JISTEM - Journal of Information Systems and Technology Management
Revista de Gestão da Tecnologia e Sistemas de Informação
Vol. 10, No. 2, May/Aug., 2013 pp.389-406
ISSN online: 1807-1775
DOI: 10.4301/S1807-17752013000200012

INTEGRACIÓN DE LOS ALGORITMOS DE MINERÍA DE DATOS 1R, PRISM E ID3 A POSTGRESQL

Yadira Robles Aranda
Anthony R. Sotolongo
Universidad de las Ciencias Informáticas, Ciudad de la Habana, Cuba

---

ABSTRACT

In this research, data mining and decision tree techniques were analyzed as well as the induction of rules to integrate their many algorithms into the database managing system (DBMS), PostgreSQL, due to the defficiencies of the free use tools avaialable. A mechanism to optimize the performance of the implemented algorithms was proposed with the purpose of taking advantage of the PostgreSQL. By means of an experiment, it was proven that the time response and results obtained are improved when the algorithms are integrated into the managing system.

Keywords : data mining, database managing system, PostgreSQL, decision trees, induction of rules.

RESUMEN

En la presente investigación se analizaron las técnicas de minería de datos de árboles de decisión y de reglas de inducción para integrar varios de sus algoritmos al sistema gestor de base de datos (SGBD) PostgreSQL, buscando suplir las deficiencias de las herramientas libres existentes. También se propuso un mecanismo para optimizar el rendimiento de los algoritmos implementados con el objetivo de aprovechar las ventajas de PostgreSQL y se comprobó, mediante un experimento, que al utilizar los algoritmos integrados al gestor, los tiempos de respuestas y los resultados obtenidos son superiores.

Palabras claves: Minería de datos, sistema gestor de bases de datos, PostgreSQL, árboles de decisión, reglas de inducción.

Manuscript first received/*Recebido em* 18/09/2012 Manuscript accepted/*Aprovado em*: 23/04/2013

Address for correspondence / Endereço para correspondência

Yadira Robles Aranda, Msc, Profesor Asistente Dpto de Ingeniería de Software, Universidad de las Ciencias Informáticas, Cuba, carretera San Antonio de los Baños km 1 ½, reparto Lourdes, Boyeros, Ciudad de la Habana. E-mail: yrobles@uci.cu

Anthony R. Sotolongo, Msc, Profesor Asistente. Dpto de PostgreSQL, Universidad de las Ciencias Informáticas, Cuba, carretera San Antonio de los Baños km 1 ½, reparto Lourdes, Boyeros, Ciudad de la Habana. E-mail: asotolongo@uci.cu

1. INTRODUCCIÓN

La minería de datos es una técnica que nos permite obtener patrones o modelos a partir de los datos recopilados. Esta técnica se aplica en todo tipo de entornos como, por ejemplo, en la rama biológica, aplicaciones educacionales y financieras, procesos industriales, policiales y políticos.

Dentro de la minería de datos existen diversas técnicas, entre las cuales se encuentran la de inducción de reglas y árboles de decisión, que según diversos estudios realizados, se encuentran entre las más utilizadas. (Moreno, 2007) (Heughes Escobar, 2007)

Existen numerosas herramientas independientes del sistema gestor de bases de datos que permiten aplicar esas técnicas a grandes volúmenes de datos, sin embargo, la mayoría de estas herramientas son propietarias y no están al alcance de las organizaciones cubanas por ser altamente costosas. Otras herramientas como WEKA o YALE RapidMiner tienen licencia GPL, pero cuando existe una gran cantidad de datos a analizar, el proceso se vuelve engorroso y lento (Soto Jaramillo, 2009). Además, se debe garantizar la seguridad de los datos pues la información viaja a través de la red.

Para solucionar estos problemas, en la actualidad, algunas empresas como Microsoft y Oracle han desarrollado módulos dentro de sus sistemas gestores de bases de datos que incluyen las técnicas de minería de datos, lo que les permite agilizar los tiempos de respuesta ya que no sería necesario transformar "datos sin formato" en "información procesable" (preparación de los datos) o importación o vinculación con la herramienta encargada de hacer el análisis. De esa forma, se evita tener que contar con personal preparado en otras herramientas de análisis de datos y se proporciona a los analistas de datos un acceso directo pero controlado, lo que acelera la productividad sin poner en riesgo la seguridad de los datos. Sin embargo, a pesar de estas ventajas, estos softwares tienen el inconveniente de ser propietarios.

Actualmente Cuba está inmersa en migrar a software de código abierto buscando garantizar la seguridad nacional y lograr su independencia tecnológica. Una de las tareas para lograr este objetivo es la migración a la tecnología de bases de datos PostgreSQL por ser el SGBD de código abierto más avanzado del mundo ya que soporta la gran mayoría de las transacciones SQL, control concurrente, ofrece modernas características como consultas complejas, disparadores, vistas, integridad transaccional y permite agregar extensiones de tipo de datos, funciones, operadores y lenguajes procedurales. (Vázquez Ortíz & Castillo Martínez, 2011) (The PostgreSQL Global Development Group, 2011). Sin embargo, este sistema no ha integrado estas técnicas de minería de datos.

Por ese motivo, es necesario lograr la independencia del sistema gestor de base de datos PostgreSQL para analizar los datos mediante las técnicas de minería de datos, reglas de inducción y árboles de decisión. De ahí que se plantee como objetivo, en la presente investigación, integrar algoritmos de las técnicas de minería de datos, reglas de inducción y árboles de decisión al sistema gestor de bases de datos PostgreSQL.

2. DESARROLLO

Minería de datos

El descubrimiento de conocimiento en bases de datos (KDD, por sus siglas en inglés: Knowledge Discovery from Databases) se ha desarrollado en los últimos años como un proceso que consta de una secuencia iterativa de etapas o fases, que son: preparación de los datos (selección y transformación), minería de datos, evaluación, interpretación y toma de decisiones.

Una de las fases más importantes dentro de este proceso es la minería de datos, que integra técnicas de análisis de datos y extracción de modelos (U. Fayyad, 1996). La minería de datos se basa en varias disciplinas, entre ellas la estadística, las bases de datos, el aprendizaje automático y otras que dependen del negocio al cual se aplica el proceso o del tipo de aplicación.

En los últimos años, muchos investigadores se han profundizado en este tema y han dado distintos conceptos sobre la minería de datos, de entre los cuales el proporcionado por Fayyad (*), "la minería de datos es un proceso no trivial de identificación válida, novedosa, potencialmente útil y entendible de patrones comprensibles que se encuentran ocultos en los datos" (U. Fayyad, 1996).

Técnicas de minería de datos

Las técnicas de minería de datos constituyen un enfoque conceptual y, habitualmente, son implementadas por varios algoritmos (Molina López & García Herrero). Estas pueden clasificarse, según su utilidad, en técnicas de clasificación, de predicción, de asociación o de agrupamiento (clustering).

- **Las técnicas de predicción** permiten obtener pronósticos de comportamientos futuros a partir de los datos recopilados, de ahí que se apliquen frecuentemente. Estas técnicas resultan útiles, por ejemplo, en aplicaciones para predecir el parte meteorológico o en la toma de decisiones por parte de un cliente en determinadas circunstancias.
- **Las técnicas de agrupamiento** concentran datos dentro de un número de clases preestablecidas o no, partiendo de criterios de distancia o similitud, de manera que las clases sean similares entre sí y distintas de las otras clases (Rodríguez Suárez, 2009). Su utilización ha proporcionado significativos resultados en lo que respecta a los clasificadores o reconocedores de patrones, como en el modelado de sistemas.
- **Las técnicas de reglas de asociación** permiten establecer las posibles relaciones o correlaciones entre distintas acciones o sucesos aparentemente independientes; pudiendo reconocer como la ocurrencia de un suceso o acción puede inducir o generar la aparición de otros (Molina López & García Herrero).
- **Las técnicas de clasificación** definen unas series de clases, en que se pueden agrupar los diferentes casos. Dentro de este grupo se encuentran las técnicas de árboles de decisión y reglas de inducción.

(*)El Dr. Usama Fayyad fue el vicepresidente ejecutivo de Yahoo y creó el grupo DMX dentro de Microsoft, dedicado a la minería de datos. Actualmente es CEO de Opens Insights. Realizó varios tutoriales sobre Minería de Datos, así como algoritmos y técnicas para el desarrollo de los negocios y business intelligence. Es editor en jefe de la revista sobre minería de datos llamada: Data mining and

Knowledge Discovery, publicada por Kluwer Academic Publishers. Participó en la programación de KDD-94 y KDD-95 (conferencia internacional de minería de datos y descubrimiento de conocimiento).

Árboles de decisión

Un árbol de decisión es un conjunto de condiciones organizadas en una estructura jerárquica, de tal manera que permite determinar la decisión final que se debe tomar siguiendo las condiciones que se cumplen desde la raíz del árbol hasta alguna de sus hojas. Los árboles de decisión se utilizan desde hace siglos, y son especialmente apropiados para expresar procedimientos médicos, legales, comerciales, estratégicos, matemáticos, lógicos, entre otros (Solarte Martínez G. R., 2009). Estos se caracterizan por la sencillez de su representación y de su forma de actuar, además de la fácil interpretación, dado que pueden ser expresados en forma de reglas de decisión.

Una de las grandes ventajas de los árboles de decisión es que, en su forma más general, las opciones posibles a partir de una determinada condición son excluyentes. Esto permite analizar una situación y siguiendo el árbol de decisión apropiadamente, llegar a una sola acción o decisión a tomar.

Entre los algoritmos de árboles de decisión se encuentran el ID3 (Induction of Decision Trees) y el C4.5 desarrollados por JR Quinlan, siendo que el ID3 es considerado un clásico de los algoritmos de aprendizaje automático.

Inducción de Reglas

Las reglas permiten expresar disyunciones de manera más fácil que los árboles y tienden a preferirse con respecto a los árboles por tender a representar "pedazos" de conocimiento relativamente independientes.

Las técnicas de Inducción de Reglas permiten generar y contrastar árboles de decisión, o reglas y patrones a partir de los datos de entrada. La información de entrada será un conjunto de casos en que se ha asociado una clasificación o evaluación a un conjunto de variables o atributos (Omar Ruiz, 2008).

Como ventajas de las reglas de inducción podemos citar las representaciones de hipótesis más "comprensibles" para el ser humano y el formalismo más popular de representación del conocimiento.

Entre los algoritmos, que implementan las técnicas reglas de clasificación se encuentran:

- Algoritmo 1R
- Algoritmo PRISM

Herramientas para aplicar técnicas de minería de datos

Para la aplicación de las técnicas de minería de datos existen diversas herramientas; algunas son independientes del sistema gestor de bases de datos y otras son nativas de un gestor de bases de datos específico.

Herramientas nativas del gestor

En los últimos años, empresas como ORACLE y SQL Server han incorporado algunos algoritmos o técnicas para el análisis de datos, buscando facilitar el proceso de descubrimiento de conocimiento para la toma de decisiones.

SQL Server Data Mining: es una herramienta que contiene las características necesarias para crear complejas soluciones de minería de datos, ya que permite:

- Aplicar soluciones de minería de datos utilizando Microsoft Excel.
- Entender cómo, cuándo y dónde aplicar los algoritmos que se incluyen en el servidor de SQL.
- Realizar la extracción de datos de procesamiento analítico en línea (OLAP).
- Utilizar SQL Server Management Studio para acceder y proteger los objetos de minería de datos.
- Utilizar SQL Server Business Intelligence Development Studio para crear y gestionar proyectos de minería de datos (MacLennan, Tang, & Crivat, 2009).

Entre las ventajas de la minería de datos de Microsoft podemos citar la integración estrecha con la plataforma de base de datos de clase mundial SQL Server, ya que aprovecha el desempeño, la seguridad y las características de optimización de SQL Server; la extensibilidad, ya que se puede extender la minería de datos de Microsoft para implementar algoritmos que no vienen incluidos en el producto.

Los algoritmos implementados por Microsoft son:

- Árboles de decisión.
- Bayes naive.
- Clústeres.
- Redes neuronales.
- Serie temporal.
- Regresión lineal.
- Clústeres de secuencia.
- Asociación.

Oracle Data Mining: permite que las empresas desarrollen aplicaciones de inteligencia de negocio avanzadas que exploten las bases de datos corporativas, descubran nuevos conocimientos e integren esa información en aplicaciones comerciales (Haberstroh, 2008).

Oracle Data Mining incorpora las siguientes funcionalidades de minería de datos para realizar clasificaciones, agrupamiento, predicciones y asociaciones.

- Agrupamiento (k-means, O-Cluster).
- Árboles de decisión.
- Atributo relevante.
- Característica de selección.
- Clasificador bayesiano (naive bayes).
- Máquinas de soporte vectorial (support vector machines).

- Modelos lineales generalizados
- Reglas de asociación (APRIORI).

Todas las funciones de los modelos son accesibles a través de una API basada en Java.

El carácter nativo de la solución es un plus fuerte, en tanto que las implementaciones de cada una de las etapas del proceso se encuentran incluidas en el motor.

Herramientas independientes del gestor

Entre las herramientas libres más utilizadas para la minería de datos se encuentran Weka (Waikato Environment for Knowledge Analysis) es una herramienta visual de distribución libre para el análisis y la extracción de conocimiento a partir de datos (V.Ramesh, 2011)

Las principales ventajas de la herramienta son:

- Es multiplataforma.
- Contiene una extensa colección de técnicas para preprocesamiento y modelado de datos.
- Es fácil de usar, gracias a su interfaz gráfica.
- Soporta varias tareas de minería de datos, especialmente preprocesamiento, agrupamiento, clasificación, regresión, visualización y selección.
- Permite combinar varios algoritmos basados en técnicas de minería de datos, para obtener mejores resultados en el descubrimiento de conocimiento.
- Es capaz de mostrar los datos en varios tipos de gráficos con el objetivo de proporcionar una mejor comprensión y un mejor análisis.

YALE RapidMiner

La herramienta fue desarrollada en Java, en 2001, por el departamento de inteligencia artificial en la universidad de Dortmund. Es multiplataforma, es un software de código abierto GNU y con licencia GPL. La última versión, incluye características como las de implicar nuevos formatos de entrada de datos con operadores para Microsoft Excel y SPSS. Desde la perspectiva de la visualización, YALE ofrece representaciones de datos en dispersión en 2D y 3D; representaciones de datos en formato SOM (Self Organizing Map); coordenadas paralelas y grandes posibilidades de transformar las visualizaciones de los datos.

De forma general, se puede decir que las herramientas de minería de datos de ORACLE y SQL Server son herramientas muy potentes, y que una de sus mayores fortalezas radica en la integración con el sistema gestor de base de datos. Sin embargo, ambas son herramientas propietarias y muy costosas para las empresas cubanas. Por otra parte, las herramientas como Weka y YALE Rapid Miner son herramientas libres, pero que tienen la desventaja de ser un proceso engorroso, puesto que requiere tiempo para la preparación, la vinculación de los datos con la herramienta, extendiendo así el tiempo de respuesta de los análisis.

3. RESULTADOS

Implementación del Algoritmo 1R

El algoritmo 1R, propuesto por Robert C. Holte en 1993, es un clasificador muy sencillo, que únicamente utiliza un atributo para la clasificación. A pesar de que el autor lo cataloga como "*Program 1R is ordinary in most respects.*" sus resultados pueden ser muy buenos en comparación con algoritmos mucho más complejos y su rendimiento promedio está por debajo de los de C4.5 en solo 5,7 puntos porcentuales de aciertos de clasificación según los estudios realizados por el autor del algoritmo (HOLTE, 1993).

```
1R (ejemplos) {
  Para cada atributo (A)
    Para cada valor del atributo (Ai)
      Contar el número de apariciones de cada clase con Ai
      Obtener la clase más frecuente (Cj)
      Crear una regla del tipo Ai -> Cj
    Calcular el error de las reglas del atributo A
  Escoger las reglas con menor error
}
```

Figura 1- Pseudocódigo del algoritmo 1R

La implementación del algoritmo 1R se realizó utilizando el pseudocódigo mostrado en la figura 1. Esta función solo permite trabajar con tablas que tengan atributos nominales y en las que no debe haber atributos con valores desconocidos para obtener el resultado deseado.

La función toma como entrada el nombre de la tabla y la clase sobre la cual se va a realizar el análisis y devuelve como resultado un conjunto de reglas para los atributos con la menor cantidad de errores.

Implementación del algoritmo PRISM

El algoritmo PRISM es un algoritmo de cubrimiento sencillo que asume que no hay ruido en los datos. Su objetivo es crear reglas perfectas que maximicen la relación p/t, siendo p la cantidad de ejemplos positivos cubiertos por la regla y t la cantidad de ejemplos cubiertos por la regla. (Chesñevar, 2009)

Este algoritmo tiene la característica de eliminar los ejemplos que va cubriendo por las reglas conformadas, por lo cual las reglas deben mostrarse e interpretarse en el orden que se van cubriendo.

```
PRISM (ejemplos) {
  Para cada clase (C)
    E = ejemplos
    Mientras E tenga ejemplos de C
      Crea una regla R con parte izquierda vacía y clase C
      Hasta R perfecta Hacer
        Para cada atributo A no incluido en R y cada valor v de A
          Considera añadir la condición A=v a la parte izquierda de R
          Selecciona el par A=v que maximice p/t
            (en caso de empates, escoge la que tenga p mayor)
        Añadir A=v a R
    Elimina de E los ejemplos cubiertos por R
```

Figura 2: Pseudocódigo del algoritmo PRISM

La implementación del algoritmo PRISM se realizó utilizando el pseudocódigo de la figura 2. Esta función solo permite atributos nominales y, para obtener el resultado deseado, no puede haber atributos con valores desconocidos..

La función toma como entrada el nombre de la tabla y la clase sobre la cual se va a realizar el análisis y devuelve como resultado un conjunto de reglas que se deben interpretar en el orden en que aparecen como lo estipula el algoritmo.

Implementación del algoritmo ID3

El ID3, propuesto por J. R Quinlan en 1986, es un algoritmo simple y, a la vez potente, que permite elaborar un árbol de decisión como un método para aproximar una función objetivo de valores discretos, que es resistente al ruido en los datos y que es capaz de hallar o aprender de una disyunción de expresiones.

Para construir el árbol, el algoritmo utiliza el análisis de la entropía, la teoría de la información (basada en la entropía) y la ganancia de información.

1. Seleccionar el atributo A_i que maximice la ganancia $G(A_i)$.
2. Crear un nodo para ese atributo con tantos sucesores como valores tenga.
3. Introducir los ejemplos en los sucesores según el valor que tenga el atributo A_i.
4. Por cada sucesor:
   a. Si sólo hay ejemplos de una clase, C_k, entonces etiquetarlo con C_k.
   b. Si no, llamar a ID3 con una tabla formada por los ejemplos de ese nodo, eliminando la columna del atributo A_i.

Figura 3 - Pseudocódigo del algoritmo ID3

La implementación de algoritmo ID3 se realizó utilizando el pseudocódigo de la figura 3. Esta función sólo permite trabajar con tablas que tengan atributos nominales y, para obtener el resultado deseado, no puede haber atributos con valores desconocidos.

La función toma como entrada el nombre de la tabla y la clase sobre la cual se va a realizar el análisis y devuelve como resultado un conjunto de reglas derivadas del árbol de decisión.

Mecanismo para optimizar el rendimiento de los algoritmos implementados.

Generalmente las tablas sobre las que se realizan análisis de minería de datos cuentan con un gran volumen de información, lo que puede retrasar el resultado de dicho estudio.

Una de las opciones que brinda PostgreSQL para mejorar el rendimiento en estos casos es el particionado de tabla, que permite obtener un mejor desempeño a la hora de consultar dichas tablas.

El particionado de tablas es una técnica que consiste en descomponer una enorme tabla (padre) en un conjunto de tablas hijas. Esta técnica reduce la cantidad de lecturas físicas a la base de datos cuando se ejecutan las consultas.

En PostgreSQL los tipos de particionado existentes son por rango y por lista (The PostgreSQL Global Development Group, 2011).

- Particionado por rango: se crean particiones mediantes rangos definidos en base a cualquier columna que no se solape entre los rangos de valores asignados a diferentes tablas hijas.
- Particionado por lista: se crean particiones por valores.

En la presente investigación se implementa una función que permite realizar el particionado de la tabla según los valores de la clase (particionado por lista). Lo cual permite agilizar la búsqueda a la hora de clasificar.

La función tiene como parámetro de entrada la tabla que se desea particionar y el nombre de la clase, creándose tantas particiones como valores tenga la clase.

Las tablas padres creadas por la función tendrán como nombre la concatenación de máster más el antiguo nombre de la tabla y las tablas hijas tendrán como nombre la concatenación del antiguo nombre de la tabla más el valor de la clase por la cual se creó la partición.

Integración de los algoritmos al SGBD PostgreSQL.

A partir de la versión 9.1, PostgreSQL brinda facilidades para que los usuarios puedan crear, cargar, actualizar y administrar extensiones utilizando el objeto de base de datos EXTENSION (PostgreSQL, 2011).

Entre las ventajas de esta nueva funcionalidad, se encuentra que, en lugar de ejecutar un script SQL para cargar objetos que estén "separados" en su base de datos, se tendrá la extensión como un paquete que contendrá todos los objetos definidos en ella, lo que trae gran beneficio al actualizarla o eliminarla, ya que, por ejemplo, se pueden eliminar todos los objetos utilizando el comando DROP EXTENSION sin necesidad de especificar cada uno de lo objetos definidos dentro de la extensión. Además de eso, se cuenta con un repositorio para obtener extensiones y contribuir con ellas (http://pgxn.org/).

La integración de los algoritmos implementados con el SGBD se va a realizar mediante la creación de una extensión por las ventajas que PostgreSQL brinda para su creación.

Creación de la extensión minería de datos

Para crear la extensión, se deben crear dos archivos. En el primero se definen las características de la extensión ; en el segundo, los objetos SQL que se desean agregar. Estos deben ubicarse en el directorio de la instalación "C:\Archivos de programa\PostgreSQL\9.1\share\extension"

En el archivo "mineria_datos.control" creado para agregar la extensión en que se cargaron las funciones de los algoritmos implementados se definieron los siguientes parámetros:

- comment: una breve descripción sobre el contenido de la extensión creada.
- Encoding: el tipo de codificación utilizado.
- default_version: la versión de la extension.
- relocatable: si se puede.
- schema: el esquema en que se almacenarán los objetos creados por la extensión.

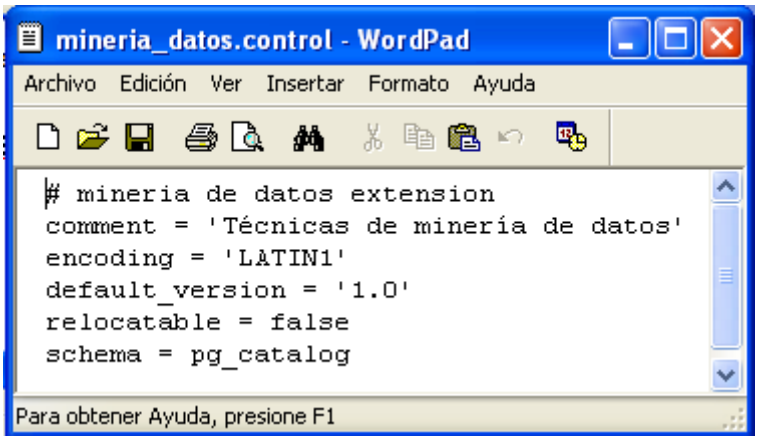

Figura 4 - Archivo que contiene las características de la extensión

Una vez que se haya definido el archivo "mineria_datos.control", se especifica el archivo que contendrá el código de las funciones desarrolladas "mineria_datos--1.0.sql"

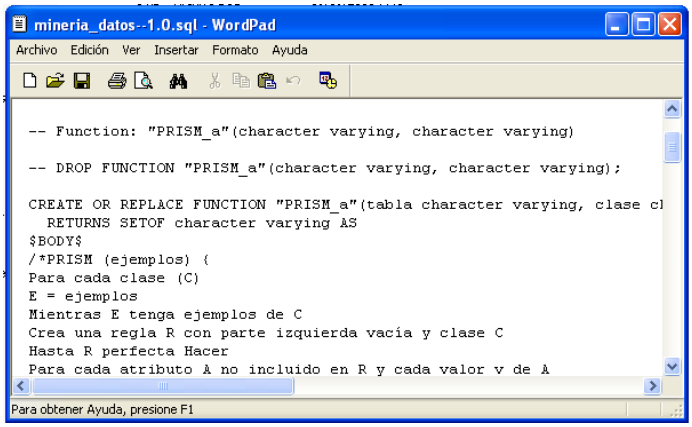

Figura 5 - Archivo que contiene el código y los objetos definidos en la extensión

Trabajo con la extensión de minería de datos.

Para que los usuarios puedan utilizar la extensión de minería de datos, simplemente deben ejecutar el comando "CREATE EXTENSION mineria_datos" que cargará la extensión como se puede apreciar en la imagen 6.

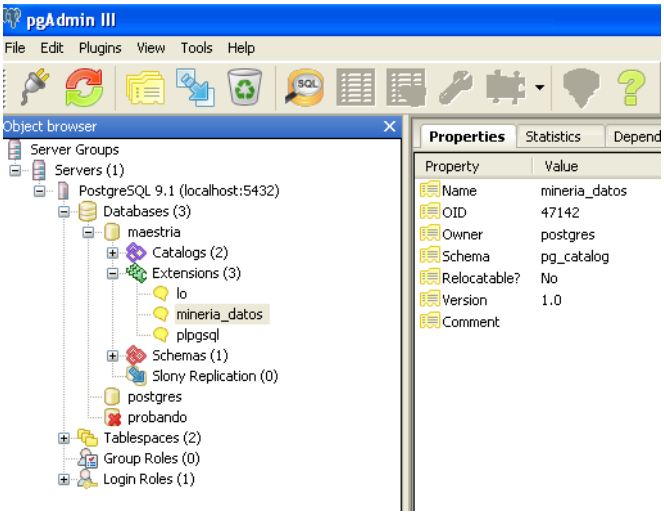

Figura 6 - La extensión "mineria_datos" creada

Para consultar las funciones agregadas por la extensión, se debe consultar, en el esquema pg_catalogo, la carpeta de funciones como se muestra en la figura 7.

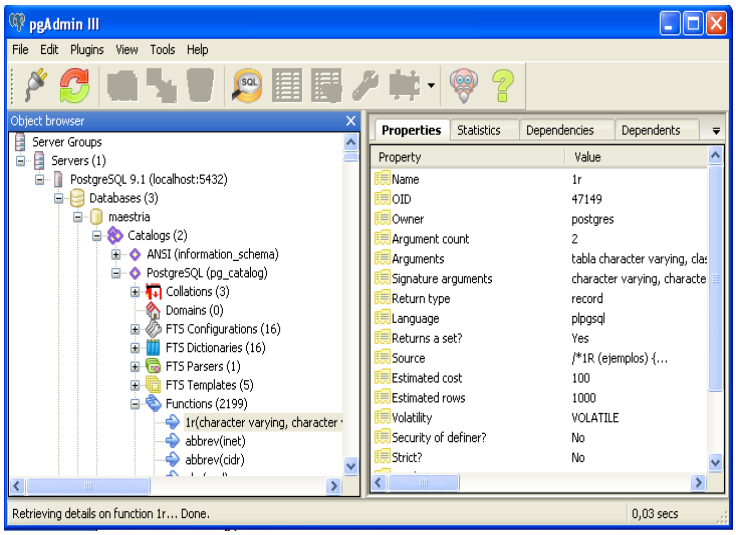

Figura 7 - Funciones de la extensión "mineria_datos" ubicada en el esquema pg_catalogo

La extensión de minería de datos creada puede ser usada a partir de la versión 9.1 de PostgreSQL.

Evaluación de los algoritmos implementados.

Para validar los algoritmos, se diseñó un experimento definido por Roberto Hernández Sampieri [1] como "un estudio de investigación en el que se manipulan deliberadamente una o más variables independientes (supuestas causas) para analizar las consecuencias que la manipulación tiene sobre una o más variables dependientes (supuestos efectos), dentro de una situación de control para el investigador" (Martínez Valenzuela, 2007).

En este caso, se definieron como variables independientes la cantidad de registros y la herramienta utilizada para aplicar la minería de datos. Como variables dependientes se identificaron el tiempo de respuesta y el resultado de los algoritmos.

Para una mejor comprensión del diseño del experimento, se resume la definición operacional en las tablas 1 y 2.

[1]**Dr Roberto Hernández Sampieri** Director del Centro de Investigación de la Universidad de Celaya y profesor en el Instituto Politécnico Nacional

Tabla 1- Operacionalización de las variables independientes

Variable	Tipo de variable	Operacionalización	Categorías
cantidad de registros	Independiente	Cantidad de filas que posee la tabla que será analizada.	- 100002 - 500010 - 1000020
Herramienta	Independiente	Herramienta utilizada para aplicar la minería de datos	- Weka - PostgreSQL(Algoritmos integrados al SGBD)
Particionado	Independiente	Si la tabla a la que se van a aplicar los algoritmos de minería de datos está o no particionada	- Particionado - No particionado

Tabla 2 - Operacionalización de las variables dependientes

Variable	Unidad de medida
Tiempo de respuesta	Intervalo de tiempo (segundos)
Resultados del algoritmo	Grado de acuerdo (sí o no)

Aplicación del experimento

Para el entorno del experimento se seleccionó una computadora Haier con Procesador Intel Celeron 2000 MHz, una memoria RAM de 1024 MB y un disco duro de 120 Gb. Además se cuenta con el servidor de PostgreSQL 9.1, la herramienta PgAdminIII y Weka 3.6.7. Esta última fue seleccionada para la comparación, ya que, según el estudio realizado en el artículo "Herramientas de Minería de Datos" publicado en la revista RCCI, es la herramienta libre más conocida y más utilizada.

Relación de la variable cantidad de registros con el tiempo de resultado.

En este primer caso se va a realizar un estudio de cómo se comporta el tiempo de respuesta en las herramientas Weka y el gestor PostgreSQL cuando se manipula la variable cantidad de registros para cada uno de los algoritmos estudiados.

Los tiempos de respuestas de Weka se van a medir desde el momento en que se establece la consulta para cargar los datos de la BD hasta el momento en que la herramienta brinda el resultado del algoritmo.

Tabla 3 - Resultado de la manipulación de la variable cantidad de registros para el algoritmo 1R

cantidad de registros	Weka (1R)	PostgreSQL(1R)
100002	**11,13**	**1,5**
500010	**17,24**	**8,18**
1000020	**---**	**16,43**

En la tabla número 3 se puede observar que a medida que se fue incrementando la cantidad de registros los tiempos de análisis para el algoritmo 1R aumentaron y, en el caso de la herramienta Weka, los tiempos de respuestas resultaron superiores con respecto al análisis realizado mediante los algoritmos integrados al SGBD PostgreSQL. Para la categoría o nivel de 1000020 registros, los análisis no se pudieron efectuar con la herramienta Weka ya que esta retornó error debido a la gran cantidad de datos.

Para una mayor comprensión de la información de la tabla 3, se representa la gráfica de la figura 8.

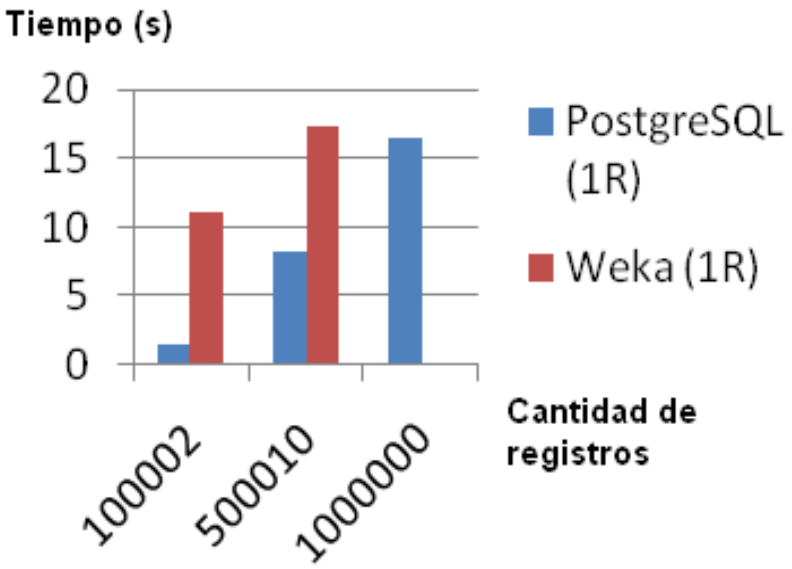

Figura 8 - Resultados de la variable cantidad de registros para el 1R

En la tabla 4 se puede apreciar que, del mismo modo que ocurrió con el algoritmo PRISM, a medida que se aumentaron la cantidad de filas, los tiempos de análisis se incrementaron. Para el caso de la categoría o nivel 500010 registros, la herramienta Weka tardó 3600 segundos ejecutando sin mostrar el resultado y, en el caso de 1000020, ocurrió error.

Tabla 4 - Resultado de la manipulación de la variable cantidad de registros para el algoritmo PRISM

cantidad de registro	Weka(PRISM)	PostgreSQL(PRISM)
100002	11,33	10,1
500010	3600	94,7
1000020	---	219,45

La figura 9 presenta los resultados del análisis realizado anteriormente.

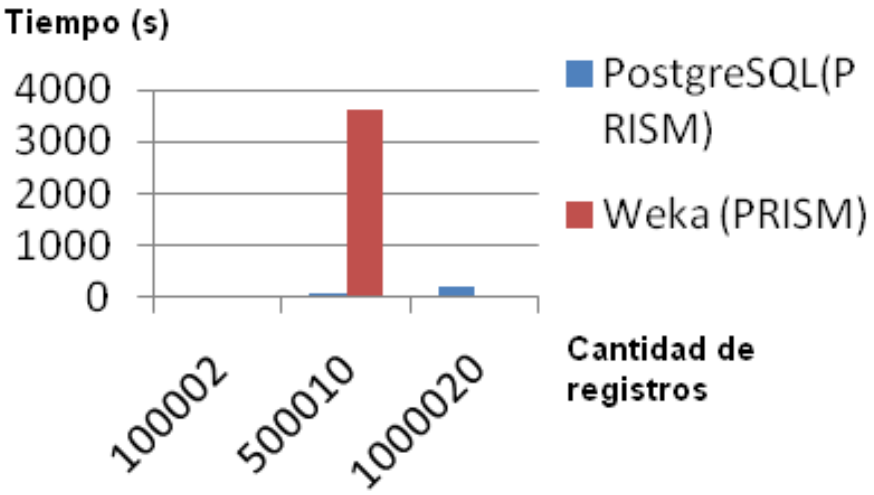

Figura 9 - Resultados de la variable cantidad de registros para el PRISM

La la tabla 5 deja evidente, al igual que las tablas anteriores, la relación directamente proporcional entre la cantidad de registros y el tiempo de análisis del algoritmo ID3, resaltando los tiempos de la solución propuesta en la investigación que son menores. En la figura 10 se puede observar el gráfico de los resultados del análisis realizado en la tabla 5.

Tabla 5: Resultado de la manipulación de la variable cantidad de registros para el algoritmo ID3

Cantidad de registro	Weka (ID3)	PostgreSQL(ID3)
100002	7,42	2,58
500010	23,78	14,36
1000020	---	41,19

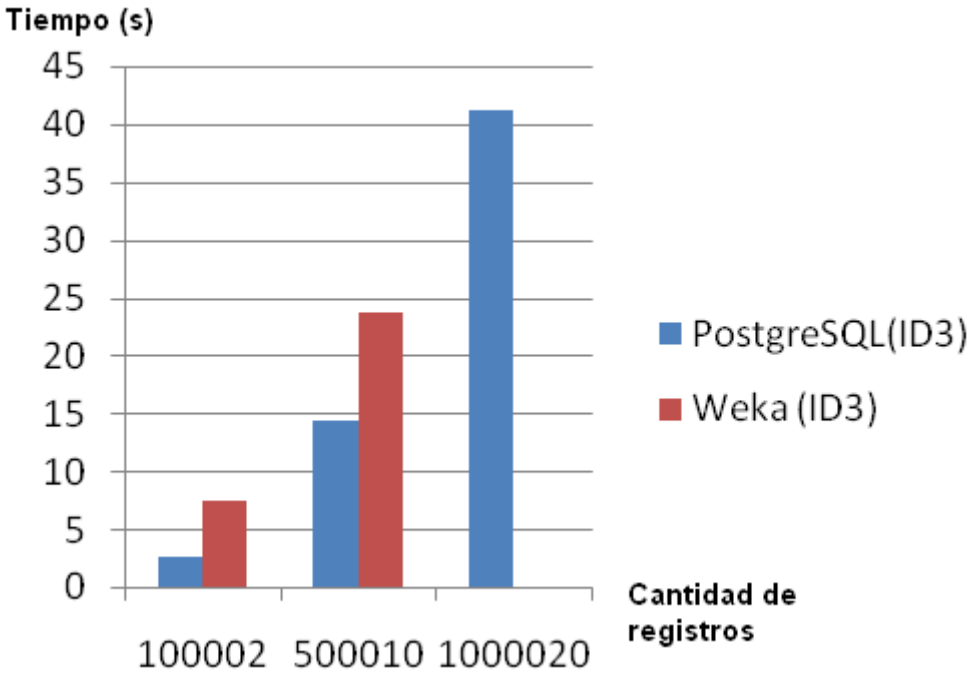

Figura 10 - Resultados de la variable cantidad de registros para el ID3

Relación de la variable cantidad de registros con las respuestas de los algoritmos.

En el caso número 2 se analiza el comportamiento de la variable resultado de los algoritmos al manipular la cantidad de registros que serán analizados .

Tabla 6 - Resultados de la relación entre la cantidad de registros y los resultados para el algoritmo 1R

Cantidad de registro	Weka (1R)	PostgreSQL(1R)
100002	Si	Si
500010	Si	Si
1 000020	No	Si

Tabla 7 - Resultados de la relación entre la cantidad de registros y los resultados para el algoritmo PRISM

Cantidad de registro	Weka(PRISM)	PostgreSQL(PRISM)
100002	Si	Si
500010	No	Si
1000020	No	Si

Tabla 8 - Resultados de la relación entre la cantidad de registros y los resultados para el algoritmo ID3

Cantidad de registro	Weka (ID3)	PostgreSQL(ID3)
100002	Si	Si
500010	Si	Si
1000020	No	Si

Al analizar los resultados de las tablas 6, 7 y 8 se puede concluir que a medida que se incrementó la cantidad de registros se dificultó el análisis de los datos por medio de la herramienta Weka.

Validación del mecanismo de particionado

Para validar que el particionado de tabla propuesto mejora el rendimiento de los algoritmos, se crearán particiones en la tabla prueba_d, que cuenta con 8000160 registros.

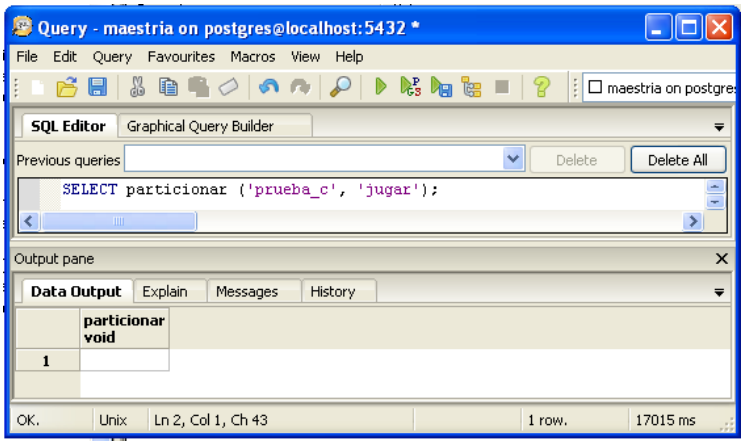

Figura 11- Particionado de la tabla prueba_c

Como resultado del particionado, se obtienen 3 tablas masterprueba_d que es la tabla padre, prueba_dsi que contiene todos los registros que su case tiene valor “si” y prueba_dno los registros con valores “no”.

Al aplicar el algoritmo 1R integrado al SGBD PostgreSQL sin crear particiones, el tiempo de respuesta es de 174,184 segundos y tras haber particionado la tabla, es de 149,98 segundo (véase el anexo 5).

La figura 12 muestra cómo el análisis en la tabla particionada por el valor de la clase es menor que en la tabla normal, lo que demuestra que el mecanismo de particionado de datos agiliza el resultado del algoritmo 1R.

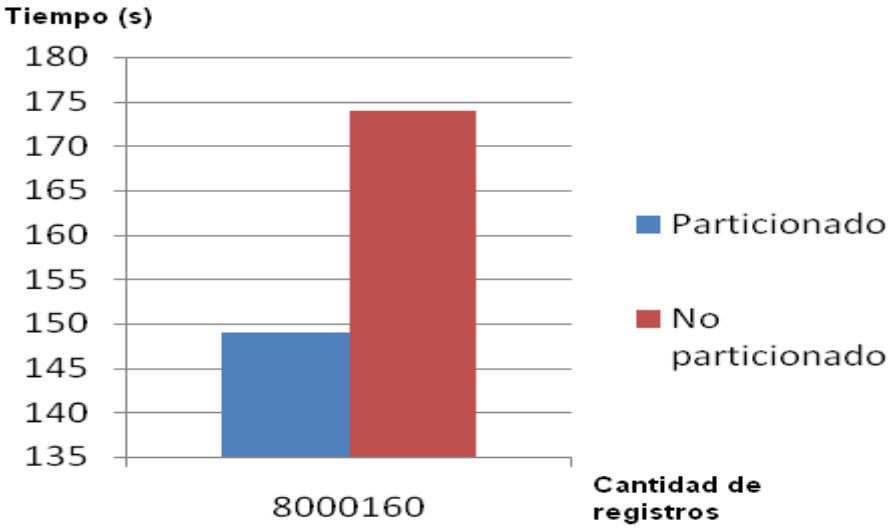

Figura 12 - Comparación de los resultados de los análisis en tablas particionada y sin particionar

4. CONCLUSIONES

Los resultados obtenidos en esta investigación permiten afirmar que: las técnicas de minería de datos árboles de decisión y reglas de inducción permiten obtener, como resultado final, reglas que, por sus características, son unas de las formas de representar que más se han divulgado y unas de las técnicas que las personas comprenden con mayor facilidad.

Además se pudo evidenciar que las herramientas libres existentes de minería de datos tienen el inconveniente de ser independientes del SGBD, razón por la cual se implementaron tres algoritmos de técnicas de clasificación y se integraron al SGBD PostgreSQL a través de la creación de una extensión, lo que contribuye a la soberanía tecnológica del país y a que el gestor sea más competitivo.

Asimismo, se desarrolló una función que permite aprovechar uno de los mecanismos de optimización del gestor para mejorar los resultados de respuesta de los algoritmos implementados.

Los algoritmos implementados fueron validados por medio de un diseño de experimento que permitió observar que los tiempos de análisis de los algoritmos integrados al SGBD son menores que los resultados de la herramienta Weka.

5. RECOMENDACIONES

Se deben integrar otros algoritmos de minería de datos al SGBD PostgreSQL de la técnica de Reglas de Asociación, ya que esta es tan descriptiva como las utilizadas en la investigación, además de figurar entre las más utilizadas.

REFERENCIA BIBLIOGRÁFICA

Chesñevar, C. I. (2009). datamining y aprendizaje automatizado. obtenido de http://cs.uns.edu.ar/~cic/dm2009/downloads/transparencias/05_dm%20(learning_rules).pdf

Haberstroh, R. (2008). Oracle ® data mining tutorial for Oracle Data Mining 11g Release 1, Oracle.

Heughes Escobar, V. (2007). Minería web de uso y perfiles de usuario: aplicaciones con lógica difusa. tesis de doctorado, Universidad de Granada.

Holte, R. C. (1993). Very simple classification rules perform well on most commonly used datasets, Machine Learning, 11, 63-91, Kluwer Academic Publishers, Boston

Maclennan, J., Tang, Z., & Crivat, B. (2009). Data mining with Microsoft SQL server 2008. Wiley Publishing, Inc., Indianapolis, Indiana.

Martínez Valenzuela, V. Et al. (octubre de 2007). Diseño experimental. Universidad Autonoma de Baja California, obtenido de http://www.slideshare.net/hayimemaishte/diseo-experimental

Molina López, J. M., & García Herrero, J. Técnicas de análisis de datos. Universidad Carlos III. Madrid. 4-5

Moreno, G. (octubre de 2007). Recuperado el enero de 2012, de http://gamoreno.wordpress.com/2007/10/03/tecnicas-mas-usadas-en-la-mineria-de-datos/

Omar Ruiz, S. B. , Bauz. Sergio, Jimenez, Maria (2009). Aplicación de minería de datos para detección de patrones en investigaciones biotecnológicas. ESPOL, Ecuador http://www.dspace.espol.edu.ec/handle/123456789/4719

PostgreSQL. (12 de septiembre de 2011), The PostgreSQL Global Development Group obtenido de http://www.postgresql.org/about/press/presskit91/es/

Rodríguez Suárez, Yuniet; Díaz Amador, Anolandy. (2011) Herramientas de minería de datos. Revista Cubana de Ciencias Informáticas, [S.l.], v. 3, n. 3-4, oct.. ISSN 2227-1899. Disponible en: http://rcci.uci.cu/index.php/rcci/article/view/78 .

Solarte Martínez, G. R. (2009). técnicas de clasificación y análisis de representación del conocimiento para problemas de diagnóstico. recuperado el diciembre de 2011, de http://www.utp.edu.co/php/revistas/scientiaettechnica/docsftp/222025177-182.pdf

Soto Jaramillo, C. M. (2009). Incorporación de técnicas multivariantes en un sistema gestor de bases de datos. Universidad Nacional de Colombia http://www.bdigital.unal.edu.co/895/1/71335481_2009.pdf

The PostgreSQL global development group. (2011). PostgreSQL 9.1.0 documentation.

U Fayyad, G. P.-S. (1996). data mining and knowledge discovery in databases: an overview, communications of acm. obtenido de http://dl.acm.org/citation.cfm?id=240464

V.Ramesh, P. P. (agosto de 2011). Performance analysis of data mining techniques for placement chance prediction. recuperado el diciembre de 2011, de http://www.ijser.org/researchpaper%5cperformance-analysis-of-data-mining-techniques-for-placement-chance-prediction.pdf

Vazquez Ortíz, Y., Mesa Reyes, Y., & Castillo Martínez, g. (2012). Comunidad técnica cubana de PostgreSQL: Arma para la migración del país a tecnologías de bases de datos de código. Uciencia.