarse en áreas más pequeñas.

3. Construye una nueva malla inicial de forma elitista.

3.3. El problema de Minería de Reglas de Asociación Cuantitativas

El procedimiento de adaptación realizado en esta investigación, consiste en realizar ajustes en el esquema de representación de los elementos de la

población. Esto es en la acción de los operadores y en la configuración de los parámetros del algoritmo VMO. La población está representada por un conjunto de nodos, donde cada nodo es una estructura que define a un individuo, y se construye a partir de un esquema de reglas de asociación numérica.

3.3.1. Esquema de Regla

El esquema de regla es una plantilla a partir de la cual se generan los individuos de la población, similar a una definición de una clase y sus objetos instanciados. Esto implica que el procedimiento de optimización se aplica a cada población derivada de su respectiva plantilla. Para su representación es necesario partir de la siguiente definición. Sea R una regla, de modo que R se describa con las condiciones c_1 y c_2. Entonces tenemos la denominación $c_1 \rightarrow c_2$, donde c_1 es la condición y c_2 es la conclusión. Un elemento de la regla se puede considerar como una expresión $A = v$ si A es una variable de tipo categórico. Mientras que, si es numérico, tenemos $A = [l, u]$, donde l y u son los umbrales inferior y superior respectivamente.

La Tabla 3.2 describe la estructura lógica de los elementos de una regla, se marca la diferencia tanto al extremo izquierdo como al derecho. Esta primera partición corresponde al antecedente y al consecuente en el orden indicado. Además, representa los atributos, que son parte de cada extremo de la regla. El esquema de una regla se puede construir de forma aleatoria, mediante la

izquierdo				derecho			
Item L1	Item L2	...	Item Ln	Item R1	Item R2	...	Item Rn

Tabla 3.2: Esquema lógico de regla

combinación controlada de cada uno de los atributos que la componen, o a partir de las especificaciones del usuario. Para el último caso es necesario una interfaz que permita el ingreso de estos parámetros. Un ejemplo, en el que se obtienen tres esquemas de reglas para clasificación se detalla a continuación, todo esto en base a una definición que puede establecer el usuario.

Dado a_1 y a_2 que representan a dos atributos numéricos y b_1 un atributo categórico, con l_x y h_x los umbrales inferior y superior del atributo x se llaman respectivamente, por lo que se puede establecer un primer esquema.

$$Regla = a_1 \in [l_1, h_1] \wedge a_2 \in [l_2, h_2] \rightarrow b_1 = C_1$$

Cada combinación de atributos permite obtener una nueva plantilla que se ingresará en el proceso de optimización. A continuación se muestra un ejemplo de posibles derivaciones.

$$Regla_1 = a_1 \in [l_1, h_1] \rightarrow b_1 = C_1$$
$$Regla_2 = a_2 \in [l_2, h_2] \rightarrow b1 = C_1$$
$$Regla_3 = a_1 \in [l_1, h_1] \ \wedge \ a_2 \in [l_2, h_2] \rightarrow b_1 = C_1$$

El proceso de optimización basado en algoritmos de población, parte de la definición de una población inicial de reglas, el objeto de mejora son los intervalos de cada atributo numérico. Como en toda convergencia, se requiere una función de evaluación de la calidad que cubra los criterios más interesantes.

3.3.2. Función de evaluación

Las personas tienen diferentes percepciones sobre la calidad de las cosas y se basan en criterios únicos o múltiples para obtener una estimación que cumpla con sus requisitos. Encontrar una relación que cumpla con la mayoría de estos requisitos es un desafío de investigación. Sin embargo, la aproximación ha sido posible a través de funciones que han incluido los indicadores de calidad comunes para el problema. En general, la importancia de una regla de asociación es relativa a los requerimientos del usuario, por lo que existen indicadores objetivos (basados en probabilidad y estructura), subjetivos (representaciones novedosas e inesperadas) y semánticos (se ajustan a la información que pueden brindar los patrones) [253].

En un estudio realizado por [254], se analizaron nueve métricas aplicables a las reglas de asociación cuantitativas. Los resultados llevaron a afirmar que las medidas *soporte*, *confianza*, *ganancia* y *precisión* son el mejor resumen de todas las estudiadas.

En el capítulo de literatura se hizo referencia a *soporte* y *confianza*. Estas son las medidas objetivas más comunes en las reglas de asociación. Han tenido una amplia aplicación en los problemas relacionados con la minería de reglas de asociación para bases de datos transaccionales. Autores de todo el mundo han adoptado su definición, y las concepciones iniciales se encuentran en [255].

La métrica *Ganancia* [256], se obtiene de la diferencia entre las medidas de *confianza de la regla* y el *soporte de RHS*. La Ecuación 3.11 muestra la relación con las otras medidas para su cálculo. Este criterio también se conoce como *valor agregado* o *variación de soporte*.

$$gain(A \rightarrow B) = conf(A \rightarrow B) - supp(B) \tag{3.11}$$

La variante propuesta por [257] con respecto a esta métrica, para estimar la calidad en reglas de asociación cuantitativas, se define en la Ecuación 3.12. La inserción del parámetro *minConf* en la función transfiere el nivel de confianza deseado por el usuario a la evaluación de la calidad.

$$gain(A \rightarrow B) = supp(A \wedge B) - minConf * supp(A) \tag{3.12}$$

En la investigación desarrollada por [270], se definió una función de evaluación considerando los criterios de cobertura, superposición, amplitud y cantidad de atributos. Los tres últimos indicadores están controlados por un peso asignado a cada uno, y es definido por el usuario. La cobertura es una medida similar al apoyo y la confianza. La Ecuación 3.13 muestra la expresión que relaciona estos indicadores con sus pesos. La función ha tenido una aplicación exitosa en la optimización de reglas de asociación cuantitativas. Sin embargo, la reducción de los parámetros introducidos por el usuario es otro objetivo de las funciones de evaluación, que hay que alcanzar.

$$f(i) = cov - (mark * \omega) - (ampl * \psi) + (nAtr * \mu) \tag{3.13}$$

La cobertura de la regla es uno de los criterios considerados en esta propuesta, y abarca las definiciones de *soporte* y *confianza*. Sin embargo, este enfoque es insuficiente en el contexto de las reglas de asociación cuantitativas, y es necesario agregar una estrategia, para explorar las áreas densas en el espacio de la solución. La relación entre la amplitud del rango y la amplitud del dominio de la variable, cuantificada en términos de poblaciones se aproxima a las áreas de mayor densidad. La función de utilidad se define en la Ecuación 3.14, donde $ratio_a$ es la proporción de densidad del intervalo en un atributo.

$$util_{ampl} = \prod_{a \in A_{num}} (1 - ratio_a) \tag{3.14}$$

Los parámetros mínimos de soporte y confianza son coeficientes que actúan sobre la cobertura, por lo que la Ecuación 3.15 compensa la cobertura con el valor de la utilidad de las amplitudes.

$$Fitness(R) = cov(R) \times util_{ampl} \tag{3.15}$$

3.3.3. Estructura del nodo

Existen dos métodos para codificar las reglas, ambos se han inspirado en la representación de cromosomas con algoritmos genéticos. El primer método, llamado *Pittsburgh* se caracteriza por la inclusión de un conjunto de reglas en un solo nodo, y se considera apropiado para problemas de clasificación. *Michigan* es la otra alternativa y se caracteriza por representar una sola regla en cada nodo. A diferencia del primero, es útil tanto en problemas de clasificación como en la identificación de patrones.

Esta investigación utiliza el método *Michigan* porque el enfoque es la clasificación e identificación de reglas cuantitativas. Con este enfoque, cada regla se representa en un nodo compuesto por más de un vector de dimensión m, el cual, entre otras propiedades, almacena tanto el umbral inferior como el superior del intervalo correspondiente al atributo i-th de la regla. La longitud es equivalente a los atributos numéricos n de la plantilla de reglas.
Otras propiedades importantes que componen el nodo son el *valor-aptitud*, que es el valor de calidad obtenido de la función objetivo para el estado actual, la ubicación de los intervalos en el conjunto de datos, el soporte de las condiciones formadas con atributos cualitativos que forman parte de la regla. La Figura 3.6 ilustra la estructura de un nodo, donde n es el número de atri-

Atributo 1	Atributo 2	...	Atributo n
UB1	UB2	...	UBn
LB1	LB2	...	LBn

Figura 3.6: Representación de los nodos en QARM_VMO

butos presentes en la plantilla de reglas que se está evaluando, por ejemplo:

Dado $r1 = (A \in [lb_1, ub_1] \wedge B \in [lb_2, ub_2] \rightarrow C = \text{"}red\text{"}$, en $r1$ hay dos atributos numéricos en el lado izquierdo y un atributo cualitativo en el lado derecho, para más detalles, el esquema obtenido es algo similar a: $(A \in [0{,}0; 0{,}0] \wedge B \in [0{,}0; 0{,}0] \rightarrow C = \text{"}red\text{"})$; observe que los valores de umbral no están definidos, ellos se establecen en la evaluación de la regla.

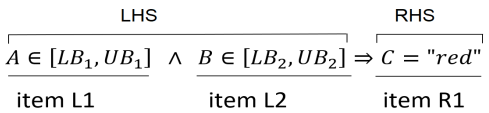

Figura 3.7: Ejemplo lógico de regla

El nodo almacena los intervalos de cada uno de los atributos numéricos de la regla, para ello, abstracciones de varios elementos son necesarios, La Figura 3.7 describe de forma general los extremos y las partes individuales de la regla para una representación lógica.

3.3.4. Problema de optimización

Sea I las observaciones del conjunto de datos, una población S tal que $S \subset I$. tanto I como S tienen elementos enteros que hacen referencia a los índices de la fila en un arreglo unidimensional para los atributos numéricos. Si l_x y h_x son los umbrales inferior y superior de un intervalo para un atributo x, y tanto l_x como h_x son elementos de x. Entonces existen los índices il_x y ih_x en una población S_x, tal que, $S_x[il_x] = l_x$ y $S_x[ih_x] = h_x$. De esta forma las variables de operación son valores discretos compuestos por $\{il_{x1}, ih_{x1}, il_{x2}, ih_{x2}, \cdots, il_{xn}, ih_{xn}\}$, siendo n el número de atributos numéricos presentes en la regla (dimensiones).

La población se inicializa mediante cálculos realizados por las Ecuaciones 3.16 y 3.17. w corresponde al cálculo de la amplitud del umbral superior ih respecto al umbral inferior il, donde n es el número de valores del dominio (observaciones en cada atributo numérico), i_s es el índice de las observaciones correspondiente a la población S y n_s el número de observaciones en la población S.

$$w(i_s) = w(i_s - 1) - \frac{i_s \cdot (n - \lfloor minSupp \cdot n \rfloor)}{n_s - 1} \tag{3.16}$$

$$f(il, ih) = \begin{cases} il & = \lfloor \mathcal{U}[0,1] \cdot (n - w) \rfloor \\ ih & = il + (w - 1) \end{cases} \tag{3.17}$$

Por cada atributo numérico de una regla se obtienen los intervalos por medio de la función $f(il, ih)$. Esto se aplica a todas las reglas que conforman la población, es necesario un proceso de validación de estos umbrales para garantizar la consistencia del intervalo. Se comprueba requisitos como: $il \leq ih$, $il \geq 0$,$ih \geq 0$, etc.. consiguiendo de esta manera inicializar los intervalos de toda la población de reglas con valores consistentes.

```
1 nodo_constructor(regla){
2     dim=obtieneDimension(regla)
3     contador=0, c=0
4     Repetir
5         varVector[c]=regla.IndiceMin[contador]
6         varVector[c+1]=regla.IndiceMax[contador]
7         c=c+2
8      Hasta (contador==dim)}
```

El *script* que se detalla arriba corresponde a uno de los constructores aplicados para inicializar el vector de las variables de operación. A continuación se describe el procedimiento en tres niveles generales:

- Definición de una plantilla de reglas
- Generación de la población de reglas
- Optimización de los intervalos en los atributos numéricos de la regla

Para activar el primer nivel se requiere la intervención del usuario y el acceso al conjunto de datos. Entonces se puede procesar la definición de una plantilla de reglas. En el último nivel mencionado, los procedimientos entregan la regla con el mayor valor obtenido por la función de evaluación para cada plantilla base.

La definición de una plantilla la establece el usuario. Configura los atributos tanto del antecedente como del consecuente. A partir de esta matriz, es posible personalizar las combinaciones entre atributos. En el tema sobre *el esquema de reglas* se detalla más específicamente este procedimiento.
Partiendo del esquema de reglas generales definido por el usuario, hemos desarrollado un procedimiento para obtener una lista de plantillas de reglas derivadas de la combinación de los atributos, respetando la estructura del esquema. El número de plantillas en la lista T_l está definido por la Ecuación (3.18), donde m es un parámetro que establece el número máximo de atributos numéricos deseados y n es la cantidad disponible en el regla.

$$nT_l = \sum_{r=2}^{m} {}_nP_r \tag{3.18}$$

El tamaño de la población es un parámetro que el usuario puede personalizar. Se permite la manipulación de este valor para encontrar un ajuste apropiado.

Los individuos de una población definida antes del proceso general se denominan población inicial y se obtienen del esquema de reglas $rTest_x$. (Línea 1 en Algoritmo 2). Es importante generar poblaciones de muy buena calidad. Para ello, la diversidad y la amplitud son dos criterios considerados, lo que lleva al uso de procedimientos basados en distribuciones aleatorias e introduciendo variables de control.

$$rTest_j^k = \bigwedge_{i=1}^{k} a_i \in [l_i, h_i] \rightarrow b_k = c_j \tag{3.19}$$

donde $k = 1 \cdots n$ y n es el número de atributos, $j = 1...m$, m es el número de clases. De esta forma se alcanza una población de reglas definidas por la Ecuación 3.19, con $k = 1$ y $j = 1$ en la clase, se tiene:
$R_q = a_1 \in [l_q, h_q] \rightarrow b_1 = c_1$ donde $q = 1 \cdots n$ y n es el tamaño de la población.

$$rTest^k = (\bigwedge_{i=1}^{kl} a_i \in [l_i, h_i]) \rightarrow (\bigwedge_{j=1}^{kr} b_j \in [l_j, h_j]) \tag{3.20}$$

En un esquema completamente numérico, se genera una explosión combinatoria controlada por el número de atributos numéricos que componen tanto LHS como RHS de la regla y, por lo tanto, se generan varias plantillas a partir de este proceso. La Ecuación 3.20 muestra una forma general de obtener cada individuo en la población, donde kl y kr son los números de atributos en LHS y RHS, respectivamente. A continuación se describe el proceso de QARM_VMO en el Algoritmo 2. La línea 1 involucra el conjunto de procedimientos para la generación de la población de reglas, posterior a la explosión combinatoria de atributos desde un esquema definido y parametrizado por el usuario. Similar a VMO original incluye los tres mecanismos de exploración:

1. Generación de nodos en dirección a extremo local de la vecindad.
2. Generación de nodos en dirección al extremo global.
3. Generación de nodos a partir de los nodos fronterizos.

En QARM_VMO la estructura del nodo está compuesto por un arreglo que almacena los indices de umbrales inferiores y superiores correspondientes a cada intervalo de los atributos numéricos que forman la regla.

Un procedimiento similar a la generación de población de reglas es utilizado para construir la función *GetNewRule(···)*. Esta función permite crear un nuevo individuo de la población cuando el algoritmo QARM_VMO lo requiere en las líneas 9,14 y 17.

Algorithm 2 QM_VMO algorithm

Require: C,k,rTest
Ensure: best_node
Generación de población de reglas R_i por Ecuación (3.19,3.20,3.16)
repeat
  foreach: $n \in N_0$ **do**
    Encontrar los k nodos cercanos usando Ecuación(3.2)
    seleccionar el mejor nodo localn_i^*
    if n_i^* es mejor que n_i **then**
      calcular factor de cercanía Pr entre n_i y n_i^* por Ecuación (3.1)
      $nfValues = f(n_i, n_i^*, Pr)$
      $newRule = GetNewRule(nfValues)$
      $n_{new} = nodeRule(newRule, nfValues)$
  foreach: $n \in N_0$ **do**
    calcular factor de cercanía Pr entre n_i y n_g por Ecuación (3.1)
    $nfValues = g(n_i, n_g, r)$
    $newRule = GetNewRule(nfValues)$
    $n_{new} = nodeRule(newRule, nfValues)$
  $nfValues = h(n_i, w)$
  $newRule = GetNewRule(nfValues)$
  $n_{frontiers} = nodeRule(newRule, nfValues)$
  Rellenar nodos en la malla total desde $n_{frontiers}$
  ordenar los nodos por función de calidad en la malla total
  Aplicar el operador de limpieza
  Construir la malla para la próxima iteración de forma elitista
until C iteraciones
best_node es obtenido desde malla (indice inicial)
return best_node

```
1 regla getNewRule(nfValues){
2      dim=obtieneDimension(nfValues)
3      r=crearRegla()
4      nAtributo=0, c=0
5      Repetir
6         indiceMin=nfValues[c]
7         indiceMax=nfValues[c+1]
8         iniciarRegla(r,indiceMin,indiceMax,nAtributo)
9         c=c+2
10      Hasta (nAtributo==dim)
11      evaluarRegla()
12      retornar(r)}
```

El vector *nfValues* recibe como parámetro los nuevos valores de convergencia para cada uno de las variables de operación. Estos valores son colocados en la nueva regla definida, y un procedimiento se encarga de la validación para que cumpla con las propiedades referentes a límites inferiores y superiores. Una vez configurado los intervalos para cada uno de sus atributos, las propiedades asociados a la regla de asociación son calculadas mediante la función de evaluación.

```
1 nfValues f(n_i, n_i^*, Pr, δ){
2   foreach: i ∈ n_i {
3     vm = calcula_vm(ni_i, nl_i)
4     if ((U[0,1]<= Pr) y (abs(vm - nl_i) >δ))
5      res_i = vm
6     elseif (abs(vm - nl_i) <= ds)
7      res_i = nl_i+U[−δ, δ]
8     else
9      res_i = U[vm, nl_i] }
10    retornar res}
```

El pseudocódigo indicado arriba corresponde a la implementación de la función de expansión f especificada en la línea 8 del algoritmo 2, y en correspondencia con la Ecuación 3.3. Esta función retorna un vector de valores para las variables de operación, con ese vector se procede a la creación de un nuevo nodo de tipo regla.

```
1 nfValues g(n_i, n_g, r){
2   foreach: i ∈ n_i {
3     vm = calcula_vm(ni_i, ng_i)
4     if (U[0,1]<Pr)
5      res_i = vm
6     else
7      res_i = U[vm, ng_i] }
8    retornar res}
```

Del mismo modo se describe el conjunto de pasos para la implementación de la función g, que es invocado en la línea 13 del Algoritmo 2. También en correspondencia con la Ecuación 3.6, la función acelera la convergencia mediante la exploración en dirección del nodo con mejor calidad, denominado *extremo global*.

```
1 nfValues h(n_i, w){
2     Y ← obtenerNodosFaltantes()
3     for i ← 1 to ⌊Y/2⌋ do
4      nodosNuevos.add = generarNodosInternos(···)
5     for p ← auxMalla.longitud downto Y do
6      nodosNuevos.add = generarNodosExternos(···)
7     retornar nodosNuevos}
```

La frontera se compone de los nodos más cercanos (conocidos como frontera interior o nodos internos) y más lejanos (conocidos como frontera exterior o nodos externos) del punto que representa el centro del espacio de búsqueda. Para detectar estos nodos, se utiliza la distancia euclidiana. Los nodos de mayor distancia componen el conjunto n_s y los de menor distancia componen el conjunto n_u. A partir de estos conjuntos, se crean $\lfloor Y/2 \rfloor$ nodos internos usando n_u y $Y - \lfloor Y/2 \rfloor$ nodos externos usando n_s, así se describe en la función h mostrada anterior a este párrafo.

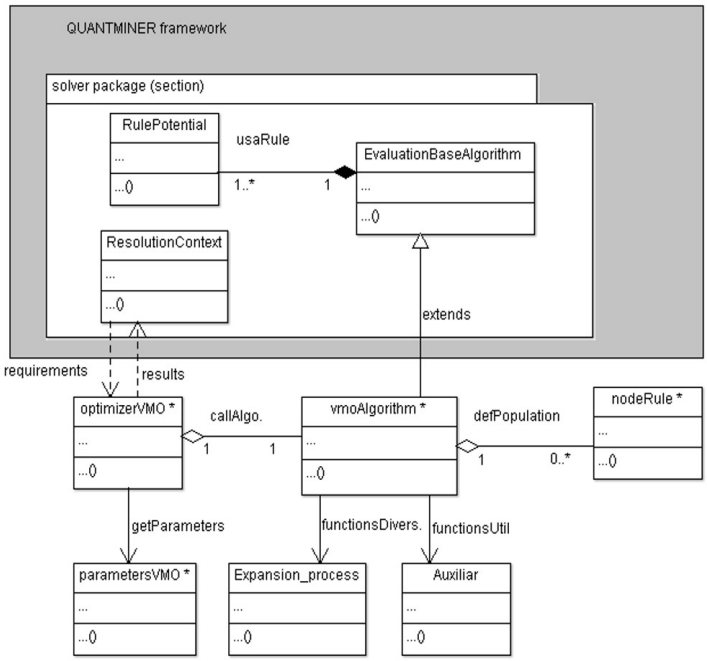

Figura 3.8: Esquema de acoplamiento de QARM_VMO con QUANTMINER

3.3.5. Acoplamiento

Es necesario esquematizar la arquitectura de acoplamiento del modelo QARM_VMO con el framework de QUANTMINER, una comprensión al detalle de su diseño ha sido un requisito indispensable para identificar y crear los mecanismos de enlace y sincronización. La Figura 3.8 ilustra una representación arquitectónica del modelo de acoplamiento.

Las clases que tienen el nombre seguido del (*) son aquellas en donde se ha introducido cantidades significativas de código. Las funciones utilizadas en QUANTMINER para enlazar QARM_VMO se encuentran dentro de las clases *ResolutionContext* y *EvaluationBaseAlgorithm*.

Similar a otros algoritmos implementados bajo este entorno, se utiliza la clase intermedia *optimizerVMO* como una interface entre QUANTMINER y QARM_VMO. Entre las funciones que debe cumplir está, el traspaso de parámetros provenientes del usuario a QARM_VMO, generar la población inicial de reglas, evaluar la cantidad de iteraciones especificada en la función objetivo, y finalmente, registrar los resultados tanto de métricas como de reglas, en la estructura correspondiente de QUANTMINER para su visualización.

Capítulo 4

Evaluación del algoritmo QARM_VMO

4.1. Introducción

En los capítulos anteriores se han presentado las técnicas para la exploración y búsqueda de reglas de asociación, además el método propuesto en esta investigación. La propuesta parte de las características que el algoritmo VMO define para la optimización de funciones continuas, esto unido al método de representación del problema de reglas de asociación en las estructuras internas del algoritmo, más otras funciones de preprocesamiento ajustadas al perfil de entrada y salida del proceso permitieron alcanzar una versión funcional para ser probada.

Los algoritmos de minería de datos son sometidos a diversas experimentaciones para evaluar el rendimiento, la estimación de la calidad en su mayor parte viene dado por métricas objetivas propias de cada técnica.
Los experimentos con algoritmos requieren de una computadora con prestaciones suficientes en cuanto a recursos, conjuntos de datos y algoritmos construidos para problemas similares.

Un ambiente de prueba bajo las mismas condiciones es alcanzable en una computadora, sin embargo, los algoritmos no presentan propiedades homogéneas en cuanto a perfiles de configuración y resultados, esto implica que las métricas de evaluación sean aplicadas apropiadamente sin favorecer ni perjudicar a alguno.

Por esos motivos, en este capítulo se detalla un conjunto de procedimien-

tos para la evaluación de la técnica propuesta. En principio un análisis de sensibilidad sobre los valores de configuración permite un ajuste del algoritmo para alcanzar resultados de mayor calidad.

Para tener una mayor variedad de datos es necesario la generación artificial de datos, es así que la inclusión de funciones para este proceso son parte de este capítulo. Conjuntos de datos reales también son tomados en cuenta para la construcción de experimentaciones rigurosas, el uso de las métricas estándar para este tipo de problemas, más otras construidas a partir de la función de evaluación propia del método, han sido los indicadores de calidad empleados tanto en el rendimiento como en la comparación con otras técnicas similares.

4.2. Determinación de los parámetros de configuración

La obtención de valores para configuración de los parámetros es un proceso necesario antes de utilizar un algoritmo. Además existe una alta variabilidad en las características que presentan los ambientes de experimentación, debido a las capacidades de hardware y propiedades de los conjuntos de datos, todo esto unido al factor humano y los objetivos perseguidos por el experimento. En consecuencia, la calidad de los estudios comparativos que se realicen en lo posterior puede verse fuertemente influenciada por la imprecisión de los parámetros o determinadas particularidades de configuración.

Parámetro	Descripción
P	Número de nodos de la población para cada iteración
T	Número de nuevos nodos requeridos para el proceso de expansión
k	Número de nodos vecinos para cada nodo de la malla
C	Número máximo de evaluaciones para la función de ajuste

Tabla 4.1: Parámetros del algoritmo VMO

Aunque los parámetros van a depender de cada técnica, algunos de los parámetros típicos que se definen en algoritmos evolutivos vienen dados por valores que definen el número de iteraciones o evaluaciones de la función de ajuste (Fitness), tamaño de la población, condiciones de parada y otros. La Tabla 4.1 describe los parámetros principales del algoritmo VMO, y también de la versión adaptada en esta tesis.

Conjuntos de datos sintéticos con propiedades personalizadas han sido generados mediante un procedimiento codificado en el software R project. Varios archivos con grandes cantidades de observaciones se utilizan en esta experimentación preliminar, también algunos conjuntos de datos reales son agregados para evaluar la veracidad de las reglas encontradas.

4.2.1. Pruebas preliminares

Tres etapas principales fueron definidos para construir un algoritmo para la generación de datos de prueba personalizados.

1. Definición de un esquema de regla
2. Extracción de variables y parámetros
3. Generación aleatoria de datos a partir del esquema de regla

La etapa *Definición de un esquema de regla* provee un mecanismo para declarar los parámetros que se utilizará en el proceso. Un archivo .csv es construido para crear un conjunto de datos, donde cada fila define a una regla y sus parámetros respectivos, se puede configurar varias reglas dentro del archivo.

Parámetro	Descripción	Ejemplo
LHS	Extremo izquierdo de la regla	{V1 = [23.0, 32.0]}
RHS	Extremo derecho de la regla	{V2 = [0.2683, 0.5835]}
supp	Soporte de la regla entre 0 y 1	0.833
conf	Confianza de la regla entre 0 y 1	0.92
lbVar1	Valor mínimo en la variable 1	22
ubVar1	Valor Máximo en la variable 1	37
mediaVar1	media en la variable 1	25
sdVar1	desviación estándar en la variable 1	2
lbVar2	Valor mínimo en la variable 2	0.1593
ubVar2	Valor Máximo en la variable 2	0.8291
mediaVar2	media en la variable 2	0.5734
sdVar2	desviación estándar en la variable 2	0.2341

Tabla 4.2: Parámetros para generación de datos sintéticos

La Tabla 4.2 detalla cada uno de los parámetros de configuración, además se agrega un ejemplo para cada definición. Los atributos relacionados con rangos de dominio, media y desviación estándar son necesarios indicar para cada variable que interviene en la regla, así una regla con más de dos variables deberá extender el conjunto de atributos en el archivo.

```
> dfRules
   extremo Nombres        linf       lsup    LINF    LSUP   media Desviacion
       LHS      V1   22.915367  32.026276 22.0000 37.0000 25.0000     2.0000
       LHS      V3 0.086823314 0.25273848  0.0494  0.3437  0.1257     0.0489
       RHS      V2   0.2574686 0.57906103  0.1593  0.8291  0.5734     0.2341
>
```

Figura 4.1: Ejemplo de esquema de regla en una estructura de R.

Los atributos LHS y RHS contienen información sobre las variables e intervalos en el caso de variables numéricas. El procedimiento no solo ha sido preparado para generar datos numéricos, también es posible definir un esquema de regla con variable categórica para fines de practicar reglas de clasificación. Tanto variables numéricas como categóricas pueden estar en cualquiera de los dos extremos de la regla.

La segunda etapa *Extracción de variables y parámetros* es un procedimiento definido en el programa R project, y corresponde a cargar los parámetros en las estructuras del programa. El tamaño de la población a generar se define para calcular las medidas de soporte tanto para la regla como de cada extremo. El ajuste al soporte del antecedente se calcula mediante el grado de confianza establecido. La figura 4.1 presenta una muestra del esquema de regla cargado en una estructura del programa R. A continuación se agregan tres líneas básicas implementadas en R para el procedimiento.

```
# definir el tamaño de la población en variable pop
```

1. `ruleTemplate=getDatos(path) #Conseguir los parámetros desde el archivo.`
2. `dfRules=getPartsRule(ruleTemplate) # Procesar esquema`
3. `#Calcular tamaños de poblaciones en la regla, LHS y RHS ajustado a soporte y confianza`

La tercera fase *Generación aleatoria de datos a partir del esquema de regla* se construye una función que permita generar de forma aleatoria los datos. El problema se reduce a generar valores desde una distribución normal truncada, con los umbrales inferior y superior especificados $a < b$. Esto puede ser hecho mediante la generación de cuantiles uniformes sobre el rango de cuantiles permitidos por el truncamiento, y entonces usar muestreo de transformación inverso para obtener los valores normales que corresponden.

Sea Φ la función de distribución acumulada de la distribución normal. Se quiere generar $X_1, \cdots, X_N$ desde una distribución normal truncada (con

media μ y varianza σ^2) con umbrales inferior y superior truncados $a < b$. Esto se define en la ecuación 4.1 y 4.2.

$$U_1, \cdots, U_N \sim IID \quad U[\Phi(\frac{a-\mu}{\sigma}), \Phi(\frac{b-\mu}{\sigma})] \tag{4.1}$$

$$X_i = \mu + \sigma \cdot \Phi^{-1}(U_i) \tag{4.2}$$

```
#Funcion para generar el nicho de la población
rtruncnorm <- function(N, mean = 0, sd = 1, a = -Inf, b = Inf) { if (a
>b) stop('Error: verifique rangos')
U <- runif(N, pnorm(a, mean, sd), pnorm(b, mean, sd))
qnorm(U, mean, sd) }
```

La función *rtruncnorm* definida previamente es invocada cuando se generan segmentos de la población en cada variable. El tamaño de estos segmentos fueron definidos en la etapa dos, y ajustados al grado de las métricas de soporte y confianza.

4.2.2. Conjuntos de datos de prueba

Para las pruebas de algoritmo se han utilizado datos sintéticos y datos reales. Por un lado los datos sintéticos se generan a partir de un algoritmo cuyo proceso se detalla en el apartado anterior, por otro lado los datos reales se han obtenido del repositorio *UCI machine learning*.

Código	Parámetro	Valor	Descripción
	Pop	100, 500 y 10000	Tres conjuntos de datos 100,500 y 10K
	LHS	{V1 = [23.0, 32.0]}	observaciones, 2 variables numéricas.
SN100	RHS	{V2 = [0.2683, 0.5835]}	Una regla definida es introducida
SN500	supp	0.833	para objeto de análisis detallado,
SN10K	conf	0.92	el código es para fines de referencia corta
	Pop	100 y 10000	Dos conjuntos de datos con 100 y 10K observaciones
	LHS	{V2 = [0.2683, 0.5835]}	y 1 variable numérica y 1 categórica.
SC100	RHS	{clase="C2"}	Un regla definida es introducida
SC10K	supp	0.45	para objeto de análisis detallado,
	conf	0.70	el código es para fines de referencia corta
	Pop	500, 1000 y 5000	tres conjuntos de datos 500,1K y 5K observaciones,
	LHS	{V1 = [23.0, 32.0],	2 variables numéricas en el
		V2 = [0.2683, 0.5835]}	antecedente y una categórica
SC500	RHS	{clase="C3"}	Un regla definida es introducida
SC1K	supp	0.833	para objeto de análisis detallado,
SC5K	conf	0.92	el código es para fines de referencia corta
	Pop	1000 y 5000	Dos conjuntos de datos 1000 y 5000 observaciones,
	LHS	{V1 = [23.0, 32.0],	2 variables numéricas en el
		V3 = [0.0868, 0.2527]}	antecedente y una en consecuente
SN1K	RHS	{V2 = [0.2574, 0.5790]}	Un regla definida es introducida
SN5K	supp	0.719	para objeto de análisis detallado,
	conf	0.908	el código es para fines de referencia corta

Tabla 4.3: Descripción de los datos sintéticos de prueba

Diez archivos con datos sintéticos fueron generados con las especificaciones indicadas en la Tabla 4.3.

Código	Nombre	Observaciones	Atributos	Clases	(Real/Entero/Nominal)
RC150	iris	150	5	3	4/0/0
RN4K	abalone	4177	9	0	9/0/0
RN96	basketball	96	5	0	5/0/0
RN345	Bupa	345	6	0	1/5/0
RN365	Dee	365	7	0	7/0/0
RC214	Glass	214	10	7	9/0/0
RN846	Vehicle	846	9	0	0/9/0
RN988	Vowel	988	13	0	10/3/0
RC178	wine	178	14	3	13/0/0

Tabla 4.4: Descripción de datos reales para prueba

Conjunto de datos reales han sido seleccionados cuidadosamente,y se detalla en la tabla 4.4. Algunos han sido aplicados en pruebas ejecutadas con otros algoritmos, y servirán como referencia. Es necesario que exista diversidad de datos para exponer a casos críticos al algoritmo. Disponer de datos que incluyan variables de clase permiten la evaluación del segmento *reglas de clasificación* que incluye la técnica.

En una primera prueba de sensibilidad, el algoritmo se ha configurado con valores por defecto en varios de los parámetros que define la técnica. En la Tabla 4.5 se describe cada parámetro. Los valores para *P0, Pf, KNear, maxAttNum ,minAttNum* se han dejado por defecto, éstos valores se han obtenido de la configuración de VMO original. Los otros parámetros se han ajustado de acuerdo a los requerimientos de la regla.

Nombre	Valor defecto	Descripción
P0	12	Tamaño inicial de nodos en la malla
Pf	32	Tamaño final de nodos en la malla ($Pf = 3 \cdot P0/2$)
KNear	3	cantidad nodos vecinos para exploración
suppMin	0.10	Umbral mínimo para el soporte de regla entre 0 y 1.
confMin	0.60	Umbral mínimo para la confianza de regla entre 0 y 1.
maxAttNum	3	Número máximo de atributos cuantitativos presentes en la regla
minAttNum	1	Número mínimo de atributos cuantitativos presentes en la regla
iterAll	NA	número de iteraciones

Tabla 4.5: Descripción parámetros del algoritmo QARM_VMO

4.2.3. Identificación de la cantidad de iteraciones para la convergencia

En la Tabla 4.6 se resume los valores mínimos, promedio y máximo del número de iteraciones para la convergencia a una solución, estos valores han sido computados desde un conjunto de (500,1000,5000 y 10000 iteraciones) para cada conjunto de datos mostrado.

Se observa que el máximo número de iteraciones requerido tiende a un crecimiento conforme se incrementan las observaciones, esto da lugar a que este parámetro sea personalizado por el usuario, sin embargo, se busca un valor por defecto. Esta claro que a mayor número de iteraciones se tiene mayor costo computacional, por lo que establecer un valor reducido de iteraciones es por una parte favorable para nuestra técnica, sin embargo, por otro lado se puede afectar la calidad de la solución.

La métrica en este caso es el número de iteraciones al que convergió el algoritmo, para capturar estos datos se sometió el algoritmo a ejecución con cada bloque de iteraciones y se extrajo el número de iteración en el que ocurrió la última actualización de su *función de ajuste*.

data	minIter	avgIter	maxIter	minSupp(%)	minConf(%)
sc100	22.50	44.40	71.00	40.00	70.00
sn100	12.00	12.00	12.00	80.00	90.00
sc500	12.00	12.00	12.00	80.00	90.00
sn500	12.00	12.00	12.00	80.00	90.00
sc1k	55.50	75.54	91.31	10.00	60.00
sn1k	38.67	77.47	114.00	10.00	60.00
sc5k	42.67	70.60	117.00	10.00	60.00
sn5k	53.08	83.32	104.25	10.00	60.00
sc10k	40.00	75.00	121.00	40.00	70.00
sn10k	58.50	131.30	177.50	10.00	60.00

Tabla 4.6: Iteraciones de convergencia en datos sintéticos

Conjuntos de datos reales también son utilizados en la prueba, los datos sintéticos ya han permitido definir un criterio sobre el número de iteraciones, sin embargo, la diversidad de características que se pueden encontrar en los datos reales hace necesario esta verificación. Los resultados de la Tabla 4.7 apoya la definición del número de iteraciones, nótese que los valores máximos de iteraciones en las que finaliza la convergencia son inferiores a 200 e incluso en varios casos inferiores a 100.

data	minIter	avgIter	maxIter	minSupp(%)	minConf(%)
abalone	76.76	91.08	124.46	10.00	60.00
basketball	49.30	68.61	79.41	10.00	60.00
bupa	52.40	62.24	72.48	10.00	60.00
glass	51.42	67.21	81.12	10.00	60.00
iris	50.46	63.66	75.22	10.00	60.00
wine	38.08	54.00	69.82	10.00	60.00

Tabla 4.7: Iteraciones de convergencia en datos reales

Con este análisis se establece el número de iteraciones por defecto en un valor de 200 iteraciones, para colecciones de datos con observaciones inferiores a 1000 se podría utilizar incluso 100 iteraciones, de esta forma alcanzar velocidades altas de respuesta.

4.2.4. Identificación del tamaño de la población

El tamaño de la población es otro de los parámetros de control tradicionales en algoritmos evolutivos, existen trabajos orientados a descartar este

parámetro mediante la introducción de un mecanismo de supervivencia basado en la edad de cada individuo de la población [258], asimismo se ha usado enfoques similares para crear algoritmos poblacionales con tamaño de población adaptativos [259]. Sin embargo, estos mecanismos podrían ser evaluados de forma consecuente a los resultado del enfoque tradicional aplicado en este algoritmo. El enfoque tradicional esta asociado a estudiar la calidad (comportamiento) cuando el tamaño de la población varía, y en este contexto una experimentación ha sido preparada.

El algoritmo QARM_VMO mediante los parámetros $P0$(Población inicial), Pf(Población final) define el número de nodos de la malla. La naturaleza del algoritmo es partir de un conjunto de nodos iniciales, y luego expandirse hasta la cantidad definida en la población final.

P0	sc100	sn100	sn500	sc500	sc1k	sn1k	sc5k	sn5k	sc10k	sn10k	R
12	**2.00**	**1.00**	**2.00**	**3.00**	**1.00**	**3.00**	**1.00**	**1.00**	**4.00**	**3.00**	**22.10**
18	3.00	2.00	4.00	4.00	2.00	4.00	4.00	2.00	3.00	2.00	30.94
24	1.00	3.00	3.00	1.00	3.00	2.00	2.00	4.00	2.00	4.00	26.08
30	4.00	4.00	1.00	2.00	4.00	1.00	3.00	3.00	1.00	1.00	25.35

Tabla 4.8: Sensibilidad al tamaño de la población en datos sintéticos

La Tabla 4.8 se construye a partir del resumen de cada conjunto de datos sintético, para ello se ha variado el tamaño de la población tanto $P0$ como Pf en las 200 iteraciones. Un valor de posición entre primer y cuarto lugar se registro a cada población definida, y el cálculo en la columna R mediante la Ecuación 4.3. Donde $R(i)$ se usó como medida para elegir la mejor prueba entre cada experimento realizado con cada uno de los conjuntos de datos, i es la observación, $k = 1 \cdots n$ son los conjuntos de datos que intervienen, y v es el valor que corresponde a la medida evaluada.

$$R(i) = \sum_{k=1}^{n} v(i)_k + STDEV(v(i)) \tag{4.3}$$

El mismo procedimiento se aplicó con una muestra de datos reales, los resultados se muestran en la Tabla 4.9.

P0	iris	basketball	bupa	wine	glass	abalone	R
12	**1.00**	**2.00**	**2.00**	**2.00**	**1.00**	**2.00**	**10.52**
18	2.00	1.00	4.00	3.00	2.00	1.00	14.17
24	3.00	4.00	1.00	1.00	4.00	3.00	17.37
30	4.00	3.00	3.00	4.00	3.00	4.00	21.55

Tabla 4.9: Sensibilidad al tamaño de la población en datos reales

Este procedimiento busca alcanzar un valor mínimo para R sobre la ejecución del algoritmo tanto en datos sintéticos como reales, este valor permite determinar el valor parametrizado de orden más alto en cuanto a calidad y de menor varianza. Es así que con valores obtenidos para $R = 22{,}10$ en el conjunto de datos sintéticos y $R = 10{,}52$ en los datos reales, el tamaño de la población que mejor favorece a la calidad de las reglas según el valor promedio obtenido de la función de ajuste es $P0 = 12$ y por tanto $Pf = 30$.

4.2.5. Identificación del número de vecinos cercanos

La definición de la cantidad de nodos vecinos influye sin duda en el rendimiento del algoritmo, la variación de este parámetro permite observar cambios en la calidad de la función de ajuste. Para esta prueba se ha configurado el mismo procedimiento experimental aplicado para ajuste del tamaño de población. Sin embargo, una prueba con cambios constantes de población inicial puede influir en la determinación de este parámetro, para evitar esa posible tendencia se ha procedido a observar el efecto en una misma población. Consiste en repetir la optimización de una regla, introduciendo la variación para $k = 1$,$k = 2$,$k = 3$,$k = 4$,$k = 5$ para analizar el efecto que produce en medida de calidad de la regla.

k	avgQualIris	avgQualbkt	avgQualSN5K
1	3.18	0.77	18.16
2	3.26	0.88	18.19
3	3.28	0.88	18.19
4	3.33	0.91	18.25
5	3.35	0.93	18.99

Tabla 4.10: Sensibilidad al número de vecinos cercanos

Los resultados mostrados en la Tabla 4.10 indican un valor de calidad que crece a medida que se incrementa este parámetro, sin embargo, también es notable que la variación tiende a reducirse, de hecho el mayor cambio se hace notable en los tres primeros valores para k. Esto nos permite concluir que el parámetro recomendable es $k = 3$, se puede de forma alternativa variar hasta $k = 5$, sin embargo, hay que considerar el costo computacional.

4.3. Evaluación del algoritmo

4.3.1. Ejemplo controlado básico de extracción de regla con QARM_VMO

Un conjunto de datos básico con diez observaciones y tres variables (dos numéricas y una categórica) se muestra en la Tabla 4.11. Este ejemplo permite hacer una primera evaluación controlada del algoritmo, y evaluar la calidad del resultado tanto con las métricas como por observación directa.

	X1	X2	Clase
1	2.62	0.11	C2
2	2.66	0.11	C2
3	2.55	0.10	C2
4	2.83	0.10	C2
5	2.78	0.11	C3
6	2.99	0.10	C1
7	3.25	0.11	C1
8	3.28	4.68	C3
9	2.96	7.57	C3

Tabla 4.11: Datos básicos para evaluación controlada del algoritmo

Los datos de entrada son configurados mediante un archivo de perfil de entrada (.prf). Este archivo contiene la ubicación del archivo de datos, el esquema de regla, y demás características descritos en la Tabla 4.5. Para demostrar que QARM_VMO tiene la capacidad de extracción de reglas desde un conjunto de datos, varios casos se han configurado como perfil de entrada. En la Tabla 4.12 se organiza los parámetros de entrada para cada uno de los casos bajo los que el algoritmo realizará el proceso de minería.

Archivo	LHS	RHS	SuppMin	confMin	Probados
SNC10_1.prf	X1	Clase	0.40	0.80	1
SNC10_2.prf	X2	Clase	0.40	0.50	1
SNC10_3.prf	X2,Clase	Clase,X2	0.40	0.50	2
SNC10_4.prf	X1,X2	Clase	0.40	0.50	3
SNC10_5.prf	X1,X2	X1,X2	0.40	0.80	2

Tabla 4.12: Descripción parámetros para el caso controlado

El perfil admite, decidir si un atributo forma parte solo del antecedente (LHS), solo del consecuente (RHS) o en ambos lados de la regla (considere que los atributos no se repiten en una misma regla). Una vez definido la participación de los atributos. Un proceso obtiene la lista de reglas a probar, mediante una combinación condicionada a la localización permitida para el

atributo, y cantidad de atributos numéricos mínimos y máximos. Así el primer caso da lugar a una sola regla de prueba, mientras que el segundo obtiene dos, debido a que $X2$ y *Clase* pueden estar tanto en LHS como en RHS sin que se repitan dentro de una misma regla.

Archivo	Rule	Supp	Conf
SNC10_1.prf	{X1 = [2.55019,2.834874]};{Clase = C2}	0.40	0.80
SNC10_2.prf	{X2 = [0.1011,0.1123]};{Clase = C2}	0.40	0.571
SNC10_3.prf	{X2 = [0.1011,0.1123]};{Clase = C2}	0.40	0.571
	{X1 = [2.55019,2.834874]};{Clase = C2}	0.40	0.80
SNC10_4.prf	{X2 = [0.1011,0.1123]};{Clase = C2}	0.40	0.571
	{X1 = [2.55019,2.834874]};{Clase = C2}	0.40	0.80
	{X1 = [2.55019,2.986227],X2 = [0.1002,0.1123]};{Clase = C2}	0.40	0.571
SNC10_5.prf	{X1 = [2.66315, 2.9043]};{X2 = [0.1011, 0.1123]}	0.40	1.0
	{X2 = [0.1034, 0.1123]};{X1 = [2.55019, 2.9043]}	0.40	1.0

Tabla 4.13: Resultados de QARM_VMO para el caso controlado

Observe el caso *SNC10_1.prf*, tiene definido en el lado izquierdo de la regla a $X1 \in [2{,}55019, 2{,}834874]$, si se analiza al detalle el conjunto de datos en la Tabla 4.11, nótese que hay cinco elementos que están dentro de este rango, es decir el soporte en LHS es 0,5 y para toda la regla es 0,4. La confianza se obtiene de la división entre 0,4 y 0,5, dando como resultado 0,8. En este ejemplo corto se ha demostrado de forma detallada que el algoritmo obtiene los resultados correctos, también otros casos citados demuestran de igual manera.

4.3.2. Pruebas de rendimiento

El incremento progresivo del número de iteraciones permite observar la convergencia mediante la función de ajuste del algoritmo. Sin embargo, reglas con diferentes características forman parte de la solución, y en cada una los valores de evaluación tienen diferentes escalas para los cálculos mediante la función de ajuste. Un primer experimento se desarrolló con cinco reglas seleccionadas del conjunto de datos *basketball*, con umbrales $suppMin = 0{,}10$ y $confMin = 0{,}60$ y los demás parámetros por defecto.

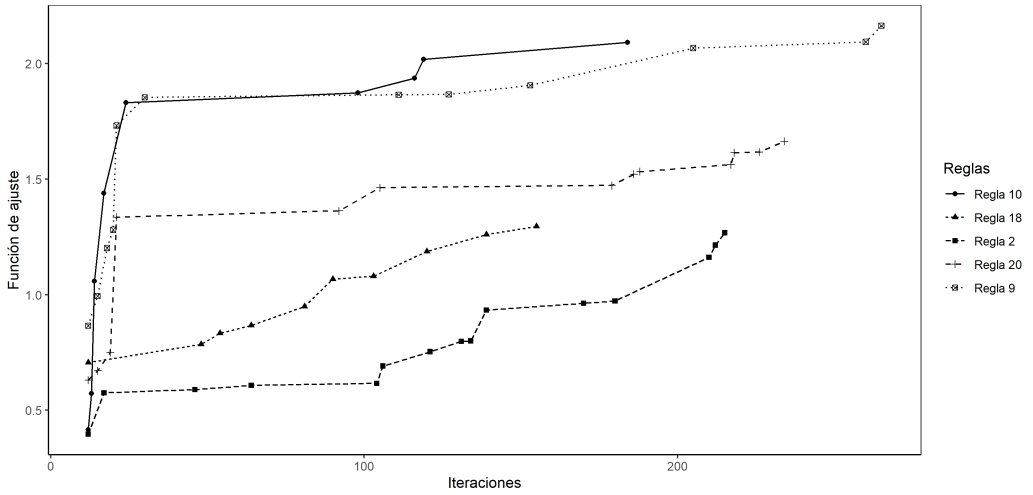

Figura 4.2: Convergencia de la función de ajuste, caso cinco reglas.

#	Regla	fit	supp	conf	lift	amp
9	{assists_per_minuteReal = [0.2084, 0.2495]} {heightInteger = [185.0, 191.0]}	2.16	0.19	0.95	2.12	0.14
10	{heightInteger = [196.0, 198.0]} {assists_per_minuteReal = [0.0528, 0.1485]}	2.09	0.21	0.95	2.13	0.19
20	{time_playedReal = [31.07, 36.67]} {points_per_minuteReal = [0.4086, 0.5885]}	1.66	0.26	0.89	1.68	0.23
18	{time_playedReal = [21.67, 26.7]} {heightInteger = [191.0, 198.0]}	1.30	0.16	0.88	1.37	0.16
2	{assists_per_minuteReal = [0.2315, 0.2771]} {ageInteger = [25.0, 29.0]}	1.27	0.14	0.93	2.12	0.21

Tabla 4.14: Cinco reglas con mejor calidad obtenidas de basketball

La Tabla 4.14 muestra cinco reglas de un total de ochenta que se probaron. Éstas fueron generadas a partir de una configuración que permite construir los esquemas, combinando todos los atributos tanto en el antecedente como en el consecuente, con un máximo de hasta tres atributos numéricos por regla. Cinco métricas que corresponde a *calidad del ajuste*, *Soporte*, *Confianza*, *Amplitud media* y *lift* son mostradas junto con la regla encontrada.

Observe la Figura 4.2, nótese que cada regla tiene su propio comportamiento. La convergencia a la solución puede ocurrir en un cierto número de iteraciones. Así por ejemplo, la regla 10 y 18 han finalizado su progreso antes de las doscientas iteraciones, mientras que la regla 9 ha alcanzado su óptimo después de las 250 iteraciones.

La medida de soporte determina porcentaje de proporción de registros que cubre la regla, es una métrica relativa al requerimiento del usuario. Así, no siempre medidas de soporte altos son las mejores, y tampoco los valores muy bajos son adecuados.

Un papel importante desempeña el umbral de soporte mínimo establecido

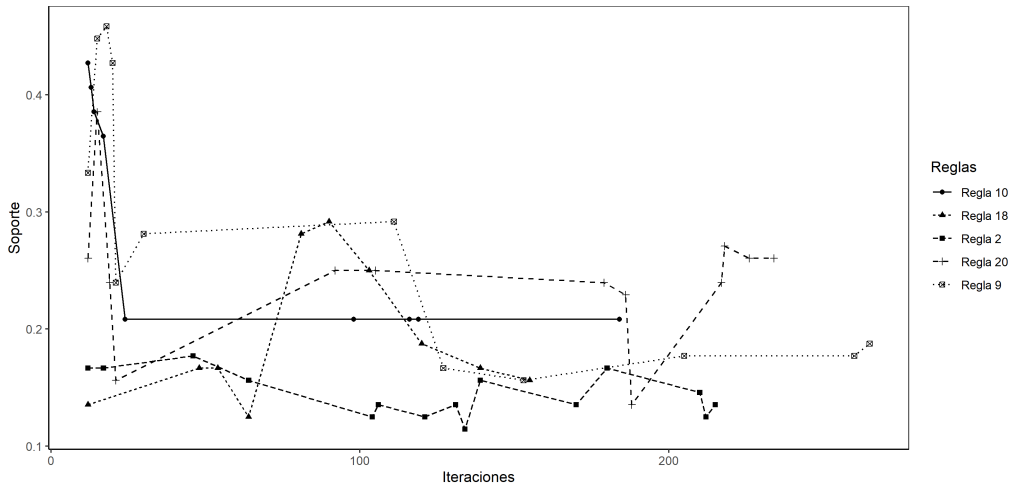

Figura 4.3: Convergencia del soporte, caso cinco reglas.

por el usuario, pues la función de evaluación favorece a intervalos con amplitudes pequeñas. Esto quiere decir que el soporte tiende hacia el umbral mínimo a medida que converge a la solución. En la Figura 4.3 se nota un patrón decreciente conforme sube el número de iteraciones, algunos picos en cierto momento parecen alcanzar el umbral mínimo, sin embargo, el objetivo no es alcanzar ese límite. El soporte de la regla es un criterio en la función de evaluación y mejorar en algún otro elemento implica que la medida de soporte tenga fluctuaciones como se observa.

Valores de confianza altos favorecen a la parte de ganancia de la función de evaluación, sin embargo, también la amplitud está en conflicto, y da lugar a fluctuaciones como se observa en la Figura 4.4.

El grado de especificación del esquema de regla contribuye significativamente a la reducción del tiempo de ejecución, sin embargo, pueden existir escenarios en donde el esquema no siempre sea tan preciso como se quisiera. Esto debido a la necesidad de explorar un conjunto de reglas a partir de una sola especificación general, lo que provoca una explosión combinatoria

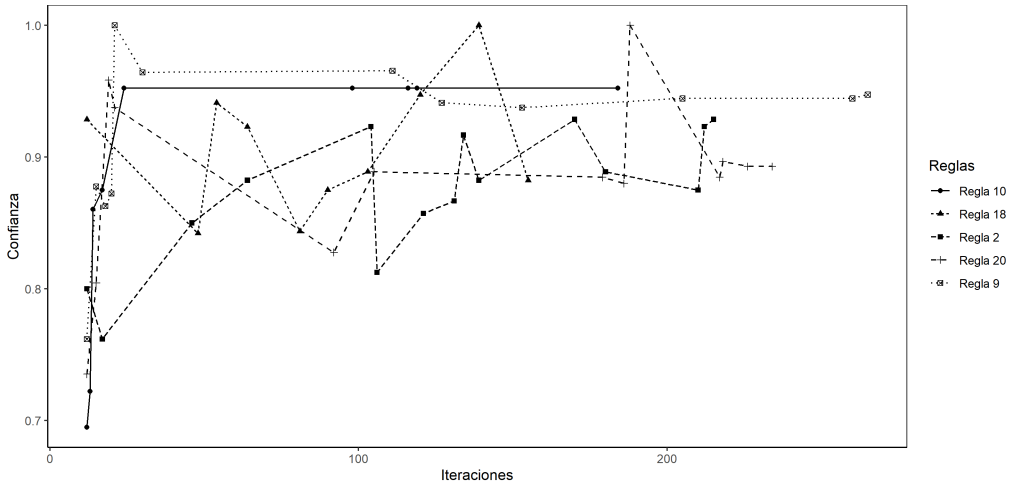

Figura 4.4: Convergencia de la confianza caso cinco reglas.

de atributos para reglas compuestas por un número mínimo $n0$ hasta un máximo nf de atributos numéricos. Por tanto se detalla un experimento para estimar el tiempo de ejecución versus el número de registros del conjunto de datos.

nombre	tipo	umbral inferior	umbral superior
valor_bienes	real	5.5	19.5
ingreso_familia	real	0.3	1.7
gastos_familia	real	0.25	1.5
ahorros_varios	real	0.0	10.0
edad	real	18.0	45.0
benficio_decision	int	1	3

Tabla 4.15: Descripción de los datos simulados para la prueba de tiempo de ejecución

Nueve archivos con datos sintéticos han sido preparados para el desarrollo de la experimentación, Los datos simulados están compuestos de seis atributos creados de forma aleatoria con el procedimiento descrito en el apartado de generación de datos de prueba; uno de los atributos es de tipo categórico y los otros son variables numéricas continuas como se describe en la tabla 4.15. Las cantidades de registros establecidas para cada archivo son (10K,20K,50K,100K,150K,200K,300K,400K y 500K); para dar cierta cohe-

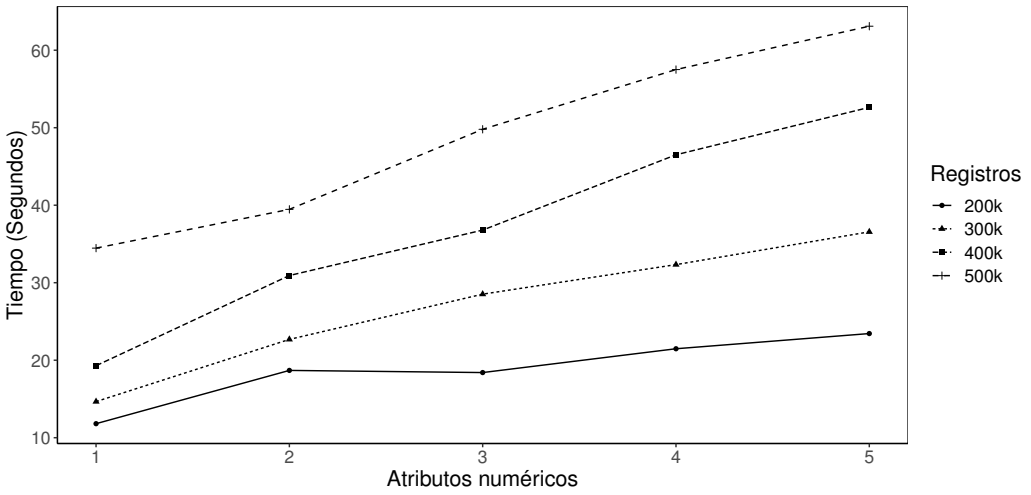

Figura 4.5: Tiempo de ejecución promedio por número de atributos numéricos.

rencia a los datos, se asume que los datos son usados para clasificar un segmento de beneficiarios del bono de solidaridad, así la variable de clase se define bajo los siguientes categorías (muy_apto, apto, no_apto) y para las variables numéricas (valor_bienes, ingreso_familia, gastos_familia, ahorros_varios y edad).

Dado que la técnica se caracteriza por la optimización intervalar de los atributos numéricos en cada regla, por consiguiente la cantidad de atributos numéricos que intervienen en la regla va ha influir en el tiempo de optimización total. Bajo este criterio, en la experimentación se introduce variación a la cantidad de atributos numéricos en la regla para $n = (1, \cdots, 5)$ además de la cantidad de registros. Como es típico de los métodos evolutivos, la definición de un único esquema para ser probado no es suficiente para seleccionar el valor de tiempo, dado que varias características del entorno experimental producen variaciones en el resultado. El esquema preparado genera la siguientes cantidades de reglas para ser probadas por el algoritmo $(n = 1 \rightarrow 10\, reglas)$,$(n = 2 \rightarrow 80\, reglas)$,$(n = 3 \rightarrow 200\, reglas)$,$(n = 4 \rightarrow 220\, reglas)$ y $(n = 5 \rightarrow 92\, reglas)$.

Para obtener el tiempo en cada caso se calcula el promedio de los tiempos de optimización alcanzado por cada regla probada en el caso respectivo, y de esta manera se puede observar en la Figura 4.6 que a mayor número de

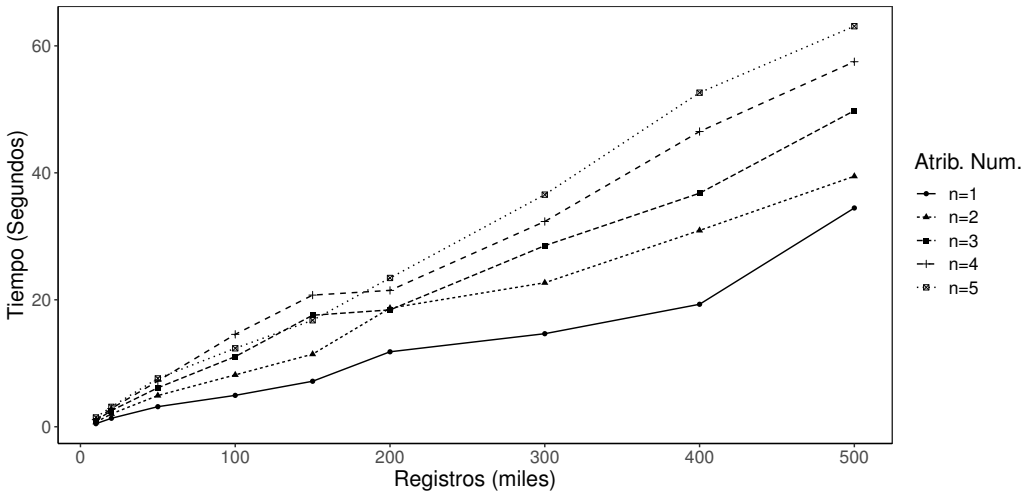

Figura 4.6: Tiempo de ejecución promedio por registros en esquemas con atributos numéricos variados .

atributos numéricos en la regla mayor es el tiempo de ejecución, y como es lógico el tiempo ejecución tiende ha crecer a medida que el número de registros aumenta. Aunque los tiempos de ejecución son aceptables cuando el número de reglas que se evalúan es pequeño. Sin embargo, en cantidades de reglas bastante grandes, la acumulación de tiempos de optimización de cada regla puede llegar a ser significativo.

n	10K	20K	50K	100K	150K	200K	300K	400K	500K
n=1	0.48	1.34	3.16	4.94	7.17	11.81	14.66	19.29	34.46
n=2	0.71	2.03	4.92	8.17	11.44	18.68	22.68	30.93	39.46
n=3	0.98	2.55	6.14	11.05	17.57	18.41	28.52	36.79	49.80
n=4	1.12	2.87	7.20	14.56	20.74	21.48	32.33	46.51	57.50
n=5	1.47	3.14	7.61	12.36	16.78	23.44	36.56	52.62	63.10

Tabla 4.16: Tiempo de optimización promedio (en segundos) por registros en reglas con cantidades variadas de atributos numéricos

La Tabla 4.16 proporciona más detalle sobre los valores obtenidos de los tiempos de optimización promedio en cada caso (cantidad de atributos numéricos por regla). Nótese que el tiempo de optimización promedio de una regla con cinco atributos numéricos en conjuntos de datos con quinientos mil registros es de 63.10 segundos, es decir que multiplicando por las 92 reglas que se evalúan se obtiene un tiempo aproximado de 96.75 minutos. Esta cantidad

ya es representativa en términos computacionales considerando de que el número de reglas a evaluar puede ser mayor. Sin embargo, es importante evaluar en relación a otras técnicas similares para tener un mayor fundamento sobre los resultados de esta métrica.

La Figura 4.5 demuestra un comportamiento similar del tiempo de optimización para el caso en que la cantidad de atributos numéricos en la regla varían. Nótese que a mayor número de atributos numéricos el algoritmo ocupa más tiempo en la optimización, de hecho por cada atributo numérico en la regla, se está agregando mayor carga en la optimización de una regla.

Datos	#Reglas	Sop(prom)	Sop(DE)	Conf(prom)	Conf(DE)	Fit(prom)	Fit(DE)
abalone	50	0.17	±0,00	0.94	±0,01	76.75	±0,57
basketball	80	0.22	±0,02	0.78	±0,01	0.27	±0,01
bupa	50	0.32	±0,01	0.76	±0,01	1.30	±0,05
dee	50	0.41	±0,01	0.71	±0,00	0.37	±0,04
glass	50	0.32	±0,01	0.84	±0,01	1.17	±0,03
iris	50	0.20	±0,01	0.84	±0,01	1.15	±0,03
pima	50	0.32	±0,01	0.75	±0,01	1.73	±0,04
wine	50	0.25	±0,02	0.78	±0,01	0.55	±0,02

Tabla 4.17: Precisión en la principales métricas evaluadas en un grupo de datos reales

Ocho conjuntos de datos reales fueron usados para evaluar la estabilidad de los resultados correspondientes a métricas de interés. Siete repeticiones se realizaron con cada conjunto datos y esquemas de regla que incluyen entre dos y cuatro atributos numéricos. La Tabla 4.17 muestra los valores promedios junto con la variación para las medidas de *Soporte*,*Confianza* y *Función de ajuste*. Observe que en casi en la totalidad de los casos se tienen variaciones pequeñas entre ±0,0 y ±0,05. Esto constituye un resultado satisfactorio sobre la confiabilidad de la solución que proporciona la técnica, entendiéndose que con siete poblaciones iniciales diferentes se aproxima con gran precisión a la solución de calidad.

4.3.3. Algoritmos de prueba

En la literatura se encuentra una gran variedad de algoritmos referentes a extracción de reglas de asociación. Cada uno tiene en parte características únicas incorporadas, así pueden existir propiedades relacionadas a los tipos de datos que cubren, mecanismo de búsqueda, estructura de representación, tipos de reglas y otros aspectos más detallados. Por una parte esta diversidad de propiedades dificulta un estudio comparativo minucioso, debido a que solo

se pueden utilizar métricas normalizadas comunes para su evaluación, y de manera eventual, estas medidas pueden tener objetivos diferentes (así para una técnica la tendencia de la métrica hacia un umbral superior es mejor, mientras que para otra puede ser todo lo contrario). Por otra parte, los conjuntos de datos utilizados en la evaluación deben ser los mismos para todos los algoritmos. Esto implica que el algoritmo sea compatible con los datos de entrada, asimismo las reglas resultantes deben estar normalizadas.

La disponibilidad de los algoritmos es otra limitación a tomar en cuenta, existen versiones binarias que se encuentran implementados tanto en software comercial como en los de libre distribución. De estas dos, las versiones implementadas en software de libre de distribución terminan siendo más atractivos por su disponibilidad y flexibilidad para ser utilizados.

Criterio	Descripción
Disponibilidad	Relacionado al grado de accesibilidad al algoritmo, de preferencia en R project o Java
Compatibilidad datos	Capacidad del algoritmo para admitir un conjunto de datos homogeneo para efecto de la comparación
Métricas comunes	Que las medidas con respecto a la calidad sean las estandarizadas, en lo posible permita agregar nuevas métricas
Posicionamiento	Existen algoritmos clásicos que están posicionados en diferentes, software y han sido usados como referentes en varios estudios.

Tabla 4.18: Criterios para seleccionar algoritmos de comparación

La participación de algoritmos clásicos en estudios comparativos es fundamental, estos son referentes de aceptación y preferencia por los usuarios, debido a las implementaciones en diferentes programas de computadoras comerciales y de libre distribución. Además, han sido considerados en múltiples estudios experimentales por diferentes autores, de hecho su alta disponibilidad hace posible que se consideren en varios estudios. En la Tabla 4.18 se resumen los cuatro criterios descritos para la selección de los algoritmos que intervienen en el estudio comparativo.

Es común encontrar suites de algoritmos sobre minería de datos que agrupan diversidad de técnicas. Así por ejemplo, jMetal[260] es un entorno basado en java con una arquitectura orientado a objetos. Este permite la experimentación con técnicas clásicas para resolver problemas de optimización con enfoques monoobjetivo y multiobjetivo, además de proveer una suite de funciones y métricas de evaluación. Asimismo, hay ambientes de desarrollo que reúnen otra variedad de algoritmos con arquitecturas flexibles, permitien-

do que nuevas técnicas reciclen las funciones y estructuras existentes, por ejemplo se tienen a QUANTMINER[261], KEEL[262, 263]. Todas son herramientas publicadas bajo GPL (Licencia Pública General), esto permite que los usuarios finales (personas, organizaciones) tengan libertad de usar, estudiar, compartir (copiar) y modificar el software sin fines comerciales.

Un proceso de evaluación con algoritmos incluye varias etapas, desde la selección misma de los datos, preprocesado, medidas de evaluación, integración y ejecución de algoritmos y generación de resultados. La integración de todas estas fases en un sola herramienta de desarrollo es posible con diversos grados de complejidad, sin embargo, se pueden reducir trabajos de implementación y codificación cuando existen librerías que abarcan gran parte de esas funcionalidades.

RKEEL[264] es una librería en R project, provee una interface para la suite de algoritmos disponibles en KEEL (java). La primera versión de este paquete abarca 110 algoritmos entre técnicas de clasificación, regresión y preprocesado. En la división de reglas de asociación se incluyen 17 algoritmos, los nombres se describen en la Tabla 4.19. El conjunto está conformado por técnicas que implementan diferentes estrategias de búsqueda de reglas de asociación numéricas.

Nombre completo	Nombre corto
APRIORI	APRIORI-A [76][265]
Association Rules Mining by means of a genetic algorithm proposed by Alatas et al.	Alatasetal-A [266]
Evolutionary Association Rules Mining with Genetic Algorithm	EARMGA-A [267]
Equivalence CLAss Transformation	Eclat-A [268][265]
Frequent Pattern growth	FPgrowth-A [269]
Genetic Association Rules	GAR-A [270]
GENetic Association Rules	GENAR-A [271]
Alcala et al Method	Alcalaetal-A [272]
Fuzzy Apriori	FuzzyApriori-A [273]
Genetic Fuzzy Apriori	GeneticFuzzyApriori-A [274]
Genetic-Fuzzy Data Mining With Divide-and-Conquer Strategy	GeneticFuzzyAprioriDC-A [275]
ARMMGA	ARMMGA-A [276]
Multi-objective differential evolution algorithm for mining numeric association rules	MODENAR-A [277]
Multi-objective rule mining using genetic algorithms	MOEA_Ghosh-A [278]
Multi-Objective Evolutionary Algorithm for Mining a Reduced Set of Interesting Positive and Negative Quantitative Association Rules	MOPNAR-A [279]
QAR_CIP_NSGAII	QAR_CIP_NSGAII-A [280]
Niching genetic algorithm to mine positive and negative quantitative association rules	NICGAR-A [281]

Tabla 4.19: Algoritmos de Reglas de Asociación disponibles en KEEL

El criterio de disponibilidad se cumple con esta suite, accesible tanto desde los entornos de Java como R Project. Los dos entornos de programación son idóneos para trabajar con las experimentaciones, sin embargo, se eligió R Project debido a que ya se ha venido usando en apartados previos, además por las bondades que este ofrece en las funciones estadísticas y visualizaciones de resultados.

Una segunda selección se resumen en la Tabla 4.20, dentro de este grupo se toma en cuenta los tres criterios restantes *posicionamiento,métricas* y *compatibilidad de datos*. Además se señala a importantes referentes y clásicos de la literatura y la industria del software, asi como relevantes de cada estrategia empleada. El conjunto elegido se conforma por *APRIORI-A [76][265],GAR-A [270],GENAR-A [271],MODENAR-A [277]* y *QAR_CIP_NSGAII-A [280]*.

A este grupo se une el algoritmo QUANTMINER[81][261], QUANTMINER es un algoritmo disponible en un framework bajo Java que recibe el mismo nombre, y como se ha mencionado en capítulos anteriores es el entorno de trabajo de base para el proyecto que se ha desarrollado.

Algoritmos	Parámetros
APRIORI-A [76][265]	Número de particiones para atributos numéricos: 4, Soporte mínimo: 0.1, Confianza mínimo: 0.6
GAR-A [270]	Número de evaluaciones: 5000, Población: 100, Número de itemsets: 100, Probabilidad de selección: 0.25, Probabilidad de cruce: 0.7, Probabilidad de mutación: 0.1, Importancia del número de registros cubiertos: 0.4, Importancia de amplitud de intervalo: 0.7, Importancia del número de atributos participantes: 0.5, Factor de amplitud: 2.0, Soporte mínimo: 0.1, Confianza mínimo: 0.6
GENAR-A [271]	Número total de reglas: 10, Número total de evaluaciones: 5000, Probabilidad de selección: 0.25, Probabilidad de mutación: 0.1, Tamaño de la población: 100, Factor de penalización: 0.7, Factor de amplitud: 2
MODENAR-A [277]	Tamaño de la población: 100, Número de evaluaciones: 50000, Tasa de Cruce (CR): 0.3, Número de soluciones no dominadas: 5, Factor de amplitud: 2, Peso para soporte: 0.8 Peso para confianza: 0.2, Peso para comprensibilidad: 0.1, Peso de amplitud del intervalo: 0.4
QAR_CIP_NSGAII-A [280]	Número de objetivos: 3, Número de evaluaciones: 2000, Tamaño de la población: 100, Probabilidad de mutación: 0.1, Factor de amplitud: 2, Umbral de diferencia: 5
QUANTMINER[81][261]	Número de evaluaciones: 100, Número mínimo de atributos numéricos: 2, Tamaño de Población: 250, Número máximo de atributos numéricos: 4, Porcentaje de cruce: 50%, Porcentaje de mutación: 40%, Soporte mínimo: 0.1, Confianza mínimo: 0.6.

Tabla 4.20: Parámetros establecidos en algoritmos de evaluación

La eficiencia de GAR y GENAR en relación a los algoritmos APRIORI y Eclat han sido demostrados en el trabajo [282]. Las pruebas revelan que estos algoritmos genéticos obtienen reglas de asociación de calidad competitivas, y se destacan notablemente con tiempos de ejecución que escalan de forma lineal cuando se incrementa la dimensión del problema.

Algoritmos genéticos con enfoque multiobjetivo han sido aplicados ampliamente en problemas de optimización. Optimización de reglas de asociación con este enfoque ha sido estudiado con diversas variantes. En [283] se muestra un estudio comparativo entre técnicas clásicas y específicamente tres algoritmos multiobjetivo MOEA_Ghosh, MOPNAR y MODENAR. El estudio señala una relación de comprensibilidad, interés y rendimiento en los resultados con el método multiobjetivo. En general es posible la personalización de los objetivos, mediante la implementación de varias funciones de optimización.

4.3.4. Resultados experimentales comparativos

Los diez conjuntos de datos reales fueron obtenidos desde [285]. La descripción general de estos datos se muestra en la Tabla 4.4, algunos de ellos contienen atributos categóricos, los cuales han sido descartados para garantizar la entrada de datos completamente numéricos a los algoritmos citados, debido a que no todos soportan procesamiento cualitativo. Todos los experimentos han sido ejecutados usando Procesador 11^{th} Gen Intel®Core™i7-1165G7 @ 2.80GHz, 2803 Mhz, 4 núcleos, 8 procesadores lógicos con 16 GB de memoria y sistema operativo Windows 10.

Los parámetros de los cinco algoritmos evolutivos y APRIORI están dados en la Tabla 4.20. Los parámetros que se muestran corresponden a los valores por defecto establecido para los algoritmos por sus autores, sin embargo, algunos valores han sido modificados para conseguir condiciones de ejecución similares, así el número de evaluaciones se a puesto en 5000, el tamaño de la población en 100 y valores mínimos para soporte y confianza en 0.10 y 0.60 respectivamente. Algoritmos evolutivos han sido ejecutados 10 veces y los valores correspondiente a las métricas han sido registradas para su comparación.

Datos	APRIORI	GAR	GENAR	MODENAR	NSGAII	QARM_VMO	QUANTMINER
basketball	0.16	0.79	0.26	0.36	0.21	0.19	0.22
iris	0.17	0.13	0.47	0.21	0.20	0.24	0.25
wine	0.13	0.57	0.16	0.17	0.17	0.21	0.26
glass	0.24	0.51	0.69	0.25	0.37	0.32	0.34
abaloneClass	0.23	0.49	0.82	0.57	0.24	0.23	0.26
dee	0.16	0.63	0.17	0.30	0.17	0.22	0.23
pima	0.20	0.70	0.58	0.42	0.20	0.27	0.28
bupa	0.33	0.70	0.77	0.18	0.44	0.31	0.35
ecoli	0.18	0.50	0.38	0.33	0.26	0.20	0.26
vehicle	0.15	0.79	0.23		0.24	0.29	0.30

Tabla 4.21: Resultados con algoritmos, métrica de soporte promedio

Algoritmos de reglas de asociación obtienen diferentes resultados para la medida de soporte, cada uno cumple con sus criterios de optimización o una estrategia de particionamiento establecida para los intervalos en el caso de los no evolutivos.

El soporte es una de las medidas fundamentales utilizados por investigadores en minería de reglas de asociación, sin embargo, algunas consideraciones de su definición citadas por [286] pueden ser significativas al momento de su análisis. Una regla de asociación se considera engañosa si se obtuvieran valores extremos, así con un valor igual a uno la regla es inútil ya que aparece en cualquier transacción, por lo que no proporciona ningún conocimiento nuevo sobre las propiedades de los datos. Por el contrario, si es cero, entonces la regla no representa ninguna transacción, por lo que se considera engañosa. La función de optimización utilizado en QARM_VMO no maximiza la medida de soporte, al contrario como efecto de la minimización de amplitudes en los intervalos se obtiene una convergencia hacia valores cercanos al umbral establecido. Con esto la eficiencia del algoritmo se focaliza en conseguir zonas de alta densidad cercanos al límite de soporte establecido.

La Tabla 4.21 muestra el valor promedio de soporte alcanzado por el conjunto de reglas extraídas por cada algoritmo en los diferentes conjuntos de datos. Nótese en el caso de QUANTMINER y QARM_VMO los valores bajos en la métrica con relación a los otros algoritmos, con excepción de APRIORI (no evolutivo basado en discretización) y algunos casos en QAR_CIP_NSGAII-A. Recuerde que el umbral mínimo de soporte en la experimentación es 0.10, y es clara la tendencia de la técnica estudiada hacia este umbral, de esta manera el usuario puede fijar el resultado deseado respecto de esta métrica.

Datos	APRIORI	GAR	GENAR	MODENAR	NSGAII	QARM_VMO	QUANTMINER
basketball	0.71	0.89	0.98	0.90	0.84	0.92	0.90
iris	0.84	0.83	0.91	0.95	0.97	0.94	0.92
wine	0.74	0.78	0.91	0.96	0.87	0.94	0.93
glass	0.87	0.90	0.97	0.98	0.91	0.92	0.92
abaloneClass	0.91	0.94	0.99	0.99	0.94	0.98	0.96
dee	0.80	0.82	0.88	0.98	0.90	0.92	0.90
pima	0.79	0.84	0.91	0.47	0.86	0.89	0.88
bupa	0.81	0.92	0.97	0.96	0.89	0.85	0.85
ecoli	0.88	0.81	0.98	0.33	0.91	0.90	0.87
vehicle	0.87	0.91	0.77	-	0.88	0.98	0.99

Tabla 4.22: Resultados con algoritmos, métrica de confianza promedio

Los algoritmos para reglas de asociación en general, buscan maximizar la

medida de confianza a valores cercanos o iguales a 1.0, un valor de confianza mayor a cero, representa el porcentaje de observaciones conteniendo tanto el antecedente como el consecuente con relación al antecedente de la regla. Importante dejar claro que esta medida es independiente de la significación con respecto al número total de observaciones del conjunto de datos, por tal razón es necesario que se use junto con la medida de soporte.

La Tabla 4.22 presenta valores promedio de la confianza en reglas obtenidas por los algoritmos en cada conjunto de datos. La mayor parte de los casos se obtienen valores superiores al 90 % y QARM_VMO es el mejor ubicado seguido de GENAR y su similar versión genética QUANTMINER, cuyos valores están dentro del cuarto cuartíl con respecto a la métrica.

Datos	APRIORI	GAR	GENAR	MODENAR	NSGAII	QARM_VMO	QUANTMINER
basketball	0.25	0.41	0.49	0.35	0.19	0.20	0.21
iris	0.25	0.17	0.49	0.26	0.14	0.17	0.20
wine	0.25	0.34	0.49	0.31	0.16	0.20	0.21
glass	0.25	0.15	0.45	0.18	0.12	0.07	0.07
abaloneClass	0.25	0.22	0.48	0.46	0.17	0.13	0.13
dee	0.25	0.35	0.48	0.35	0.20	0.21	0.21
pima	0.25	0.28	0.47	0.40	0.10	0.13	0.12
bupa	0.25	0.22	0.48	0.18	0.17	0.11	0.11
ecoli	0.25	0.29	0.43	0.16	0.14	0.18	0.21
vehicle	0.25	0.17	0.46	-	0.16	0.04	0.05

Tabla 4.23: Resultados con algoritmos, amplitud de intervalos promedio

La amplitud promedio de intervalos en los atributos que forman parte de la regla es una medida que ha sido agregado en esta experimentación, y para cada uno de los algoritmos ha sido calculada a partir de las reglas generadas. Aunque la amplitud no es encontrado comúnmente en la literatura como una métrica estandarizada, varias técnica hacen uso de una forma indirecta, así los enfoques basados en discretización por amplitud fijan de forma temprana el valor, lo que puede dar lugar tanto a zonas densas como poco densas, perdiendo de esta forma posibles reglas valiosas. Técnicas evolutivas también han usado este concepto como una estrategia para evitar reglas solapadas en la definición de las funciones objetivo [270][287].

En la experimentación llevada a cabo se observa en la Tabla 4.23 que QAR_VMO, QUANTMINER y QAR_CIP_NSGAII-A obtienen los valores más bajos de amplitud. Note que valores bajos son los esperados por nuestra técnica para esta medida de acuerdo con la función de evaluación. Con valores de amplitud dentro del primer cuartíl QARM_VMO alcanza zonas de mayor concentración en correspondencia con los umbrales de soporte y confianza

fijados.

Lift es una métrica alternativa propuesto por diferentes autores. Lift mostrará la correlación de dos elementos dada una regla. Es un factor por el cual la ocurrencia de los dos elementos como un conjunto supera la probabilidad de que esos elementos ocurran juntos de forma independiente. Lo que implica que cuanto mayor sea el valor de lift, mayor será la probabilidad de que los dos elementos en cuestión aparezcan juntos. Esta métrica podría ser definida como una medida de correlación, calculando el grado de dependencia entre el antecedente y el consecuente, así los valores menores a uno indican una dependencia negativa, valores mayores a uno corresponde a una dependencia positiva y cuando son iguales a uno se define como independencia.

Datos	APRIORI	GAR	GENAR	MODENAR	NSGAII	QARM_VMO	QUANTMINER
basketball	3.56	1.01	1.16	1.07	1.92	1.58	1.60
iris	4.12	2.74	1.70	2.24	3.42	2.75	2.59
wine	3.53	1.07	1.87	1.91	3.67	2.40	1.99
glass	3.18	1.32	1.00	1.02	1.87	1.79	1.67
abaloneClass	3.31	1.57	1.05	1.26	3.90	3.42	2.87
dee	3.56	1.08	1.43	1.39	2.63	2.14	1.97
pima	3.19	1.01	1.03	0.99	2.14	1.58	1.51
bupa	2.25	1.12	1.01	1.12	1.29	1.31	1.25
ecoli	2.82	1.38	1.44	1.00	4.01	2.27	2.11
vehicle	3.17	1.05	1.11		3.36	2.72	2.79

Tabla 4.24: Resultados con algoritmos, métrica elevación(Lift) promedio

La Tabla 4.24 muestra valores de *lift* para los siete algoritmos de prueba. En la mayor parte de los casos se nota valores promedio relativamente superiores a uno, esto indica, que esos conjuntos aparecen una cantidad de veces superior a lo esperado, bajo condiciones de independencia (por lo que se puede intuir que existe una relación, que hace que los ítems se encuentren en el conjunto más veces de lo normal).
También se ha verificado valores cercanos a uno, pero no menores. Con la misma base que define la medida, se puede decir que estos valores están más cercanos a una independencia entre el antecedente y consecuente de las reglas encontradas por los algoritmos.

En el ámbito de algoritmos evolutivos, QARM_VMO es el segundo algoritmo por detrás de QAR_CIP_NSGAII-A con valores mas altos en esta métrica, seguido también de QUANTMINER, que se ubica en el tercer lugar. El más bajo puntaje lo tiene GENAR y GAR.

En general, es decir cuando todos los tipos de algoritmos son considera-

dos, Apriori ocupa el primer lugar, pero este lugar es alcanzado a costa del sacrificio de otras métricas importantes.

Datos	APRIORI	GAR	GENAR	MODENAR	NSGAII	QARM_VMO	QUANTMINER
basketball	0.32	0.13	0.00	1.28	4.82	1.43	1.35
iris	1.09	0.61	0.04	1.00	10.06	3.92	3.40
wine	0.33	0.63	0.00	0.02	10.94	4.48	3.76
glass	0.04	1.34	0.00	0.35	5.43	5.98	6.03
abaloneClass	6.80	8.15	0.00	6.45	226.50	155.11	146.66
dee	1.50	1.50	0.00	0.50	14.64	7.56	6.65
pima	0.34	2.34	0.00	0.04	46.26	13.84	13.70
bupa	0.26	0.91	0.00	0.32	5.67	5.53	5.23
ecoli	0.16	3.39	0.00	0.00	17.61	6.25	5.50
vehicle	0.24	1.93	0.00	-	36.46	67.79	68.65

Tabla 4.25: Resultados con algoritmos, función de evaluación QARM_VMO promedio

En la Tabla 4.25 se tienen valores correspondientes a la función de ajuste implementada en QARM_VMO, con el objetivo de reunir los criterios de soporte, confianza y amplitud, se agrega como una medida de comparación adicional, sin embargo, un procedimiento de normalización fue requerido para ser insertado en todas la técnicas. Esta es una función sujeta a maximización, por tanto valores altos señalan mejor calidad, y valores bajos (incluyendo inferiores a cero en algunos casos) indican menor calidad de resultado.

Para utilizar esta medida propia de QARM_VMO en los demás algoritmos, una función fue construida en R-Project, de tal forma, que con los resultados de las medidas alcanzadas por cada técnica se crea la función. La inserción de un procedimiento para el cálculo de la amplitud de la regla, y su normalización fue requerido como parte del proceso.

Se esperaban resultados superiores en relación a todas las otras técnicas, sin embargo, se puede apreciar que QAR_CIP_NSGAII-A está por encima de QARM_VMO y seguido de cerca por la versión genética QUANTMINER. Estos resultados permiten afirmar que QAR_CIP_NSGAII-A optimiza eficientemente los criterios considerados en la FO de la investigación. Mientras que respecto al caso de QUANTMINER, es claro que la técnica basada en VMO está dando mejores resultados en cuanto a calidad.

A continuación se muestran los resultados alcanzados para la medida (FO de QARM_VMO) con algoritmos evolutivos que tienen implementaciones basados en una función monoobjetivo. El detalle de los datos se puede ver en la Tabla 4.26. Mientras más alto son los valores, estos tienen mayor calidad.

Datos	GAR	GENAR	QARM_VMO	QUANTMINER
basketball	0.13	0.00	1.43	1.35
iris	0.61	0.04	3.92	3.40
wine	0.63	0.00	4.48	3.76
glass	1.34	0.00	5.98	6.03
abalone	8.15	0.00	155.11	146.66
dee	1.50	0.00	7.56	6.65
pima	2.34	0.00	13.84	13.70
bupa	0.91	0.00	5.53	5.23
ecoli	3.39	0.00	6.25	5.50
vehicle	1.93	0.00	67.79	68.65

Tabla 4.26: Función de evaluación QARM_VMO promedio, algoritmos evolutivos simple objetivo

4.3.5. Prueba de hipótesis

Los criterios sobre el rendimiento de los algoritmos no pueden estar sujetos a simples observaciones. Para garantizar las diferencias con respecto a alguna medida de calidad, se conducen pruebas para determinar diferencias estadísticamente significativas entre el comportamiento de los algoritmos. Las pruebas de hipótesis de Friedman y post hoc Friedman-Nemenyi, fueron utilizados mediante las funciones

- friedman.test
- PMCMR::posthoc.friedman.nemenyi.test

La prueba de Friedman se aplicó a los datos contenidos en la Tabla 4.25, estos son datos obtenidos por la función de ajuste que implementa QARM_VMO y que ha sido calculado también en las otras técnicas.

La prueba de Friedman es un análisis de varianza de bloques aleatorios no paramétricos. Es decir, es una versión no paramétrica de un ANOVA unidireccional con medidas repetidas. Eso significa que, si bien una prueba ANOVA simple requiere los supuestos de una distribución normal y varianzas iguales (de los residuos), la prueba de Friedman está libre de esas restricciones. El precio de esta libertad paramétrica es la pérdida de potencia (de la prueba de Friedman en comparación con las versiones paramétricas ANOVA).

Las hipótesis para la comparación entre medidas repetidas son:

- H0: Las distribuciones (sean las que sean) son las mismas en medidas repetidas
- H1: las distribuciones entre medidas repetidas son diferentes

La Tabla 4.27 detalla los resultados del análisis post-hoc para una prueba de Friedman. En el grupo de evolutivos monoobjetivo, se descubre que hay diferencia significativa entre los cuatro algoritmos (p-value = 0.000005), y el análisis post-hoc muestra que esa diferencia es principalmente debido a que QARM_VMO tiene diferencia significativa en cuanto a calidad con algoritmos *GAR* y *GENAR*. También se puede evidenciar diferencias entre QARM_VMO y el algoritmo *APRIORI* (p-value = 0.001565), y en relación a la comparación de la técnica propuesta con dos algoritmos multiobjetivo, se descubre diferencia significativa entre las tres técnicas (p-value = 0.00016), donde también interviene QARM_VMO con *MODENAR*.

Grupo	Algoritmos	Valor p
Evolutivos simple objetivo	GENAR - GAR	0.306957
	QUANTMINER - GAR	0.160126
	QARM_VMO - GAR	**0.009855**
	QUANTMINER - GENAR	0.000772
	QARM_VMO - GENAR	**0.000009**
	QARM_VMO - QUANTMINER	0.726353
Clásico	**QARM_VMO - APRIORI**	**0.001565**
Evolutivos Multi objetivo	NSGAII - MODENAR	0.000182
	QARM_VMO - MODENAR	**0.020002**
	QARM_VMO - NSGAII	0.372027

Tabla 4.27: Prueba de Friedman Post hoc para medida calidad de FO QARM_VMO

Para corroborar el test de Friedman, que de hecho resulta con diferencias significativas. Adicional se realiza las comparaciones por pares mediante el test de Wilcoxon con corrección de holm que se muestra en la Tabla 4.28, se nota que coincide en la significación, además podría tener diferencia con QUANTMINER (*p=0.064* que es casi significativo). Es notable la diferencia que existe con el algoritmo clásico *APRIORI*, y se verifica la no existencia de diferencias entre QAR_CIP_NSGAII-A y la técnica propuesta (QARM_VMO)

		GAR	GENAR	QUANTMINER
Evolutivos	GENAR	0.012	-	-
Simple	QUANTMINER	0.012	0.012	-
Objetivo	**QARM_VMO**	**0.012**	**0.012**	0.064
		APRIORI		
Clasico	**QARM_VMO**	**0.002**		
		MODENAR	NSGAII	
Evolutivos	NSGAII	0.0059	-	-
multiobjetivo	**QARM_VMO**	**0.0059**	0.084	

Tabla 4.28: Prueba de Wilcoxon rangos con signo para medida de calidad de FO QARM_VMO

Capítulo 5

Conclusiones

En esta Tesis se ha abordado el problema de la minería de reglas de asociación desde una perspectiva de optimización, utilizando la metaheurística VMO para la optimización de intervalos en los atributos numéricos, considerando un esquema de regla definido por el usuario, que puede componerse tanto de atributos numéricos como categóricos. Primeramente, se ha introducido la problemática relativa a esta área de investigación. Luego, se han analizado las diferentes técnicas sobre el problema enfocado a minería de reglas de asociación, haciendo énfasis en las cuantitativas. Se cubre, desde los principales enfoques aplicados de forma clásica por la práctica de minería de datos, hasta los trabajos actuales basados en técnicas de optimización mediante computación evolutiva. Como resultados de esta Tesis se ha presentado: Un algoritmo para extracción de reglas de asociación numéricas, un método novedoso de búsqueda de reglas basado en un esquema definido por el usuario, y un estudio comparativo de la técnica propuesta con otros algoritmos de su clase. En los siguientes puntos se presentan las conclusiones de cada uno de los trabajos y métodos expuestos.

5.1. Conclusiones respecto al estado del arte

Desde la aparición del enfoque de minería de reglas de asociación cuantitativas, muchas técnicas han sido estudiadas para dar solución a problemas derivados de los conjuntos de datos numéricos. ARM desde su origen se planteó como un problema de tipo combinatorio en donde intervienen elementos discretos, pero a medida que su importancia y utilidad fueron creciendo, este se ha ido adaptando a conjuntos de datos diversos, y a varias técnicas de procesamiento, que unidos a las capacidades de cómputo actuales, se obtienen algoritmos bastante competitivos.

En general, lo que dificulta el tratamiento de un atributo cuyo dominio es muy amplio, como el caso de una variable numérica, es la necesidad de convertir estos datos a un tipo manejable sin alterar su información, pues el problema está en encontrar los intervalos más óptimos para la formulación de reglas de calidad; de esta forma se tiene varias técnicas, tales como: *particionamiento, agrupamiento, estadísticos, técnicas difusas y algoritmo evolutivos.* A continuación se describen algunas conclusiones importantes de cada uno.

La *discretización de la variable numérica* es el primer método aplicado por [76], para tratar este problema, en sus primeros trabajos sobre ARM, ya se incluye una técnica para lidiar con variables numéricas. Los atributos numéricos tienen características diferentes a los atributos categóricos, por lo que requieren otros métodos para su estudio. Es así que la discretización de las variables numéricas ha sido el fundamento teórico utilizado para desarrollar los primeros aportes en esta área. Sin embargo, esta técnica trae consigo el problema de la sensibilidad del *soporte y confianza* , debido a que la presencia de muchos intervalos ocasiona valores de *soporte* bajos para cada intervalo, dando lugar a la generación de muy pocas reglas; por otra parte, si el tamaño individual de los intervalos se incrementa, la perdida de información persiste en términos de *confianza.* La selección del método de discretización es otro problema para la técnica de particionamiento, que se ha tratado de lidiar mediante el conocimiento a priori de su aplicación posterior a la discretización. De allí que una técnica supervisada puede ser una opción para fines de clasificación.

El *análisis estadístico de las variables* es otra estrategia utilizada para determinar patrones en las reglas de asociación numérica. En este caso el consecuente de la regla puede corresponder a la distribución de las variables numéricas del antecedente.
Algunos investigadores consideran que la distribución de los valores en los atributos numéricos puede tener una representación generalizada de una definición categórica cuando se utilizan las medidas estadísticas adecuadas [83]. Es así que basado en esta teoría se introduce una definición de partición de dominio y proponen técnicas de bi-partición basadas en la media, mediana y minimización de la desviación estándar. Aunque la estrategia permite identificar zonas densas y extraordinarias, sin embargo, presentan la dificultad para tratar con reglas de más de una dimensión y que combinen atributos categóricos.

La *teoría de agrupación*, que generalmente se atribuye a la clasificación en minería de datos, se ha extendido a los procedimientos de discretización. En este contexto estrategias eficientes como el agrupamiento jerárquico basado en la variación de densidad han sido aplicados para resolver el problema de la partición de intervalos. Este método encuentra reglas de asociación cuantitativas difusas eficientes, alcanzando una división exitosa de atributos numéricos en los mejores intervalos.

El concepto de *lógica difusa* también es ampliamente usado en la resolución de problemas de minería de reglas de asociación cuantitativas, así se tienen aportes relevantes como la construcción de un algoritmo con capacidad para incluir términos ambiguos tanto en el antecedente como en el consecuente. La calidad de solución en una técnica basada en métodos difusos puede verse comprometida cuando la selección del conjunto difuso no es adecuada. Introducción de procedimientos de normalización en los conjuntos difusos logran mejoras en cuanto a la diversidad de reglas, aunque el impacto no se ha reflejado significativamente en la calidad.

Finalmente, la *computación evolutiva* se ha utilizado con éxito para optimizar las reglas de asociación tanto cualitativas como cuantitativas. Los distintos algoritmos existentes en este segmento se han adaptado y ajustado al problema a escala en términos de calidad y variedad. Los algoritmos genéticos han sido los primeros métodos evolutivos aplicados en esta problemática, posteriormente la introducción de metaheurísticas basadas en población en los últimos años, han dado resultados satisfactorios, pues se ha conseguido, reducir los costos computacionales asociados a los algoritmos genéticos, debido a su rápida convergencia hacia las soluciones; [89] desarrollan un algoritmo denominado MOPAR, basado en PSO para hallar reglas de asociación numéricas, usando un enfoque multiobjetivo, los resultados mostraron que MOPAR extrae AR numéricas confiables (con valores de confianza cercanos al 95 %), comprensibles e interesantes.

5.2. Conclusiones respecto a la técnica aplicada

Se ha estudiado el entorno de QUANTMINER, un framework de libre distribución para implementación de técnicas basadas en reglas de asociación. Su implementación basado en lenguaje de programación Java ha facilitado la reutilización de funciones significativas tales como: las interfaces de usuario tanto para entrada y salida de información, la estructura de las reglas de asociación y procedimientos para el cálculo de métricas asociadas a la calidad.

El ambiente QUANTMINER es apropiado para agregar implementaciones de algoritmos para minería de reglas de asociación. Sus módulos permiten lidiar con conjuntos de datos tanto cualitativos como cuantitativos, con un mecanismo de ingreso de parámetros bastante intuitivo, permite la personalización de los atributos para la definición de un esquema de regla. Sin embargo, para implementaciones que no requieran de la definición de esquemas, los costos de adaptación pueden llegar a ser demasiado altos.

Los esquemas de regla son mecanismos que permiten la personalización de los atributos y dimensiones de las reglas de asociación. Este enfoque reduce relativamente el tiempo de optimización, cuando el usuario establece directamente las especificaciones de la regla, así el algoritmo no tiene que ocupar recursos en la exploración de reglas innecesarias. Sin embargo, esta estrategia puede incurrir en incrementos considerables de tiempo, cuando se ha generado un conjunto de sub-esquemas de reglas a partir de la plantilla general, debido a que varias reglas se encolan para la optimización.

La función objetivo utilizada permite la extracción de reglas de asociación en función de los umbrales de soporte y confianza mínimo establecidos por los usuarios, como consecuencia el algoritmo extrae reglas que convergen al valor mínimo de soporte establecido y por encima del valor de confianza definido, tienden al máximo valor de la métrica, es decir uno.

QARM_VMO genera resultados satisfactorios, en la experimentación controlada fueron construidos artificialmente conjuntos de datos a partir de reglas de asociación definidas de manera a priori, los resultados obtenidos en relación a las medidas de soporte y confianza tienen similitud con los valores esperados, esto puede verificarse en las tablas 4.12 y 4.13.

Las pruebas de sensibilidad para los parámetros de QARM_VMO se realizaron tanto en conjuntos de datos sintéticos como reales, y los resultados definen que el número de iteraciones recomendado está sobre el valor de 200, que es donde la convergencia de la función de evaluación alcanza una mayor estabilidad, esto es apreciado también en las medidas de soporte y confianza. El tamaño de la población inicial se determina en el valor de 12, por tanto el número de nodos en la malla final es 36, este resultado es conseguido por una función que evalúa la menor varianza respecto de la posición en cuanto a calidad, alcanzado en la experimentación con datos sintéticos de diferente dimensión y en seis conjuntos de datos reales.

5.3. Conclusiones respecto al modelo de adaptación

Optimización por Mallas Variables (VMO) es una metaheurística poblacional, que ha sido exitosamente utilizado en su versión continua, para la optimización de funciones unimodales y multimodales. En el proceso de adaptación se han modificado el algoritmo hacia un enfoque de optimización discreta, así la estructura del *nodo* (elemento que constituye el individuo de la población) ha sido adaptado para trabajar con las variables de operación del problema de optimización. Adicional a esto, se incluye mecanismos de construcción de reglas de asociación a partir del esquema durante la etapa de actualización de los valores de operación.

El acoplamiento del algoritmo QARM_VMO con las estructuras de QUANTMINER se consigue por medio de una clase denominada *optimizer* y *EvaluationBase*", que es utilizado en la herramienta con todos los algoritmos que tiene registrado. *Optimizer* contiene métodos y propiedades que permite la recepción de parámetros y del esquema de regla que proviene del módulo principal, adicional tiene implementaciones para construir las soluciones en el formato de regla de asociación propio de QUANTMINER. La clase *EvaluationBase* tiene funciones y propiedades para la evaluación de la función de ajuste, cálculo de métricas de calidad, gestión de las reglas de asociación potenciales definidas en la población.

5.4. Conclusiones respecto al estudio comparativo

Pruebas comparativas de rendimiento fueron efectuados con diez conjuntos de datos reales y seis algoritmos para extracción de reglas. Este grupo se compone del tradicional *APRIORI*, dos algoritmos evolutivos multiobjetivo *MODENAR-A* y *QAR_CIP_NSGAII-A*, dos algoritmos monoobjetivo *GAR* y *GENAR*, también se agrega la versión genética de QUANTMINER. Se consiguió integrar los resultados de las pruebas en la herramienta R-project mediante el uso de la librería RKeel, que tienen las implementaciones de los algoritmos en el segmento de reglas de asociación, mientras que para QUANTMINER y QARM_VMO los datos fueron generados desde Java y leídos desde R-project.

Entre los algoritmos evolutivos monoobjetivo, QARM_VMO alcanzó el valor más alto de calidad obtenido por la función de ajuste, la misma que fue implementada como una función adicional en R-project para el cálculo en las otras técnicas probadas. En comparación con el algoritmo *GAR*, para el peor de los casos QARM_VMO está 1.31 unidades por encima, en el conjunto de datos *basketball*, y en el mejor de los casos está 146.95 unidades por encima en el conjunto de datos *abalone*. En relación al algoritmo *GENAR*, este algoritmo obtuvo valores muy bajos para la función de ajuste, cediendo ventaja considerable a QARM_VMO. Finalmente la versión genética de QUANTMINER está ligeramente por debajo de QARM_VMO en la mayor parte de casos con una diferencia menor a uno, y en dos conjuntos de datos de los diez, QUANTMINER resulta ligeramente superior.

Para la función de calidad implementada en la técnica propuesta y normalizada en el resto de técnicas probadas. Por una parte, en comparación con las versiones genéticas monoobjetivo, QARM_VMO demuestra superioridad en comparación a *GAR* y *GENAR*, y con respecto a QUANTMINER, aunque es ligeramente superior en los resultados, no tiene una diferencia significativa. Por otro lado con respecto al algoritmo clásico *APRIORI* se evidencia que QARM_VMO es superior, y finalmente en comparación con dos técnicas multiobjetivo, se pudo evidenciar que es superior a *MODENAR*, y no hay diferencia significativa con *QAR_CIP_NSGAII-A*.

Bibliografía

[1] Raghavendra Kune, Pramod Kumar Konugurthi, Arun Agarwal, Raghavendra Rao Chillarige, and Rajkumar Buyya. The anatomy of big data computing. *Software: Practice and Experience*, 46(1):79–105, 2016.

[2] Chunhua Fu, Xiaojing Wang, Lijun Zhang, and Liying Qiao. Mining algorithm for association rules in big data based on hadoop. *AIP Conference Proceedings*, 1955(1):040035, 2018.

[3] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery in databases. *AI Magazine*, 17(3):37, Mar. 1996.

[4] Marcos D Assunção, Rodrigo N Calheiros, Silvia Bianchi, Marco AS Netto, and Rajkumar Buyya. Big data computing and clouds: Trends and future directions. *Journal of Parallel and Distributed Computing*, 79:3–15, 2015.

[5] David E Goldberg. Genetic and evolutionary algorithms come of age. *Communications of the ACM*, 37(3):113–120, 1994.

[6] Kwok-leung Tsui. Association Rules and Market Basket Analysis. In *Response*, chapter 15, pages 1–47. John Wiley and Sons, Indianapolis, 3 edition, 2009.

[7] Etsuko Sugawara and Hiroshi Nikaido. Properties of AdeABC and AdeIJK efflux systems of Acinetobacter baumannii compared with those of the AcrAB-TolC system of Escherichia coli. *Antimicrobial Agents and Chemotherapy*, 58(12):7250–7257, 2014.

[8] Patrice Degoulet and Marius Fieschi. Medical Decision Support Systems. In *Introduction to Clinical Informatics*, volume 6, chapter 25, pages 153–167. Springer, Netherlands, 1997.

[9] Yue Huang, Paul McCullagh, Norman Black, and Roy Harper. Evaluation of outcome prediction for a clinical diabetes database. In Jesús A López, Emilio Benfenati, and Werner Dubitzky, editors, *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)*, volume 3303, pages 181–190, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[10] Kasun S. Perera, Bijay Neupane, Mustafa Amir Faisal, Zeyar Aung, and Wei Lee Woon. Mining Intelligence and Knowledge Exploration. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 8284, pages 370–382, Cham, 2013. Springer International Publishing.

[11] Dhrubajit Adhikary and Swarup Roy. Trends in quantitative association rule mining techniques. In *2015 IEEE 2nd International Conference on Recent Trends in Information Systems, ReTIS 2015 - Proceedings*, pages 126–131, Kolkata, India, 2015. IEEE.

[12] Hernando Camargo and Mario Silva. Dos caminos en la búsqueda de patrones por medio de Minería de Datos : SEMMA y CRISP. *Revista de Tecnología*, 9(1):11–18, 2010.

[13] Bing Liu, Wynne Hsu, Yiming Ma, and Blwhy Ma. Integrating Classification and Association Rule Mining. In *Knowledge Discovery and Data Mining*, KDD'98, pages 80–86. AAAI Press, 1998.

[14] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB 1994, pages 487–499, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc.

[15] Jochen Hipp, Ulrich Güntzer, and Gholamreza Nakhaeizadeh. Algorithms for association rule mining - a general survey and comparison. *SIGKDD Explor. Newsl.*, 2(1):58–64, June 2000.

[16] Tabeen Tasneem, Tazeen Tasneem, and Mir Md. Jahangir Kabir. Performance analysis of classical and evolutionary algorithms for mining association rules. In *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, pages 1–6, 2019.

[17] Jesus Alcala-Fdez, Nicolo Flugy-Pape, Andrea Bonarini, and Francisco Herrera. Analysis of the effectiveness of the genetic algorithms based

on extraction of association rules. *Fundam. Inf.*, 98(1):1–14, January 2010.

[18] Pradnya A. Vikhar. Evolutionary algorithms: A critical review and its future prospects. In *2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC)*, pages 261–265, 2016.

[19] Amilkar Puris, Rafael Bello, Daniel Molina, and Francisco Herrera. Variable mesh optimization for continuous optimization problems. *Soft Computing*, 16(3):511–525, 2012.

[20] Amilkar Yudier Puris Cáceres, Pavel Novoa Hernández, and Byron Oviedo Bayas. Desarrollo de metaheurísticas poblacionales para la solución de problemas complejos, 2020.

[21] Gregory Piatetsky-Shapiro. Knowledge discovery in real databases: A report on the ijcai-89 workshop. *AI Magazine*, 11(4):68, Dec. 1990.

[22] Gartner Inc. gartner it glosary datamining. `http://www.gartner.com/it-glossary/data-mining`. Accessed: 18-Oct-2016.

[23] Chinta Someswara Rao, D. Ravi Babu, . Shiva Shankar, V. Pradeep Kumar, J. Rajanikanth, and Ch. Chandra Sekhar. Mining Association Rules Based on Boolean Algorithm - a Study in Large Databases. *International Journal of Machine Learning and Computing*, 3(Icmlc):347–351, 2013.

[24] ISO Central Secretary. Information technology – Database languages – SQL – Part 2: Foundation. Standard ISO/IEC CD 9075-2:2021, International Organization for Standardization, Geneva, CH, 2021.

[25] K.C. Laudon and J.P. Laudon. *Sistemas de información gerencial*, chapter 1, pages 2–32. Pearson Educación, Mexico,MX, 12 edition, 2016.

[26] Paul Zikopoulos, Chris Eaton, Dirk Deroos, Tom Deutsch, and George Lapis. *What Is Big Data? Hint: You're a Part of It Every Day.* McGraw-Hill, USA, 1 edition, 2012.

[27] S. Kannan, S. Karuppusamy, A. Nedunchezhian, P. Venkateshan, P. Wang, N. Bojja, and A. Kejariwal. Chapter 3 - big data analytics for social media. In Rajkumar Buyya, Rodrigo N. Calheiros, and Amir Vahid Dastjerdi, editors, *Big Data*, pages 63–94. Morgan Kaufmann, 2016.

[28] Doug Laney et al. 3d data management: Controlling data volume, velocity and variety. *META group research note*, 6(70):1, 2001.

[29] Charles W. Bachman. The evolution of storage structures. *Commun. ACM*, 15(7):628–634, July 1972.

[30] E. F. Codd. A relational model of data for large shared data banks. *Commun. ACM*, 13(6):377–387, June 1970.

[31] Ossi Ylijoki and Jari Porras. Perspectives to definition of big data: a mapping study and discussion. *Journal of Innovation Management*, 4(1):69–91, 2016.

[32] Rob Kitchin. Big data, new epistemologies and paradigm shifts. *Big data & society*, 1(1):2053951714528481, 2014.

[33] Viktor Mayer-Schönberger and Kenneth Cukier. *Big data: A revolution that will transform how we live, work, and think.* Houghton Mifflin Harcourt, 2013.

[34] Thomas M. Mitchell. *Machine Learning.* McGraw-Hill, Inc., USA, 1 edition, 1997.

[35] Alex Guazzelli, Wen-Ching Lin, and Tridivesh Jena. *PMML in Action: Unleashing the Power of Open Standards for Data Mining and Predictive Analytics.* CreateSpace, Scotts Valley, CA, 2nd edition, 2012.

[36] Marco Loog. Chapter 5 - supervised classification: Quite a brief overview. In Enrico Camporeale, Simon Wing, and Jay R. Johnson, editors, *Machine Learning Techniques for Space Weather*, pages 113–145. Elsevier, 2018.

[37] Wonkook Kim and Shunlin Liang. *Unsupervised Classification*, pages 1–5. American Cancer Society, 2017.

[38] Maria-Florina Balcan and Nika Haghtalab. Noise in Classification. *arXiv e-prints*, page arXiv:2010.05080, oct 2020.

[39] SAS Institute Inc. Sas enterprise miner: Reveal valuable insights with powerful data mining software, 2015.

[40] Rama Shankar. *Process improvement using six sigma: a DMAIC guide.* Quality Press, 2009.

[41] Daniel T Larose and Chantal D Larose. *Discovering knowledge in data: an introduction to data mining*, volume 4. John Wiley & Sons, 2014.

[42] Ivar Jacobson, Grady Booch, and James Rumbaugh. *The Unified Software Development Process*. Addison-Wesley Longman Publishing Co., Inc., USA, 1999.

[43] EMC Digital Universe with Research & Analysis by IDC. The digital universe of opportunities: Rich data and the increasing value of the internet of things, 2014.

[44] Reed T Sutton, David Pincock, Daniel C Baumgart, Daniel C Sadowski, Richard N Fedorak, and Karen I Kroeker. An overview of clinical decision support systems: benefits, risks, and strategies for success. *NPJ digital medicine*, 3(1):1–10, 2020.

[45] Sophie Pilleron, Diana Sarfati, Maryska Janssen-Heijnen, Jérôme Vignat, Jacques Ferlay, Freddie Bray, and Isabelle Soerjomataram. Global cancer incidence in older adults, 2012 and 2035: a population-based study. *International journal of cancer*, 144(1):49–58, 2019.

[46] Kyu-Won Jung, Young-Joo Won, Hyun-Joo Kong, and Eun Sook Lee. Prediction of cancer incidence and mortality in korea, 2019. *Cancer research and treatment: official journal of Korean Cancer Association*, 51(2):431, 2019.

[47] Saba Maleki Birjandi and Seyed Hossein Khasteh. A survey on data mining techniques used in medicine. *Journal of Diabetes & Metabolic Disorders*, pages 1–17, 2021.

[48] J Ferlay, M Colombet, I Soerjomataram, T Dyba, G Randi, M Bettio, A Gavin, O Visser, and F Bray. Cancer incidence and mortality patterns in europe: Estimates for 40 countries and 25 major cancers in 2018. *European journal of cancer*, 103:356–387, 2018.

[49] Samir S Yadav, Shivajirao M Jadhav, Snigdha Nagrale, and Niraj Patil. Application of machine learning for the detection of heart disease. In *2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, pages 165–172. IEEE, 2020.

[50] Keerti Shrivastava and Varsha Jotwani. A comparative analysis of various data mining techniques to predict heart disease. In *Expert Clouds and Applications*, pages 283–296. Springer, 2022.

[51] Mafizur Rahman, Maryam Mehzabin Zahin, and Linta Islam. Effective prediction on heart disease: Anticipating heart disease using data mining techniques. In *2019 International Conference on Smart Systems and Inventive Technology (ICSSIT)*, pages 536–541. IEEE, 2019.

[52] Jiaxin Li, Zijun Zhou, Jianyu Dong, Ying Fu, Yuan Li, Ze Luan, and Xin Peng. Predicting breast cancer 5-year survival using machine learning: A systematic review. *PloS one*, 16(4):e0250370, 2021.

[53] Habib Dhahri, Eslam Al Maghayreh, Awais Mahmood, Wail Elkilani, and Mohammed Faisal Nagi. Automated breast cancer diagnosis based on machine learning algorithms. *Journal of healthcare engineering*, 2019, 2019.

[54] Jiande Wu and Chindo Hicks. Breast cancer type classification using machine learning. *Journal of personalized medicine*, 11(2):61, 2021.

[55] Ashish Goyal, Maheshwar Kuchana, and Kameswari Prasada Rao Ayyagari. Machine learning predicts live-birth occurrence before in-vitro fertilization treatment. *Scientific reports*, 10(1):1–12, 2020.

[56] Jiahui Qiu, Pingping Li, Meng Dong, Xing Xin, and Jichun Tan. Personalized prediction of live birth prior to the first in vitro fertilization treatment: a machine learning method. *Journal of translational medicine*, 17(1):1–8, 2019.

[57] Boran Sekeroglu, Kamil Dimililer, and Kubra Tuncal. Student performance prediction and classification using machine learning algorithms. In *Proceedings of the 2019 8th International Conference on Educational and Information Technology*, ICEIT 2019, pages 7–11, New York, NY, USA, 2019. Association for Computing Machinery.

[58] J Dhilipan, N Vijayalakshmi, S Suriya, and Arockiya Christopher. Prediction of students performance using machine learning. In *IOP Conference Series: Materials Science and Engineering*, volume 1055, page 012122. IOP Publishing, 2021.

[59] Argelia Berenice Urbina Nájera and Jorge de la Calleja Mora. Brief review of educational applications using data mining and machine learning. *Revista electrónica de investigación educativa*, 19(4):84–96, 2017.

[60] Leila Ismail, Huned Materwala, and Alain Hennebelle. Comparative analysis of machine learning models for students'performance prediction. In Tatiana Antipova, editor, *Advances in Digital Science*, pages 149–160, Cham, 2021. Springer International Publishing.

[61] Feng Guo and Huilin Qin. Data mining techniques for customer relationship management. *Journal of Physics: Conference Series*, 910:012021, oct 2017.

[62] Kun Wu and Feng-ying Liu. Application of data mining in customer relationship management. In *2010 International Conference on Management and Service Science*, pages 1–4, 2010.

[63] Loredana MOCEAN and Ciprian Marcel POP. Marketing Recommender Systems: A New Approach in Digital Economy. *Informatica Economica*, 16(4):142–149, 2012.

[64] Wei Deng, Yong Shi, Zhengxin Chen, Wikil Kwak, and Huimin Tang. Recommender system for marketing optimization. *World Wide Web*, 23(3):1497–1517, 2020.

[65] Chunyong Yin, Shilei Ding, and Jin Wang. Mobile marketing recommendation method based on user location feedback. *Human-centric computing and information sciences*, 9(1):1–17, 2019.

[66] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, SIGMOD ́93, pages 207–216, New York, NY, USA, 1993. Association for Computing Machinery.

[67] Jiawei Han, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011.

[68] Pan-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining-Instructors Solution Manual*. Pearson, 2006.

[69] Jong Soo Park, Ming-Syan Chen, and Philip S. Yu. An effective hash-based algorithm for mining association rules. SIGMOD '95, pages 175–186, New York, NY, USA, 1995. Association for Computing Machinery.

[70] Sergey Brin, Rajeev Motwani, Jeffrey D. Ullman, and Shalom Tsur. Dynamic itemset counting and implication rules for market basket data. In *Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data*, SIGMOD 1997, pages 255–264, New York, NY, USA, 1997. Association for Computing Machinery.

[71] Ashoka Savasere, Edward Omiecinski, and Shamkant B. Navathe. An efficient algorithm for mining association rules in large databases. In

Proceedings of the 21th International Conference on Very Large Data Bases, VLDB 1995, pages 432–444, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[72] Hannu Toivonen. Sampling large databases for association rules. In T. M. Vijayaraman, Alejandro P. Buchmann, C. Mohan, and Nandlal L. Sarda, editors, *Proceedings of the 22nd International Conference on Very Large Data Bases (VLDB'96)*, pages 134–145, Förenta Staterna (USA), September 1996. Morgan Kaufmann Publishers.

[73] Sajid Mahmood, Muhammad Shahbaz, and Aziz Guergachi. Negative and positive association rules mining from text using frequent and infrequent itemsets. *The Scientific World Journal*, 2014, 05 2014.

[74] Sikha Bagui and Probal Chandra Dhar. Mining positive and negative association rules in hadoop's mapreduce environment. In *Proceedings of the ACMSE 2018 Conference*, ACMSE 2018, New York, NY, USA, 2018. Association for Computing Machinery.

[75] Maria-Luiza Antonie and Osmar R. Zaïane. Mining positive and negative association rules: An approach for confined rules. In Jean-François Boulicaut, Floriana Esposito, Fosca Giannotti, and Dino Pedreschi, editors, *Knowledge Discovery in Databases: PKDD 2004*, pages 27–38, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[76] Ramakrishnan Srikant and Rakesh Agrawal. Mining Quantitative Association Rules in Large Relational Tables. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, SIGMOD '96, pages 1–12, New York, NY, USA, 1996. ACM.

[77] R Ulrich, Lothar Richter, Stefan Kramer, and Technische Universit. Quantitative Association Rules Based on Half-Spaces : An Optimization Approach. In *Data Mining, 2004. ICDM'04. Fourth IEEE International Conference on*, pages 507–510, Brington, UK, 2004. IEEE.

[78] R J Miller and Y Yang. Association Rules over Interval Data. *Proceedings of the 1997 ACM SIGMOD international conference on Management of data*, pages 452–461, 1997.

[79] Keith C. C. Chan and Wai-Ho Au. An effective algorithm for mining interesting quantitative association rules. *Proceedings of the 1997 ACM symposium on Applied computing - SAC '97*, pages 88–90, 1997.

[80] Chunyao Song, Tingjian Ge, Chunyao Song, and Tingjian Ge. Discovering and managing quantitative association rules. *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management - CIKM '13*, pages 2429–2434, 2013.

[81] Ansaf Salleb-Aouissi, Christel Vrain, and Cyril Nortet. QuantMiner: A genetic algorithm for mining quantitative association rules. In *IJCAI International Joint Conference on Artificial Intelligence*, IJCAI'07, pages 1035–1040, Hyberadad, India, 2007. Morgan Kaufmann Publishers Inc.

[82] Dhrubajit Adhikary and Swarup Roy. A new equivalence class based approach for discretizing quantitative data using Point Shift Mechanism. In *2015 International Symposium on Advanced Computing and Communication (ISACC)*, pages 174–180, Silchar, India, sep 2015. IEEE.

[83] Yonatan Aumann and Yehuda Lindell. A statistical theory for quantitative association rules. *Journal of Intelligent Information Systems*, 20(3):255–283, 2003.

[84] Gong Mi Kang, Yang Sae Moon, Hun Young Choi, and Jinho Kim. Bipartition techniques for quantitative attributes in association rule mining. *IEEE Region 10 Annual International Conference, Proceedings/TENCON*, pages 1–6, 2009.

[85] Been-chian Chien and Zin-long Lin. An efficient clustering algorithm for mining fuzzy quantitative association rules. *Joint 9th IFSA World Congress and 20th NAFIPS International Conference*, 3(C):1306–1311, 2002.

[86] Yang Junrui and Feng Zhang. An effective algorithm for mining quantitative associations based on subspace clustering. In *Networking and Digital Society (ICNDS), 2010 2nd International Conference on*, volume 1, pages 175–178. IEEE, 2010.

[87] J Yang, X Hu, and Y Fu. Fuzzy Association Rules Mining Algorithm FMFFI Based on Bidirectional Search Technique. In *2015 7th International Conference on Intelligent Human-Machine Systems and Cybernetics*, volume 2, pages 440–443, 2015.

[88] Mahtab Hossein Afshari, Mohammad Naderi Dehkordi, and Mehdi Akbari. Association rule hiding using cuckoo optimization algorithm. *Expert Systems with Applications*, 64:340–351, 2016.

[89] Vahid Beiranvand, Mohamad Mobasher-Kashani, and Azuraliza Abu Bakar. Multi-objective PSO algorithm for mining numerical association rules without a priori discretization. *Expert Systems with Applications*, 41(9):4259–4273, 2014.

[90] Liqiang Geng and Howard J. Hamilton. Interestingness measures for data mining: A survey. *ACM Comput. Surv.*, 38(3):9–es, September 2006.

[91] Pang-Ning Tan, Vipin Kumar, and Jaideep Srivastava. Selecting the right interestingness measure for association patterns. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD 2002, pages 32–41, New York, NY, USA, 2002. Association for Computing Machinery.

[92] Sergey Brin, Rajeev Motwani, and Craig Silverstein. Beyond market baskets: Generalizing association rules to correlations. In *Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data*, SIGMOD 1997, pages 265–276, New York, NY, USA, 1997. Association for Computing Machinery.

[93] Gregory Piatetsky-Shapiro. Discovery, analysis, and presentation of strong rules. In *Knowledge Discovery in Databases*, pages 229–248. AAAI/MIT Press, 1991.

[94] Nada Lavrac, Peter A. Flach, and Blaz Zupan. Rule evaluation measures: A unifying view. In *Proceedings of the 9th International Workshop on Inductive Logic Programming*, ILP '99, pages 174–185, Berlin, Heidelberg, 1999. Springer-Verlag.

[95] Fernando Berzal Galiano, Ignacio J. Blanco, Daniel Sánchez, and María Amparo Vila. Measuring the accuracy and interest of association rules: A new framework. *Intell. Data Anal.*, 6:221–235, 2002.

[96] Philippe Lenca, Patrick Meyer, BenoÃ®t Vaillant, and StÃ©phane Lallich. On selecting interestingness measures for association rules: User oriented description and multiple criteria decision aid. *European Journal of Operational Research*, 184(2):610–626, 2008.

[97] José María Luna, M Ondra, Habib M Fardoun, and Sebastián Ventura. Optimization of quality measures in association rule mining: an empirical study. *International Journal of Computational Intelligence Systems*, 12(1):59–78, 2018.

[98] M. Martínez-Ballesteros and J. C. Riquelme. Analysis of measures of quantitative association rules. In *International Conference on Hybrid Artificial Intelligence Systems*, volume 6679 LNAI, pages 319–326, Springer Berlin Heidelberg, 2011.

[99] Mehrdad Almasi and Mohammad Saniee Abadeh. Rare-pears: A new multi objective evolutionary algorithm to mine rare and non-redundant quantitative association rules. *Knowledge-Based Systems*, 89:366–384, 2015.

[100] Anirban Mukhopadhyay, Ujjwal Maulik, Sanghamitra Bandyopadhyay, and Carlos Artemio Coello Coello. A survey of multiobjective evolutionary algorithms for data mining: Part i. *IEEE Transactions on Evolutionary Computation*, 18(1):4–19, 2014.

[101] Youcef Djenouri, Philippe Fournier-Viger, Asma Belhadi, and Jerry Chun-Wei Lin. *Metaheuristics for Frequent and High-Utility Itemset Mining*, pages 261–278. Springer International Publishing, Cham, 2019.

[102] Anirban Mukhopadhyay, Ujjwal Maulik, Sanghamitra Bandyopadhyay, and Carlos A. Coello Coello. Survey of multiobjective evolutionary algorithms for data mining: Part ii. *IEEE Transactions on Evolutionary Computation*, 18(1):20–35, 2014.

[103] Sebastián Ventura and José María Luna. *Genetic Programming in Pattern Mining*, pages 87–117. Springer International Publishing, Cham, 2016.

[104] Sujatha Srinivasan and Sivakumar Ramakrishnan. Evolutionary multi objective optimization for rule mining: A review. *Artif. Intell. Rev.*, 36(3):205–248, October 2011.

[105] Bodrunnessa Badhon, Mir Md Jahangir Kabir, Shuxiang Xu, and Monika Kabir. A survey on association rule mining based on evolutionary algorithms. *International Journal of Computers and Applications*, pages 1–11, 2019.

[106] Maráa J. del Jesus, José A. Gámez, Pedro González, and José M. Puerta. On the discovery of association rules by means of evolutionary algorithms. *WIREs Data Mining and Knowledge Discovery*, 1(5):397–415, 2011.

[107] Seyed Mohssen Ghafari and Christos Tjortjis. A survey on association rules mining using heuristics. *WIREs Data Mining and Knowledge Discovery*, 9(4):e1307, 2019.

[108] Akbar Telikani, Amir H. Gandomi, and Asadollah Shahbahrami. A survey of evolutionary computation for association rule mining. *Information Sciences*, 524:318–352, 2020.

[109] Esmat Rashedi, Elaheh Rashedi, and Hossein Nezamabadi-pour. A comprehensive survey on gravitational search algorithm. *Swarm and Evolutionary Computation*, 41:141–158, 2018.

[110] Yunfei Cui, Zhiqiang Geng, Qunxiong Zhu, and Yongming Han. Review: Multi-objective optimization methods and application in energy saving. *Energy*, 125:681–704, 2017.

[111] Ali Husseinzadeh Kashan. League championship algorithm (lca): An algorithm for global optimization inspired by sport championships. *Applied Soft Computing*, 16:171–200, 2014.

[112] Wengdong Wang and Susan M Bridges. Genetic algorithm optimization of membership functions for mining fuzzy association rules. *Department of Computer Science Mississippi State University*, 2, 2000.

[113] Rupali Haldulakar and Jitendra Agrawal. Optimization of association rule mining through genetic algorithm. *International Journal on Computer Science and Engineering (IJCSE)*, 3(3):1252–1259, 2011.

[114] M. Saggar, A.K. Agrawal, and A. Lad. Optimization of association rule mining using improved genetic algorithms. In *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583)*, volume 4, pages 3725–3729 vol.4, 2004.

[115] Hong Guo and Ya Zhou. An algorithm for mining association rules based on improved genetic algorithm and its application. In *2009 Third International Conference on Genetic and Evolutionary Computing*, pages 117–120, 2009.

[116] W. Soto and A. Olaya-Benavides. A genetic algorithm for discovery of association rules. In *2011 30th International Conference of the Chilean Computer Science Society*, pages 289–293, Los Alamitos, CA, USA, nov 2011. IEEE Computer Society.

[117] Xiaowei Yan, Chengqi Zhang, and Shichao Zhang. Armga: Identifying interesting association rules with genetic algorithms. *Applied Artificial Intelligence*, 19(7):677–689, 2005.

[118] Mir Md. Jahangir Kabir, Shuxiang Xu, Byeong Ho Kang, and Zongyuan Zhao. A new multiple seeds based genetic algorithm for discovering a set of interesting boolean association rules. *Expert Systems with Applications*, 74:55–69, 2017.

[119] Bilal Alataş and Erhan Akin. An efficient genetic algorithm for automated mining of both positive and negative quantitative association rules. *Soft Comput.*, 10(3):230–237, February 2006.

[120] Yang Xu, Mingming Zeng, Quanhui Liu, and Xiaofeng Wang. A genetic algorithm based multilevel association rules mining for big datasets. *Mathematical Problems in Engineering*, 2014, 2014.

[121] Chun-Hao Chen, Tzung-Pei Hong, and Vincent S. Tseng. An improved approach to find membership functions and multiple minimum supports in fuzzy data mining. *Expert Syst. Appl.*, 36(6):10016–10024, August 2009.

[122] Tony Cheng-Kui Huang. Discovery of fuzzy quantitative sequential patterns with multiple minimum supports and adjustable membership functions. *Information Sciences*, 222:126–146, 2013. Including Special Section on New Trends in Ambient Intelligence and Bio-inspired Systems.

[123] M. Kaya and R. Alhajj. A clustering algorithm with genetically optimized membership functions for fuzzy association rules mining. In *The 12th IEEE International Conference on Fuzzy Systems, 2003. FUZZ '03.*, volume 2, pages 881–886 vol.2, 2003.

[124] M. Kaya and R. Alhajj. Genetic algorithm based framework for mining fuzzy association rules. *Fuzzy Sets and Systems*, 152(3):587–601, 2005.

[125] Chun-Hao Chen, Tzung-Pei Hong, and Vincent S. Tseng. A cluster-based fuzzy-genetic mining approach for association rules and membership functions. In *2006 IEEE International Conference on Fuzzy Systems*, pages 1411–1416, 2006.

[126] Chun-Hao Chen, Tzung-Pei Hong, and Vincent S Tseng. A cluster-based genetic-fuzzy mining approach for items with multiple minimum

supports. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 864–869. Springer, 2008.

[127] Chun-Hao Chen, Tzung-Pei Hong, Yeong-Chyi Lee, and Vincent S. Tseng. Finding active membership functions for genetic-fuzzy data mining. *International Journal of Information Technology & Decision Making*, 14(06):1215–1242, 2015.

[128] Yi-Chung Hu. Determining membership functions and minimum fuzzy support in finding fuzzy association rules for classification problems. *Knowledge-Based Systems*, 19(1):57–66, 2006.

[129] Lotfi Asker Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy sets and systems*, 1(1):3–28, 1978.

[130] Rafael Alcala, Jesus Alcala-Fdez, M.J. Gacto, and Francisco Herrera. Genetic learning of membership functions for mining fuzzy association rules. In *2007 IEEE International Fuzzy Systems Conference*, pages 1–6, 2007.

[131] Ana María Palacios, José Luis Palacios, Luciano Sánchez, and Jesús Alcalá-Fdez. Genetic learning of the membership functions for mining fuzzy association rules from low quality data. *Inf. Sci.*, 295(C):358–378, February 2015.

[132] Chuan-Kang Ting, Ting-Chen Wang, Rung-Tzuo Liaw, and Tzung-Pei Hong. Genetic algorithm with a structure-based representation for genetic-fuzzy data mining. *Soft Computing*, 21(11):2871–2882, 2017.

[133] Ferrante Neri and Carlos Cotta. Memetic algorithms and memetic computing optimization: A literature review. *Swarm and Evolutionary Computation*, 2:1–14, 2012.

[134] Ferrante Neri, Ville Tirronen, and Tommi Karkkainen. Enhancing differential evolution frameworks by scale factor local search - part ii. In *2009 IEEE Congress on Evolutionary Computation*, pages 118–125, 2009.

[135] Chun-Hao Chen, Tzung-Pei Hong, Vincent S. Tseng, and Chang-Shing Lee. A genetic-fuzzy mining approach for items with multiple minimum supports. In *2007 IEEE International Fuzzy Systems Conference*, pages 1–6, 2007.

[136] Chun-Hao Chen, Tzung-Pei Hong, and Vincent S. Tseng. Speeding up genetic-fuzzy mining by fuzzy clustering. In *2009 IEEE International Conference on Fuzzy Systems*, pages 1695–1699, 2009.

[137] Tzung-Pei Hong, Yeong-Chyi Lee, and Min-Thai Wu. Using the master-slave parallel architecture for genetic-fuzzy data mining. In *2005 IEEE International Conference on Systems, Man and Cybernetics*, volume 4, pages 3232–3237 Vol. 4, 2005.

[138] Tzung-Pei Hong, Yeong-Chyi Lee, and Min-Thai Wu. An effective parallel approach for genetic-fuzzy data mining. 41(2):655–662, feb 2014.

[139] Youcef Djenouri, Ahcene Bendjoudi, Djamel Djenouri, and Marco Comuzzi. Gpu-based bio-inspired model for solving association rules mining problem. In *2017 25th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*, pages 262–269, 2017.

[140] Halina Kwasnicka and Kajetan Switalski. Discovery of association rules from medical data-classical and evolutionary approaches. *Annales UMCS Sectio AI Informatica.*, 4:163–177, 01 2006.

[141] Bing Wang, Kathryn E. Merrick, and Hussein A. Abbass. Co-operative coevolutionary neural networks for mining functional association rules. *IEEE Transactions on Neural Networks and Learning Systems*, 28(6):1331–1344, 2017.

[142] Peddi Kishor and Porika Sammulal. Association rule mining using an unsupervised neural network with an optimized genetic algorithm. In *International Conference on Communications and Cyber Physical Engineering 2018*, pages 657–669. Springer, 2018.

[143] Feng Wen, Guo Zhang, Lingfeng Sun, Xingqiao Wang, and Xiaowei Xu. A hybrid temporal association rules mining method for traffic congestion prediction. *Computers & Industrial Engineering*, 130:779–787, 2019.

[144] Amir Hossein Gandomi, Xin-She Yang, Siamak Talatahari, and Amir Hossein Alavi. 1 - metaheuristic algorithms in modeling and optimization. In Amir Hossein Gandomi, Xin-She Yang, Siamak Talatahari, and Amir Hossein Alavi, editors, *Metaheuristic Applications in Structures and Infrastructures*, pages 1–24. Elsevier, Oxford, 2013.

[145] Ilhem BoussaÃ¯d, Julien Lepagnot, and Patrick Siarry. A survey on optimization metaheuristics. *Information Sciences*, 237:82–117, 2013. Prediction, Control and Diagnosis using Advanced Neural Computations.

[146] K. Hirasawa, M. Okubo, H. Katagiri, J. Hu, and J. Murata. Comparison between genetic network programming (gnp) and genetic programming (gp). In *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546)*, volume 2, pages 1276–1282 vol. 2, 2001.

[147] Shingo Mabu, Ci Chen, Nannan Lu, Kaoru Shimada, and Kotaro Hirasawa. An intrusion-detection model based on fuzzy class-association-rule mining using genetic network programming. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 41(1):130–139, 2011.

[148] Kaoru Shimada, Kotaro Hirasawa, and Jinglu Hu. Class association rule mining with chi-squared test using genetic network programming. In *2006 IEEE International Conference on Systems, Man and Cybernetics*, volume 6, pages 5338–5344, 2006.

[149] J.M. Luna, J.R. Romero, and S. Ventura. Grammar-based multi-objective algorithms for mining association rules. *Data & Knowledge Engineering*, 86:19–37, 2013.

[150] Anirban Mukhopadhyay, Ujjwal Maulik, Sanghamitra Bandyopadhyay, and Carlos Artemio Coello Coello. A survey of multiobjective evolutionary algorithms for data mining: Part i. *IEEE Transactions on Evolutionary Computation*, 18(1):4–19, 2014.

[151] Reda Alhajj and Mehmet Kaya. Multi-objective genetic algorithms based automated clustering for fuzzy association rules mining. *Journal of Intelligent Information Systems*, 31(3):243–264, 2008.

[152] Mehmet Kaya and Reda Alhajj. Multi-objective genetic algorithm based method for mining optimized fuzzy association rules. In Zheng Rong Yang, Hujun Yin, and Richard M. Everson, editors, *Intelligent Data Engineering and Automated Learning – IDEAL 2004*, pages 758–764, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[153] M. Kaya and R. Alhajj. Integrating multi-objective genetic algorithms into clustering for fuzzy association rules mining. In *Fourth IEEE*

International Conference on Data Mining (ICDM-2004), pages 431–434, 2004.

[154] Chun-Hao Chen, Ji-Syuan He, and Tzung-Pei Hong. Moga-based fuzzy data mining with taxonomy. *Knowledge-Based Systems*, 54:53–65, 2013.

[155] Cheng-I Chen and Yuan-Chieh Chin. Extended real model of kalman filter for power system harmonic measurements. In *2010 Asia-Pacific Power and Energy Engineering Conference*, pages 1–5, 2010.

[156] Chun-Hao Chen, Tzung-Pei Hong, and VincentS. Tseng. Finding pareto-front membership functions in fuzzy data mining. *International Journal of Computational Intelligence Systems*, 5:343–354, 2012.

[157] Chun-Hao Chen, Tzung-Pei Hong, Vincent S. Tseng, and Lien-Chin Chen. A multi-objective genetic-fuzzy mining algorithm. In *2008 IEEE International Conference on Granular Computing*, pages 115–120, 2008.

[158] Mehmet Kaya. Multi-objective genetic algorithm based approaches for mining optimized fuzzy association rules. *Soft computing*, 10(7):578–586, 2006.

[159] Cristóbal José Carmona, Pedro Gonzalez, María José del Jesus, and Francisco Herrera. Nmeef-sd: Non-dominated multiobjective evolutionary algorithm for extracting fuzzy rules in subgroup discovery. *IEEE Transactions on Fuzzy Systems*, 18(5):958–970, 2010.

[160] Stephen G. Matthews, Mario A. Gongora, and Adrian A. Hopgood. Evolving temporal fuzzy association rules from quantitative data with a multi-objective evolutionary algorithm. In Emilio Corchado, Marek Kurzyński, and Michał Woźniak, editors, *Hybrid Artificial Intelligent Systems*, pages 198–205, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[161] Stephen G. Matthews, Mario A. Gongora, and Adrian A. Hopgood. Evolving temporal fuzzy itemsets from quantitative data with a multi-objective evolutionary algorithm. In *2011 IEEE 5th International Workshop on Genetic and Evolutionary Fuzzy Systems (GEFS)*, pages 9–16, 2011.

[162] B. Minaei-Bidgoli, R. Barmaki, and M. Nasiri. Mining numerical association rules via multi-objective genetic algorithms. *Information Sciences*, 233:15–24, 2013.

[163] M. MartÃnez-Ballesteros, I.A. Nepomuceno-Chamorro, and J.C. Riquelme. Discovering gene association networks by multi-objective evolutionary quantitative association rules. *Journal of Computer and System Sciences*, 80(1):118–136, 2014.

[164] María Martínez-Ballesteros, Jaume Bacardit, Alicia Troncoso, and José C. Riquelme. Enhancing the scalability of a genetic algorithm to discover quantitative association rules in large-scale datasets. 22(1):21–39, January 2015.

[165] K.Y. Fung, C.K. Kwong, K.W.M. Siu, and K.M. Yu. A multi-objective genetic algorithm approach to rule mining for affective product design. *Expert Systems with Applications*, 39(8):7411–7419, 2012.

[166] María Martínez-Ballesteros, A Troncoso, Francisco Martínez-Álvarez, and José C Riquelme. Improving a multi-objective evolutionary algorithm to discover quantitative association rules. *Knowledge and Information Systems*, 49(2):481–509, 2016.

[167] Rainer Storn and Kenneth Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997.

[168] Ferrante Neri and Ville Tirronen. Recent advances in differential evolution: a survey and experimental analysis. *Artificial intelligence review*, 33(1):61–106, 2010.

[169] Swagatam Das and Ponnuthurai Nagaratnam Suganthan. Differential evolution: A survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation*, 15(1):4–31, 2011.

[170] Rawaa Dawoud Al-Dabbagh, Ferrante Neri, Norisma Idris, and Mohd Sapiyan Baba. Algorithmic design issues in adaptive differential evolution schemes: Review and taxonomy. *Swarm and Evolutionary Computation*, 43:284–311, 2018.

[171] Iztok Fister, Andres Iglesias, Akemi Galvez, Javier Del Ser, Eneko Osaba, and Iztok Fister. Differential evolution for association rule mining

using categorical and numerical attributes. In Hujun Yin, David Camacho, Paulo Novais, and Antonio J. Tallón-Ballesteros, editors, *Intelligent Data Engineering and Automated Learning – IDEAL 2018*, pages 79–88, Cham, 2018. Springer International Publishing.

[172] Anshu Zhang and Wenzhong Shi. Mining significant fuzzy association rules with differential evolution algorithm. *Applied Soft Computing*, 97:105518, 2020.

[173] Xin-She Yang and Mehmet Karamanoglu. 1 - swarm intelligence and bio-inspired computation: An overview. In Xin-She Yang, Zhihua Cui, Renbin Xiao, Amir Hossein Gandomi, and Mehmet Karamanoglu, editors, *Swarm Intelligence and Bio-Inspired Computation*, pages 3–23. Elsevier, Oxford, 2013.

[174] Marco Dorigo, Mauro Birattari, and Thomas Stutzle. Ant colony optimization. *IEEE computational intelligence magazine*, 1(4):28–39, 2006.

[175] M. Dorigo and L.M. Gambardella. Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1):53–66, 1997.

[176] R.J. Kuo, S.Y. Lin, and C.W. Shih. Mining association rules through integration of clustering analysis and ant colony system for health insurance database in taiwan. *Expert Systems with Applications*, 33(3):794–808, 2007.

[177] R.J. Kuo and C.W. Shih. Association rule mining through the ant colony system for national health insurance research database in taiwan. *Computers & Mathematics with Applications*, 54(11):1303–1318, 2007.

[178] Tzung-Pei Hong, Ya-Fang Tung, Shyue-Liang Wang, and Yu-Lung Wu. A multi-level ant-based algorithm for fuzzy data mining. In *NAFIPS 2009 - 2009 Annual Meeting of the North American Fuzzy Information Processing Society*, pages 1–5, 2009.

[179] Tzung-Pei Hong, Ya-Fang Tung, Shyue-Liang Wang, Min-Thai Wu, and Yu-Lung Wu. An acs-based framework for fuzzy data mining. *Expert Systems with Applications*, 36(9):11844–11852, 2009.

[180] Min-Thai Wu, Tzung-Pei Hong, and Chung-Nan Lee. An improved ant algorithm for fuzzy data mining. In Jeng-Shyang Pan, Shyi-Ming Chen,

and Ngoc Thanh Nguyen, editors, *Computational Collective Intelligence. Technologies and Applications*, pages 344–351, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[181] Min-Thai Wu, Tzung-Pei Hong, and Chung-Nan Lee. A continuous ant colony system framework for fuzzy data mining. *Soft Computing*, 16(12):2071–2082, 2012.

[182] J. L. Olmo, J. R. Romero, and S. Ventura. Classification rule mining using ant programming guided by grammar with multiple pareto fronts. *Soft Comput.*, 16(12):2143–2163, December 2012.

[183] Juan Luis Olmo, José María Luna, José Raul Romero, and Sebastián Ventura. Association rule mining using a multi-objective grammar-based ant programming algorithm. In *2011 11th International Conference on Intelligent Systems Design and Applications*, pages 971–977, 2011.

[184] Juan Luis Olmo, José Raúl Romero, and Sebastián Ventura. Single and multi-objective ant programming for mining interesting rare association rules. *International Journal of Hybrid Intelligent Systems*, 11(3):197–209, 2014.

[185] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of ICNN 1995 - International Conference on Neural Networks*, volume 4, pages 1942–1948 vol.4, 1995.

[186] Bilal Alatas and Erhan Akin. Multi-objective rule mining using a chaotic particle swarm optimization algorithm. *Knowledge-Based Systems*, 22(6):455–460, 2009.

[187] Mayank Agrawal, Manuj Mishra, and Shiv Pratap Singh Kushwah. Association rules optimization using improved pso algorithm. In *2015 International Conference on Communication Networks (ICCN)*, pages 395–398, 2015.

[188] K Indira and S Kanmani. Association rule mining through adaptive parameter control in particle swarm optimization. *Computational Statistics*, 30(1):251–277, 2015.

[189] Shang Qianxiang and Wu Ping. Association rules mining based on improved pso algorithm. In *2017 2nd IEEE International Conference on Computational Intelligence and Applications (ICCIA)*, pages 145–149, 2017.

[190] Tong Su, Haitao Xu, and Xianwei Zhou. Particle swarm optimization-based association rule mining in big data environment. *IEEE Access*, 7:161008–161016, 2019.

[191] R.J. Kuo, C.M. Chao, and Y.T. Chiu. Application of particle swarm optimization to association rule mining. *Applied Soft Computing*, 11(1):326–336, 2011.

[192] Manish Gupta. Application of weighted particle swarm optimization in association rule mining. 2012.

[193] Mansour Sheikhan and Maryam Sharifi Rad. Using particle swarm optimization in fuzzy association rules-based feature selection and fuzzy ARTMAP-based attack recognition. *Security and Communication Networks*, 6(7):797–811, aug 2012.

[194] Shweta Tyagi and Kamal K. Bharadwaj. Enhancing collaborative filtering recommendations by utilizing multi-objective particle swarm optimization embedded association rule mining. *Swarm and Evolutionary Computation*, 13:1–12, 2013.

[195] R. J. Kuo, Monalisa Gosumolo, and Ferani E. Zulvia. Multi-objective particle swarm optimization algorithm using adaptive archive grid for numerical association rule mining. *Neural Computing and Applications*, pages 1–14, 2017.

[196] Imam Tahyudin and Hidetaka Nambo. The combination of evolutionary algorithm method for numerical association rule mining optimization. In Jiuping Xu, Asaf Hajiyev, Stefan Nickel, and Mitsuo Gen, editors, *Proceedings of the Tenth International Conference on Management Science and Engineering Management*, pages 13–23, Singapore, 2017. Springer Singapore.

[197] Imam Tahyudin and Hidetaka Nambo. The rule extraction of numerical association rule mining using hybrid evolutionary algorithm. In *2017 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, pages 1–6, 2017.

[198] K.N.V.D. Sarath and Vadlamani Ravi. Association rule mining using binary particle swarm optimization. *Engineering Applications of Artificial Intelligence*, 26(8):1832–1840, 2013.

[199] Chun-Wei Lin, Lu Yang, Philippe Fournier-Viger, Tzung-Pei Hong, and Miroslav Voznák. A binary pso approach to mine high-utility itemsets. *Soft Computing*, 21:5103–5121, 2017.

[200] Jerry Chun-Wei Lin, Lu Yang, Philippe Fournier-Viger, Ming-Thai Wu, Tzung-Pei Hong, and Leon Shyue-Liang Wang. A swarm-based approach to mine high-utility itemsets. In Leon Wang, Shiro Uesugi, I-Hsien Ting, Koji Okuhara, and Kai Wang, editors, *Multidisciplinary Social Networks Research*, pages 572–581, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

[201] Jerry Chun-Wei Lin, Lu Yang, Philippe Fournier-Viger, Jimmy Ming-Thai Wu, Tzung-Pei Hong, Leon Shyue-Liang Wang, and Justin Zhan. Mining high-utility itemsets based on particle swarm optimization. *Engineering Applications of Artificial Intelligence*, 55:320–330, 2016.

[202] Youcef Djenouri and Marco Comuzzi. Combining apriori heuristic and bio-inspired algorithms for solving the frequent itemsets mining problem. *Information Sciences*, 420:1–15, 2017.

[203] Youcef Djenouri, Djamel Djenouri, Asma Belhadi, Philippe Fournier-Viger, and Jerry Chun-Wei Lin. A new framework for metaheuristic-based frequent itemset mining. *Applied Intelligence*, 48(12):4775–4791, December 2018.

[204] G Maragatham and M Lakshmi. A weighted particle swarm optimization technique for optimizing association rules. In *International Conference on Computing and Communication Systems*, pages 655–664. Springer, 2011.

[205] Veenu Mangat and Renu Vig. Novel associative classifier based on dynamic adaptive pso: Application to determining candidates for thoracic surgery. *Expert Systems with Applications*, 41(18):8234–8244, 2014.

[206] Kshitij Tayal and Vadlamani Ravi. Particle swarm optimization trained class association rule mining: Application to phishing detection. In *Proceedings of the International Conference on Informatics and Analytics*, pages 1–8, 2016.

[207] Jitendra Agrawal, Shikha Agrawal, Ankita Singhai, and Sanjeev Sharma. Set-pso-based approach for mining positive and negative association rules. *Knowledge and information systems*, 45(2):453–471, 2015.

[208] Danfeng Yan, Xuan Zhao, Rongheng Lin, and Demeng Bai. Ppqar: parallel pso for quantitative association rule mining. *Peer-to-Peer Networking and Applications*, 12(5):1433–1444, 2019.

[209] Baris Yuce, Michael S. Packianather, Ernesto Mastrocinque, Duc Truong Pham, and Alfredo Lambiase. Honey bees inspired optimization method: The bees algorithm. *Insects*, 4(4):646–662, 2013.

[210] Reza Akbari, Alireza Mohammadi, and Koorush Ziarati. A novel bee swarm optimization algorithm for numerical function optimization. *Communications in Nonlinear Science and Numerical Simulation*, 15(10):3142–3155, 2010.

[211] Y. Djenouri, H. Drias, Z. Habbas, and H. Mosteghanemi. Bees swarm optimization for web association rule mining. In *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, volume 3, pages 142–146, 2012.

[212] Youcef Djenouri, Habiba Drias, and Zineb Habbas. Bees swarm optimisation using multiple strategies for association rule mining. 6(4):239–249, September 2014.

[213] Qingchen Zhang, Laurence T. Yang, Zhikui Chen, and Peng Li. A survey on deep learning for big data. *Information Fusion*, 42:146–157, 2018.

[214] Youcef Djenouri, Ahcene Bendjoudi, Malika Mehdi, Nadia Nouali-Taboudjemat, and Zineb Habbas. Gpu-based bees swarm optimization for association rules mining. *The Journal of Supercomputing*, 71(4):1318–1344, 2015.

[215] Youcef Djenouri, Djamel Djenouri, and Zineb Habbas. Intelligent mapping between gpu and cluster computing for discovering big association rules. *Applied Soft Computing*, 65:387–399, 2018.

[216] Dervis Karaboga. An idea based on honey bee swarm for numerical optimization, technical report - tr06. *Technical Report, Erciyes University*, 01 2005.

[217] Qi Liu, Gengzhong Feng, Nengmin Wang, and Giri Kumar Tayi. A multi-objective model for discovering high-quality knowledge based on data quality and prior knowledge. *Information Systems Frontiers*, 20(2):401–416, April 2018.

[218] D.T. Pham, A. Ghanbarzadeh, E. KoÃ§, S. Otri, S. Rahim, and M. Zaidi. The bees algorithm, a novel tool for complex optimisation problems. In D.T. Pham, E.E. Eldukhri, and A.J. Soroka, editors, *Intelligent Production Machines and Systems*, pages 454–459. Elsevier Science Ltd, Oxford, 2006.

[219] Mojtaba Asadollahpour Chamazi and Homayun Motameni. Finding suitable membership functions for fuzzy temporal mining problems using fuzzy temporal bees method. *Soft Comput.*, 23(10):3501–3518, May 2019.

[220] Xin-She Yang. *A New Metaheuristic Bat-Inspired Algorithm*, pages 65–74. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[221] Xin-She Yang and Amir Hossein Gandomi. Bat algorithm: a novel approach for global engineering optimization. *Engineering computations*, 2012.

[222] Rizk M Rizk-Allah and Aboul Ella Hassanien. New binary bat algorithm for solving 0–1 knapsack problem. *Complex & Intelligent Systems*, 4(1):31–53, 2018.

[223] Kamel Eddine Heraguemi, Nadjet Kamel, and Habiba Drias. Association rule mining based on bat algorithm. In Linqiang Pan, Gheorghe Păun, Mario J. Pérez-Jiménez, and Tao Song, editors, *Bio-Inspired Computing - Theories and Applications*, pages 182–186, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[224] Kamel Eddine Heraguemi, Nadjet Kamel, and Habiba Drias. Association rule mining based on bat algorithm. *Journal of Computational and Theoretical Nanoscience*, 12(7):1195–1200, 2015.

[225] Kamel Eddine Heraguemi, Nadjet Kamel, and Habiba Drias. Multi-swarm bat algorithm for association rule mining using multiple cooperative strategies. *Applied Intelligence*, 45(4):1021–1033, 2016.

[226] Xin-She Yang. Firefly algorithms for multimodal optimization. In Osamu Watanabe and Thomas Zeugmann, editors, *Stochastic Algorithms: Foundations and Applications*, pages 169–178, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[227] Xin-She Yang and Suash Deb. Cuckoo search via lÃ©vy flights. In *2009 World Congress on Nature Biologically Inspired Computing (NaBIC)*, pages 210–214, 2009.

[228] Amir Hossein Gandomi, Xin-She Yang, and Amir Hossein Alavi. Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Engineering with computers*, 29(1):17–35, 2013.

[229] Uroš Mlakar, Milan Zorman, Iztok Fister Jr, and Iztok Fister. Modified binary cuckoo search for association rule mining. *Journal of Intelligent & Fuzzy Systems*, 32(6):4319–4330, 2017.

[230] Xiangtao Li, Jie Zhang, and Minghao Yin. Animal migration optimization: an optimization algorithm inspired by animal migration behavior. *Neural Computing and Applications*, 24(7):1867–1877, 2014.

[231] Le Hoang Son, Francisco Chiclana, Raghavendra Kumar, Mamta Mittal, Manju Khari, Jyotir Moy Chatterjee, and Sung Wook Baik. Armamo: An efficient association rule mining algorithm based on animal migration optimization. *Knowledge-Based Systems*, 154:68–80, 2018.

[232] Youcef Gheraibia and Abdelouahab Moussaoui. Penguins search optimization algorithm (pesoa). In Moonis Ali, Tibor Bosse, Koen V. Hindriks, Mark Hoogendoorn, Catholijn M. Jonker, and Jan Treur, editors, *Recent Trends in Applied Artificial Intelligence*, pages 222–231, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[233] Youcef Gheraibia, Abdelouahab Moussaoui, Youcef Djenouri, Sohag Kabir, and Peng Yeng Yin. Penguins search optimisation algorithm for association rules mining. *Journal of computing and information technology*, 24(2):165–179, 2016.

[234] IBM Corporation. Ibm spss modeler for windows.

[235] Oracle Corp. Oracle data miner.

[236] SAS Institute Inc. Sas enterprise miner.

[237] TIBCO Software Inc. Tibco data science-statistica software.

[238] Frontline Systems Inc. Analytic solver datamining.

[239] Microsoft Inc. Microsoft analysis services.

[240] University of Ljubljana. Orange datamining.

[241] Ian H Witten, Eibe Frank, Leonard E Trigg, Mark A Hall, Geoffrey Holmes, and Sally Jo Cunningham. Weka: Practical machine learning tools and techniques with java implementations. 1999.

[242] Rakotomalala Ricco. Tanagra: a free software for research and academic purposes, 2005.

[243] Universidad de Waikato. Weka.

[244] Rapidaminer Inc. Rapidminer.

[245] Graham J Williams. Rattle: A Data Mining GUI for R. *The R Journal*, 1(2):45–55, 2009.

[246] Philippe Fournier-Viger, Jerry Chun-Wei Lin, Antonio Gomariz, Ted Gueniche, Azadeh Soltani, Zhihong Deng, and Hoang Thanh Lam. The spmf open-source data mining library version 2. In Bettina Berendt, Björn Bringmann, Élisa Fromont, Gemma Garriga, Pauli Miettinen, Nikolaj Tatti, and Volker Tresp, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 36–40, Cham, 2016. Springer International Publishing.

[247] Konstantinos Malliaridis, Stefanos Ougiaroglou, and Dimitris A. Dervos. Webapriori: A web application forÂ association rules mining. In Vivekanandan Kumar and Christos Troussas, editors, *Intelligent Tutoring Systems*, pages 371–377, Cham, 2020. Springer International Publishing.

[248] KNIME.com AG. Knime.

[249] Iztok Fister and Iztok Fister Jr au2. uarmsolver: A framework for association rule mining, 2020.

[250] Michael Hahsler, Bettina GrÃ¼n, and Kurt Hornik. arules - a computational environment for mining association rules and frequent item sets. *Journal of Statistical Software*, 14(15):1–25, 2005.

[251] Christian Borgelt and Gil Gonzalez Rodriguez. Frida -a free intelligent data analysis toolbox. In *2007 IEEE International Fuzzy Systems Conference*, pages 1–5, 2007.

[252] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y. P. Chen, A. Auger, and S. Tiwari. Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization. Technical report, Nanyang Technological University, Singapore, 2005.

[253] Liqiang Geng and Howard J. Hamilton. Interestingness measures for data mining. *ACM Computing Surveys*, 38(3):1–32, 2006.

[254] M. Martínez-Ballesteros, F. Martínez-Álvarez, A. Troncoso-Lora, and J.C. Riquelme Santos. Selecting the best measures to discover quantitative association rules. *Neurocomputing*, 126:3–14, 2014.

[255] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. *ACM SIGMOD Record*, 22(May):207–216, 1993.

[256] Fernando Berzal, Juan-Carlos Cubero, and Aí da Jiménez. Interestingness Measures for Association Rules. *Intell. Data Anal.*, 17(2):298–307, 2010.

[257] Takeshi Fukuda, Yasuhiko Morimoto, Shinichi Morishita, and Takeshi Tokuyama. Mining Optimized Association Rules for Numeric Attributes. *Journal of Computer and System Sciences*, 58(1):1–12, 1999.

[258] A. E. Eiben, M. C. Schut, and A. R. de Wilde. Is self-adaptation of selection pressure and population size possible? a case study. In *Proceedings of the 9th International Conference on Parallel Problem Solving from Nature*, PPSN 2006, pages 900–909, Berlin, Heidelberg, 2006. Springer-Verlag.

[259] Thomas Bäck, A. E. Eiben, and N. A. L. van der Vaart. An empirical study on gas "without parameters". In *Proceedings of the 6th International Conference on Parallel Problem Solving from Nature*, PPSN VI, pages 315–324, Berlin, Heidelberg, 2000. Springer-Verlag.

[260] Juan J. Durillo and Antonio J. Nebro. jmetal: A java framework for multi-objective optimization. *Advances in Engineering Software*, 42(10):760–771, 2011.

[261] Ansaf Salleb-Aouissi. QuantMiner for Mining Quantitative Association Rules. *The Journal of Machine Learning Research*, 14(1):3153–3157, 2013.

[262] Isaac Triguero, Sergio González, Jose M. Moyano, Salvador García, Jesús Alcalá-Fdez, Julián Luengo, Alberto Fernández, María José del Jesús, Luciano Sánchez, and Francisco Herrera. Keel 3.0: An open source software for multi-stage analysis in data mining. *International Journal of Computational Intelligence Systems*, 10:1238–1249, 2017.

[263] J. Alcalá-Fdez, L. Sánchez, S. García, M. D. Jesús, S. Ventura, J. M. Guiu, J. Otero, C. Romero, J. Bacardit, V. Santos, Juan Carlos Fernández, and F. Herrera. Keel: a software tool to assess evolutionary

algorithms for data mining problems. *Soft Computing*, 13:307–318, 2009.

[264] Jose Moyano and Luciano Sanchez. Rkeel: Using keel in r code. In *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 257–264, 2016.

[265] Christian Borgelt. Efficient implementations of apriori and eclat. In *Proc. 1st IEEE ICDM Workshop on Frequent Item Set Mining Implementations (FIMI 2003, Melbourne, FL). CEUR Workshop Proceedings 90*, page 90, 2003.

[266] Bilal Alataş and Erhan Akin. An efficient genetic algorithm for automated mining of both positive and negative quantitative association rules. *Soft Comput.*, 10(3):230–237, February 2006.

[267] Xiaowei Yan, Chengqi Zhang, and Shichao Zhang. Genetic algorithm-based strategy for identifying association rules without specifying actual minimum support. *Expert Systems with Applications*, 36(2, Part 2):3066–3076, 2009.

[268] M.J. Zaki. Scalable algorithms for association mining. *IEEE Transactions on Knowledge and Data Engineering*, 12(3):372–390, 2000.

[269] Jiawei Han, Jian Pei, Yiwen Yin, and Runying Mao. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data mining and knowledge discovery*, 8(1):53–87, 2004.

[270] Jacinto Mata, José-Luis Alvarez, and José-Cristobal Riquelme. Discovering numeric association rules via evolutionary algorithm. In Ming-Syan Chen, Philip S. Yu, and Bing Liu, editors, *Advances in Knowledge Discovery and Data Mining*, pages 40–51, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[271] J. Mata, J. L. Alvarez, and J. C. Riquelme. An evolutionary algorithm to discover numeric association rules. In *Proceedings of the 2002 ACM Symposium on Applied Computing*, SAC '02, pages 590–594, New York, NY, USA, 2002. Association for Computing Machinery.

[272] Jesús Alcalá-Fdez, Rafael Alcalá, María José Gacto, and Francisco Herrera. Learning the membership function contexts for mining fuzzy association rules by using genetic algorithms. *Fuzzy Sets and Systems*, 160(7):905–921, 2009. Theme: Modeling and Learning.

[273] TZUNG-PEI HONG, CHAN-SHENG KUO, and SHENG-CHAI CHI. Trade-off between computation time and number of rules for fuzzy mining from quantitative data. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 09(05):587–604, 2001.

[274] Tzung-Pei Hong, Chun-Hao Chen, Yu-Lung Wu, and Yeong-Chyi Lee. A ga-based fuzzy mining approach to achieve a trade-off between number of rules and suitability of membership functions. *Soft Comput.*, 10(11):1091–1101, June 2006.

[275] Tzung-Pei Hong, Chun-Hao Chen, Yeong-Chyi Lee, and Yu-Lung Wu. Genetic-fuzzy data mining with divide-and-conquer strategy. *IEEE Transactions on Evolutionary Computation*, 12(2):252–265, 2008.

[276] Hamid Reza Qodmanan, Mahdi Nasiri, and Behrouz Minaei-Bidgoli. Multi objective association rule mining with genetic algorithm without specifying minimum support and minimum confidence. *Expert Systems with Applications*, 38(1):288–298, 2011.

[277] Bilal Alatas, Erhan Akin, and Ali Karci. Modenar: Multi-objective differential evolution algorithm for mining numeric association rules. *Applied Soft Computing*, 8(1):646–656, 2008.

[278] Ashish Ghosh and Bhabesh Nath. Multi-objective rule mining using genetic algorithms. *Inf. Sci.*, 163(1-3):123–133, jun 2004.

[279] Diana Martín, Alejandro Rosete, Jess Alcalá-Fdez, and Francisco Herrera. A new multiobjective evolutionary algorithm for mining a reduced set of interesting positive and negative quantitative association rules. *IEEE Transactions on Evolutionary Computation*, 18(1):54–69, 2014.

[280] D. MartÃn, A. Rosete, J. AlcalÃ¡-Fdez, and F. Herrera. Qar-cip-nsga-ii: A new multi-objective evolutionary algorithm to mine quantitative association rules. *Information Sciences*, 258:1–28, 2014.

[281] D. Martín, J. Alcalá-Fdez, A. Rosete, and F. Herrera. Nicgar. *Inf. Sci.*, 355(C):208–228, August 2016.

[282] Jesus Alcala-Fdez, Nicolo Flugy-Pape, Andrea Bonarini, and Francisco Herrera. Analysis of the effectiveness of the genetic algorithms based on extraction of association rules. *Fundam. Inf.*, 98(1):1–14, January 2010.

[283] Diana MartÃn, Alejandro Rosete, Jesus AlcalÃ¡-Fdez, and Francisco Herrera. A multi-objective evolutionary algorithm for mining quantitative association rules. In *2011 11th International Conference on Intelligent Systems Design and Applications*, pages 1397–1402, 2011.

[284] Ashish Ghosh and Bhabesh Nath. Multi-objective rule mining using genetic algorithms. *Inf. Sci.*, 163(1-3):123–133, June 2004.

[285] J. Alcalá-Fdez, Alberto Fernández, J. Luengo, J. Derrac, and S. García. Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *J. Multiple Valued Log. Soft Comput.*, 17:255–287, 2011.

[286] J. M. Luna, M. Ondra, H. M. Fardoun, and S. Ventura. Optimization of quality measures in association rule mining: an empirical study. *International Journal of Computational Intelligence Systems*, 12:59–78, 2018.

[287] Jayashree Piri and Raghunath Dey. Quantitative association rule mining using multi-objective particle swarm optimization. *Int J Sci Eng Res*, 5(10):155–161, 2014.