ISSN: 1692-7257 - Volumen 1 - Número 15 - Año 2010

Revista Colombiana de Tecnologías de Avanzada

Recibido: 14 de agosto de 2009
Aceptado: 22 de octubre de 2009

POLARIS: A WEB USAGE MINING TOOL

POLARIS: UNA HERRAMIENTA PARA MINERIA DE USO DE LA WEB

PhD. Ricardo Timarán Pereira, Ing. Johana Daza Burbano, Ing. Alejandra Zuleta Medina, Ing. Diana Angulo Urbano

Universidad de Nariño. Departamento de Sistemas de la Facultad de Ingeniería
Ciudad Universitaria Torobajo, San Juan de Pasto, Nariño, Colombia
E-mail: ritimar@udenar.edu.co, {jeda13, alejazul07}@gmail.com, agaphe70@hotmail.com }

Abstract: This paper presents the first version of Polaris, a Web Usage Mining tool, developed in the KDD laboratory of Department of Systems at the University of Nariño. Polaris' architecture is composed of three modules: Utilities, Kernel and Graphical User Interface. Web Usage Mining Algorithms were implemented: Apriori, FPGrowth and EquipAsso for Association task; C4.5 and Mate-tree Algorithms for classification task and HPG algorithm for Learning Navigation Patterns. Polaris' functionality is tested with the analysis and evaluation of algorithms performance for association rules and classification.

Keywords: Web mining, web usage mining, web mining tools.

Resumen: En este artículo se presenta la primera versión de Polaris, una herramienta para Minería de Uso de la Web, desarrollada en el laboratorio KDD del departamento de Sistemas de la Universidad de Nariño. La arquitectura de Polaris está compuesta por tres módulos: Utilidades, Kernel e Interfaz Grafica de Usuario. Se implementaron los algoritmos de minería de uso de la web: Apriori, FPGrowth y EquipAsso para la tarea de Asociación; C4.5 y Mate-tree para la tarea de Clasificación y el algoritmo HPG para Aprendizaje de Patrones de Navegación. Se prueba su funcionalidad con el análisis y evaluación de desempeño de los algoritmos de reglas de Asociación y Clasificación.

Palabras clave: Minería web, minería de uso, herramientas de minería web.

1 INTRODUCCION

La Internet es sin duda el mecanismo más importante, práctico y altamente difundido para el intercambio de información de todo tipo convirtiéndose en un recurso de disposición pública y general al cual acceden las personas en busca de satisfacer sus necesidades de información, obtener algún recurso en particular o realizar algún tipo de transacción (Hernández *et al.*, 2004). Sin embargo, el gran volumen de información contenido en la Web hace cada vez más complejo y caótico el proceso de búsqueda de la misma, sobre todo cuando no se cuentan con los recursos necesarios para llegar a ella o se desconoce una forma de acceso rápida que garantice la obtención del recurso deseado en medio de un mar de sobrecarga de información.

Una solución a este problema, es la aplicación de técnicas de Minería de Datos sobre los datos contenidos en la World Wide Web, denominada Minería Web (*Web Mining*). A diferencia de los datos almacenados en una base de datos, la Web es un gran repositorio de hipertexto, donde los documentos contienen datos de muy diverso tipo (texto, imágenes, audio, video, entre otros) que son no estructurados o semiestructurados.

Esta diversidad permite el descubrimiento de patrones basándose en tres conceptos: el contenido, la estructura y el uso.

En este artículo se presenta la primera versión de una herramienta para realizar Minería de Uso de la Web denominada Polaris. Esta herramienta está compuesta por tres módulos: el módulo de utilidades con clases y bibliotecas comunes que permiten la recuperación de datos desde los archivos logs, el módulo Kernel donde se implementaron los filtros para el preprocesamiento de los datos y los algoritmos de Minería de Uso, y el modulo de la interfaz grafica que permite la interacción amigable entre el usuario y la herramienta. En Polaris se encuentran implementados los algoritmos Apriori (Agrawal y Srikant, 1994), FPGrowth (Han *et al*., 2004) y EquipAsso (Timarán y Millán, 2005a; Timarán y Millán, 2005b) para la tarea de Asociación, los algoritmos C4.5 (Quinlan, 1993) y Mate-tree (Timarán, 2007) para la tarea de Clasificación y el algoritmo HPG (Hernández *et al*., 2004) para aprendizaje de patrones de navegación. Se prueba la funcionalidad de Polaris con el análisis y evaluación de desempeño del algoritmo de reglas de Asociación EquipAsso con respecto a los algoritmos Apriori y FPGrowth y el algoritmo de Clasificación Mate-tree con respecto al algoritmo C4.5.

El resto del artículo está organiza en secciones. En la sección 2, se abordan los conceptos preliminares de Minería Web. En la sección 3 se describe la arquitectura de la herramienta Polaris. En la sección 4, se presentan algunos aspectos de la implementación de Polaris. En la sección 5, se muestran los resultados de las pruebas realizadas con esta herramienta y finalmente en la sección 6 se dan las conclusiones y trabajos futuros.

2 CONCEPTOS PRELIMINARES

2.1 Minería Web

La minería web es el proceso de descubrir patrones interesantes y potencialmente útiles en la estructura, el contenido y la utilización de los sitios Web (Bamshad et al., 1997). Con base a esta definición, la minería web se clasifica en minería de la estructura web, minería del contenido de la web y minería de uso de la web.

La Minería de la Estructura de la Web (*Web Structure Mining*) analiza la topología de los hipervínculos de la estructura de un sitio web con el fin de categorizar sus páginas web (Tingshao, 2001).

Minería del Contenido de la Web (*Web Content Mining*) identifica patrones relativos a los contenidos textuales y gráficos de los documentos web y a las búsquedas que se realizan sobre los mismos (Filocamo y Chesñevar, 2004).

Minería de Uso de la Web (*Web Usage Mining*) encuentra patrones sobre el uso que se le da a la web a través del análisis de los registros de los servidores (*log files*) derivados de la interacción de los usuarios con la web (Villena *et al*., 2002).

2.2 El Proceso de Minería de Uso de la Web

Minería de Uso de la Web es el proceso de aplicar técnicas de minería de datos para descubrir patrones de uso a partir de archivos bitácoras (*log files*) de los servidores web, con el fin de entender el comportamiento y hábitos de los usuarios del servidor y mejorar el diseño del sitio web (Villena *et al*., 2002). Este proceso no es trivial, compuesto por una secuencia iterativa de las fases de preprocesamiento de datos, descubrimiento de patrones y análisis de patrones.

El preprocesamiento de datos es el primer paso de la minería de uso y trata con los archivos bitácoras de los servidores. Incluye subprocesos específicos como limpieza de datos, identificación de páginas vistas, identificación de usuarios, identificación de sesiones (también llamado sesionalización), inferencia de referencias pérdidas debido a problemas de caché y la identificación de transacciones (ó episodios).

El descubrimiento de patrones es la fase de minería de uso web, donde, una vez los datos han sido preparados, se aplican técnicas estadísticas y de Aprendizaje de Máquina (*Machine Learning*), para extraer patrones de uso. Existen dos aproximaciones principales para extraer patrones de navegación de los usuarios desde los datos *logs*: las técnicas estándares de minería de datos como reglas de asociación, clasificación, agrupación (*clustering*) y patrones secuenciales, y las técnicas de aprendizaje de patrones de navegación (Hernández *et al*., 2004).

El análisis de patrones se ocupa de desarrollar técnicas y herramientas de visualización que faciliten al usuario, la comprensión del conocimiento extraído y su contraste con el conocimiento descubierto anteriormente sobre el problema tratado.

3 ARQUITECTURA DE POLARIS

La arquitectura de la herramienta Polaris la componen tres módulos, el de utilidades, el Kernel y la interfaz grafica de usuario, que juntos soportan el proceso de minería de uso de la web. La arquitectura de esta herramienta se muestra en la fig. 1.

3.1 Módulo de Utilidades

Este módulo es el encargado de realizar la conexión a los archivos *log* de acceso a servidores web Apache e IIIS. Reconoce archivos *logs* de formato común (CLF), formato extendido (ELF) o formato IIIS. Permite almacenar y recuperar un proyecto realizado en Polaris en los formatos *.pol y *.xml. Escoge las extensiones de documentos que se van a analizar en el los *logs* de acceso, tales como .html, .htm, .php, .jsp, .doc, .pdf, .asp, .aspx, .php4, .txt y .xml.

Por otra parte, en este módulo se mantiene una colección de clases principales y librerías que son utilizadas por otras clases para el desarrollo de tareas comunes como manipulación, visualización e intercambio de datos, permitiendo así, la reutilización de código.

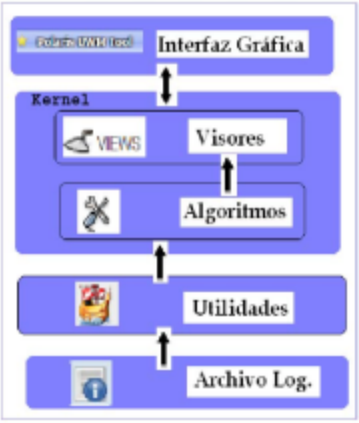

Fig. 1. Arquitectura de la herramienta Polaris

3.2 Módulo de Kernel

El objetivo de este módulo es realizar el tratamiento de los archivos *log* para posteriormente aplicar las técnicas y los algoritmos de minería de datos respectivos. Está compuesto por los submódulos de Preprocesamiento de datos y Algoritmos.

El módulo de Preprocesamiento contiene todos los filtros necesarios para poder realizar la limpieza y transformación de los archivos *log*. El proceso de limpieza se hace automáticamente, en el momento de seleccionar el archivo *log*. Se eliminan del archivo *log* todos aquellos datos erróneos o aquellos datos que no contribuyen al análisis que se va a realizar, como por ejemplo, se elimina todas aquellas peticiones que no son exitosas, es decir aquellas que comiencen con los códigos de error de servidor 400 y 500, se elimina todos los archivos con extensiones .gif, .jpg, .jpeg, .png, .swf, .map y se elimina peticiones realizadas por los *robots*, como Googlebot, InfoSeek Robot 1.0, Infoseek Sidewinder, PerlCrawler 1.0, Scooter, SpiderBot, Site Searcher, entre otros.

El proceso de transformación incluye los siguientes filtros:

Inter.Session transforma el archivo *log* a un archivo con sesiones por intervalos. El intervalo por defecto es de 30 minutos por sesión.

GAP Session transforma el archivo *log* a un archivo con sesiones GAP, es decir, separación por tiempo muerto.

Discretize Se hace una transformación de las sesiones en una nueva sesión, transformando los valores continuos a discretos.

El módulo de Algoritmos contiene las clases necesarias para la aplicación de los algoritmos de minería de datos que permiten extraer patrones de uso para las tareas de Asociación, Clasificación y para el aprendizaje de patrones de navegación.

Para la tarea de asociación se encuentran implementados los algoritmos:

A priori tiene como objetivo reducir el número de conjuntos considerados, generando un conjunto de *itemsets* frecuentes a partir de *itemsets* candidatos (Agrawal y Srikant, 1994).

FP–Growth utiliza una estructura de datos llamada árbol de patrones frecuentes (*FP-tree)*, la cual es una estructura que permite calcular los patrones frecuentes sin generar patrones candidatos (Han, J. et al., 2004).

EquipAsso es un algoritmo, para el cálculo de los itemsets frecuentes basado en dos operadores del álgebra relacional para Asociación: *Associator* y *EquiKeep* e implementado en el lenguaje SQL mediante las primitivas SQL *Associator Range* y *EquiKeep On* (Timarán y Millán, 2005a; Timarán y Millán, 2005b).

Para la tarea de clasificación se encuentran implementados algoritmos:

C4.5: construye un árbol de arriba hacia abajo recursivamente utilizando la manera de divide y vencerás y una métrica basada en la entropía, conocida como ganancia de información (Quinlan, 1993).

Mate-tree: es un algoritmo basado en el operador algebraico relacional *Mate* que conjuntamente con los operadores agregados *Entro* y *Gain* facilitan el cálculo de la *Ganancia de Información* y con el operador algebraico relacional *Describe Classifier*, la construcción del árbol de decisión (Timarán, 2007).

Para el aprendizaje de patrones de navegación se implementó el algoritmo HPG (*hypertext probabilistic gramar*) que representa las sesiones de navegación de los usuarios inferidas desde los archivos *log*, como una gramática probabilística de hipertexto, tal que las cadenas generadas por la gramática con mayor probabilidad corresponden a los caminos preferidos por los usuarios (Hernández *et al*., 2004).

3.3 Módulo de Interfaz Gráfica de Usuario

Este módulo proporciona el ambiente gráfico necesario para que el usuario pueda interactuar de una forma fácil y agradable con los diferentes componentes de la herramienta Polaris. Además da soporte visual al kernel, a través del módulo de visores para construir y desplegar las estructuras para la visualización gráfica y dinámica de los resultados obtenidos después de la aplicación de las diferentes técnicas y algoritmos de minería de datos. El conocimiento extraído se lo puede visualizar en forma de tablas, árboles, gráficas estadísticas tipo barras, líneas, pastel, y grafos. En la figura 2 se muestra el entorno gráfico de Polaris.

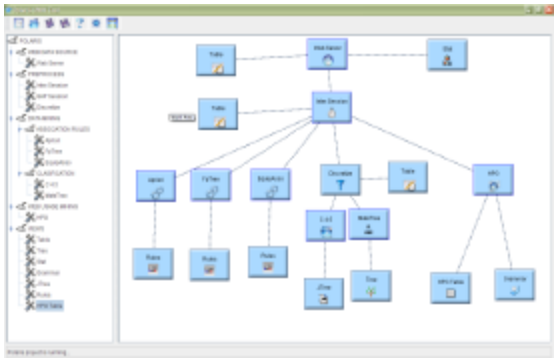

Fig . 2. Entorno gráfico de la herramienta Polaris

4 ASPECTOS DE IMPLEMENTACIÓN DE LA HERRAMIENTA POLARIS

Polaris se desarrolló bajo el sistema operativo Fedora Core versión 6 bajo una arquitectura de procesador a 64 bits, utilizando el lenguaje de programación Java 6.0, lo que la convierte en una herramienta independiente a la plataforma donde se ejecute. Para su construcción, se usaron herramientas de software libre con aplicaciones como el IDE de desarrollo NetBeans 5.5, Subversion, para el control de versiones, The Gimp y KIconEdit, para la manipulación de imágenes e iconos, PostgreSQL 8.2, para las pruebas de conexión y evaluación, entre otros.

La herramienta Polaris se diseño utilizando el análisis y diseño orientado a objetos con el lenguaje de modelamiento unificado UML. Su diseño es modular con el fin de permitir el crecimiento continuo de esta herramienta. Nuevos filtros, algoritmos y vistas pueden ser fácilmente implementados e incluidos en Polaris siguiendo este diseño. La estandarización de las entradas de los algoritmos, provenientes del módulo de utilidades o del kernel, y sus salidas, hacia los esquemas de visualización, facilitó al grupo de desarrollo la implementación de los algoritmos, trabajando de manera distribuida, haciendo uso de aplicaciones para el control de versiones.

5 PRUEBAS DE FUNCIONALIDAD DE POLARIS

Las pruebas de funcionalidad y rendimiento de los algoritmos implementados en la herramienta Polaris se realizaron en un computador Intel Pentium IV de 3.0. GHz, Disco Duro de 180 GB, memoria de 2 Gigas y tarjeta de video de 64 megas. Los conjuntos de datos utilizados en las pruebas pertenecen a diferentes archivos *log* de repositorios reales de la Universidad de Nariño y de la NASA disponibles en:
http://www.nasa.gov/centers/kennedy/home/index.html

5.1 Evaluación de rendimiento de los Algoritmos de Asociación

Para evaluar el rendimiento de los algoritmos de Asociación Apriori, Fp-Growth y EquipAsso se utilizaron los archivos *log* de uno de los servidores de la Universidad de Nariño. En la tabla 1 se muestran los archivos *log* utilizados.

Las pruebas consistieron en medir el tiempo de estos algoritmos con diferentes soportes mínimos. Los resultados de las pruebas sobre el archivo *www.access_log* se muestran en la tabla 2 y fig. 3.

Tabla1. Archivos log Universidad de Nariño

Conjunto de datos Asociación		
Nombre del Archivo	**Número de Registros**	**Tamaño KB**
access_log	176	457
ainfo_acces_log	188	1923
matriculas_access_log	476	5958
estudiantes_access_log	492	4467
akane_access_log	570	2037
personal_access_log	1013	768
ci_access_log	5443	10,068
www.access_log	19425	93759

Tabla 2. Rendimiento algoritmos Asociación con el archivo www.access log

www.access_log			
Soporte %	**tiempo milisegundos**		
	Apriori	**FPGrowth**	**EquipAsso**
1	17345	16799	15390
0,75	18585	14468	12532
0,5	15296	13437	8531
0,25	16906	12547	9625
0,05	19344	14781	9485

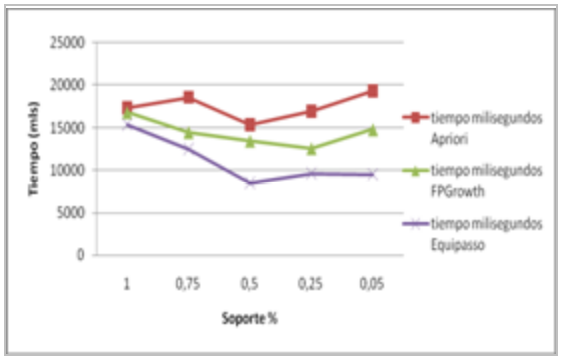

Fig. 3. Comportamiento algoritmos Asociación con el archivo www_access_log

Analizando los resultados de las pruebas realizadas se puede decir que el algoritmo EquipAsso es el que mejor comportamiento tiene, contrario al algoritmo Apriori cuyo rendimiento es el peor. En general, en soportes altos (mayores o iguales a 1) los tiempos empleados por los tres algoritmos son similares. La diferencia entre ellos se empieza a notar a medida que se disminuyen los soportes y los valores se hacen pequeños (≤ 0.5).

5.2 Evaluación de rendimiento de los Algoritmos de Clasificación

Para evaluar el rendimiento de los algoritmos de Clasificación C4.5 y Mate-tree se utilizaron los archivos *log* de los repositorios de la NASA. En la tabla 3 se muestran los archivos *log* utilizados.

Las pruebas consistieron en medir el tiempo de estos algoritmos con diferentes atributos clase. Los resultados de las pruebas sobre el archivo *log1.log* se muestran en la tabla 4 y figura 4.

Tabla 3. Archivos log NASA

Conjunto de datos Clasificación		
Nombre del Registro	**Numero de Registros**	**Tamaño KB**
kk.log	10	2
kk2.log	12	1
Documentolog.log	33	13
log1.log	7011	3102
access_la_nave.log	11797	19387
acces_server.log	29771	40030

Tabla 4. Rendimiento algoritmos de Clasificación con el archivo log1.log

log1.log			
Discretización	**tiempo milisegundos**		
	Atributo Clase	**C4.5**	**Mate**
Default	IP type	406	4750
	Schedule	422	4375
	Week day	0.10	3973
	Duration	219	4969
	Request Amount	437	4500
	Download size	594	4500

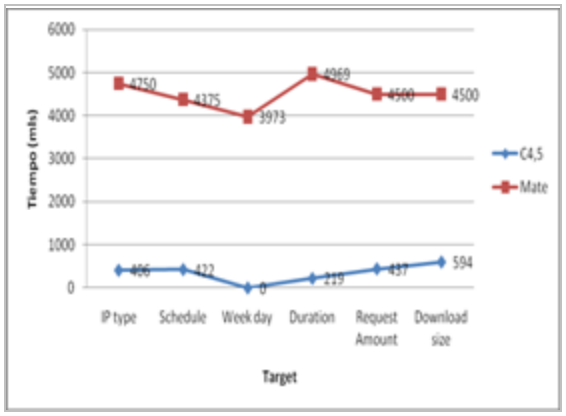

Fig . 4. Comportamiento algoritmos Clasificación con el archivo log1.log

Analizando los resultados de las pruebas se puede observar que la elección del atributo clase es determinante para que aumenten o disminuyan los tiempos de ejecución en ambos algoritmos dependiendo de la relevancia de dicho atributo dentro del conjunto. El rendimiento del algoritmo Mate-tree se ve considerablemente afectado por el número de registros que se estén analizando. El tiempo empleado por el mismo es directamente proporcional al número de registros analizados. En términos generales el algoritmo C4.5 tiene un mejor comportamiento que el algoritmo Mate-tree.

6 CONCLUSIONES Y TRABAJOS FUTUROS

Se cuenta con la primera versión de Polaris, una herramienta para minería de uso de la web, desarrollada en el laboratorio KDD del departamento de Ingeniería de Sistemas de la Universidad de Nariño (Colombia), bajo software libre con licencia GPL v2.0. Polaris soporta todas las fases del proceso de minería de uso de la web (procesamiento de datos, descubrimiento de patrones y análisis de patrones). En la fase de descubrimiento de patrones soporta las tareas de Asociación, con los algoritmos Apriori, FPGrowth y EquipAsso; Clasificación con los algoritmos C4.5 y Mate-tree y aprendizaje de patrones de navegación con el algoritmo HPG. Las diferentes pruebas de funcionalidad realizadas con Polaris, demuestran que es una herramienta fiable para ser utilizada en proyectos reales de minería de uso de la web.

Como trabajos futuros están el de implementar en Polaris otras tareas y algoritmos de minería de uso de la web, como clustering y patrones secuenciales. Además, complementar esta herramienta para que Polaris sea capaz de soportar minería de la estructura de la web y minería del contenido de la web.

REFERENCIAS

Agrawal R. y Srikant R. (1994). Fast Algorithms for Mining Association Rules, VLDB Conference, Santiago, Chile.

Bamshad M., Cooley R. y Srivastava J. (1997). Web Mining: Information and Pattern Discovery on the World Wide Web, Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence ICTAI, Newport Beach, California, USA.

Filocamo G. y Chesñevar C.(2003). Formalización de Web Mining como Conocimiento Estructurado, Anales del V Workshop de Investigadores en Ciencias de la Computación WICC 2003, Universidad Nacional del Centro de la Provincia de Buenos Aires, Argentina.

Han J., Pei J. y Yin Y. (2000). Mining Frequent Patterns without candidate Generation, Proceedings ACM SIGMOD, Dallas, TX, USA.

Hernandez O.J., Ramirez Q.M. y Ferri R.C. (2004). Introducción a la Minería de Datos, Editorial Pearson Prentice Hall, Madrid, España.

Quinlan J.R. (1993). C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers.

Timarán P.R. y Millán M. (2005a). EquipAsso: un Algoritmo para el Descubrimiento de Reglas de Asociación basado en Operadores Algebraicos, Libro de Memorias de 4ª Conferencia Iberoamericana en Sistemas, Cibernética e Informática CISCI 2005, pp. 343—348. Orlando, Florida, USA.

Timarán P.R. y Millán M. (2005b). EquipAsso: an Algorithm based on New Relational Algebraic Operators for Association Rules Discovery, Proceedings of the Fourth IASTED International Conference on Computational Intelligence, ACTA Press, Calgary, Alberta, Canada.

Timarán P.R. (2007). Mate-tree: un Algoritmo para el Descubrimiento de Reglas de Clasificación basado en Operadores Algebraicos Relacionales, Libro de Memorias de 6ª Conferencia Iberoamericana en Sistemas, Cibernética e Informática CISCI 2007, pp. 196—201. Orlando, Florida, USA.

Tingshao Z. (2001). Web Usage Mining for Internet Recommendation, Proposal for Ph.D. Candidacy Examination. University of Alberta, Canada.

Villena J., González J.C., Barceló E. y Velasco, J.R. (2002). Minería de Uso de la Web mediante Huellas y Sesiones, Memorias de la VIII Conferencia Iberoamericana de Inteligencia Artificial IBERAMIA , Sevilla, España.