UNIVERSIDAD DE
GRANADA

UNIVERSIDAD TECNOLOGICA
METROPOLITANA
Santiago de Chile

Aplicaciones del Soft Computing al análisis de ficheros log de sitios Web

Autor: Ricardo Valenzuela Gaete

Director de Tesis: Dr. Juan Luis Castro Peña

2006

Editor: Editorial de la Universidad de Granada
Autor: Ricardo Alberto Valenzuela Gaete
D.L.: Gr. 2310 - 2006
ISBN: 84-338-4166-1

Tras cursar el programa de doctorado “Computación y Sistemas Inteligentes: aplicaciones a Internet y al Comercio Electrónico” durante los cursos 2001-2003, y haber presentado el proyecto de tesis **Aplicaciones del Soft Computing al análisis de ficheros log de sitios Web**

,

Ricardo Valenzuela Gaete, presenta esta memoria para optar al grado de doctor por la Universidad de Granada, con la conformidad del Director de la mismo Dr. D. Juan Luis Castro doctor.

En Granada a 25 de Octubre de 2006

Ricardo Valenzuela Gaete	Juan Luis Castro Peña
Autor de la Tesis	Director de la Tesis

INDICE

INTRODUCCION

Las técnicas de Descubrimiento de Conocimiento o KDD, como también las del Aprendizaje Automático no pueden ser directamente aplicadas al análisis de los datos disponibles en Internet dado su diversidad y variedad. Las bases de datos tradicionales manejan patrones de datos estructurados, en cambio la información contenida en Internet es evidentemente heterogénea, no tiene etiquetas o marcas de búsqueda, esta dividida o distribuida, es variada o de diverso tipo, tiene un bajo grado de estructuración, es cambiante en el tiempo y es de alta dimensionalidad.

Esta información almacenada en distintas computadoras puede ser conceptualizada o modelada entendiéndola como una gran base de datos no estructurados distribuida sobre una red global de computadoras ó *b_ internet*, la cual contiene distintos tipos de *objetos primitivos* de información tales como: patrones, símbolos, conceptos, imágenes, audio, video y *objetos contenedores* que los almacenan y permiten su visualización tales como páginas web, sitios web o bodegas de objetos que sirvan de contenedores para su posterior despliegue y visualización.

Toda esta información queda disponible para un usuario, en distintos servidores que son dispuestos sobre la red global o web; servidores sobre los cuales se implementan soluciones web con sus respectivos métodos de acceso a los servicios implementados en estas con el objetivo fundamental de que los objetos que forman parte de estas soluciones, puedan ser visualizados, rescatados, entendidos o percibidos por los usuarios y clientes. Esta visualización o *percepción de los contenidos* se realiza en la *ventana virtual* que el usuario dispone para el despliegue de los objetos como por ejemplo un objeto página web al cual se accede por algún programa de búsqueda y un medio físico o sistema de comunicación, es desplegado y visualizado en esta ventana Este conjunto de elementos compuesto por: la microelectrónica, los sistemas de comunicación, sistemas de software, bases de datos y otros medios, forman en conjunto con las personas naturales y empresas un gran mercado o plaza de intercambio de información de datos, adquisición de bienes y servicios, intercambio de dinero y sofisticadas formas de comunicación para la adquisición de conocimiento. A medida que las tecnologías de información avanzan hacia nuevas etapas en su desarrollo, se presenta una paradoja en cuanto al explosivo aumento de la información almacenada en INTERNET, dado que en el orden en que esta aumenta; de manera proporcional se pierde la capacidad de manejar o interpretar sus

contenidos, siendo muy difícil su manejo y recuperación con el objetivo de obtener conocimiento subyacente a los datos.

I.1. El problema del análisis de los web logs

Si nos detenemos en primer lugar en una valorización descriptiva del conocimiento que el ser humano puede extraer de b_internet, podemos mencionar que una de sus principales características es que, en muchos casos, es un conocimiento de alto nivel que el ser humano expresa con el lenguaje natural, utilizando términos, conceptos y relaciones imprecisos. Un ejemplo de esto es el conocimiento que se puede extraer de los ***ficheros logs***. Estos ficheros, almacenan los eventos que se producen en los servidores web como consecuencia de la navegación de los usuarios sobre el sitio web que este implementa; por ejemplo una petición de un servicio o de un objeto almacenado en el sitio web genera el algunos casos una infinidad de eventos que se almacenan como registros en este tipo de ficheros, estos registros contienen información como la ip del usuario, la fecha y la hora de la petición, los objetos solicitados y el resultado de la operación solicitada,. información que es almacenada de acuerdo a un formato conocido. A modo de ejemplo la información almacenada en los ficheros logs de un sitio web orientado al e-commerce, acumula datos básicos como los objetos solicitados, los clientes o ip y el instante de tiempo de una petición, información desde la cual se pueden determinar características derivadas como la duración de la visita, los caminos recorridos sobre la estructura del sitio, los productos con mayor venta o las relaciones entre productos y clientes. Como puede observarse, muchas de las cuestiones que resultan de interés conocer a través del análisis de ficheros logs son de naturaleza precisa, pero muchos otros son de naturaleza imprecisa, ya sea porque la propia cuestión es imprecisa, o porque la forma natural de describir la respuesta lo es. Por ejemplo, la cuestión de cuantos usuarios ven con detalle la página del producto X, es una cuestión de naturaleza imprecisa ("ver con detalle" es una expresión que no esta definida con precisión), y donde la respuesta natural debería ser expresada al mismo nivel de imprecisión, algo así como "muy pocos, la mayoría no se fijan en los detalles".

El razonamiento expuesto anteriormente, nos lleva a plantear una hipótesis de trabajo que se base en estas características, es decir un proceso de obtención de conocimiento que considere a los *ficheros logs* como fuente de datos o dominio de conocimientos, debe necesariamente considerar el hecho que la información a analizar es substancialmente no estructurada y que debe obtener conocimiento que pueden ser descrito como información vaga o con

incertidumbre e imprecisión. Otro ejemplo adicional de este hecho puede ser descrito tomando como ejemplo un evento asociado a la petición de un objeto página web en un instante de tiempo T_0; este evento no implica necesariamente que la *pagina fue vista* por el usuario o ip registrado en ese evento o registro, lo que agrega un grado de complejidad adicional a cualquier proceso e análisis de los ficheros logs.

Las herramientas de la Computación Flexible o Soft Computing resultan apropiadas para sustentar el desarrollo de nuevas metodologías de análisis de los datos contenidos en los ficheros logs en el sentido que indicamos, por ejemplo la Lógica Difusa permite manejar conceptos vagos y ser asociados a variables lingüísticas, permitiendo por ejemplo establecer si una visita a un sitio web fue corta mediana o alta. Estos análisis para una empresa que apoya su funcionamiento en un sitio web corporativo generalmente están dirigidos a determinar el comportamiento de sus clientes, páginas visitadas, productos adquiridos, comportamientos anómalos de usuarios, personalización o adaptación del sitio web a las necesidades de los usuarios; o a particularidades propias del negocio o del conocimiento que se pretenda obtener el cual puede ser utilizado en la toma de decisiones.

Los procesos asociados con la extracción de conocimiento a partir de la información almacenada en los ficheros logs pueden ser establecidos por medio de la utilización y adaptación de las metodologías y técnicas desarrolladas por la Minería de Datos y aplicarlas a los web logs o Web Usage Mining; no obstante lo anterior debemos destacar que dado el bajo nivel de estructuración de los datos contenidos en este tipo de ficheros y a características como redundancias, referencias de tiempo anómalas u otras causas propias es necesario adaptar las técnicas y métodos del Data Mining para ser aplicadas a los ficheros logs. Este hecho a convertido a este tipo de estudio en una área fértil para la investigación y desarrollo de nuevas tecnologías o métodos que permitan mejores resultados en cuanto a la interpretación del conocimiento extraído. Si ampliamos este análisis al total de la información almacenada en Internet, asimilada en nuestro caso a una base conceptual de datos no estructurados o b_internet , necesariamente la creación de nuevas herramientas y técnicas es casi una obligación.

Los ficheros logs son creados de manera automática por procesos especializados que se ejecutan directamente en la plataforma de trabajo de la solución web o fuente primaria y son el resultado de las peticiones de objetos realizadas por sujetos reales o virtuales a esta, luego por tanto es posible admitir en otro nivel de abstracción que las fuentes primarias web o soluciones

web, están creadas por una serie de objetos contenedores como páginas web, bases de datos de objetos, bodegas de objetos y por objetos primitivos como ficheros de texto (pdf, doc, ..), objetos no-textuales como audio y video, objetos gráficos o de imágenes y otros no clasificados; objetos en los cuales se almacena información que puede ser transformada en conocimiento por medio de procesos particularizados a dar respuesta a problemas específicos como por ejemplo determinar si un objeto página web fue visto o percibido por el sujeto visitante. Basándonos en esta hipótesis de trabajo se admite que el Web está constituido por una gran variedad de objetos almacenados en distintas computadoras distribuidas sobre la red global, siendo la principal características de estos objetos el estar diseñados u orientados a la visualización o percepción de los mismos por parte de los usuarios.

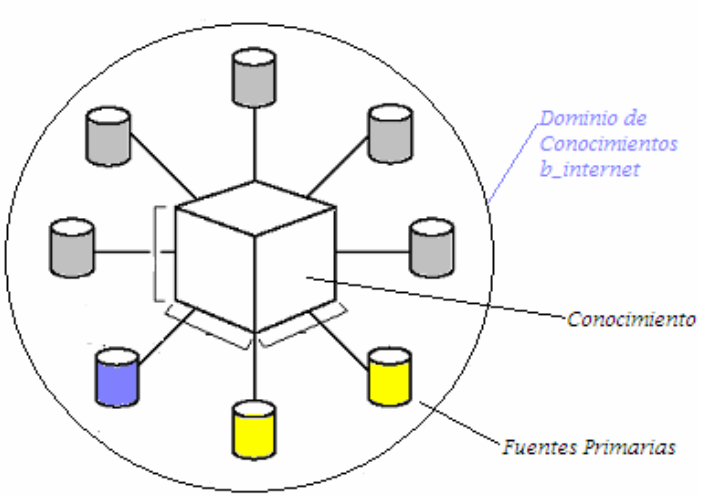


Figura I.1 Fuentes Primarias de Datos y Objetos

Un objeto contenedor página web por ejemplo, se construye orientando sus contenidos a la visualización de los objetos primitivos que este almacena, estos objetos primitivos a su vez contienen la información que está destinada a los usuarios, información que es desplegada y observada por medio de la ventana virtual implementada sobre la computadora del cliente o sobre algún medio físico en donde es factible visualizar los objetos solicitados a la fuente primaria. Hessen J. en su "Teoría sobre el Conocimiento" [68], establece que: el conocimiento "es la imagen percibida por el sujeto conforme al objeto observado, dado que le es imposible conocer la realidad completa del objeto. De acuerdo con este planteamiento asumiremos que la suma total de objetos contenidos en el web, objetos que están orientados principalmente a la visualización o percepción de los usuarios que los solicitan a las distintas fuentes primarias

disponibles en la red global, construyen un Dominio de Conocimiento Virtual o Dominio Universal de Conocimiento. Este Dominio Universal de Conocimientos o Internet construido bajo la forma de objetos primitivos, objetos contendores y agrupaciones de ambos almacenados en diversas fuentes primarias almacena conocimiento de distinto tipo el cual puede ser estudiado desde variados puntos de vista seleccionando los datos apropiados al problema a resolver.

Un dominio de información de datos crudos como los ficheros logs o dominio de conocimientos genérico, en si solo representa datos agrupados desde distintas fuentes o servidores como por ejemplo logs de servidores web, de servidores proxy o de servidores de contenido, estos deben ser depurados, clasificados o interpretados con la finalidad de "entender" la información que contienen, es decir deben ser transformados en información, la cual a su vez tendrá un fundamento o propósito de acuerdo al problema que se pretende estudiar o resolver. Esta información procesada y aplicada en el marco de referencia de un usuario real o artificial junto a la percepción que este adquiere de la visualización de los mismos se convierte en conocimiento el cual puede ser aplicado con objetivos precisos para establecer una verdad o una verdad racional, la cual cumple con el objetivo al ser almacenada e interpretada como inteligencia o sabiduría en el caso de una persona o inteligencia racional (o artificial) en el caso de un sistema artificial.

De acuerdo a los planteamientos y, desde el punto de vista de la ontología del conocimiento almacenado en el web, podemos mencionar que este es creado por los individuos u organizaciones que implementan soluciones web y por los usuarios reales y virtuales que "navegan" buscando recursos siguiendo sus preferencias, intereses, modas o patrones de comportamiento; o en su defecto sin un objetivo determinado. Esta navegación o búsqueda de recursos queda representada y registrada en cada servidor bajo la forma de un evento que se almacena en un fichero logs o web logs de forma automática, tarea realizada por programas de software especialmente diseñados para este efecto. Es destacable mencionar que junto a los usuarios reales, existen usuarios virtuales o programas especializados conocidos como robots, arañas (spiders), agentes, agentes inteligentes que gatillan eventos idénticos al solicitar conocer los recursos disponibles en las distintas fuentes primarias con el objetivo de construir listas o índices de los recursos disponibles en la red global. Ambos conjuntos de datos es decir los creados con la finalidad de implementar una solución web como los generados automáticamente por consecuencia de la navegación de los usuarios de los mismos constituyen

el que hemos denominado en esta tesis como Dominio de Conocimientos Universal o B_Internet. Se ha considerado esta definición genérica constituida por distintos tipos de datos agrupados en conjuntos o sub-dominios de conocimiento, con la finalidad de facilitar el estudio para la resolución de problemas específicos como por ejemplo la Personalización de un Sitio Web.

I.2 Web mining: El enfoque de Data Mining

Los Dominios de Conocimiento definidos anteriormente, agrupan datos con la finalidad de orientarlos al estudio, modelación o abstracciones con el objetivo de solucionar determinados problemas asociados al web, como por ejemplo el determinar o recuperar patrones que puedan ser empleados en la Personalización de Sitios Web, en la Detección de Intrusos o de Comportamientos Anormales, en Estudios de Usos o Usabilidad, en Análisis de Redes y su Tráfico, en Sistemas de Soporte a la Decisión, en la mejora del Diseño de Estructuras de Sitios Web, es evidente por tanto que el estudio de cualquier tipo de navegación de los usuarios reales o virtuales representa un significativo interés para todos aquellos que implementan soluciones web, sean estos empresas, gobiernos o personas naturales. El conjunto de datos aplicables a estas abstracciones o modelos de solución, proviene de la información que almacenan los ficheros web logs y puede ser entendido como un *dominio de conocimiento genérico* , es indudable por tanto que la eficiente recuperación de patrones a partir de este dominio de datos y la obtención de conocimiento a partir del mismo, se ha convertido en un tema de vital importancia para todos aquellos que emplean el Web. Este problema puede ser abordado desde varios puntos de vista siendo algunos de estos:

Detección de patrones: lingüísticos, patrones crudos, simbólicos...otros

Desarrollo de técnicas basadas en inteligencia artificial para la interpretación del código del significado de los datos o patrones disponibles.

Recuperación de información de datos no estructurados o no textuales

Mejoras en los programas de búsqueda, dado que estos solo manejan consultas crisp

Implementación de ventanas virtuales de acuerdo al perfil de comportamiento de un usuario o personalización web.

Clasificación de información por relevancia y construcción de bases de conocimiento.

Obtención de Conocimiento y Descubrimiento de Conocimiento.

Realidad Virtual (comercio, ocio, gobierno, relaciones humanas)

Consideremos a modo de ejemplo el problema de la Personalización Automática de un Sitio Web; Según Mobasher B. [125] - "La Personalización de un Sitio Web, puede ser descrita como las acciones basadas en la experiencia de navegación de los usuarios tendientes a adaptar la solución web a sus preferencias o gustos". Los elementos principales asociados con la personalización de un Sitio Web corresponden a la modelación de objetos requeridos y sujetos que los solicitan con la finalidad de determinar por ejemplo el uso de un objeto en particular y su relación con o los sujetos que lo solicitan. El dominio de datos para este estudio se encuentra en los ficheros logs o en aquellos ficheros usados para el registro de eventos o peticiones de recursos.

Son ampliamente aceptados los planteamientos realizados por Cooley R. [39] relacionados con el estudio del web desde el punto de vista del "uso" de los recursos disponibles en los diversos sitios web de la red global. Cooley plantea la necesidad de aplicar las técnicas del Data Mining a los repositorios de datos y ficheros web logs en un proceso que denomina Web Usage Mining, en donde considera que los datos pueden ser clasificados en un proceso de Web Usage Mining en:

Content (Contenido), los datos reales contenidos en una página web.

Structure (Estructura), los datos que describen la organización de los contenidos de una página. Estructuras HTML, XML.

Usage (Uso), datos que describen los patrones de uso de las páginas web, como direcciones ip, fecha y hora de acceso u otras relacionadas con el formato de los ficheros web logs..

User Profile (Perfiles de Usuarios), datos que suministran información demográfica de los usuarios del sitio web

Mobasher M. et. al [126] plantea que estos dominios de datos o información deben ser tratados, ordenados y empleados para construir modelos u abstracciones que permitan determinar: *users*, *page view, click-streams, server session and episodes,* para luego aplicarlos a procesos de Data Mining con el objetivo establecer soluciones o modelos aplicados al web. Cooley

define que una página vista o page view corresponde a todos los archivos que son presentados en el lado del cliente como consecuencia de un simple click del ratón; una tormenta de clicks o click-streams es una secuencia de page view que son accesadas por un usuario; una sesión de servidor es el click-streams para una visita única del usuario a un Sitio Web y finalmente un episodio es un sub-conjunto de page view de una sesión.

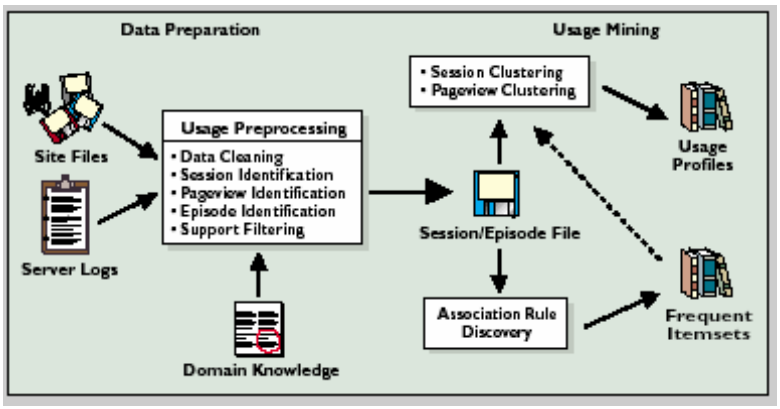


Figura I.2 Arquitectura de un Proceso Web Usage Mining.

Arotatei D. y Mitra S. [7] indican que el Web Mining se refiere al uso de las técnicas del Data Mining que automáticamente recuperan, extraen y evalúan (análisis / generalización) información para el descubrimiento de conocimiento desde documentos y servicios almacenados en el web, planteando una taxónomia de procesos que se indica en la Figura I..3

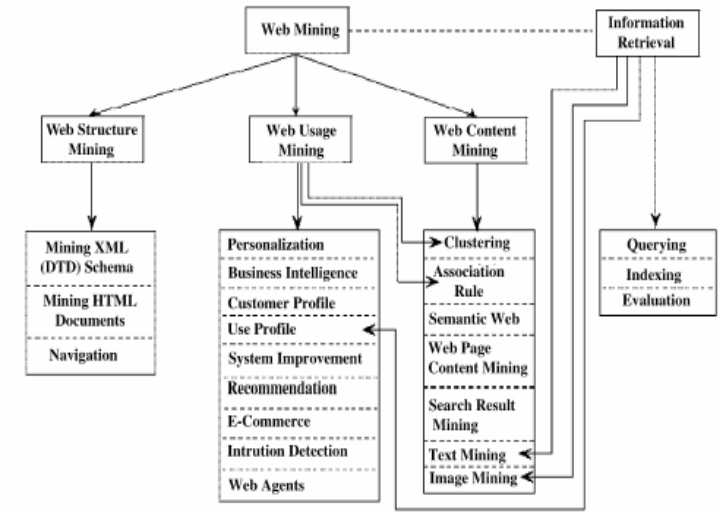


Figura I.3 Taxónomia del Web Mining

De acuerdo a los planteamientos [7,27,39,125,126], la personalización de un sitio web es aplicable a cualquier actividad de búsqueda de recursos en el web, y puede ser definida como cualquier acción hecha a la medida basada en la experiencia de un usuario o conjunto de estos, las acciones pueden abarcar desde realizar su visita al sitio web mas agradable para un usuario, hasta anticipar sus necesidades suministrándole información personalizada. Los procesos asociados con el Data Mining aplicado al web consideran tareas de alto nivel para realizar la limpieza de los datos con este objetivo, la identificación de los usuarios, identificación de sesiones, identificación de "page view" y "path completion" se encuentran ente las tareas previas a ejecutar antes de emprender un proceso de Minería de Datos

El Data Mining aplicado al web o Web Mining se caracteriza por la necesidad de disponer de datos confiables para obtener resultados confiables, desafortunadamente los datos almacenados en los web logs raramente mantienen una calidad que permitan aplicar sus técnicas asociadas de forma directa, es conveniente destacar que una fuente de información proviene de los ficheros logs y del formato con que estos registran los eventos http generados por peticiones de recursos al servidor web. Una de las primeras dificultades que presentan los "logs", es determinar por ejemplo la identificación de los usuarios que solicitan recursos, es común que muchos usuarios accedan a un sitio web desde un mismo punto que es el caso de un servidor proxy. Por otra parte el empleo del concepto *pageview* para determinar que página es relevante o ha sido visitada presenta la dificultad adicional que es dependiente de la estructura del sitio, en donde cada archivo HTML tiene una correlación determinada con el pageview, lo cual puede incrementar ficticiamente esta medida. Es evidente que es necesario implementar nuevas técnicas para aplicar un Data Mining a los datos disponibles en el web o realizar modificaciones a las herramientas y algoritmos empleados en el Data Mining con el objetivo de que puedan ser adaptados al tipo de datos a analizar.

Si observamos al web desde el punto de vista de la recuperación de la información (IR) contenida en los distintos objetos, observaremos en primer lugar que el término Información Retrieval fue definido por Salton G. [152] como: *"la recuperación de información es un campo que se preocupa de la estructura, análisis, organización, almacenamiento, búsqueda y recuperación de información";* ha sido aceptado que el concepto Recuperación de Información se refiere a los procesos que permiten la Recuperación de Documentos y la Recuperación de Texto. Los paradigmas de IR se refieren a establecer modelos conceptuales que permitan la eficiente recuperación de

información relevante. Estos modelos según Kraaij W. [93] pueden ser agrupados en las siguientes clases: Modelos Lógicos, Modelos de Espacios Vectoriales, Modelos Probabilísticos. Arotatei D. y Mitra S. [7] plantean en su Taxónomia del Web Mining que la IR es un proceso asociado al Web Mining, en donde la recuperación de texto (Text Mining) o recuperación de imágenes (Image Mining) se incluyen como parte del Web Content Mining, ambas técnicas a menudo son consideradas en la comunidad IR como *multimedia retrieval* y se basan o refieren a la extracción automática de partes textuales o habladas de los documentos contenidos en Internet. La mayoría de los modelos de IR son de naturaleza estadística, estos modelos asumirán explícitamente o implícitamente que dentro de un texto existen ciertas distribuciones de datos textuales que tienen propiedades estadísticas o marcas por medio de las cuales es posible sacar inferencias estadísticas, empleando distribuciones como la Normal o Gaussiana, Binominal, Modelos de Markov, Distribución Multinomial, Distribución de Poisson o técnicas probabilísticas. Nos referimos a estas técnicas para graficar el problema del análisis de los datos contenidos en Internet, por ejemplo en el caso de la recuperación de información *multimedial* los datos son *no-textuales* lo cual complica la adaptación de técnicas como el Data Mining a este nuevo dominio de datos. y en especial cuando los datos son aquellos relacionados con el uso de los recursos almacenados en un fichero web logs; mencionamos con anterioridad que la principal característica de la información contenida en Internet es que evidentemente es heterogénea, no tiene etiquetas o marcas de búsqueda, esta dividida o distribuida, es variada o de diverso tipo, tiene un bajo grado de estructuración, es cambiante en el tiempo y es de alta dimensionalidad.

Definir o restringir el dominio de conocimiento es critico para abordar cualquier estudio relacionado con el uso de los recursos almacenados en las distintas fuentes primarias distribuidas en el web, y dado su importancia en el capítulo 2 de esta tesis, se entregará una descripción más detallada del origen de los datos contenidos en *b_internet*, sus principales características, formatos y otras particularidades. Esta tesis esta restringida al dominio de conocimiento genérico constituido por el conjunto de datos contenidos en los denominados ficheros logs o archivos web logs, por lo cual abordaremos este tema de manera detallada con el objetivo de precisar el contexto de trabajo de cualquier investigación que se base en este tipo de archivo.

I.3 El paradigma del Soft Computing

Hemos mencionado anteriormente que el descubrimiento de conocimiento desde los ficheros web logs hay componentes intrínsicamente difusos, por tanto las técnicas de Descubrimiento de Conocimiento o KDD, como también las del Aprendizaje Automático no pueden ser directamente aplicadas al análisis de los datos almacenados en este tipo de ficheros, siendo necesario su modificación o ser adaptadas como por ejemplo añadiéndoles técnicas y conceptos del Soft Computing. Consideramos como hipótesis que las herramientas de la Computación Flexible en donde la Lógica Difusa juega un rol fundamental son aplicables al estudio de este dominio universal de conocimiento con el objetivo de resolver tres problemas fundamentales o genéricos asociados al dominio de conocimiento genérico incrustado en los web logs, estos problemas han sido definidos en esta tesis como:

> *El problema de la Obtención de Conocimiento Racional[8] a partir de la información almacenada en el web.*
>
> *El problema de la búsqueda de información por medio de un Lenguaje Natural o búsquedas con vaguedad.*
>
> *El problema de la Interpetración de los Contenidos de los Datos no-textuales contenidos en las diversas fuente primarias distribuidas en el web*

En el mundo del conocimiento basado en el web el "concepto relevancia" juega un rol fundamental, concepto que nuevamente tiene asociado incertidumbre, tanto en la búsqueda del recurso u objeto del interés del usuario, como en la percepción basada en la visualización de este. Tratándose de un objeto contenedor página web y de la percepción del mismo y de sus contenidos, el problema se transforma en un problema complejo dado que la percepción del contenido representada en nuestra propuesta con la pregunta *¿la página web fue vista o visitada?* considera nuevamente la relevancia de los contenidos de la misma.

El rol de la Lógica Difusa en el análisis de los datos disponibles en el web es fundamental para establecer el paradigma de Inteligencia Aplicada al Web, este razonamiento se basa en que el conocimiento basado en la percepción de la información almacenada en el web entendiendo por percepción a la visualización de los objetos que contienen información relevante o con relevancia para el usuario que la solicita, relevancia que finalmente se resuelve en el cerebro del usuario es esencialmente imprecisa. Nuestra pregunta base planteada con anterioridad: ¿el

objeto página web fue visitado por el usuario? ó ¿los contenidos de la página web fueron percibidos por el usuario?, tiene el objetivo preciso de graficar el problema genérico que hemos denominado : *El Problema de la Obtención de Conocimiento Racional a partir de la información almacenada en el Web.*

De acuerdo a estos argumentos podemos deducir que los problemas genéricos planteados sobre el dominio de conocimientos universal o Internet sugieren que las respuestas en escala humana o de conocimiento deberán contener elementos imprecisos, como por ejemplo una respuesta a la pregunta base puede ser:

"el sitio web fue visitado por alrededor de diez minutos por un total de 20 usuarios los cuales solicitaron de preferencia la página web que contenía un objeto descrito visualmente".

Una respuesta de esta naturaleza utilizando técnicas que empleen lógicas crisp, técnicas probabilísticas o herramientas estadísticas convencionales es compleja de obtener; las técnicas estadísticas o demográficas entregan respuestas a la pregunta base, en términos de cantidad, como por ejemplo: los hits sobre la página estudiada corresponden a 456 o la página más solicitada es pw_favorita. Es evidente que en una escala de conocimiento basado en la información, los términos lingüísticos formalizados por intermedio de la Lógica Difusa, facilitan las respuestas a preguntas que contienen incertidumbre o basadas en los datos contenidos en el dominio de conocimientos universal o los sub dominios definidos con anterioridad.

Zadeh L.. A. [192,193,195] plantea un concepto que provee una base para la solución del problema de la obtención de conocimiento del web y la construcción de un web inteligente es el denominado *Precisiated Natural Language (PNL),* el cual se basa en la lógica difusa.

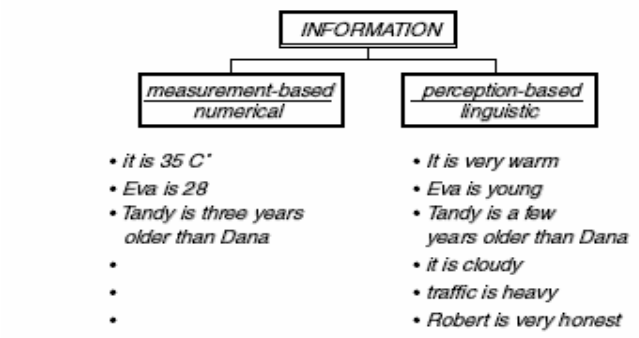


Figura I.4 La información basada en Medidas y basada en Términos Lingüísticos

Zadeh L.. A. [194] indica en el trabajo referido que el "Precisiated Natural Language (PNL)", permite describir las percepciones de manera similar a un lenguaje natural. Se asume que una preposición ***p*** en un lenguaje natural ***NL***, puede ser representada como una restricción generalizada de la forma : ***X isr R***; en donde ***X*** es la variable restringida, ***R*** es una relación que restringe (no crisp) y ***r*** es una variable de indexación

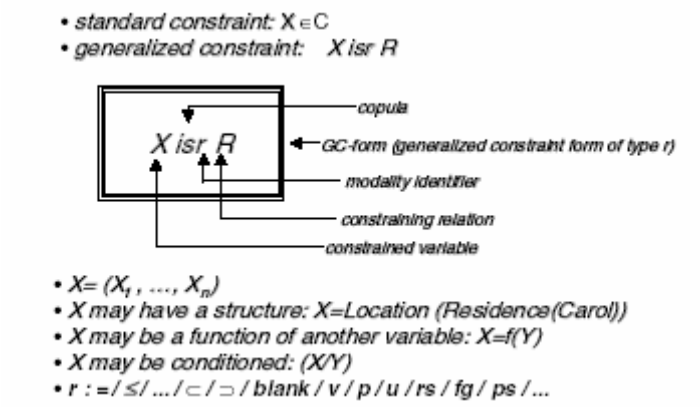


Figura I.5 Restricciones de una Variable

Un elemento importante a considerar en la respuesta a nuestra pregunta base es la relevancia, dado que una respuesta que considere términos lingüísticos como breve, prolongada o corta, respuesta que podemos ejemplificar como: la visita fue muy corta luego el contenido no fue percibido, encierra el concepto de relevancia referido al contenido del objeto página web.

La relevancia junto a la redundancia construyen una realidad o contexto cotidiano a los usuarios que buscan información almacenada en el web, y actúan en conjunto sobre los datos almacenados en los ficheros logs, dado que una página será desplegada solo una vez que el usuario escoja de una lista redundante aquella URL que contiene la información requerida o relevante para el mismo, provocando con su decisión una visita al sitio web con el respectivo evento asociado, evento que es registrado y almacenado en un fichero logs. El denominado PageView o Hits de Paginas, empleado por herramientas como los analizadores de logs u otras herramientas dedicadas al análisis de estos ficheros no entregan una respuesta apropiada a nuestra hipótesis de trabajo: ¿ el objeto página web fue visitado por el usuario? o ¿ los contenidos de la página web fueron percibidos por el usuario?.

Este problema relacionado con la visita efectiva a un sitio web por parte de un usuario es una de la motivaciones de esta tesis, problema que será abordado en el desarrollo de la misma, desde

los puntos de vista de las soluciones de data mining y sus dificultades, desde el punto de vista de los métodos empleados por trabajos relacionados, desde el estado del arte de las técnicas web mining y de las principales dificultades que plantea el análisis de los datos contenidos en los ficheros logs, manteniendo el contexto de esta tesis que corresponde al planteamiento de una nuevo algoritmo para el análisis delos datos contenidos en los ficheros logs, algoritmo que hemos denominado Minero P*

I.4 Objetivo

Considerando el paradigma de un *Web Inteligente* planteado por Zadeh L. [192], el cual basa sus modelos en las herramientas de Computación Flexible, nuestra pregunta base o hipótesis de trabajo asociada a la obtención de conocimiento de un dominio de datos restringido, dominio que se refiere en la presente tesis al análisis de los datos almacenados en los ficheros logs para determinar conocimiento que permita responder una serie de preguntas bases o genéricas representadas por:

> ¿ el objeto página web fue visitado por el usuario?
>
> *¿ los contenidos de la página web fueron percibidos por el usuario? o*
>
> *¿ la página fue vista?*

La(s) pregunta(s) genérica(s) anterior(es) refleja(n) un problema real que ha motivado la atención de muchos trabajos científicos, el cual es determinar el comportamiento de los usuarios de un sitio web por medio del análisis de los ficheros logs, o en palabras mas simples en la obtención de conocimiento a partir de la información almacenada en los ficheros logs. Esta tesis trata se enmarca en esa dificultad, la cual ha sido abordada desde el punto de vista de la metodología y de las herramientas a utilizar las cuales se basan en la Computación Flexible o Soft Computing.

Descripción del problema

Este tiene su inicio al momento que un usuario solicita a un buscador un determinado símbolo o concepto, recibiendo como respuesta una lista o índice de los conocidos o rescatados del web por parte del buscador, es claro que los mejores motores de búsqueda no tienen capacidad

deductiva o la capacidad de responder a consultas vagas con respuestas relevantes obtenidas de la base de conocimientos universal o Internet. Nuestra simple pregunta: ¿ el objeto página web fue visto por el usuario?, agrega una nueva dimensión de complejidad, en donde la "relevancia en la búsqueda de los documentos", objetos, símbolos u otros conceptos no es resuelta por el motor de búsqueda, sino que finalmente será el usuario el que dirigirá su atención a la lista de objetos presentada y seleccionara una URL de su interés. Esta lista generalmente voluminosa y ambigua contiene el concepto o símbolo solicitado presentándose la paradoja: mayor volumen-menor relevancia, lo cual obliga al usuario a emprender su propia búsqueda por relevancia empleando un método de ensayo-error sobre las distintas fuentes primarias identificadas como las URL que contienen la información solicitada de acuerdo a sus necesidades. Este proceso ensayo-error deja un rastro en los servidores visitados, rastro que es registrado bajo la forma de un evento sobre un fichero web logs. Una vez seleccionada la URL que contiene el concepto o símbolo requerido, , nuestro "hombre imaginario" usuario web real o virtual, realizará una petición vía http al servidor que contiene el objeto cuya relevancia fue determinada de acuerdo al criterio del usuario petición que finalmente será gestionada retornando hacia su ventana virtual el objeto requerido para su visualización y posterior percepción. Esta visualización o percepción del contenido se llevara a cabo solo si el documento es relevante para el usuario con respecto a ciertas necesidades de información relacionadas con el tópico de la consulta, más precisamente el contenido de la página web será visualizada solo si el contenido de esta responde a las necesidades del usuario. Desde el lado del servidor o fuente primaria de objetos web, se genera y almacena un evento que refleja las peticiones del usuario real o virtual, estos eventos almacenados en un fichero logs contienen los datos del objeto requerido, la hora y la fecha de la solicitud, el éxito o fracaso de la petición y constituyen el dominio genérico de conocimiento que es la base para el descubrimiento de patrones con el objetivo de dar respuesta a las preguntas formuladas con anterioridad. El mayor problema que se presenta al intentar establecer si un objeto que forma parte de una solución web fue visualizado, visitado o percibido por ejemplo, corresponde al contexto de la pregunta y su respuesta inmediata, este contexto esta definido en la escala humana del conocimiento con preguntas y respuestas como:

La visita fue corta y el usuario abandono el sitio web, por tanto el objeto no fue visto.

El usuario solicito el objeto y mantuvo una sesión efectiva por 30 minutos.

¿La página más solicitada efectivamente es la de mayor interés para nuestros clientes?.

El comportamiento de navegación de nuestros clientes demuestra que adquieren productos solo de la página de ofertas.

Los objetos que tienen una descripción por medio de una imagen son los primeros en ser requeridos.

¿Cuál es el horario de mayor visita?.¿ Que zona de la ventana virtual es la más apropiada para disponer un objeto?.¿Cuál es el objeto mas solicitado?

La mayoría de estas preguntas y sus respuestas tienen una característica común, son imprecisas luego el tratamiento o modelación basados en técnicas, estadísticas, posibilísticas o lógicas bivaluadas o crisp es inapropiado, dado que algunas respuestas son parcialmente verdaderas y una gran cantidad de preguntas contienen incertidumbre. Deteniéndonos en el análisis de una pregunta que motiva el trabajo de esta tesis: ¿ el objeto página web fue visitado por el usuario?, sus posibles respuestas esta basadas en modelos, heurísticas o sistemas que se basan en la información contenida en los ficheros web logs, es decir en eventos generados de manera aleatoria o caótica en el lado del servidor por causa de peticiones realizadas por usuarios reales o virtuales. Es justificable considerar que las fronteras de las posibles respuestas están por tanto en el cerebro del sujeto que solicita el recurso, dado que es incierto un comportamiento lineal o exacto del mismo. La percepción del contenido de un objeto página web, el tiempo de visita por este, el seguimiento de sus click o clickstreams empleados por el usuario nos pueden dar una visión de si el objeto fue visto o percibido. Es claro que en la actualidad no existe un único paradigma para estudiar este caso, o el problema general relacionado con el "uso de los recursos u objetos que forman parte de una solución web".

I.5 Desarrollo de la memoria

Esta tesis considera algunas nuevas definiciones o abstracciones generales que serán utilizadas en el desarrollo de la misma, conceptos que serán resumidos en un glosario de *términos y conceptos*. En conjunto con los nuevos planteamientos se considera la utilización de términos y conceptos ampliamente aceptados por la comunidad y grupos de trabajo relacionados con el estudio y análisis de los datos contenidos en los ficheros web logs con el objetivo de determinar características derivas que puedan ser empleadas en la obtención de conocimiento

Junto a las definiciones de términos y conceptos anteriores, se consideran los siguientes restricciones generales o hipótesis:

1. El trabajo se enmarca en la Obtención de Características Derivadas para la Obtención de Conocimiento o el problema de la Obtención de Conocimiento Racional[112] a partir de la información almacenada en los web logs

2. .La información contenida en los ficheros web logs, no tiene restricciones y es almacenada de acuerdo a un formato conocido.

3. Es conocida la estructura de páginas, objetos y vínculos contenidos del o los sitios web empleado como referencia u origen de los datos.

Relacionado con la estructura de la presente tesis en el Capítulo 1 esta dedicado a un análisis exhaustivo de los de datos contenidos en los ficheros logs, orientándose este al análisis descriptivo del contenido de los registros de eventos almacenados en este tipo de ficheros, se realiza una descripción y comparación de los distintos formatos empleados por este tipo de ficheros como por ejemplo el CLF, XCLF, NCSA, W3C.. En este capítulo se plantea el problema real que tiene el análisis de los datos contenidos en este tipo de ficheros; se entregan ejemplos de como un evento idéntico genera distintas calidades en la información almacenada por este tipo de fichero, granularidad que es dependiente del formato escogido. Se indican las principales herramientas estadísticas o software analizadores de log sus ventajas y desventajas y los principales problemas que estas herramientas no pueden resolver o dar respuestas apropiadas. Este capitulo tiene el objetivo principal de dar una visión general del problema del análisis de la información contenida en este tipo de archivo y sus implicancias en los métodos de análisis y herramientas escogidas para este efecto.

El Capítulo 2 se enmarca en las principales herramientas de análisis de los datos almacenados en los de ficheros web logs, orientadas al descubrimiento de conocimiento, enfocándose en tres técnicas fundamentales el Data Mining, el Web Mining y el Soft Computing. Este capítulo se orienta a las descripción de los principales métodos y herramientas empleadas por el Data Mining, como por ejemplo redes neuronales, algoritmos de agrupamiento y otras que se analizan desde el punto de vista de un estado del arte, el cual se extiende al Web Mining en donde se realiza una comparación de ambas técnicas aplicadas sobre el universo de datos almacenado en los ficheros web logs. Se indican los algoritmos más significativos y las técnicas más significativas en un proceso de Web Mining y Data Mining, como son por ejemplo clustering, reglas de asociación o las redes neuronales sus ventajas y sus desventajas. Este capítulo adicionalmente entrega una

nueva visión conceptual relativa a las distintas fases involucradas en un proceso de Web Mining, visión asociada a términos lingüístico para un mejor entendimiento de los distintos procesos asociados a cada etapa.

El Capítulo 3 tiene por nombre *Análisis inteligente de sitios Web*, propuesta que se orienta a la construcción de un sistema formal para el análisis de sitos web. Este capitulo se concentra en las definiciones, hipótesis y enunciados necesarios que serán aplicados en un método de análisis de web logs. Se realiza una abstracción de los conceptos que intervienen en el problema: objeto página, sesión, visita, …; y se define el problema del análisis inteligente como el descubrimiento de "la realidad" en este sistema formal.

En el capítulo 4 se aborda el concepto de percepción de una página y se realiza el desarrollo de un nuevo algoritmo de preprocesamiento de datos que se a denominado Minero P*, algoritmo que en conjunto a herramientas del Soft Computing permiten dar respuesta a la pregunta ¿la página fue vista o percibida?. Para ello se asocia a cada objeto página tres conceptos propios, el concepto de visita corta, media y larga a esa página, lo que se traduce en una cualificación de la página en función de los hábitos en tiempos de visita de los usuarios. Se introduce un algoritmo para calcular de forma dinámica los valores de esos conceptos, y a partir de estos valores se realiza una aproximación al concepto de percepción de una página

Finalmente, en el Capítulo 5 se establece una metodología general para el análisis de web logs, mediante la integración de las técnicas de web mining clásicas y los nuevos conceptos derivados del desarrollo del capítulo 4. Se trata de extender el conjunto de información que se puede obtener en el análisis de los ficheros, para incorporar información que pueda hacer referencia a los conceptos introducidos en el capítulo 4.

La tesis concluye con las principales conclusiones obtenidas de esta memoria y sus proyecciones futuras visión que apunta al denominado "web inteligente".

CAPÍTULO 1: *PLANTEAMIENTO DEL PROBLEMA*

La gran red global de computadoras o World Wide Web se ha convertido en el mayor repositorio o bodega de conocimiento de la humanidad, este éxito espectacular de Internet se basa en dos pilares fundamentales: el protocolo HTTP y el lenguaje HTML, herramientas que simplifican el desarrollo e implementación de complejos sitios web en los servidores web que los alojan, estas soluciones web han transformado a la red global en una gran plaza de intercambio de información, servicios y productos, en donde las soluciones de *Ecommerce* o *Comercio Electrónico por Internet* pueden ser descritas como un gran mercado de comercio virtual en donde los clientes y las empresas virtuales y reales intercambian bienes, producto y servicios. Como Internet es una red de cobertura mundial, disponible las 24 horas del día y al alcance de la gran mayoría de las personas, es posible entender que las nuevas empresas virtuales y sus productos están al alcance de una distancia web para el cliente o usuario de estos mercados, distancia virtual que no presenta barreras geográficas o de otros tipos para la adquisición de objetos físicos u objetos de bits, siendo necesario disponer por ejemplo de un dinero virtual creado, transportado, cambiado o gastado en forma electrónica, para la intercambio de bienes y servicios. En esencia Internet ofrece a sus usuarios un nuevo canal de distribución y de comercialización lo cual admite en teoría, una relación virtual directa entre fabricantes de objetos reales y fabricantes de bits u objetos virtuales y los consumidores; lo que se traduce en la practica en información de datos o simplemente datos que son transportados desde una fuente primaria hasta un cliente y viceversa utilizando como medio de transporte un protocolo de comunicación de datos. Estos elementos operando integralmente, han permitido el desarrollo de distintas soluciones web o especializaciones, las cuales es posible identificar genéricamente como las E- áreas de especialización, que pueden ser referidas en nuestro idioma español como: I-Comercio, I-Estudio, I-Compras, I-área, en donde la letra "I" representa a la red de redes o Internet. Todas estas fuentes primarias almacenan información de datos, bajo la forma de un objeto virtual conformado o fabricado por medio de bit, los cuales se disponen en distintos ficheros o páginas web para su visualización.

Cualquier persona que desee publicar información puede hacerlo de manera sencilla con tan solo colocarla en un servidor Web, información que puede ser solicitada o accesada por medio de consultas o peticiones usando navegadores Web. Los procedimientos para proporcionar la

información requerida por un cliente de una solución web, son relativamente fáciles desde el punto de vista del usuario (proveedor / cliente), pero no lo es tanto desde el punto de vista tecnológico o de los recursos de la red involucrados.

Una característica del intercambio y recuperación de información en la red global, es que esta se establece por medio de un canal de comunicaciones entre una fuente y un destino (sujeto-objeto, entre ETD, receptor –emisor, software cliente-servidor, proceso que es establecido o implementado por los distintos dispositivos de comunicación utilizando un protocolo de comunicación de datos que para el caso de Internet corresponde a la suite o conjunto de protocolos denominados genéricamente como TCP/IP. Este intercambio de información de datos puede ser registrado empleando procesos o programas de software especializados que capturan los eventos producidos y los almacenan en archivos de registros de eventos o " log files" (por sus siglas en ingles), los cuales quedan disponibles y almacenados en servidores web y equipos electrónicos de comunicación de datos. Estos ficheros logs, constituyen una fuente de datos importante y son creados generalmente en el lado del servidor de forma automática como resultado del proceso de intercambio de información que se realiza entre un sujeto-objeto.

Este dominio de datos constituido por los ficheros logs será estudiado detalladamente en el presente capítulo con el objetivo de establecer el grado de complejidad que requiere su análisis o en otras palabras es el planteamiento del problema del análisis de la información contenida en los ficheros. A modo de ejemplo podemos indicar que en este tipo de ficheros quedan registradas las peticiones de objetos de un usuario en cuanto al resultado de las mismas, peticiones de objetos y recursos realizadas a un servidor web el cual las atiende y como un resultado secundario genera un evento que es almacenado en este tipo de fichero, a partir de esta fuente de datos se podrán establecer relaciones u asociaciones, como por ejemplo entre el tiempo de petición de una página por parte de un "cliente" y la permanencia de este en el servidor.

La información que contienen estos archivos "logs" es del tipo alfa-numérica y se refiere a datos como fecha, tiempo de conexión, dirección ip del cliente, archivo de destino en el servidor, dirección ip del servidor, puerto del servidor y otras que dependerán del formato empleado. Es evidente que la eficiente recuperación de patrones y la obtención de conocimiento desde el Web se ha convertido en un tema de vital importancia para todos aquellos que emplean este dominio

virtual. La información almacenada en distintas computadoras que están físicamente distribuidas en Internet, puede ser asociada conceptualmente con una gran base de datos distribuida que contiene información no estructurada. Esta base de datos conceptual o B_Internet, contiene distintos tipos de información la cual puede ser clasificada como: distribuida, sin etiquetas, heterogénea, medianamente mezclada, semi-estructurada, cambiante en el tiempo y de alta dimensionalidad; información disponible para un usuario en la forma de objetos que son solicitados por medio de algún programa de búsqueda y que se despliegan sobre una ventana virtual implementada en el computador del usuario. Conceptualmente el análisis de los datos contenidos en B_Internet puede ser referido al desarrollo de técnicas basadas en sistemas inteligentes, que permitan determinar las necesidades potenciales de un cliente, su comportamiento, como también el descubrimiento de conocimiento desde el web, empleando como fuente de origen los datos proporcionados por los programas o procesos que registran los eventos ocurridos por la navegación de los usuarios en Internet, eventos que son almacenados en archivos de registro de eventos o "web logs". La gran red global de computadoras o World Wide Web se ha convertido e en el mayor repositorio o bodega de conocimiento de la humanidad, este éxito espectacular de Internet se basa en dos pilares fundamentales: *el protocolo HTTP y el lenguaje HTML*.

1.1 Los datos almacenados en los Ficheros Logs

Una característica del intercambio y recuperación de información en la red global, es que esta se establece por medio de un canal de comunicaciones entre una fuente y un destino, entre un sujeto y un objeto, entre ETDs o equipos electrónicos de comunicación de datos, entre sistemas receptor–emisor; empleando para este intercambio de información protocolos de comunicación, programas de software cliente-servidor u otros programas especializados. Es destacable mencionar el rol fundamental en el desarrollo de la red global de computadoras que le corresponde al conjunto de protocolos de comunicación de datos TCP/IP, conjunto de protocolos sobre el cual se basa el intercambio de información en la red global de computadoras o Internet. En este intercambio de información de datos, intervienen distintos equipos electrónicos de comunicación, programas de software y protocolos de comunicación los cuales permiten transportar la información de datos empleando para este efecto un canal de comunicaciones sobre el cual se establece una comunicación efectiva entre un sujeto y un objeto. Este intercambio de información va generando distintos eventos o sucesos relacionados con los objetos requeridos, el puerto de destino y en forma más generalizada con el éxito o

fracaso de la comunicación entre dos o más entidades, cómo son por ejemplo un cliente y un servidor y viceversa. Estos sucesos o eventos de importancia pueden ser capturados y posteriormente registrados por medio de programas diseñados para cumplir este rol, programas que se ejecutan en los distintos servidores o en los equipos de comunicación de datos siendo estas aplicaciones las encargadas de capturar los distintos *eventos* de importancia que ocurren tanto a nivel físico o nivel del hardware, como en los distintos niveles de software que intervienen en una comunicación efectiva del "sujeto" con los "objetos" almacenados en un servidor de red.

Un evento puede ser entendido como el resultado de una operación realizada por un servidor ante una solicitud o requerimiento por parte de un cliente, como por ejemplo la solicitud y posterior retorno de un documento, la ejecución de un script, la bajada de un archivo, u otras solicitudes de servicios. Estos requerimientos de los usuarios hacia el servidor y las clases de respuestas que este entrega como resultado de las operaciones solicitadas pueden ser registradas empleando programas especializados que capturan el resultado de la operación, almacenando los distintos componentes asociados al tipo de solicitud y las respuesta entregadas por el servidor en un archivo de registro de eventos; este proceso automático de captura y registro de eventos es normalmente conocido como el "loggins de eventos" del servidor. Considerando la arquitectura de Web más sencilla, cada vez que un cliente realiza un requerimiento a un servidor de red, es enviado un paquete HTTP (Hypertext Transfer Protocol) sobre la red, desde el cliente hacia el servidor nombrado en el campo URL (Universal Resource Locator) de la solicitud. Luego el servidor retorna uno o más paquetes conteniendo o bien la respuesta, o bien un código de error.

Desde el punto de vista de la topología de redes, existen cuatro lugares posibles desde donde capturar los eventos o transacciones: sobre el servidor de red, sobre los servidores proxy, sobre los clientes, sobre la red.

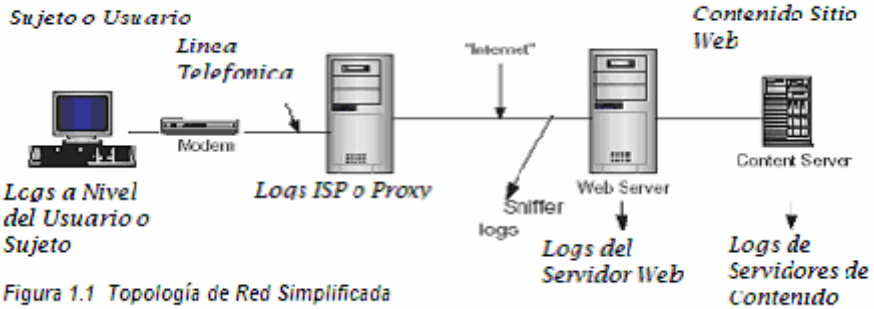


Figura 1.1 Topología de Red Simplificada

Una vez producido un evento o suceso como por ejemplo "***servidor no disponible***"; este es capturado por un programa especializado y registrado posteriormente en un "archivo de registro de eventos" o "log files" (por sus siglas en ingles), en estos archivos se almacenan el total de los sucesos relevantes que ocurren en un servidor web, siendo esta información almacenada descriptiva del tipo de evento o suceso ocurrido. En estos archivos por ejemplo pueden quedar registradas las páginas requeridas por un usuario, el tiempo de conexión del usuario con el servidor web, los puertos de acceso utilizados u otros parámetros referidos al uso del sitio web o del servidor de red que lo almacena. A partir de la información almacenada en los archivos logs, es posible establecer relaciones u asociaciones posteriores como por ejemplo: la hora y fecha cuando fue requerida una página web por parte de un "cliente" del servidor web, el total de páginas requeridas por un usuario en particular, la página más visitada o establecer relaciones más sofisticadas a partir del conocimiento subyacente en la información de datos contenidas en los diversos archivos que registran los eventos de un servidor web.

Otro tipo de información posible de ser almacenada en archivos de eventos, corresponde a los datos provenientes de las denominadas ***"galletas o cookies"*** (por sus siglas en ingles); código fuente generado por el administrador del servidor web con el objetivo de individualizar clientes, realizar un seguimiento automático del uso del sitio web u otras funciones mas sofisticadas. Las cookies "confían" en la "colaboración" implícita del usuario y son transportadas desde el servidor hasta el equipo del cliente generando conjuntos de datos que usualmente se denominan como ***datos provenientes del cliente,*** esta información puede ser recopilada con el empleo de un agente remoto, modificando el código fuente del motor de búsqueda o por medio de "*técnicas colaborativas voluntarias"* basadas en incentivos a los usuarios del servidor web. Las técnicas colaborativas, generalmente emplean ***applets java*** o códigos especializados y están orientadas a una mejor identificación del usuario y de su comportamiento sobre el uso del sitio web almacenado en el servidor. Una desventaja de este método es que requiere ejecutar código fuente en el equipo del cliente afectando el rendimiento del mismo. Un browser modificado y aceptado voluntariamente por el usuario tiene una mayor eficacia dado que aumenta la versatilidad de registrar distintos tipos de información del uso del sitio web.

En el caso del empleo de ***Servidores Proxy***, estos equipos suelen ser utilizados para mejorar el rendimiento o tiempo de acceso a páginas web; reduciendo la carga de la red tanto desde el lado del servidor web como desde el lado del cliente; estos servidores almacenan las páginas de mayor uso debiendo ***"predecir"*** el comportamiento del sitio, con la finalidad de cargar aquellas

páginas más solicitadas por los usuarios y almacenarlas a fin de reducir el tiempo de acceso a las mismas.

1.1.1 Archivos Logs basados en un Servidor de Web

Los servidores de Web pueden recolectar las solicitudes de documentos, objetos y otros servicios por parte de sus clientes, registrando cada solicitud http y sus resultados (el evento). Este proceso de captura de información es realizado de manera automática por una o varias aplicaciones que se ejecutan en tiempo real en el servidor, los resultados obtenidos ante un requerimiento de un cliente del servidor web son monitoreados y capturados por estas aplicaciones sean estos exitosos o hayan terminado en un error, resultando eventos que son almacenados en la forma de registros en un archivo dedicado a esta única función, estos archivos generalmente son conocidos como archivos de registro de eventos o archivos logs. Es destacable mencionar que la información almacenada en estos archivos no es del todo precisa o confiable dado que: las aplicaciones encargadas de capturar los eventos pueden ser personalizadas por el administrador del servidor, tanto en el tamaño máximo en bytes del archivo en donde son almacenados los eventos, como en otros parámetros relacionados con la captura de los mismos. A modo de ejemplo si el archivo definido en nombre y tamaño, completa su capacidad máxima de almacenamiento, algunos sucesos de importancia pueden no ser registrados. Por otra parte el empleo de Servidores Proxy afecta la calidad de los datos registrados en los archivos de registros de eventos del servidor principal, entre otras razones porque es posible que las solicitudes de los clientes queden registradas como anónimas o con el nombre del Proxy. Otro problema que se puede presentar es que la información almacenada en los logs del servidor excluye aquellos documentos (paginas web) que están temporalmente almacenado en un "*cache de páginas*". En caso de existir más de un servidor que proporciona un servicio, existirá la posibilidad que un mismo evento sea almacenado en dos archivos diferentes. Se deberá tener especial cuidado que el registro de los eventos en caso de dos o más servidores que prestan el mismo servicio, no se interfieran unos con los otros almacenando registros de eventos redundantes. Existe la posibilidad que para dar cumplimiento a normativas relativas a la privacidad, no sea posible recolectar los nombres de los clientes

1.1.2 Logs basados en un Servidor Proxy

El empleo de Servidores Proxy permite una serie de mejoras en la calidad de los servicios ofrecidos a los clientes de un servidor web, el conjunto de solicitudes de los clientes cuyos

buscadores o browsers están configurados para utilizar el proxy, son resueltas por este equipo de manera similar a como un servidor web resuelve esas solicitudes, es posible por tanto capturar los eventos generados a partir de las solicitudes de los "clientes del proxy" y almacenarlas en un archivo logs. Este registro de eventos no esta ausente de problemas, los cuales pueden ser resumidos como sigue:

En la actualidad los Web Browser utilizan un servidor proxy solo si el usuario se lo indica expresamente, es decir el buscador deberá estar configurado para utilizar un proxy. Como resultado de los anterior los archivos logs de los servidores proxy podrían almacenar información parcial o incompleta de sus clientes.

Es necesario considerar como una dificultad adicional, que es necesario que los usuarios del servidor proxy conozcan de su existencia y tengan la capacitación necesaria para configurar sus buscadores con el objetivo de utilizarlos.

Generalmente los buscadores o browsers crean en la memoria y discos del cliente "caches" de objetos y por tanto aquellas solicitudes referidas a estos objetos son resueltas por el equipo en donde se esta ejecutando el buscador (browsers) y no son enviadas al servidor proxy

1.1.3 Logs basados en un Cliente

El registro de eventos realizado sobre la misma máquina cliente de un servidor de red, tiene varias ventajas respecto a los métodos expuestos anteriormente, dentro de estas ventajas se encuentra que es posible registrar aquellas solicitudes que fueron resueltas por el cache de memoria o disco de la máquina cliente. En el caso de este tipo de registro de eventos, es necesario disponer de una aplicación que realice el monitoreo de la estación de trabajo del usuario, siendo esto último una considerable limitación, dado que es necesario constar con la colaboración explicita del usuario o cliente. Algunas debilidades de este método pueden ser resumidas como sigue:

Se requiere de la colaboración explicita del usuario

El formato de los archivos de registro es dependiente de la plataforma de trabajo del usuario, del buscador utilizado y de las configuraciones propias de la estación de trabajo.

No existe un estándar para archivos logs de clientes

Es necesario utilizar buscadores modificados.

El rendimiento de la máquina cliente se deteriora proporcionalmente al tipo de monitoreo que se realiza sobre esta

Con el empleo de aplicaciones de monitoreo y registro de eventos sobre la estación de trabajo del cliente, este puede ver afectada su privacidad. registro de eventos

1.1.4 Archivos Logs basados en el Monitoreo de la Red

Es posible disponer programas o equipos especializados para monitorear el trafico de red entre clientes y servidores, los programas que realizan esta función son conocidos como analizadores de tráfico o vulgarmente como "olfateadotes o sniffers". En este tipo de registro de eventos se puede emplear un completo equipo de monitoreo especializado como lo es por ejemplo un analizador de trafico Ethernet, o en su defecto disponer de un simple programa alojado en un host que escucha pasivamente todo el tráfico que viaja en la red. El método consiste en capturar y analizar el contenido de los distintos paquetes que transitan por el segmento de red en donde es alojado el analizador o el host que contiene el programa, para luego identificar los paquetes en transito que contienen en su interior partes de mensajes previamente seleccionados, como por ejemplo aquellos que contienen "llamados HTTP" con la finalidad de construir un archivo log de las URLs solicitadas en dichos paquetes. Una de las debilidades de este método consiste en que solo es posible "escuchar" a los clientes que están en el mismo segmento de red que el equipo que realiza la "escucha" o del host que almacena al programa de monitoreo, con lo cual la ubicación del equipo especializado o programa de monitoreo es de mucha importancia. Junto a lo anterior se presenta la dificultad adicional que el monitoreo debe de ser realizado a nivel de paquetes en tránsito siendo necesario indicar un tiempo de muestreo o frecuencia de captura de paquetes con la finalidad de no saturar la entrada de datos del equipo o del programa "sniffers". La principal desventaja del monitoreo de red es la necesidad de que la red permita broadcasting o paquetes destinados a todo el universo de la red o bien realizar el truco de configurar el monitor de red como una puerta de encaminamiento o enlace (gateway) sobre un enlace de red punto-a-punto. Las principales ventajas que encierra esta técnica están en el hecho que un programa monitor de red puede utilizar el encabezado de los mensajes capturados para realizar cálculos basados en múltiples paquetes, y de esta manera producir más información respecto de la utilización del Web de la que se puede obtener con el formato de logs estándar de los servidores proxy o de otros servidores. Adicionalmente esta técnica no

afecta a los servidores de red ni a sus clientes. El monitoreo de red puede estar diseñado para capturar distintos protocolos de red, es decir es independiente del tipo de protocolo utilizado. Otra ventaja considerable es que esta técnica puede utilizarse para verificar información registrada por otros mecanismos de logging. de eventos pudiendo programarse al monitor de red para registrar información de múltiples protocolos, como ftp, pop, http, smtp, tcp, tftp y otros.

1.1.5 Monitores de Red para el Web Plataforma Unix

En los sistemas operativos ***Unix*** existen de manera usual dos aplicaciones especializadas denominadas casi por regla general como ***httpfilt*** y ***httpdump,*** las cuales permiten recopilar eventos y registrarlos en archivos logs preestablecidos. . Estos programas son ejecutados en tiempo real en el sistema operativo y habitualmente se consideran como parte de este, su diseño permite registrar logs de todas las solicitudes HTTP que ocurren en la red a la cual están estos proceso dirigen su monitoreo. El proceso *httpfilt* por ejemplo esta diseñado para registrar solicitudes HTTP, mientras que *httpdump* provee información más extensa de cada solicitud y puede ser extendida para registrar tráfico proveniente de otros protocolos diferentes de HTTP.

1.1.6 Como se crea un Fichero Logs.

Una administración eficiente de un Servidor Web necesariamente se basa en monitorear y registrar aquellos eventos o sucesos importantes relativos por ejemplo al rendimiento del servidor, a la actividad que están realizando los usuarios sobre este o en su defecto registrar cualquier problema de hardware o software que pudiera ocurrir en un instante dado. En esencia un sistema de registro de eventos que forma parte del sistema operativo, proporciona una forma estandarizada para que diversas aplicaciones de un servidor "registren" los eventos de hardware o software de importancia en un archivo cuyo formato es conocido por estas aplicaciones. El sistema esta compuesto básicamente por una aplicación o proceso que se ejecuta en tiempo real en el servidor, aplicación dedicada a la captura y posterior registro en un archivo que responde a un tipo de formato estandarizado. En términos generales es posible clasificar los tipos de registros de eventos en las categorías siguientes: *Registros de Errores, Registros de Acceso, Registros de Seguridad o Alarmas de Seguridad, Registros Múltiples o Combinados, Vaciado de Memoria (Dump Memory o Crash)*

La información que contienen estos archivos "logs" es principalmente del tipo numérica y se refiere a datos como fecha, tiempo de conexión, dirección ip del cliente, el fichero o página de destino en el servidor, puerto de destino del servidor. A modo de ejemplo el ***Servidor Web Apache*** registrará cualquier evento por causa de peticiones de recursos, objetos y servicios implementados sobre los ficheros *access_log* y *error_log*, definidos por medio de una directiva de sistema; Esta directiva o comando de sistema tiene la forma indicada en el ejemplo siguiente:

Ejemplo 1.1. Registro de eventos

Directiva de sistema que define nombre de archivo log en Servidor Web Apache
Fuente: http://httpd.apache.org/docs-2.0/es/mod/core.html#errorlog
La directiva (el comando) Error Log fija el nombre del archivo en donde el servidor registrará cualquier evento o suceso de error.

Sintaxis: ErrorLog file-path|syslog[:facility]

Ejemplo: ErrorLog /var/log/httpd/error_log

En el caso del servidor web de ***Microsoft*** conocido como ***IIS o Microsoft Internet Información Server*** este incluye capacidades adicionales como es la de registrar los eventos o sucesos ocurridos con la actividad de los usuarios del sitio web o la supervisión del rendimiento del servidor; en estas capacidades "adicionales" es posible observar que los registros pueden incluir información acerca de quien visito el sitio web, cuales fueron los contenidos visualizados, cuál fue la última información visualizada. En ***IIS*** es posible registrar aquellos eventos generados a partir de los intentos fallidos o exitosos en el acceso del o los componentes del sitio web implementado en el servidor, componentes tales como carpetas virtuales, archivos con información u otros objetos. En el caso del Servidor ***Web IIS***, los errores o sucesos producidos por las aplicaciones que emplean el protocolo http, son capturados y controlados de manera automática por un proceso especializado denominado ***API HTTP.*** El registro de errores que esta aplicación realiza se configura por medio parámetros, proceso que es necesario de realizar previamente por medio de la escritura de una clave ***HTTP/ Parameters*** en la base de datos de configuración del sistema operativo o Registro Windows; por medio de la asignación de estos parámetros asociados a esta clave, es posible definir el tamaño del archivo logs, ubicación del mismo, el formato del archivo logs, como también su nombre y ubicación.

Ejemplo 1.2 Clave del Registro HTTP/Parameters

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\HTTP\Parameters

Valor del Registro	**Descripción**
EnableErrorLogging	Valor **DWORD** que puede establecer en TRUE para habilitar el registro de errores o en FALSE para deshabilitarlo. El valor predeterminado es TRUE
ErrorLogFileTruncateSize	Valor **DWORD** que especifica el tamaño máximo de los archivos de registro de errores, en bytes. El valor predeterminado es un MB (0x100000).
ErrorLoggingDir	Valor **String** (cadena) que especifica la carpeta en la que la API HTTP coloca los archivos de registro. La API HTTP crea una subcarpeta **HTTPERR** en la carpeta especificada y almacena los archivos de registro en ella. Esta subcarpeta y los archivos de registro reciben a misma configuración de permisos. El administrador y las cuentas del sistema local reciben acceso completo. Los demás usuarios no tienen acceso.

1.1.7 El formato de los archivos logs. Limitaciones

Una de las principales fuentes de información en la cual puede fijarse el dominio de estudio para la extracción de conocimiento del web (Web Usage Mining) corresponde a los conjuntos de datos provenientes de los procesos que capturan y registran los eventos ocurridos sobre un **Servidor Web,** eventos que son almacenados en los denominados "archivos logs" o más simplemente "web logs". La información generada por un evento es capturada y posteriormente almacenada bajo la forma de un registro que se almacena en un archivo ASCII de texto, empleando para este fin un formato conocido; el formato corresponde a la forma en como se almacenan los datos provenientes del evento como por ejemplo el formato de la hora y la fecha.

Uno de los formatos más empleados corresponde al desarrollado por NCSA (National Center for Supercomputing Applications) o ***Formato Común de Registro*** o ***Common Logs Format (CLF)*** y su extensión o ampliación denominada ***Formato de Registro Combinado o Combined Logs Format (XCLF)***, por medio del empleo de estos formatos el Servidor Web Apache almacena los registros que contienen la información de un evento. La empresa Microsoft en cambio, para su servidor web denominado Microsoft Internet Information Server

o IIS, suministra la posibilidad de grabar en un fichero log o de registro todos los eventos o entradas de los usuarios que se conectan al mismo en tres formatos diferentes: ***Microsoft IIS Log File Format***, el ***NCSA Common Log File Format***, y ***W3C Extended Log File Format***, además de la posibilidad de grabar los datos directamente en un servidor de datos mediante un enlace ODBC con la opción ODBC Logging. El formato que más información ofrece al administrador del servidor web es mediante la utilización del formato W3C Extended, ya que permite recoger hasta un total de 20 datos diferentes, incluyendo las direcciones IP del cliente y del servidor, la fecha / hora de la conexión, los bytes enviados / recibidos, los recursos visitados, la consulta solicitada y otras opciones.

Los formatos por tanto son relativamente libres y descriptivos del tipo de error o suceso, de la aplicación que los captura y de las características de diseño del servidor web aplicadas por su fabricante o desarrollador; en términos generales todos los formatos incluyen datos como la fecha y hora del error, la dirección ip del cliente u otra información relevante, no obstante a lo anterior, en la mayoría de los archivos de registro de eventos o logs es común emplear un formato estándar para indicar la fecha y la hora del suceso, basado en la norma ***ISO 8601*** que se refiere a recomendaciones para la representación de la fecha y la hora estos para el intercambio de información. En un archivo logs típico los registros se dispondrán de manera similar al ejemplo siguiente:

Ejemplo 1.3 Registro Típico Almacenado en un Archivo Logs

(Plataforma Unix, Servidor Web Apache)
http://httpd.apache.org/docs-2.0/es/logs.html

```
[Wed Oct 11 14:32:52 2000] [error] [client 127.0.0.1] client denied by server configuration:
/export/home/live/ap/htdocs/test
```

Una característica importante de mencionar respecto a los servidores web comerciales, se refiere al como se implementan el o los sistemas de registros de eventos que emplean estos servidores. Estos sub-sistemas, están diseñados para capturar y registrar eventos empleando programas o procesos especializados; estos programas generalmente incluyen diversas capacidades que permiten personalizar o depurar la captura de los sucesos ocurridos en el servidor, junto a lo anterior los archivos en donde se registrarán los eventos emplean alguno de los distintos formatos descritos anteriormente, formatos que son utilizados a discreción por los distintos fabricantes de acuerdo a criterios de diseño propios del sistema. Esta diversidad de captura de información, nos permite concluir que la información disponible en estos archivos logs puede

ser de contenido distinto entre servidores web implementados con un mismo producto de software.

Considerando al ***Servidor Web Apache*** como ejemplo; podemos indicar que este producto de software suministra al administrador del servidor las herramientas necesarias para establecer un reporte de errores o reportes de acceso altamente configurables. El servidor emplea el formato conocido *como* ***Formato Común de Registro o Common Logs Format (CLF)***, o su extensión o ampliación denominada ***Formato de Registro Combinado o Combined Logs Format (XCLF)*** para disponer los distintos registros de los eventos o sucesos ocurridos en el servidor en un archivo cuyo nombre y ubicación puede ser definida por el administrador del servidor web. Una configuración típica por medio de una línea de comando la cual fija el formato del registro de acceso o logs de acceso del Servidor Apache corresponde a:

Ejemplo 1.4 Directiva LogFormat "%h %l %u %t \"%r\" %>s %b" common
CustomLog logs/access_log common

Con esto se define el apodo (nickname) *common* y se lo asocia con un determinado formato. El formato consiste en una serie de directivas o comandos con tantos por ciento; cada uno de estos le dice al servidor que registre una determinada información en particular. El formato también puede incluir caracteres literales, que se copiarán directamente en el registro. La directiva o comando *CustomLog*, crea un nuevo fichero de registro usando el apodo definido. El nombre del fichero de registro de acceso se asume que es relativo al valor especificado en *ServerRoot* a no ser que empiece por una barra (/).

La configuración de arriba escribirá las entradas en el registro con el formato conocido como Formato Común de Registro (CLF). Este formato estándar lo pueden generar muchos servidores web diferentes y lo pueden leer muchos de los programas que analizan registros. Las entradas de un fichero de registro que respetan ese formato común tienen una apariencia parecida a esta:

Ejemplo 1.5 Formato CLF.

127.0.0.1 - frank [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif HTTP/1.0" 200 2326

El llamado Formato de Registro Combinado tiene la forma siguiente:

Ejemplo 1.6 Formato de Registro Combinado

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-agent}i\"" combined
CustomLog log/access_log combined
```

Es exactamente igual que el Formato Común de Registro, pero añade dos campos. Cada campo adicional usa la directiva ***%{header}i***, donde *header* puede ser cualquier cabecera de petición HTTP. El registro de acceso cuando se usa este formato tendrá este aspecto:

Ejemplo 1.7 Registro de Acceso Formato de Registro Combinado

```
127.0.0.1 - frank [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif HTTP/1.0" 200 2326
"http://www.example.com/start.html" "Mozilla/4.08 [en] (Win98; I ;Nav)"
```

Los campos adicionales son:

```
"http://www.example.com/start.html" (\"%{Referer}i\")
```

La cabecera de petición de HTTP "Referer" (sic). Muestra el servidor del que proviene el cliente.

```
"Mozilla/4.08 [en] (Win98; I ;Nav)" (\"%{User-agent}i\")
```

La cabecera de petición HTTP "User-Agent". Es la información de identificación que el navegador del cliente incluye sobre sí mismo.

1.2 Limitaciones en el análisis de la información contenida en los ficheros logs

El contenido de información que es almacenada en los archivos log, por medio de registros o entradas de eventos, pueden ser significativamente diferente. Las siguientes líneas de ejemplo proceden de un registro de errores de la API HTTP de IIS de Microsoft:

Ejemplo 1.8 Un registro de errores típico de la API http

```
2002-07-05 18:45:09 172.31.77.6 2094 172.31.77.6 80 HTTP/1.1 GET /qos/1kbfile.txt 503 – ConnLimit 2002-07-
05 19:51:59 127.0.0.1 2780 127.0.0.1 80 HTTP/1.1 GET /ThisIsMyUrl.htm 400 – Hostname 2002-07-05 19:53:00
127.0.0.1 2894 127.0.0.1 80 HTTP/2.0 GET / 505 - Version_N/S 2002-07-05 20:06:01 172.31.77.6 64388
127.0.0.1 80 - - - - - Timer_MinBytesPerSeco
```

El formato utilizado por la API HTTP del servidor IIS es normalmente el mismo formato que los de W3C, con la diferencia de que los archivos de registro de errores de la API HTTP no tienen encabezados de columna. Cada línea de un registro de errores de la API HTTP registra un error. Los campos aparecen en un orden específico. Un carácter de espacio único (0x0020)

separa cada campo del anterior. En cada campo, el signo más (0x002B) reemplaza a los caracteres de espacio, tabuladores y caracteres de control no imprimibles.

La interpretación de la información contenida en los distintos campos de este registro esta dada por la tabla siguiente:

Tabla 1.1 Campos de un Registro de Error de la API http

Campo	Descripción
Fecha	El campo Fecha sigue el formato de W3C. Este campo se basa en el Horario universal coordinado (UTC). El campo Fecha tiene siempre diez caracteres, con el formato AAAA-MM-DD. Por ejemplo, el 1 de mayo de 2003 se expresa como 2003-05-01.
Hora	El campo Hora sigue el formato de W3C. Este campo se basa en el UTC. El campo Hora tiene siempre ocho caracteres, con el formato MM:HH:SS. Por ejemplo, las 5:30 p.m. (UTC) se expresa como 17:30:00.
Dirección IP del cliente	Dirección IP del cliente afectado. El valor de este campo puede ser una dirección IPv4 o una dirección IPv6. Si la dirección IP del cliente es IPv6, en la dirección se incluye también el campo ScopeId.
Puerto del cliente	Número de puerto del cliente afectado.
Dirección IP del servidor	Dirección IP del servidor afectado. El valor de esta campo puede ser una dirección IPv4 o una dirección IPv6. Si la dirección IP del servidor es IPv6, en la dirección se incluye también el campo ScopeId
Puerto del servidor	Número de puerto del servidor afectado.
Versión del protocolo	Versión del protocolo que se utiliza. Si la conexión no se ha analizado lo suficiente como para determinar la versión del protocolo, se utiliza un guión (0x002D) como marcador de posición del campo vacío. Si el número de la versión principal o el de la versión secundaria analizadas es igual o mayor a 10, la versión se registra como HTTP/?.?.
Verbo	Estado del verbo que pasa la última solicitud analizada. Se incluyen los verbos desconocidos, pero si el tamaño de un verbo es de más de 255 bytes, se trunca en esta longitud. Si no hay ningún verbo disponible, se utiliza un guión (0x002D) como marcador de posición del campo vacío.
CookedURL + Consulta	La dirección URL y cualquier consulta asociada se registran como un campo separado por un signo de interrogación (0x3F). Este campo se trunca en su longitud límite, 4096 bytes. Si esta dirección URL se ha analizado (es decir, es una versión canónica de caracteres anchos, "cooked"), se registra con conversión de la página de código local y se trata como campo Unicode. Si esta dirección URL no se ha analizado ("cooked") en el momento del registro, se copia tal como es, sin conversión Unicode. Si la API HTTP no puede analizar esta dirección URL, se utiliza un guión (0x002D) como marcador de posición del campo vacío.
Estado del protocolo	El estado del protocolo no puede ser mayor de 999. Si el estado del protocolo de la respuesta a una solicitud está disponible, se registra en este campo. Si el estado del protocolo no está disponible, se utiliza un guión (0x002D) como marcador de posición del campo vacío.
Id. del sitio	No se utiliza en esta versión de la API HTTP. En este campo aparece siempre un guión (0x002D) como marcador de posición.
Reason Phrase	Este campo contiene una cadena que identifica el tipo de error que se está registrando. Este campo no se deja vacío nunca.

1.2.1 Comparación de los datos almacenados en distintos formatos de ficheros logs

En los siguientes ejemplos se indican cómo se registran los datos de las transacciones de mensajes y se presentan en cada uno de los cuatro formatos. Hemos considerado para este ejemplo, las entradas de registro de las transmisiones SMTP las cuales normalmente incluyen varios registros. Sólo se muestra un registro para cada formato, a fin de permitir ver las diferencias entre los formatos.

Ejemplo 1.9 Ejemplo de Registro de Eventos en distintos formatos

Maria Bonita de Nwtraders.com envía un mensaje a un colega de Micropalt el 18 de agosto de 1997, aproximadamente a las 13:17 p.m.

Formato de archivo de registro IIS de Microsoft
10.456.789.1, Nwtraders.com, 8/18/97, 13:17:37, SMTPSVC1, MAIL01, 19.200.200.1, 90, 42, 0, 250, 0, MAIL FROM, -, FROM:*mariabonita@Nwtraders.com*

Formato de archivo de registro común NCSA
10.456.789.1 - Nwtraders.com [18/Ago/1997:13:17:37 - 0800] "MAIL FROM -? DE: mariabonita HTTP/1.0 250 0
Formato de archivo de registro extendido W3C
En el ejemplo siguiente sólo se incluye la configuración predeterminada. Puede agregar otras opciones.
13:17:37 10.456.789.1 MAIL FROM - 250

Es fácilmente observable las diferencias entre los "datos crudos" de uno y otro formato o de los registros contenidos en los archivos logs entre los distintos formatos. Estos registros que almacenan la información del evento o suceso son capturados por la aplicación encargada, aplicación que en el caso de los servidores web mayormente utilizados como el Servidor Apache (demonio httpd) y el Servidor IIS (API http) son configurables, es decir permiten definir el tipo de suceso o eventos a registrar. Por ejemplo la API HTTP registra las repuestas de error a clientes, los tiempos de espera de conexión, las solicitudes huérfanas y las conexiones interrumpidas que no se controlan correctamente. Se describen los tipos de sucesos factibles de ser capturados por la API http:

Respuestas a clientes: La API HTTP envía una respuesta de error a un cliente, por ejemplo, un error 400 debido a un error de análisis en la última solicitud recibida. Una vez que la API HTTP envía la respuesta de error, finaliza la conexión.
Tiempos de espera de conexión: La API HTTP finaliza la conexión tras el tiempo de espera. Si hay una solicitud pendiente cuando se excede el tiempo de espera de la conexión, la solicitud se utiliza para proporcionar más información sobre la conexión en el registro de errores.

Solicitudes huérfanas: Un proceso en modo de usuario se detiene inesperadamente cuando todavía hay solicitudes en cola dirigidas a dicho proceso. La API HTTP registra las solicitudes huérfanas en el registro de errores.

Los errores concretos se describen mediante *Cadenas Reason Phrase* que aparecen siempre como el último campo de cada línea de error. En la tabla siguiente se enumeran las cadenas Reason Phrase de la API HTTP:

Tabla 1.2 Cadenas Reason Phrase de la API HTTP

Reason Phrase	Descripción
AppOffline	Se ha producido un error de servicio no disponible (error 503 de HTTP). El servicio no está disponible porque los errores de la aplicación han hecho que la aplicación quede sin conexión.
AppPoolTimer	Se ha producido un error de servicio no disponible (error 503 de HTTP). El servicio no está disponible porque el proceso del grupo de aplicaciones está ocupado y no puede atender la solicitud.
AppShutdown	Se ha producido un error de servicio no disponible (error 503 de HTTP). El servicio no está disponible porque la aplicación se ha cerrado automáticamente en respuesta a la directiva del administrador.
BadRequest	Se ha producido un error de análisis al procesar la solicitud.
Connection_Abandoned_By_AppPool	Un proceso de trabajo del grupo de aplicaciones que se ha cerrado inesperadamente o ha dejado una solicitud pendiente al cerrar el controlador.
Connection Dropped	Reservado. No se utiliza actualmente.
ConnLimit	Se ha producido un error de servicio no disponible (error 503 de HTTP). El servicio no está disponible porque se ha alcanzado o excedido el límite de conexión de nivel de sitio.
Disabled	Se ha producido un error de servicio no disponible (error 503 de HTTP). El servicio no está disponible porque un administrador ha dejado la aplicación sin conexión.
EntityTooLarge	Una entidad supera el tamaño máximo permitido.
FieldLength	Se ha excedido el límite de longitud del campo.
Forbidden	Durante el análisis se ha encontrado un elemento o secuencia prohibido.
Header	Se ha producido un error de análisis en un encabezado.
Hostname	Se ha producido un error de análisis al procesar un nombre de host.
Internal	Se ha producido un error de servidor interno (error 500 de HTTP).
Invalid_CR/LF	Se ha producido un retorno de carro o salto de línea no permitido.
LengthRequired	Falta un valor de longitud requerido.
No disponible	Se ha producido un error de servicio no disponible (error 503 de HTTP). El servicio no está disponible porque se ha producido un error interno (como por ejemplo un error de asignación de memoria).
N/I	Se ha producido un error de no implementación (error 501 de HTTP), o un error de servicio no disponible (error 503 de HTTP) a causa de una codificación de transferencia desconocida.
Number	Se ha producido un error de análisis al procesar un número.
Precondition	Falta una precondición requerida.
QueueFull	Se ha producido un error de servicio no disponible (error 503 de HTTP). El servicio no está disponible porque la cola de solicitudes de la aplicación está llena.
RequestLength	Se ha excedido el límite de longitud de la solicitud.
Timer_AppPool	La conexión ha caducado porque una solicitud esperó demasiado tiempo en una cola del grupo de aplicaciones a que una aplicación de servidor la sacara de la cola y la procesara. La

Timer_ConnectionIdle	duración del tiempo de espera es ConnectionTimeout. De manera predeterminada, este valor se establece dos minutos. La conexión ha caducado y sigue inactiva. La duración predeterminada de ConnectionTimeout es de dos minutos.
Timer_EntityBody	La conexión ha caducado antes de que llegara el cuerpo de la entidad de la solicitud. Cuando está claro que una solicitud tiene un cuerpo de entidad, la API HTTP activa el contador Timer_EntityBody. Inicialmente, el límite de este contador se establece en el valor de ConnectionTimeout (normalmente 2 minutos). Cada vez que se recibe otra indicación de datos en esta solicitud, la API HTTP vuelve a establecer el contador para dar a la conexión dos minutos más (o el tiempo especificado en ConnectionTimeout).
Timer_HeaderWait	La conexión ha caducado porque el análisis de encabezado de una solicitud ha superado el límite predeterminado de dos minutos.
Timer_MinBytesPerSecond	La conexión ha caducado porque el cliente no recibía la respuesta a una velocidad razonable. La velocidad de la respuesta era menor que el valor predeterminado de 150 bytes por segundo.
Timer_Response	Reservado. No se utiliza actualmente.
URL	Se ha producido un error de análisis al procesar una dirección URL.
URL_Length	Una dirección URL ha excedido el tamaño máximo permitido.
Verb	Se ha producido un error de análisis al procesar un verbo.
Version_N/S	Se ha producido un error de versión no compatible (error 505 de HTTP).

1.3 Algunos problemas en el análisis de logs tradicional.

El análisis de los archivos "web logs" por medio de herramientas de software comerciales o de libre disposición esta limitado generalmente a la determinación de resultados estadísticos a partir de la información almacenada en los ficheros de registros de eventos o "ficheros logs" de los servidores web. Estos ficheros recogen como hemos mencionado anteriormente todas las "peticiones" hacia el servidor y las almacenan como un evento o registro, conjunto de datos en el cual queda descrito el suceso o la solicitud realizada. La información recopilada en estos archivos, no es del todo fiable, producto de varios factores, como por ejemplo las páginas visitadas que están almacenadas en caches y proxies no son registradas como un evento al momento de ser requeridas, dado que no son suministradas por el servidor web sino por máquinas que participan en la comunicación efectiva con el usuario con la finalidad de reducir el trafico de red, esto es de mucha importancia dado que el empleo de servidores caches y proxies, permite un significativo ahorro de costos de ancho de banda en la comunicación.

```
192.168.217.117 - - [06/Jun/2006:23:08:21 -0400] "GET / HTTP/1.1" 200 11189
192.168.217.117 - - [06/Jun/2006:23:08:21 -0400] "GET /funciones_dw.js HTTP/1.1" 304 -
192.168.217.117 - - [06/Jun/2006:23:08:21 -0400] "GET /estilo/estiloportal.css HTTP/1.1" 304 -
```

Ejemplo 1.10 .Fichero Logs filtrado por dirección IP (Fuente: U.Virtual UTEM)

El empleo de servidores proxies y caches tiene por consecuencia que gran cantidad de las solicitudes de páginas de los usuarios de un sitio web, llegan a su destino suministradas directamente desde el servidor cache del proveedor de acceso a Internet o desde el servidor proxy de la institución o empresa, también es factible que estas páginas sean suministradas directamente por el cache del navegador que el usuario este utilizando. Este efecto puede ser relevante al momento de emplear herramientas de análisis de logs dado en sitios con mucho tráfico, la proporción de páginas entregadas por el servidor web del sitio puede llegar a ser significativamente menor que aquellas entregadas por los servidores de apoyo a la gestión del sitio (caches y proxys) distorsionando cualquier análisis estadístico de la información almacenada en los logs.

Otro defecto importante de los "analizadores de logs"surge de la propia naturaleza de la red Internet; el denominado WWW o red global de computadoras se comporta para un usuario como un canal de comunicaciones unidireccional desde el punto de la información o de los recursos requeridos por este; esta búsqueda unidireccional de información o de recursos, obliga a los usuarios a navegar por distintos servidores disponibles en la red, ya sea de manera manual indicando la URL (http:\\ ubicación) o de forma asistida por medio de programas buscadores especializados o motores de búsqueda. Los motores de búsqueda, son poderosas herramientas de software especializadas en la búsqueda de recursos e información disponible en Internet, asisten a los usuarios de la red global a buscar información especifica desde una fuente disponible en la red global, red que se caracteriza por las grandes cantidades de información que almacena, por la dinámica de cambio de las fuentes de información disponibles, tanto en su actualización como en su obsolescencia y por la variedad de los tipos de información, variedad que transforma la búsqueda de los recursos o información de interés de un usuario en un trabajo lento, complejo e improductivo. Los motores de búsqueda se construyen (o se diseñan) con un objetivo fundamental: facilitar la búsqueda de recursos o información disponible en la red Internet o en palabras más simples *"encontrar lo que la gente o usuarios de Internet desean"*.. La red global puede ser entendida como un sistema de información textual o de hipertextos distribuida, en donde los usuarios buscan información o recursos empleando enlaces o "link" hacia estos recursos; estos enlaces quedan determinados por la ubicación física de los servidores que contienen estos recursos o información, la naturaleza de la información contenida en el web es por tanto descentralizada. Un motor de búsqueda deberá realizar dos funciones fundamentales: construir índices de la información contenida en el web y navegar de manera automática por demanda o petición. por medio del empleo de agentes de indexación de índices de los

documentos registrados en el Web, construyendo listas de los documentos registrados en los distintos servidores web. Para la construcción de estas listas, los browsers o "buscadores" generan consultas o peticiones a los servidores que almacenan la información requerida o solicitada por un usuario, peticiones que quedan reflejadas como un evento en un archivo logs. Motores de búsqueda como por ejemplo *Webcrawler*, *Lycos*, *InfoSeek*, *Excite*, *Altavista*, *Google*, *Yahoo*, *Mosaic* indexan hasta valores cercanos al 90% de los documentos del web, lo que los trasforma en "súper usuarios" de los servidores web disponibles en la red, dado que permanentemente están realizando peticiones con la finalidad de establecer las listas de los documentos disponibles en estos servidores. Estos *agentes de indización* son por consecuencia súper generadores de eventos dado que permanentemente realizan peticiones a los servidores web. Una consulta o un criterio de búsqueda aplicado por un usuario a un motor de búsqueda, tiene por resultado una larga lista con cientos y miles de "índices" relacionados con la consulta efectuada; este resultado entregado al usuario por el buscador ha tenido previamente un efecto adicional sobre los servidores web que almacenan la información requerida por el usuario, la causa de este efecto corresponde a una petición del agente de indexación del browser y la consecuencia de esta petición es la generación de eventos en cada uno de los servidores que almacenaban la información de su interés. El resultado o lista de índices obliga al usuario a emprender una tediosa selección uno a uno de los documentos relacionados con su consulta lo cual prácticamente anula la efectividad de los browsers y por consecuencia el registro de los eventos generados por los mismos sobre un servidor y su posterior análisis por medio de herramientas comerciales.

Los agentes de indización empleados por los browsers desde el punto de vista de los eventos registrados en un fichero log de un servidor web pueden ser asociados como "usuarios virtuales" que generan eventos en grandes cantidades, distorsionando la información almacenada en los logs; esta distorsión corresponde esencialmente al volumen o porcentaje de información almacenada por esta causa pudiéndose considerar estos datos almacenados como "ruido" propio de la naturaleza del web.

Otras fuentes generadoras de eventos corresponden a los *agentes inteligentes* y a los *robots* o *spiders,* estos "autómatas de software" recorren los distintos servidores web con el objetivo de registrar un documento, esta técnica de búsqueda es esencialmente automática y se basa en programas de software que recorren de forma autónoma la red de acuerdo a ciertos parámetros determinados por la información contenida en el programa realizado por su creador. Un programa robot o un

agente inteligente de búsquedas esta capacitado para recorrer el *hiper-espacio textual* constituido por las páginas propuestas por los propios usuarios de la red, quienes pueden escribir en sus páginas web determinados códigos para que puedan ser capturados por los robots de búsquedas y así obtener visibilidad en el web. Independiente de lo anterior los *robots* o *spiders* generalmente tienen la capacidad de visitar a los servidores web con lo cual se constituyen en grandes generadores de eventos mientras realizan su rutina de recorrer el web con el objetivo de registrar un documento. Un robot o un agente inteligente difiere fundamentalmente de un browser en un aspecto: la autonomía en la búsqueda; un browser normalmente debe de ser operado por un usuario real o en palabras más precisas por una persona humana, en cambio un agente inteligente es un dispositivo de software que tiene incorporado técnicas de aprendizaje y opera bajo ciertas características de programación en la búsqueda de documentos de acuerdo a ciertos criterios, entregando sus resultados directamente al usuario generador, resultados que no están destinados a la construcción de listas de índice o al almacenamiento en bases de datos de búsqueda. Los robots recorren el web y a partir de estos recorridos aprenden caminos que permitirán recomendar a los usuarios los sitios en donde podrán encontrar la información requerida, es destacable mencionar que agentes inteligentes, robots y spiders pueden ser considerados como grandes fuentes generadoras eventos siendo necesario identificarlos ya sea para bloquear su acceso al servidor web o en su defecto para que los eventos generados por sus peticiones sean auditables por medio de programas de análisis de eventos.

1.3.1 Como opera un robot, como genera eventos o ruido de eventos

Los robots, spiders y agentes inteligentes son programas de búsqueda de información especializados, reúnen la información para la cual están diseñados o en su defecto reúnen los enlaces que las contienen, almacenando los resultados en bases de datos. Esta Información almacenada puede ser el texto completo requerido por el diseñador del robots, o solamente el titulo o el enlace que contiene la información solicitada.

Los motores de búsqueda habitualmente emplean estos programas robots para implementar los servicios de búsqueda que ofrecen a los usuarios del web, por ejemplo *Google* emplea el robot conocido como *GoogleBot* el cual cumple entre otras funciones las siguientes: indexa páginas web , extrae información de ficheros PDF, PS, XLS, DOC., la frecuencia que este robot accede a un servidor web (sitio web) dependerá de un parámetro definido por Google como *PageRank* el cual indica la frecuencia de acceso del robot al sitio o página web. Sitios en el cual este parámetro es asignado como alto (PR10) tienen por resultado que el servidor web (sitio web) es

visitado en algún momento del día por el robot *GoogleBot* y como consecuencia de estos accesos la generación de eventos en los ficheros logs del servidor, eventos que hemos considerado o definido en esta tesis como "*ruido de eventos" o "ruido proveniente del web"*

Cualquier análisis de los eventos contenidos de los ficheros logs deberá necesariamente incluir la detección de los accesos de los robots al servidor y/o de los eventos relacionados con robots/spiders, agentes inteligentes, y otros programas que recorren el web en busca de información. Por ejemplo para determinar si el robot GoogleBot a visitado un sitio web; se deberá revisar los ficheros logs del servidor y buscar los registros o eventos almacenado en estos archivos en los que aparezca el string "googlebot"; por lo general el nombre del robot vendrá acompañado de otros identificadores como por ejemplo:

Servidor	Dirección IP
crawl1.googlebot.com	216.239.46.20
crawl2.googlebot.com	216.239.46.39
crawl3.googlebot.com	216.239.46.61
crawler2.googlebot.com	64.68.86.55
xxx.googlebot.com	x.y.z.w

1.4 Análisis de Ficheros Logs de servidores http y Analizadores de Logs

Hemos mencionado que las distintas fuentes primarias que almacenan objetos en el web o mas simplemente servidores web (los de FTP, *proxy-cache,* etc. son fuentes primarias) guardan, si están configurados para ello, registros de eventos por peticiones de recursos realizadas vía http, estos ficheros son almacenados en el disco duro en donde se aloja el sistema operativo, o en la ubicación seleccionada por el administrador del sistema. En estos ficheros logs se anotan todos los eventos que ocurren durante el funcionamiento normal del servicio web o sitio web. En ellos podemos encontrar por ejemplo el registro de las operaciones que han fallado, incluyendo algunas veces el motivo del fallo, estos ficheros de forma general son conocidos como error _ log.

```
68.142.212.170 - - [09/Jul/2006:04:06:24 -0400] "GET /de_imagespag/nestorjpg.jpg HTTP/1.0" 304 -
69.79.179.83 - - [09/Jul/2006:04:06:55 -0400] "GET /plataforma/aulavirtual/foro_s_p.php?llaveid=34271
HTTP/1.1" 200 4388
69.79.179.83 - - [09/Jul/2006:04:07:07 -0400] "GET /plataforma/aulavirtual/foro_s_p.php?llaveid=34270
```

Junto al error _ log, normalmente se activa el registro de accesos al servidor, en donde encontraremos las peticiones de recursos y objetos realizadas por usuarios reales o virtuales al servidor, por norma general estos ficheros adoptan el nombre de log de acceso o access_log. Es destacable mencionar en este punto que la captura de eventos puede ser realizada sobre el servidor web, sobre los servidores proxy, sobre la computadora del cliente o directamente sobre la red, lo cual agrega varias dimensiones al problema del análisis de los datos contenidos en este tipo de fichero, tarea que puede ser realizada por medio de programas especializados o analizadores de logs. Es común que en las implementaciones de soluciones web se consideren parámetros de calidad en cuanto a las velocidades de acceso al sitio, como disponibilidad de los recursos u otras característica, consideraciones que dan por resultado, la implementación de servidores proxy cache o aceleradores de navegación de paginas, como también la implementación de servidores frontales que replican las peticiones a servidores especializados o proxy reverso (reverse proxy). Es claro por tanto que para un mismo usuario y una misma petición el registro del eventos asociado a su solicitud puede ser almacenado en distintos servidores o en cada uno de los que respondan efectivamente a la petición.

Los datos y el formato de los mismos que vamos a poder encontrar los ficheros de logs *dependerán* como hemos planteado del tipo de servidor web utilizado en la implantación de la "solución web" a modo de ejemplo algunos de los posibles: Apache, Internet Información Server, AOLServer, Roxen, Caudium, THTTPD, Jetty, por norma general estos servidores web guardan los eventos en los ficheros logs utilizando el formato CLF o XCLF formatos descritos con anterioridad, en términos generales la información factible de ser almacenada en este tipo de ficheros podemos resumirla en:

Tabla 1.3. Resumen de Información de Datos contenidas en un registro Web Logs

1	Número de peticiones recibidas (*hits*).
2	Volumen total en bytes de datos y ficheros servidos.
3	Número de peticiones por tipo de fichero (por ejemplo, HTML).
4	Direcciones de clientes diferentes atendidas y peticiones para cada una de ellas.
5	Número de peticiones por dominio (a partir de dirección IP).
6	Número de peticiones por directorio o fichero.
7	Número de peticiones por código de retorno HTTP.
8	Direcciones de procedencia (*referrer*).
9	Navegadores y versiones de éstos usados.

En términos muy generales las aplicaciones para analizar logs, se diseñan para trabajar con los distintos formatos disponibles como son el NCSA, W3C u otros formatos empleados para el almacenamiento de registros de eventos en archivos ASCII o de texto; estas aplicaciones comerciales o de libre disposición generan diversas estadísticas a partir de la información rescatada de estos archivos, estadísticas que se basan principalmente en la información factible de obtener a partir de los datos almacenados, en la **Tabla 1.3** se entrega un resumen de la información contenida en los formatos más ampliamente utilizados que son el CLF y XCLF. A pesar de que las informaciones que podemos obtener del análisis de los ficheros de log están restringidas a los formatos de los eventos son numerosas, destacaremos en este punto unas cuantas que no se pueden obtener de forma directa por medio de un programa Analizador de Logs. De ellas, destacaremos la siguientes por su especial interés a los planteamientos del paradigma Web Usage Mining:

Tabla 1.4 Información Complicada de Obtener

1	Identidad de los usuarios, excepto en aquellos casos en los que el usuario se identifique por petición del servidor.
2	Número de usuarios. A pesar de tener el número de direcciones IP distintas, no podemos saber de forma absoluta el número de usuarios, y más si tenemos en cuenta la existencia de servidores *proxy-cache*. Una dirección IP puede representar a muchos usuarios
3	Un robot, araña u otro programa de navegación automático (por ejemplo, los usados por los buscadores como Google).
4	Un usuario individual con un navegador en su ordenador.
5	Un servidor *proxy-cache,* que puede ser usado por cientos de usuarios.
6	Datos cualitativos: motivaciones de los usuarios, reacciones al contenido, uso de los datos obtenidos, etc.
7	Ficheros no vistos.
8	Qué visitó el usuario al salir de nuestro servidor. Este dato quedará recogido en los *log* del servidor donde el usuario fue después del nuestro.
	Otras...

1.5 Errores comunes en la interpretación de los logs

La limpieza de los datos almacenados en los ficheros logs y la depuración de los mismos, procesos que en la minería de datos aplicada al web es conocida como Data Cleaning o Limpieza de Datos, presenta grandes problemas, como por ejemplo convertir la información almacenada en los ficheros logs en sesiones de usuarios, o en su defecto determinar episodios; los procesos asociados contienen grados de dificultad considerables y han sido abordados en diferentes trabajos científicos relacionados con el Web Usage Mining. En estas tareas se incluyen el mezclado de logs de fuentes diversas o múltiples servidores, remover los accesos a archivos gráficos y el análisis gramatical de los contenidos. Por otra parte la información puede

quedar registrada de forma parcial, lo que puede conducirnos a interpretaciones erradas o erróneas de los datos. Gran parte de dichas inconsistencias proceden del cache que realizan los propios navegadores, como también del que realizan servidores proxy-cache intermedios, o proxy-reverse. Estos problemas que se presentan en el Data Mining aplicado al Web no son abordados por la mayoría de los Analizadores de Logs, dado que estos tan solo consideran de manera general estadísticas asociadas a los datos contenidos en los logs. Dado que la información contenida en los ficheros de log presenta dificultades en su análisis e interpretación indicaremos a continuación algunos errores de interpretación comunes:

> *Los hits no equivalen a visitas. Una página puede generar más de un hit, ya que contiene imágenes, hojas de estilo, etc., que corresponden a otro hit.*
>
> *Las sesiones de usuario son fáciles de aislar y contar. Las sesiones, si no existe un mecanismo específico de seguimiento (cookies, etc.), se obtienen normalmente considerando todos los accesos provenientes de la misma dirección durante un lapso de tiempo consecutivo como perteneciente a la misma sesión. Esto no tiene en cuenta, ni la existencia de servidores proxy-cache, ni la posibilidad de que un usuario se mantenga un tiempo detenido (consultando otras fuentes de información, etc.).*
>
> *Datos como las medias de páginas por visita y las listas de páginas más visitadas se obtienen a partir de las sesiones de usuario. Dada la dificultad de calcular éstas, los valores obtenidos no tendrán excesiva fiabilidad. Además, la existencia de servidores proxy-cache tiene un efecto altamente perjudicial en las listas de páginas más visitadas. Precisamente al ser las más visitadas, tendrán más posibilidades de estar almacenadas en los servidores de cache.*
>
> *Es difícil deducir la ubicación geográfica de los usuarios a partir de las direcciones IP. En muchos casos, ubicaremos todo un bloque de direcciones en la ciudad donde tiene la sede principal el proveedor de servicios de Internet de un usuario, a pesar de que éste puede encontrarse en un lugar distinto.*

Las Tablas [1.5] y [1.6] contienen respectivamente un resumen con los principales Analizadores de Logs o los más populares, tanto comerciales como de libre disposición disponibles en diversos sitios web de Internet y una comparación de sus funcionalidades

Tabla 1.5 Analizadores de Logs.

Nombre	Multiplataforma Web Servers	Formatos de Logs Soportados	Tipos de Reportes	Notas
AWStats 6.3	Si	NCSA combinado/XLF/ELF común/CLF), WC3	Estadísticas Avanzadas	
SRG 1.3.1	No	CLF	Muy básicos	Dedicado a Proxy SRG
XpoLog 2.2	Si	Múltiples	Avanzadas	Permite poblar base de datos para análisis posteriores
Cronolog 1.6.2	No (Apache)	Solo entradas estándar CLF	Muy básicos	
ModLogAn 0.8.2	SI	Múltiples	Genera “Template”	
Analog 5.2.4				
Syslog-ng 1.4.9				
Calamaris 2.52				
Ktail 0.5.1				
Radius Report 0.3.b6				
Ftp logger 1.5				
Webalizer	No (Apache)	NCSA combinado/XLF/ELF Logs común/CLF)	Estadísticos	
HitBox				

Tabla 1.6 Tabla de Comparación de Log Analyzers

Features/Software’s	AWStats	Analog	Webalizer	HitBox
Version - Date	6.3 - December 2004	5.32 - April 2003	2.01-10 - April 2002	NA
Language	Perl	C	C	Embedded HTML tag
Available on all platforms	Yes	Yes	Yes	NA
Sources available	Yes	Yes	Yes	No
Price/Licence	Free/GPL	Free/GPL	Free/GPL	Free with adverts/Proprietary
Works with Apache combined (XLF/ELF)	Yes	Yes	Yes	NA
Works with Apache common (CLF) log format	Just some features	Just some features	Just some features	NA
Works with IIS (W3C) log format	Yes	Yes	Need a patch	NA
Works with personalized log format	Yes	Yes	No	NA
Analyze Web/Ftp/Mail log files	Yes/Yes/Yes	Yes/No/No	Yes/No/No	NA/No/No
Update of statistics from	command line (CLI) and/or a browser (CGI)	command line (CLI) and/or a browser (CGI)	command line	NA
Internal reverse DNS lookup	Yes	Yes	Yes	NA
DNS cache file	Static and dynamic	Static or dynamic	Static or dynamic	NA
Process logs spitted by load balancing systems	Yes	Yes	No	No
Report number of "human" visits	Yes	No	Yes	Yes
Report unique "human" visitors	Yes	No	No	Yes
Report session duration	Yes	No	No	Yes

Not ordered records tolerance and reorder for visits	Yes	Visits not supported	No	?
Statistics for visits are based on	Pages *****	Not supported	Pages *****	Pages *****
Statistics for unique visitors are based on	Pages *****	Not supported	Not supported	Pages *****
Report countries	From IP location or domain name	Domain name	Domain name	?
Report regions (US and Canada states)	Need Maxmind Regions database	No	No	No
Report cities	Need Maxmind Cities database	No	No	No
Report ISP	Need Maxmind ISP database	No	No	No
Report Organizations name	Need Maxmind Organizations database	No	No	No
Report hosts	Yes	Yes	Yes	Yes
Report WhoIs information's on hosts	Yes	No	No	No
Report authenticated users	Yes	Yes	No	No
Report/Filter robots (nb detected)	Yes/Yes (335**)	Yes / Yes (8**)	No/No	No/No
Report/Filter worms (nb of families detected)	Yes/Yes (5)	No / No	No/No	No/No
Report rush hours	Yes	Yes	Yes	Yes
Report days of week	Yes	Yes	Yes	Yes
Report most often viewed pages	Yes	Yes	Yes	Yes
Report entry pages	Yes	No	Yes	Yes
Report exit pages	Yes	No	Yes	Yes
Not ordered records tolerance and reorder for entry/exit pages	Yes	Entry/Exit not supported	No	?
Detection of CGI pages as pages (and not just hits)	Yes	Only if prog ends by a defined value	Only if prog ends by a defined value	Yes
Report pages by directory	No	Yes	No	No
Report pages with last access time/average size	Yes/Yes	Yes/No	No/No	No/No
Dynamic filter on hosts/pages/referers report	Yes/Yes/Yes	No/No/No	No/No/No	No/No/No
Report web compression statistics (mod_gzip,mod_deflate)	Yes	No	No	No
Report file types	Yes	Yes	No	No
Report by file size	No	Yes	No	No
Report OS (nb detected)	Yes (35)	Yes (29)	No (0)	?
Report browsers (nb detected)	Yes (98*)	Yes (9*)	Yes (4*)	Yes (<20*)
Report details of browsers versions	Major and minor versions	Major versions only	Major an minor versions	Major and minor versions
Report screen sizes	Yes	No	No	Yes
Report tech supported by browser for Java/Flash/PDF	Yes/Yes/Yes	No/No/No	No/No/No	No/No/No
Report audio format supported by browser for	Yes/Yes/Yes	No/No/No	No/No/No	No/No/No

Real/QuickTime/Mediaplayer				
Report search engines used (nb detected)	Yes (115***)	Yes (24)	No (0)	Yes (<20 ***)
Report keywords/keyphrases used on search engines (nb detected)	Yes/Yes (111***)	Yes/No (29***)	No/Yes (14***)	Yes/No (<20***)
Report external refering web page with/without query	Yes/Yes	No/No	No/Yes	Yes/No
Report HTTP Errors	Yes	Yes	Yes	No
Report 404 Errors	Nb + List last date/referer	Nb only	Nb only	No
Report 'Add to favorites' statistics	Yes	No	No	No
Other personalized reports for miscellanous/marketing purpose	Yes	No	No	No
Daily statistics	Yes	Yes	Yes	Yes
Monthly statistics	Yes	Yes	Yes	Yes
Yearly statistics	Yes	Yes	Yes	Yes
Benchmark with no DNS lookup in lines/seconds (full features enabled, with XLF format, cygwin Perl 5.8, Athlon 1Ghz)	5200****	39000****	12000****	NA No program to run
Benchmark with DNS lookup in lines/seconds (full features enabled, with XLF format, cygwin Perl 5.8, Athlon 1Ghz)	80****	80****	80****	NA No program to run
Analyzed data save format (to use with third tools)	Structured text file or XML	No database built Need full log scan for each report	Flat text file	Not possible
Export statistics to PDF	Experimental	No	No	No
Graphical statistics in one page / several / or frames	Yes/Yes/Yes	Yes/No/No	Yes/Yes/No	No/Yes/Yes

* This number is not really the number of browsers detected. All browsers (known and unknown) can be detected by products that support user agent listing (AWStats,Analog,Webalizer,HitBox). The 'browser detection feature' and number is the number of known browsers for which different versions/ids of same browser are grouped by default in one browser name.

** AWStats can detect robots visits: All robots among the most common are detected, list is in robotslist.txt (250Kb). Products that are not able to do this give you false information, above all if your site has few visitors. For example, if you're site was submitted to all famous search engines, robots can make 500 visits a month, to find updates or to see if your site is still online. So, if you have only 2000 visits a month, products with no robot detection capabilities will report 2500 visits (A 25% error !). AWStats will report 500 visits from robots and 2000 visits from human visitors.

*** AWStats has url syntax rules for the most popular search engines (that's the 'number detected'). Those rules are updated with AWStats updates. But AWStats has also an algorithm to detect keywords of unknown search engines with unknown url syntax rules.

**** Most log analyzers have poor (or not at all) robots, search engines, os or browsers detection capabilities and less features (no or poor visits count, no filter rules, etc...).
It is not possible to add all AWStats features to other log analyzers, so don't forget that benchmarks results are for 'different features'.
For this benchmark, I did just complete Webalizer and Analog robots or search engines databases with part of AWStats database. So Webalizer config file was completed with this file, Analog config file was completed with this file. Note that without this very light add (using default conf file), Webalizer speed is 3 times faster, Analog is 15% faster).
Benchmark was made on a combined (XLF/CLF) log record on an Athlon 1GHz.
You must keep in mind that all this times are without reverse DNS lookup. DNS lookup speed depends on your system, network and Internet but not on the log analyzer you use. For this reason, DNS lookup is disabled in all log analyzer benchmarks. Don't forget that DNS lookup is 95% (even with a lookup cache) of the time used by a log analyzer, so if your host is not already resolved in log file and DNS lookup is enable, the total time of the process will be nearly the same whatever is the speed of the log analyzer.

***** Some visitors use a lot of proxy servers to surf (ie: AOL users), this means it's possible that several hosts (with several IP addresses) are used to reach your site for only one visitor (ie: one proxy server download the page and 2 other servers download all images). Because of this, if stats of unique visitors are made on "Hits", 3 users are reported but it's wrong. So AWStats, like HitBox,

considers only HTML pages to count unique visitors. This decrease the error (not totally, because it's always possible that a proxy server download one HTML frame and another one download another frame).

[F_1] Fuente : **AWStats official web site** http://awstats.sourceforge.net/docs/awstats_compare.html

De las tablas anteriores es posible deducir que la mayoría de estos productos responden a las preguntas básicas como: quien, donde, cuanto, cual; respondiendo con estadísticas asociadas, no establecen vínculos o relaciones entre distintos "objetos" almacenados en los registros de un archivo logs como por ejemplo: los usuarios que visitaron la página ***A*** también compraron el producto ***B***. Es claro por tanto que las motivaciones de diversos trabajos científicos realizados con la finalidad de aplicar técnicas de minería de datos a los datos contenidos en estos archivos web logs es una necesidad real dado las crecientes necesidades de las empresas y personas naturales que utilizan el web como plaza de intercambio de información, bienes y servicios.

CAPÍTULO 2: *WEB MINING*

El camino hacia un "web Inteligente", necesariamente requiere del desarrollo de herramientas y sistemas que sean capaces de manejar situaciones de manera similar a la vida cotidiana de una persona con el mundo real, dominio en donde la ambigüedad y la incertidumbre son características naturales o propias del ser humano, las redes neuronales y la lógica difusa se han constituido en instrumentos fundamentales en nuestros días para el desarrollo de aplicaciones que permitan la construcción de sistemas artificiales inteligentes que sean capaces de emular en parte el comportamiento humano permitiendo manejar incertidumbre, consultas ambiguas y la obtención de conocimiento. Un web inteligente requiere de soluciones que puedan integrarse por ejemplo a los navegadores existentes y potenciar su funcionamiento con el objetivo que puedan ser empleados por personas comunes que navegan por Internet facilitando su trabajo; el aporte del web inteligente debería estar reflejado en el desarrollo de herramientas que permitan a un usuario común búsquedas exitosas o con relevancia en el tiempo mínimo o el empleo de un lenguaje natural como interfaz de navegación. Las empresas en cambio requieren de un exhaustivo análisis del comportamiento comercial de su sitio web, dado que en la actualidad su diseño y comportamiento son modernas herramientas de competitividad y de diferenciación comercial. Una empresa típica que apoya su funcionamiento en un sitio web corporativo necesita conocer por ejemplo el como se relacionan los distintos anuncios, servicios, productos vendidos, páginas visitadas, con los "clientes virtuales" del sitio web, rescatando la información de navegación y petición de recursos desde los distintos servidores que implementan los sistemas de comercio electrónico y otros sistemas tanto al interior de la empresa como fuera de esta. La tarea consiste en desarrollar y emplear técnicas que sustituyan el ***análisis de datos dirigido a la verificación estadística*** por un enfoque de ***análisis de datos orientado al descubrimiento de conocimiento***, con la finalidad de determinar perfiles, patrones de navegación, preferencias, conducta o comportamiento, como también el uso de los recursos, servicios y objetos de la solución web. Desde el punto de vista de la empresa u organización es posible definir ***conocimiento*** como la información que posee valor, es decir aquella que representa algún particular punto de interés y es potencialmente útil para la empresa, este "conocimiento" es almacenado como conjuntos de datos en diversas fuentes tanto al interior de la empresa como en el web o Internet. El problema por tanto consiste principalmente en emplear algunas técnicas de análisis para grandes

volúmenes de datos que permitan su agrupamiento automático, clasificación, asociación de atributos, detección de patrones, detección de datos anómalos, desviaciones o tendencias.

Los mecanismos mas habitualmente empleados para la obtención de conocimiento a partir de grandes volúmenes de información consisten en el empleo de herramientas de software o sistemas constituidos por varios programas que analizan de manera automática un universo o conjunto de entrenamiento, buscando tendencias, desviaciones o en otras palabras la clasificación supervisada o no supervisada de datos para posteriormente ser particularizados a casos o situaciones especificas, las cuales pueden ser visualizadas a posteriori, proceso denominado como Data Mining (DM) o Minería de Datos, que en principio es una fase dentro del denominado Descubrimiento de Conocimiento en Bases de Datos (KDD).

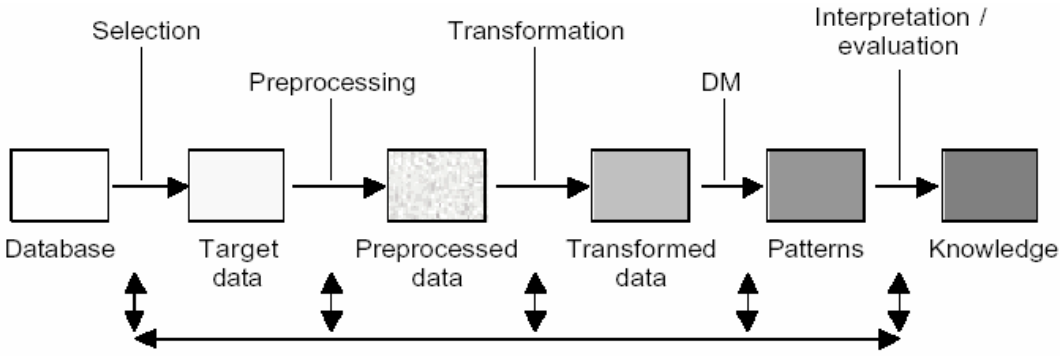


Figura 2.1 Fases de un Proceso KDD

2.1 Data Mining: Una mirada desde el punto de vista del conocimiento

Desde la mirada de las organizaciones publicas como también de las empresas privadas, se puede definir conocimiento como la información que posee valor para ella, es decir aquella información que permite establecer relaciones por medio de las cuales se puedan satisfacer demandas de los clientes o mercado a los cuales están dirigidos los objetivos comerciales de la organización. La Minería de datos puede entenderse como el proceso del *"descubrimiento de conocimiento en bases de datos"* o como *"la extracción no trivial de información implícita, previamente desconocida y potencialmente útil a partir de los datos"* y corresponde a una de las fases del denominado descubrimiento de conocimiento en bases de datos o *KDD: Knowledge Discovery in Database*. Es

importante destacar que la organización por si sola no es capaz de crear conocimiento, sino que son las personas las que establecen las percepciones, razonamientos y relaciones que permiten alcanzar el "saber de la organización" entendiendo como saber al proceso que involucra el análisis del medio ambiente y su modelación por medio de observaciones, con el objetivo de codificar estas observaciones como un conjunto de información de datos, información que posteriormente se transformara en conocimiento al momento de ser analizada.

Estableceremos algunas precisiones en torno a los elementos que constituyen la cadena de información, la cual esta constituida esencialmente por los datos. En el capítulo 1analizamos los datos desde el punto de vista de su naturaleza o generación, en esta oportunidad los observaremos desde el punto de vista del conocimiento que es posible inferir a partir de la información subyacente a los mismos. Mencionábamos en el párrafo anterior que el saber de una organización consiste en la modelación conceptual de su realidad en un *dominio de conocimientos*, constituyéndose o generándose este dominio con datos provenientes de distintas fuentes. El proceso o cadena de procesos que permitirán descubrir conocimiento se entenderá por tanto como la transformación de los datos almacenados, los cuales no tienen un sentido o significado por si mismos, y de los cuales solo tenemos una referencia contextual, por tanto es necesario agruparlos, ordenarlos y analizarlos para ser interpretados a fin de obtener y entender la información potencial o subyacente que contienen. De acuerdo con esto podemos definir:

> *Dato: punto en el espacio el cual cuenta con referencias absolutas y carece de sentido.*
>
> *Información: Conjunto o reunión de datos con algún tipo de asociación de acuerdo a una medida o atributo lo cual permite generar discernimiento en torno a ellos.*
>
> *Conocimiento: Provee el fundamento de cómo cambia la información contenida en los datos, esto puede ser percibido como los patrones de comportamiento bajo un contexto, es decir una relación de relaciones.*
>
> *Verdad o Certeza: La verdad o certeza se fundamenta en la evidencia y la evidencia nos es otra cosa que la presencia patente de la realidad*

Los conceptos anteriores reflejan los procesos asociados a los datos en su transformación en conocimiento, conceptos asociados a procesos que son esencialmente jerárquicos o consecutivos, siendo posible interpretarlos gráficamente como se indica en la Figura 2.2

Figura 2.2 Modelo de Capas Jerárquico para la Extracción de Conocimiento

Este modelo de capas de especialización jerárquicas, se define con la finalidad de asociar estas jerarquías o estratos a procesos de software que permitan intercambiar datos o información a través de puntos de acceso conocidos y estandarizados. A partir de lo anterior podemos definir ciertos grados de especialización en cuanto al modelo e implementación del o los procesos de extracción de conocimiento desde un gran conjunto de datos sean estos estructurados o no. Cuando los datos son clasificados por medio de alguna herramienta de la manera en que se transformen o conviertan en información no trivial, estamos en el proceso que se ha denominado como *Minería de Datos*.

Una de las principales tareas de la *Minería de Datos* o *Data Mining* es la de agrupar datos, identificando grupos o reuniones de estos por medio de algunas técnicas, como por ejemplo semejanza o atributos en el contexto de un conjunto de datos multidimensional, o el agrupamiento de clases de datos identificando los lugares de dispersión o de acumulamiento, y con esto se descubren patrones de comportamiento o distribuciones de tipos o clases de patrones, permitiendo disponer de aquellos ordenados o clasificados en el contexto o marco de referencia de una persona a fin de transformarlos en conocimiento; este conocimiento una vez validado y empleado con algún objetivo generara una verdad o certeza la cual puede ser entendida como una interpretación de la realidad o del modelo conceptual definido a priori.

2.2 Conceptos y Procesos Involucrados en Minería de Datos

Hemos mencionado con anterioridad que la Minería de Datos corresponde a *"la extracción no trivial de información implícita, previamente desconocida y potencialmente útil a partir de los datos"* o en otras palabras responder preguntas como ¿ Qué elementos contiene él diagnóstico de un médico?, ¿

La sentencia de un juez corresponde a la gravedad de los hechos?, ¿Cuál es el comportamiento de nuestros clientes? . Estas preguntas tienen una respuesta genérica común: la verdad o certeza, conocer la verdad o tener certeza de algo no es una tarea fácil, de hecho la credibilidad que le otorgamos a nuestros propios conocimientos admite varios grados o niveles de calificación o de "atributos" según sea el punto de vista que se analice el concepto de conocimiento, estos niveles de abstracción los hemos definido como: la duda, la opinión, la certeza, el escepticismo y la verdad.

> ***La Duda**: en la duda se fluctúa entre la negación o la afirmación de una determinada proposición.*
>
> ***La Opinión**: corresponderá a la adhesión a una proposición sin tener en cuenta el grado de verdad o falsedad de la misma.*
>
> ***La Certeza**: se fundamenta en la evidencia y la evidencia no es otra cosa que la interpretación de la realidad.*
>
> ***El Escepticismo**: corresponde a la postura en la cual se niega la capacidad de alcanzar la verdad, o en otras palabras corresponde al hecho en que una vez analizados los eventos se concluye que no hay certeza.*
>
> ***La Verdad**: será construida a partir de la certeza y el escepticismo*

Los conceptos anteriores nos permitirán definir las tareas necesarias a aplicar a un conjunto de datos con la finalidad de extraer conocimientos de estos, y pueden ser traducidos a procesos o tareas que forman parte de proceso de Descubrimiento de Conocimientos en donde la Minería de Datos dentro del proceso de obtención de conocimiento. Estas tareas pueden ser agrupadas en:

> *Estimación y Predicción: Consistirá en la revisión de los atributos de un conjunto de datos, entidades, procesos y roles, asignándoles valores cuantitativos (la duda). El concepto "predicción" se utilizara en conjunto con la valoración de un atributo, a fin de definir un dominio de conocimiento (o datos) sobre el cual se aplicará el estudio, como por ejemplo predecir el costo de un proceso productivo o el comportamiento de un grupo de usuarios.*
>
> *Categorización: La categorización consiste en el examen de los atributos de un dominio de datos o entidades y asignarles una categoría o clase predefinida (la opinión).*

Descubrimiento de Asociaciones y Agrupamiento: El descubrimiento de asociaciones consiste en identificar que atributos están asociados con otros en un determinado dominio; el agrupamiento se realizara como una consecuencia o resultado del descubrimiento de estas asociaciones o relación de relaciones y tiene por objetivo formar grupos o particiones de objetos clasificados basándose en técnicas como distancia, homogeneidad, heterogeneidad (la certeza)

Visualización de los datos y Exploración Visual de los datos: La visualización corresponde a la compleja tarea de presentar los resultados basándose en premisas de certeza y escepticismo, donde será necesario manejar "bases de datos de conocimiento o de teorías" las cuales pueden formase a partir de reglas determinadas del análisis de los datos o del experto humano o de algún software que las defina. La exploración visual de los datos se desprende como una extensión interactiva de la visualización pudiéndose establecer distintos escenarios (dominios vs reglas) que permitan alcanzar mejores grados de conocimiento el cual puede ser empleado como apoyo a la toma de decisiones (la verdad).

Las etapas y definiciones anteriores están asociadas a un proceso de Descubrimiento de Conocimiento desde el punto de vista del paradigma web inteligente, y están asociadas a conceptos lingüísticos con la finalidad de una mejor interpretación contextual de las técnicas necesarias de aplicar al conjunto o dominio de datos seleccionado para su estudio. Esta hipótesis permite independizar al proceso de KDD y la etapa de Minería de Datos de la estructuración o no de los datos y así poder aplicar sus métodos a los datos contenidos en B_Internet, es evidente que no es posible aplicar de forma directa las técnicas de minería de datos a los datos almacenados en el web, dado que estas técnicas han sido desarrolladas para ser aplicadas a bases de datos que contienen datos estructurados o a bodegas de datos (data warehouse) disponibles en la organización como repositorio de los mismos, luego por tanto siempre será necesario modificar o adaptar las técnicas seleccionadas al problema a resolver.

El Descubrimiento de Conocimiento o KDD, es el proceso de descubrimiento automático de patrones previamente desconocidos, reglas o asociaciones, y otras regularidades contenidas en un dominio de datos o de conocimientos, la Minería de Datos en cambio expresa el descubrimiento de patrones de un dominio de datos previamente preparados de una forma especifica, la minería de datos es habitualmente usada como sinónimo para KDD, sin embargo y en rigor es simplemente una fase dentro de un proceso KDD. Algunas consideraciones importantes a tener en cuenta en un proceso KDD o Minería de Datos:

El descubrimiento de conocimiento es un proceso, *y no una respuesta del sistema KDD para la acción de un usuario en un tiempo dado.*

Como cualquier otro proceso, tiene su ambiente, sus fases, y marcha bajo ciertas suposiciones y ciertas restricciones.

Las suposiciones básicas necesarias son que existe un dominio de conocimiento (una base de datos) con su diccionario de datos o descripción de estos (formato ECL), y que el usuario quiere descubrir algunos patrones de este dominio.

Del dominio de conocimiento se podrán establecer conjuntos de datos, selecciones o restricciones a fin aplicar técnicas de Data Mining para el ordenamiento, clasificación y otras técnicas, con el objetivo de descubrir patrones

La salida del proceso de Data Mining es en general un conjunto de patrones en donde no todos son útiles, luego es necesario escoger aquellos no triviales o interesantes, para posteriormente analizarlos, interpretarlos o evaluar todo lo que estos patrones descubrieron.

Los patrones seleccionados e interpretados representarán el conocimiento descubierto del dominio datos escogido.

Las fases de un proceso KDD o descubrimiento de conocimiento son:

Selección *(estimación o predicción, la duda)*

Esta fase consiste en el caso de una base de datos en seleccionar de esta el conjunto de aquellos sobre los cuales se aplicara el proceso KDD. En nuestro caso de estudio esta base queda representada por B_Internet, luego un proceso de Web KDD considerara en esta etapa la construcción del dominio de datos o de conocimientos sobre los cuales se aplicara el proceso.

Preprocesamiento *(categorización-la opinión).*

La siguiente fase, el preprocesamiento también conocida como limpieza de datos tiene por objetivo eliminar el ruido de los datos, como por ejemplo el ruido web de los ficheros logs. En esta etapa se manipulan los datos erróneos, inexactos, imprecisos, conflictivos, excepcionales, como también las ambigüedades de los mismos. Adicionalmente en esta fase es posible integrar datos de distintas fuentes como por

ejemplo ficheros logs provenientes de distintos servidores. El propósito es construir un conjunto de datos y prepararlos en términos de secuencias específicas generadoras de información de datos, el resultado es un conjunto de información procesada.

***Transformación** (categorización-la opinión).*

La siguiente fase es la transformación de la preinformación procesada de una forma adecuada o entendible para las técnicas, algoritmos u otras herramientas que serán empleadas en la fase siguiente o minería de datos.

Minería de Datos** (data mining-la certeza**)

La minería de datos corresponde a la aplicación de diferentes tecnologías que resuelven problemas típicos de agrupamiento automático, clasificación, detección de patrones o actividades que son realizadas sobre el conjunto de datos transformados en busca de patrones, guiado por el tipo de conocimiento que desea ser descubierto.. La salida de la fase de minería de datos es, en general, un conjunto de patrones.

***Interpretación y Evaluación** (el escepticismo)*

*La meta de interpretar y evaluar todo lo que los patrones descubrieron es conservar sólo aquellos patrones que son interesantes y útiles para el usuario, descartando los que no son útiles a los objetivos del estudio. EL resultado obtenido es un conjunto de patrones los cuales representan el descubierto conocimiento **(la verdad)***

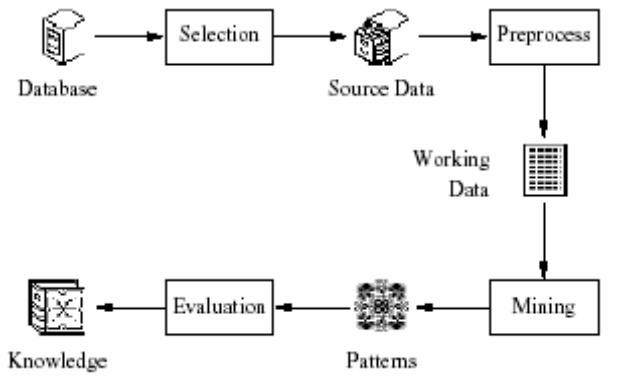


Figura 2.3 Fases de un Proceso de Minería de Datos

2.2.1 Técnicas para Minería de Datos

La minería de datos es una herramienta ampliamente utilizada cuando se trata de analizar datos con el objetivo de obtener conocimiento de los mismos; esta tesis considera dentro de sus objetivos la obtención de características derivadas del dominio de datos constituido por los denominados ficheros web logs , luego considerando este objetivo se abordaran con más detalle la generosa definición de Minería de Datos ampliamente adoptada por la comunidad KDD que indica: *el proceso de búsqueda y extracción de información implícita, previamente desconocida y potencialmente útil a partir de los datos almacenados en una base de datos.* Esta amplia definición, implica la selección, exploración y modelación de grandes de volúmenes de datos con el objetivo de descubrir patrones desconocidos, e información finalmente comprensible desde bases de datos grandes.

La minería de datos usa a una amplia familia de métodos computacionales que incluyen análisis estadístico, los árboles de decisiones, las redes neuronales artificiales, las reglas de inducción, el clustering, el refinamiento de conjuntos, la lógica difusa, los algoritmos genéticos, los sistemas neurodifusos y la visualización gráfica. Estas herramientas de minería de datos han estado disponibles por mucho tiempo, siendo los avances mas significativos en la actualidad los aportados por las redes neuronales y las técnicas de la visualización de datos. La extracción de patrones o características derivadas de un conjunto de datos es la principal motivación de cualquier actividad de minería de datos y eso se ocupa en esencia esta etapa de un proceso KDD. Formalmente, un *patrón* puede ser definido de acuerdo Fayyad et. al. [50]:

Una declaración S en L que describe relaciones entre un subconjunto dado Fs de hechos, de un conjunto dado de hechos F con alguna certeza Cx, donde S es la semejanza más sencilla para la enumeración de todos los hechos en F.

Las tareas o métodos de la minería de datos se usan para extraer patrones de conjuntos de datos grandes o voluminosos. Una taxónomia de las diversos métodos o técnicas empleados por la minería de datos es planteada por Shaw M.J. et. al [164] indicando que estas pueden dividirse de manera amplia en cinco grupos o categorías resumidos en la Figura. 2.4. Esta taxonomía refleja el papel emergente de visualización de los datos considerando a esta tarea como un área de interés o especialización. Las diferentes tareas de la minería de datos son agrupadas en estas categorías a merced del tipo de conocimiento extraído por las técnicas empleadas.

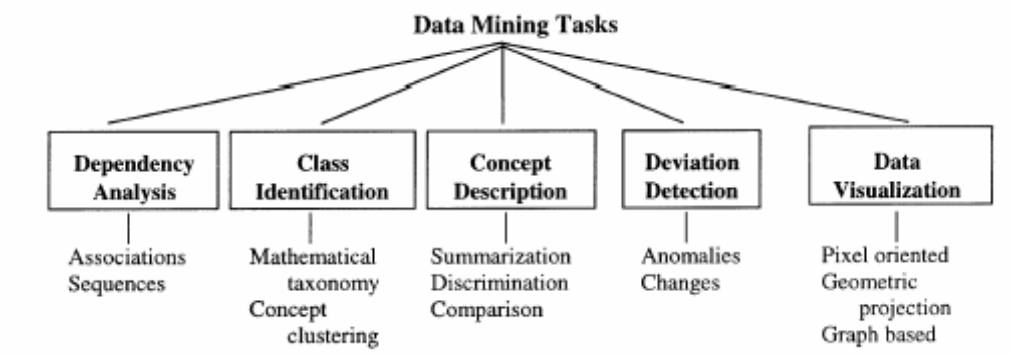


Figura 2.4 Taxónomia de los Procesos de Minería de Datos

El análisis de la dependencia

El tipo primario de conocimiento de dependencia corresponde a la asociación entre conjuntos de artículos (items) indicados con alguna confianza especificada mínima. Éste es también llamado " análisis de la canasta de mercado " y nos reporta la relación entre productos diferentes comprados por un cliente. Este tipo de conocimiento puede ser útil para elaborar estrategias sobre comercializar les para promocionar productos que tienen relaciones de dependencia en las mentes de los clientes. Por ejemplo, puede determinar que los clientes que compran pañales, compran cerveza al mismo tiempo.

La identificación de clase

La identificación de clases agrupa a los clientes en clases que están definidas por adelantado. Hay dos tipos de tareas de la identificación de clase - la taxonomía matemática y el agrupamiento de conceptos o clustering. Por ejemplo la taxónomia matemática de algoritmos de agrupamiento se orientan a determinar clases con máxima o minina similitud. Por ejemplo, una tienda de comestibles puede clasificar a sus clientes basados en su ingreso o después de las cantidades de compra y luego puede apuntar a sus esfuerzos mercadotécnicos consecuentemente. Un inconveniente de esta tarea es su incapacidad para usar información de background o experiencia. El clustering de conceptos vence esta limitación y determina grupos según la similitud de atributo así como también la cohesión conceptual definido por el dominio de conocimiento. Los usuarios proveen el conocimiento de dominio identificando características útiles del clustering. Por ejemplo, basados en los ficheros logs (proxy logs) es

posible determinar como los usuarios navegan por Internet, pudiéndose clasificar estos en términos lingüísticos como "usuarios serios", "usuarios de entrenamientos" o "usuario x"

La descripción de conceptos

La descripción de conceptos es una técnica para agrupar a los clientes basados en el dominio de conocimiento y la base de datos, sin definiciones forzadas de los grupos. La descripción del concepto puede estar usada para resumir, discriminar, comparaciones de mercadeo o establecer conocimiento del cliente. La reducción de datos es un proceso característico de este método y permite establecer un subconjunto de datos característico que es interesante con relación al dominio de conocimiento. Técnicamente, el resumen de un concepto *A* es realizado escudriñando todos las tuplas que satisfacen A y procesando todos los campos del registro. Usando identificación de clases, un comercializador puede enterarse de las características del cliente agrupándolas según su ocupación, ingreso u otras clases. La discriminación por ejemplo describe calidades lo suficiente como para diferenciar registros de una clase de otra. La comparación describe la clase de un modo que facilita comparación y análisis con otros registros.

La detección de la desviación

Las desviaciones son útiles para el descubrimiento de anomalías y cambios. Las anomalías son cosas o eventos que son diferentes a la normalidad. Por ejemplo, si comparamos a un grupo de vendedores comerciales similares e identifiqué esos que se destacan sobre el promedio, ya sea en una tendencia positiva o una negativa, luego será necesario ajustar los grupos. Las anomalías pueden ser detectadas por el análisis del significado, desviaciones estándar o típicas, y medidas de volatilidad de los datos. Además de las anomalías, las variables o los atributos pueden tener valores significativamente diferentes de las transacciones previas para el mismo cliente o el grupo de clientes. Una compañía de tarjetas de crédito puede encontrar un alza rápida en las compras a crédito de un cliente individual. Este cambio en el comportamiento puede ser un resultado de un cambio en el estado del cliente, y no necesariamente un fraude. Así, la confirmación de este cambio (o lo que cambió) está hecha después de la investigación y el conocimiento está actualizado.

La visualización de datos

El software de visualización de datos permite mirar complejos patrones que pueden ser obtenidos por los procesos de minería de datos, asignándolos a vistas tridimensionales, asociándoles colores u otras técnicas con el objetivo de transformarlos en "objetos visuales" que puedan ser interpretados por el usuario o experto (humano). Estos software tiene capacidades de manipulación avanzadas sobre los objetos visuales permitiendo por ejemplo cortar en rodajas, rotarlos con el objetivo de observar mejor sus detalles. La visualización de datos puede ser usada aisladamente o en colaboración con otras tareas como el análisis de dependencia. Kim [85] provee un análisis elaborado de las técnicas de visualización para bases de datos grandes y clasifica estas técnicas de visualización en: técnicas de proyección orientada al pixel, técnicas de proyección geométrica y técnicas basadas en gráficas.

2.2.2 Algoritmos de Minería de Datos y Técnicas Asociadas

A continuación presentamos algunas técnicas de minerías de datos estudiados a lo largo de la elaboración de esta memoria.

Árboles de Decisión o de Clasificación

Los árboles de decisión o clasificación son técnicas inductivas que son empleadas para descubrir reglas de clasificación para un atributo seleccionado de un conjunto de datos, subdividiendo sistemáticamente la información contenida en el conjunto o dominio.

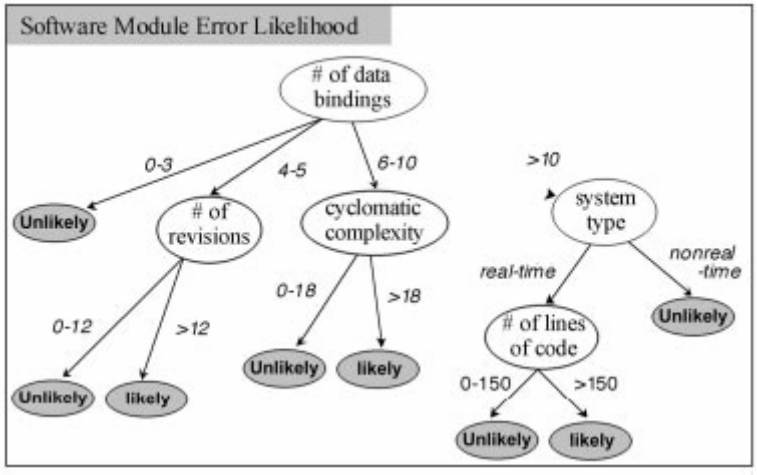


Figura 2.5 Ejemplo de un Árbol de Clasificación. [DACS]

Los algoritmos usados para construir árboles de clasificación, emplean un atributo seleccionado como ejemplo junto a un valor asignado y buscan encontrar esos atributos y esos valores segregando el máximo de registros de datos del conjunto de datos en cada nivel del árbol. El razonamiento es el siguiente mientras más información tiene un árbol más pequeñas serán las ramas de este. Quinlan J.R. [144] propuso quizás uno de los algoritmos mas populares el cual definió como ID3.

1. *Seleccione un atributo desde la raíz del árbol, haga ramas o nodos para todos los valores que este atributo puede tener;*

2. *Use el árbol generado para clasificar el conjunto de entrenamiento. Si todos los ejemplos en un nodo particular de la hoja tienen el mismo valor para el atributo, están clasificados. Este hoja o nodo será etiquetado con este valor. Si todas las hojas están etiquetadas con un valor, entonces el algoritmo termina.*

3. *De otra manera etiquete el nodo con el atributo que no cumple desde el camino de este hasta la raíz y regrese al paso 2.*

Este algoritmo trabaja por medio de la selección de un atributo para cada rama del árbol, esta selección se realiza por medio de heurísticas que tienen por objetivo realizar esta selección. Una vez determinado el atributo y su valor de referencia, el conjunto de datos puede dividirse en dos subconjuntos que definen dos caminos al cual se le aplica una técnica determinada para la correcta clasificación de acuerdo a algún criterio, como por ejemplo la formula derivada de la teoría de la información:

$$I(p_i, n_i) = -\frac{p_i}{p_i + n_i}\log_2\left(\frac{p_i}{p_i + n_i}\right) - \frac{n_i}{p_i + n_i}\log_2\left(\frac{n_i}{p_i + n_i}\right)$$

En donde la información necesaria para clasificar un elemento de un conjunto de datos ***S*** usando un atributo ***A*** esta dada por:

$$E(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p + n} I(p_i, n_i)$$

Esto es la medida ponderada de la información necesaria para clasificar elementos en subárboles *Si* definidos por el atributo A valorizado en

{V1,V2,…,Vv}.

Descubrimiento de Asociaciones

Esta técnica supone la extracción de información por coincidencias en el conjunto de datos. El descubrimiento de conocimiento tiene lugar cuando estas coincidencias son previamente desconocidas, poco triviales, e interpretable por un experto de dominio. Generalmente aplican herramientas de la teoría de probabilidades para estimar la coincidencia de un evento sobre el conjunto de datos. Con respecto a la minería de datos usualmente se utilizan *funciones de interés* para determinar que tan *interesantes* son los datos. Una de esta funciones viene dad por la siguiente formula:

$$Interestingness_{I}(A,B) = |P(A|B) - P(A)| = \left| \frac{P(A|B)}{P(B)} - P(A) \right| = \left| \frac{P(A|B) - P(A)P(B)}{P(B)} \right|,$$

Figura 2.6 Función de Interés para descubrimiento de Asociaciones.

El interés puede emplear adicionalmente técnicas estadísticas u otros criterios para analizar los datos y determinar que relaciones son datos útiles y no triviales.

Técnicas de Agrupamiento o Clustering de Decisión o de Clasificación

El Clustering o Agrupamiento esta entre las técnicas precursoras de la Minería de Datos, el concepto es muy fácil de entender, por ejemplo si uno quiere ubicar rápidamente un utensilio de cocina los ordena por su funcionalidad, los platos separados de los vasos, las cucharas de lo tenedores etc, agrupándolos en lo posible en lugares separados. Lo mismo ocurre con los datos. En el clustering no existen etiquetas predefinidas, por lo que el tipo de aprendizaje que usa es no _ supervisado. El objetivo fundamental del clustering es el de descubrir las relaciones de similaridad y disimilaridad que existen en un determinado conjunto de datos. Para ello, básicamente, lo que hace es identificar regiones densas y dispersas en el espacio definido por el conjunto de datos, aplicando conceptos de distancia entre los registros de datos. Una distancia

puede ser por ejemplo la cantidad de modificaciones realizadas a los registros, la métrica por tanto quedara definida en *#número* de modificaciones y será una diferencia numérica entre los registros seleccionados, como se indica en la Figura 2.7..

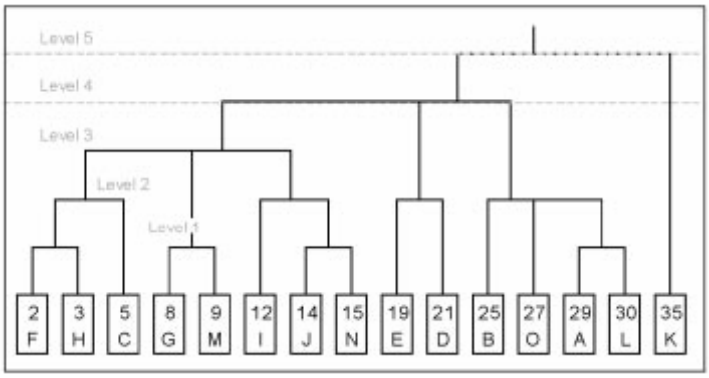


Figura 2.7 Distancia o Métrica Simple

Las distancias más empleadas corresponde a la Euclidiana y la Manhattan, las cuales extendidas a espacios N dimensiónales que es el caso de registros con múltiples atributos toman la formas siguientes:

$$\text{Euclidean}(B, D) = \sqrt{(a_1(B) - a_1(D))^2 + \ldots + (a_N(B) - a_N(D))^2}$$

$$\text{Manhattan}(B, D) = |a_1(B) - a_1(D)| + \ldots + |a_N(B) - a_N(D)|$$

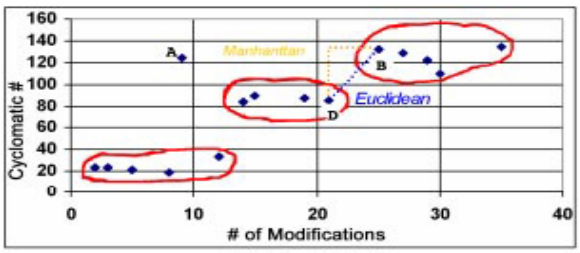


Figura 2.8 Distancias Euclidianas y Manhattan en dos Dimensiones

Algoritmos de Agrupamiento o Clustering

Es posible clasificar los algoritmos en dos tipos jerárquicos y no jerárquicos, por otra parte la metodología empleada para agrupar datos puede ser descrita de acuerdo a sus procedimientos o

técnicas asociadas por ejemplo es posible establecer agrupamiento de clases por particiones o grados de pertenencia, método que emplea lógica difusa como herramienta u otros métodos; por densidad o cercanía, por mapas autoorganizados provenientes.

Los algoritmos que emplean técnicas no jerárquicas requiere de algunas decisiones previas como por ejemplo el numero de cluster deseados o el mínimo de cercanía requerido al cluster, esto limita de alguna manera al experto en el dominio dado que no puede escoger el mejor cluster basado en su opinión subjetiva. El algoritmo mas común utilizado corresponde al conocido como el método K-means propuesto por MacQuen [113] este trabaja de la forma siguiente:

1. *Seleccione un numero K para el cluster deseado*
2. *Tome un registro para ser el centroide de cada uno de los grupos seleccionados*
3. *Experimente con el conjunto de datos, asignándolos al cluster mas cercano*
4. *Recalcule el centroide (o la media) para el nuevo cluster*
5. *Repita pasos 3 y 4 hasta que haya reubicación mínima de registros entre los grupos.*

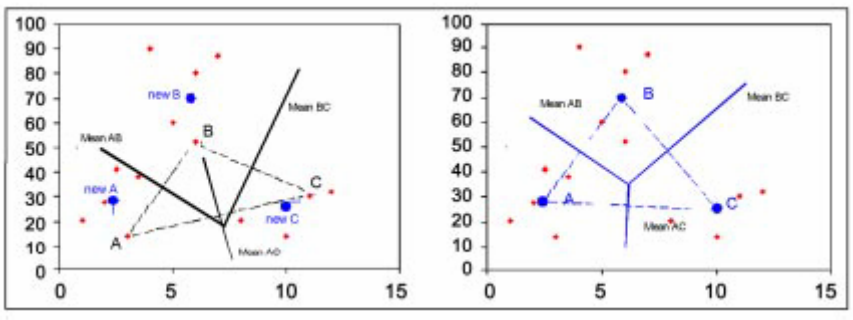


Figura 2.9 Método de las K-Means

Es importante destacar que el numero de cluster establecidos por el método K-Means es fijado a priori en el paso uno del algoritmo, este es un pequeño inconveniente cuando se obtiene un numero diferentes de cluster al deseado. Un algoritmo jerárquico en cambio crea jerarquías de cluster del grupo de datos; estos algoritmos trabajan aglutinando o dividiendo cluster, el

algoritmo de mayor uso corresponde al método de aglutinación el cual opera de manera general de la forma siguiente:

1. *Cree un grupo para cada registro en el conjunto de datos.*
2. *Mezcle por cercanía o proximidad formando grandes grupos.*
3. *Repita el proceso hasta que un único grupo sea formado*

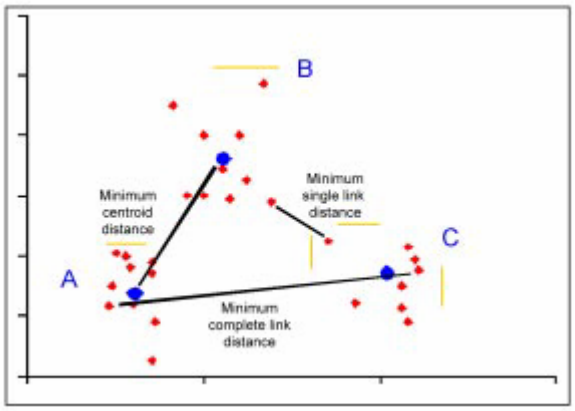


Figura 2.10 Mediciones usadas por el Método de Aglutinado

Las distancias pueden ser expresadas por cuatro vías: método del enlace simple, método del enlace completo, método del centroide y método de Ward

1. Método del enlace simple: *la distancia entre dos grupos es igual a la distancia entre los dos registros más cercanos en ellos.*
2. Método del enlace completo: *la distancia entre dos grupos es igual a la distancia entre los dos registros más distantes en ellos.*
3. Método de centroide: *la distancia entre dos grupos es igual a la distancia entre sus centroides.*
4. Método de Ward: *La distancia total entre los registros del grupo y su centroide es computada para cada una de las posibles mezclas;.y se anexa con el total menor resultando una distancia es seleccionada como la siguiente mezcla.*

Redes Neuronales Artificiales

Los primeros estudios para modelar una neurona artificial fueron realizados en 1943 por el neurofisiólogo Warren McCulloch y el matemático Walter Pitts [112], estudios qué se consideraba al cerebro como una máquina de calcular u organismo computacional, definieron en su trabajo un modelo de neurona al cual denominaron "neurona formal" el cual fue representado por medio de un circuito eléctrico semejante al de la Figura. 2.11 En este modelo de implementación física, las entradas corresponden a funciones de voltajes *V(x),* el peso de la entrada esta relacionado con una resistencia acoplada de manera serial; y el proceso de activación de la salida esta dado por la función de transferencia referida al voltaje V_o establecida como se indica en la figura.

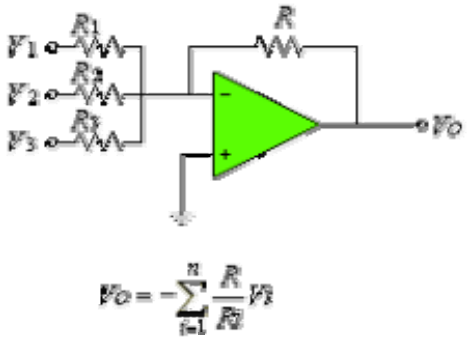


Figura 2.11 Neurona Artificial Implementada como un Circuito Analógico

En el modelo de la Figura. 2.12 la neurona artificial puede ser representada como un elemento procesador o caja negra, en donde sus entradas son valores ***x(n)***, y cada una de estas posteriormente es multiplicada por un peso ***w(n)*** asignado a la misma, el cual representa la fuerza sináptica de la conexión.

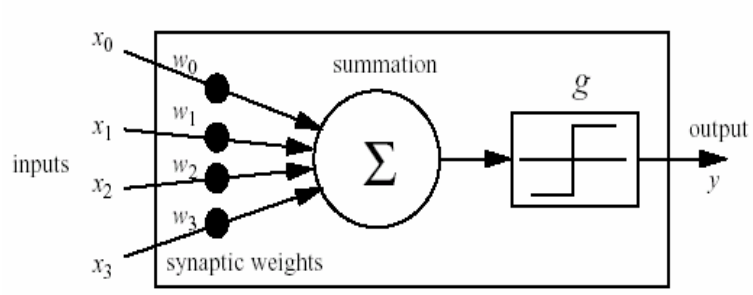


Figura 2.12 Neurona Artificial Básica Modelo Matemático

Una red neuronal artificial tiene una topología básica como la indicada en la Figura. 2.13 esta topología se basa en grupos o capas de neuronas dispuestas jerárquicamente, las cuales están conectadas entre si. Estas capas de neuronas reciben entradas las procesan y luego activan sus salidas que pasan a ser las entradas de nuevas capas de neuronas artificiales

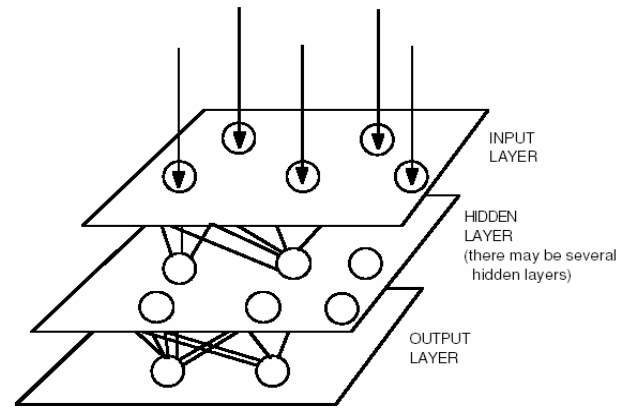


Figura 2.13 Arquitectura Simple de una Red Neuronal

La arquitectura de conexión de neuronas artificiales pasa por tanto a ser un elemento clave al momento de querer realizar algunas aplicaciones como por ejemplo: funciones de aprendizaje, detección de patrones, procesamiento de señales, reconocimiento de texto u otras. De la arquitectura y topología dependerá su eficiencia, rendimiento y aplicación en la solución de problemas del mundo real. La forma en que una neurona se conecta a otras neuronas artificiales, afectara o influenciara el funcionamiento de la neurona que la precede o antecede según sea el caso; inhibiendo o activando la líneas que las interconectan; por ejemplo es posible que el diseñador desee que una neurona inhiba a otra en la misma capa de red o en su defecto que inhiba una de sus salidas. La inhibición de una neurona por medio de otra que pertenece a la misma capa se conoce como"inhibición lateral" ; la inhibición de una salida como por ejemplo el reconocimiento de un carácter o símbolo cuya probabilidad sea superior a un rango se conoce como "inhibición por competencia" o simplemente "inhibición".

El aprendizaje de la red puede ser realizado de manera ***supervisado*** o ***no-supervisado***. Un entrenamiento supervisado consiste en suministrar a la red las entradas como también las salidas deseadas, realizándose un proceso de comparación de la entrada con la salida obtenida, propagándose el error resultante hacia atrás del la arquitectura o sistema, hasta que el sistema se ajuste a la salida deseada. De acuerdo al método en como se ajustan los pesos, el aprendizaje supervisado puede ser clasificado en: por corrección de error, por refuerzo o estocástico.

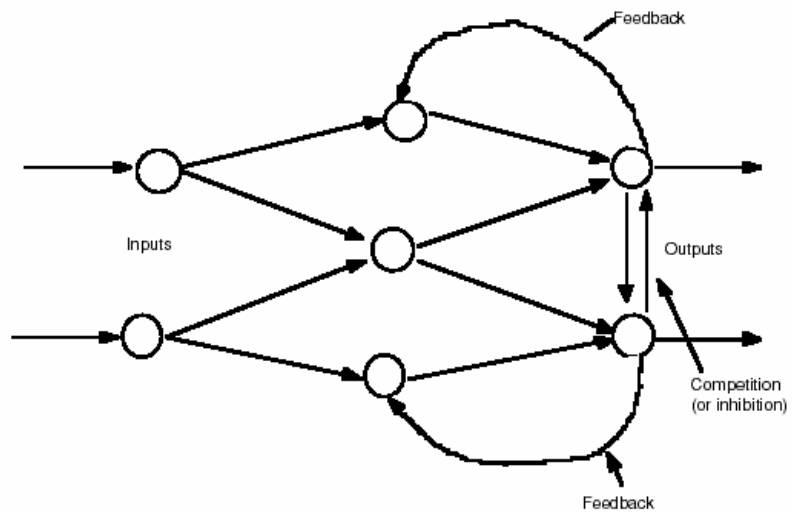


Figura 2.14 Red Neuronal Simple con Feedback and Competición

Redes Neuronales Artificiales: Aprendizaje Supervisado

En el caso de aprendizaje supervisado el ajuste de pesos por medio de la corrección del error a evolucionado y sus métodos son evidentemente de inspiración abstracta matemática, más de que inspiración biológica, por ejemplo un tratamiento básico del error esta dado por la expresión siguiente:

$$\Delta w_{ji} = \alpha \gamma_i (d_j - \gamma_j)$$

en donde :

Δw_{ji} es la variación del peso de la conexión que une dos neuronas;

γ_i es el valor de salida;

d_j es el valor de salida deseado para una neurona j;

γ_j corresponde al valor obtenido en la salida de a neurona j, y

α es el factor que regula la velocidad del aprendizaje.

Una expresión más compleja y depurada que la anterior la cual considera el error cuadrático mínimo como algoritmo de ajuste de los pesos o regla delta, método planteado por B. Widrow y M.Hoff. [186] corresponde a la expresión:

$$Error_{global} = \frac{1}{2P} \sum_{k=1}^{p} \sum_{j=1}^{n} (\gamma_j^{(k)} - d_j^{(k)})^2$$

donde:

p es el número con la información o datos que la red debe aprender
n corresponde al número de neuronas de salida.

Una contribución destacada en el proceso de aprendizaje desde un punto de vista biológico fue planteada por Donald Hebb en 1949 [67] en los términos siguientes:

*"cuando un axon de una celda **A** esta suficientemente cerca como para conseguir excitar una celda **B**, y en forma persistente toma parte en su activación; algún proceso de crecimiento o cambio metabólico tiene lugar en una o ambas celdas, de tal forma que la eficiencia de A, cuando la celda a activar es B, aumenta ."*

El planteamiento o regla de Hebb puede ser interpretado por medio de un modelo artificial como: si una neurona recibe una entrada de otra neurona y ambas están en actividad el peso entre estas neuronas es aumentado.

Matemáticamente esto puede ser expresado asignando un valor numérico a las conexiones de salida de las neuronas, asignando un valor positivo cuando la neurona esta activa y un valor negativo cuando su salida esta inhibida. Como Hebb indica que al estar ambas neuronas activas el peso es aumentado, esto puede ser expresado como:

$$\Delta w_{if} = \gamma_i \gamma_j$$

Redes Neuronales Artificiales: Aprendizaje No Supervisado

En el caso de un entrenamiento no supervisado, a la red solo se le suministran las entradas pero no las salidas deseadas, la red por tanto no recibe ninguna información del entorno que le indique que la salida generada como una respuesta a las entradas es o no correcta; esta característica es conocida como una capacidad de la red para auto-organizarse. Este tipo de redes neuronales artificiales que se ***auto-organizan***; permiten al carecer de un "supervisor" para determinar de las entradas suministradas regularidades, características, clasificaciones o categorías; elementos que están presentes de manera implícita en las entradas presentadas o de la información de datos que estas contienen.

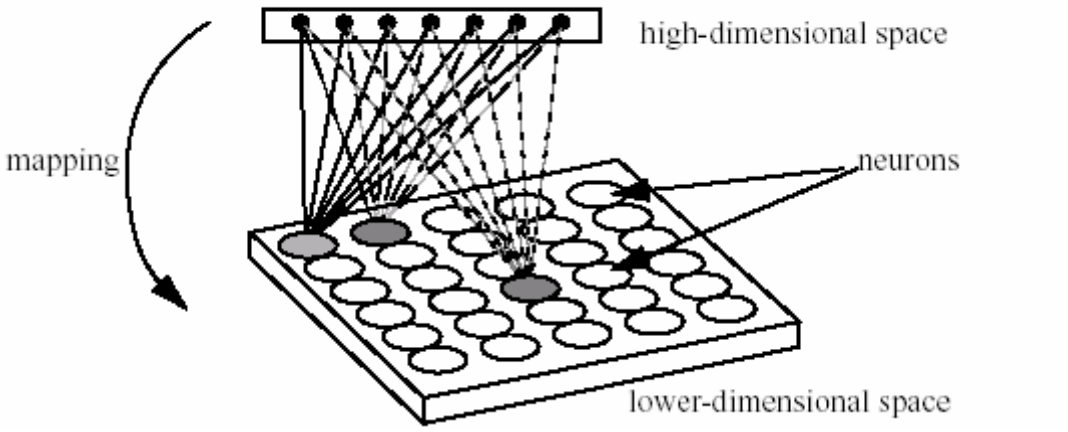
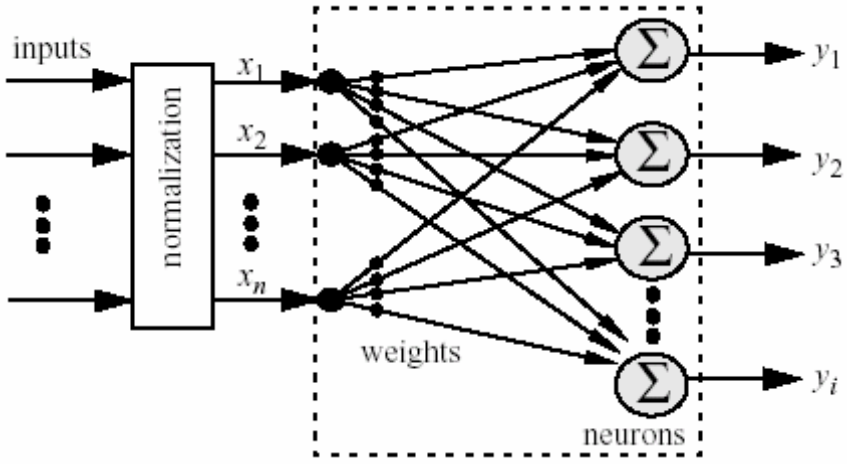


Figura 2.15 Modelo de Mapas Topológicos o de Características Cognitivas

Un investigador destacado en aplicar estas capacidades de las neuronas biológicas a un modelo artificial de red neuronal es Tuevo Kohonen [89], plantea en el trabajo referido un modelo de red neuronal con capacidad para formas mapas de características similar es a lo que ocurre en el cerebro, la hipótesis de su trabajo esta basada en el supuesto de que un estimulo o información de entrada a una arquitectura de una red neuronal artificial en la cual su diseño y funcionalidad es conocido, seria suficiente para la formación mapas topológicos de la información recibida en la entrada.



$$y_i = \sum_{j=1}^{n} m_{i,j}\, x_j,$$

Figura 2.16 Arquitectura de red de T. Kohonen

La arquitectura esta basada en dos capas (layer) con N neuronas de entrada y M de salida, en donde cada una de las N neuronas de entrada se conecta a las M neuronas de salida en una topología de conexión hacia delante; la arquitectura considera inhibición lateral implícita entre las neuronas de salida, esta característica significa que cada neurona influenciara a su vecina cercana, y esta "influencia" será cuantificada por medio de la distancia entre ellas, si esta distancia es muy pequeña significa que la neurona esta muy alejada de aquella en referencia.

Respecto al aprendizaje del modelo de Kohonen, se distinguen dos etapas: una etapa de aprendizaje *stand alone* (*off line o fuera de servicio*) y una etapa de funcionamiento. Es destacable por tanto mencionar que la red previamente a su entrada en operación se auto-organiza, para posteriormente entrar en pleno funcionamiento por medio de un algoritmo de aprendizaje que puede ser el siguiente:

1. *Inicializar los pesos w(n)*
2. *Presentar los vectores de entrada* E_k
3. *Se activa salida en neurona "vencedora" por competitividad, esto significa el patrón más similar a la información presentada en las entradas. Para esto se calculan las distancias o diferencias entre los vectores entrada y salida de cada una de las neuronas de salida empleando la técnica de la distancia euclidea o expresiones similares, como por ejemplo eliminando la raíz:*

$$d_j = \sum_{i=1}^{N} (e_i^k + w_{ji})^2 \quad 1 \le j \le M$$

Una vez localizada la neurona vencedora se actualizan los pesos entre las neuronas de entrada y la neurona vencedora; y así como las conexiones entre las neuronas de entrada y las neuronas vecinas a la vencedora; con esta técnica de diseño se consigue asociar la información de las entradas a un zona de neuronas (Figura. 2.17) en la capa de salida; esta zona puede ser reducida por medio del ajuste de los pesos por iteraciones de la red.

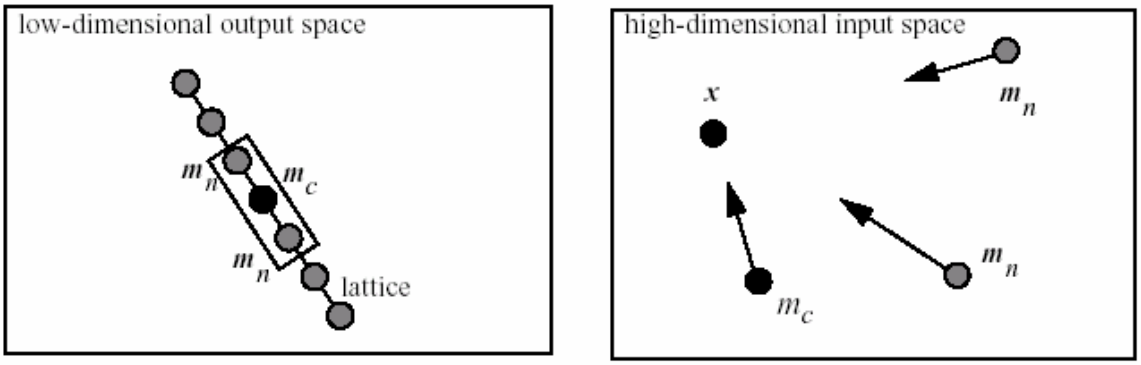


Figura 2.17 Vecindad a la neurona vencedora

El proceso debe ser repetido volviendo a presentar los patrones E_k a la entrada empleando como referencia un coeficiente de aprendizaje estimativo ó $\alpha(t)$ el cual es un valor de adaptación o ganancia de la red, cuyo valor es:

$$0 \leq \alpha(t) \leq 1$$

Al repetir el proceso se van construyendo zonas $N_c(t_{1....n})$ de manera consecutiva reduciéndose su espacio dimensional en cada iteración; en la Figura. 2.18 se grafica como evoluciona el mapa de características para una serie de iteraciones

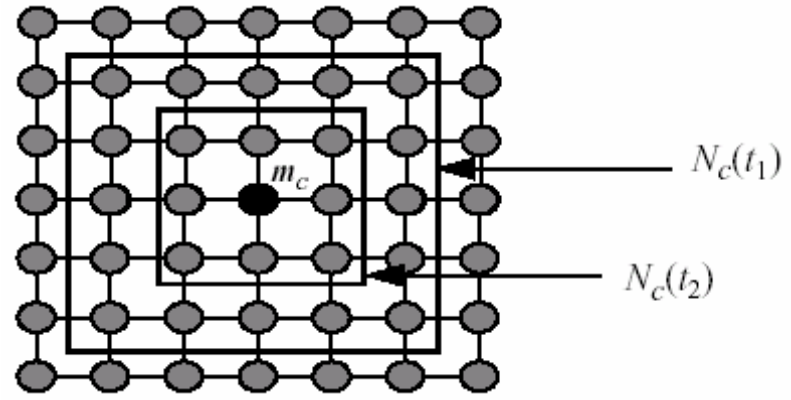


Figura 2.18 Vecindad o Zona de evolución de una Arquitectura de Kohonen

La importancia del modelo de red que se auto-organiza o redes auto-asociativas, radica en el hecho de que una arquitectura artificial que emplea aprendizaje no supervisado, podría en teoría realizar interpretaciones de la realidad y generar por ejemplo el concepto de memoria

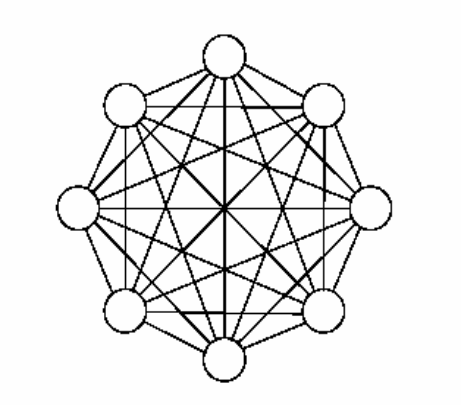


Figura 2.19 Arquitectura de Red de J. Hopfield

Las redes de tipo asociativas pueden ser empleadas como memorias y ser aplicadas en el problema de reconocer patrones, destacándose el trabajo de J. Hopfield presentado en 1982 a la Nacional Academy of Sciences, trabajo en que da un fuerte impulso a este tipo de redes, presentando un modelo conceptual de red basado en principios de la física como la energía y los sistemas físico estadísticos.

El concepto básico de modelamiento empleado por Hopfield para establecer una memoria asociativa es interpretar que el sistema evoluciona en movimientos sucesivos hasta alcanzar su estado de ajuste o estabilización, estado en el cual el patrón presentado es idéntico al almacenado o semejante en un cierto grado con respecto al almacenado; lo cual permitiría su posterior tratamiento y correcto ensamblaje. Este concepto puede ser referido como *"memoria asociativa de Hopfield"*

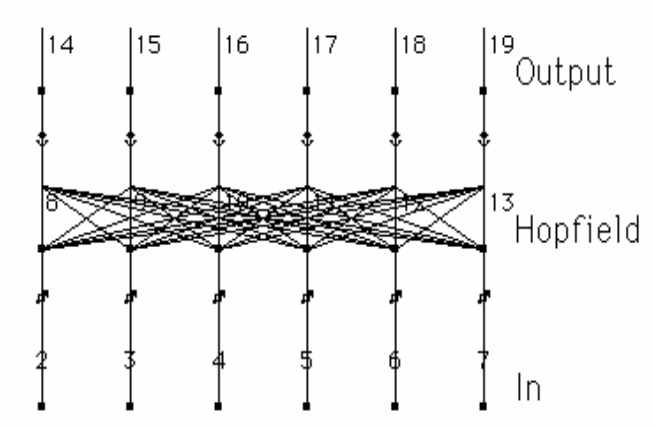


Figura 2.20 Memoria de Contenidos Direccionable (Hopfield Network)

Función de Energía: Una característica funcional importante de las redes neuronales artificiales corresponde conceptualmente al algoritmo que se emplea para que la red alcance su convergencia hacia un valor estable en la salida, es en esta área en que el aporte de Hopfield al desarrollo de las redes neuronales artificiales es destacable. Lo significativo de su trabajo esta en la aplicación por parte de este, de modelos matemáticos para ser empleados en el establecimiento de los puntos de estabilidad de una red neuronal; Hopfield expresa en su trabajo una función o modelo matemático para representar los posibles puntos de equilibrio de la red estableciendo una similitud formal con la energía mecánica clásica para construir esta función, función denominada a partir de este concepto como "función de energía de la red". La función de energía para una red Hopfield discreta puede ser expresada como:

$$E = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1, i\neq j}^{N} w_{ij} s_i s_j + \sum_{i=1}^{N} \theta_i s_i$$ o $$E = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1, j\neq i}^{n} w_{ij} y_i y_j - \sum_{i=1}^{n} x_i y_i + \sum_{i=1}^{n} \theta_i y_i$$

en donde

w_{ij} *: es el peso de la conexión entre las neuronas* **i** *y* **j;**

s_i *: valor en la salida de la neurona* **i;**

s_j *: valor en la salida de la neurona* **j;**

θ_i *: umbral de la función de activación de la neurona* **i**

La función de energía puede interpretarse como una superficie que presenta una cierta cantidad de valores mínimos, esta superficie puede ser asociada por ejemplo con el mapa de una cadena montañosa, en donde los mínimos pueden ser sindicados con los valles o planos de esta superficie. Se puede deducir de esta arquitectura de red que una vez almacenados *N* patrones o información, los posibles estados de estabilidad de la red son también *N* y estos son los mínimos de la función de energía; estados estables que se alcanzan al presentar información de datos en las entradas de la red y esta "ajusta" sus valores hasta obtener un mínimo, generando de esta manera una salida estable.

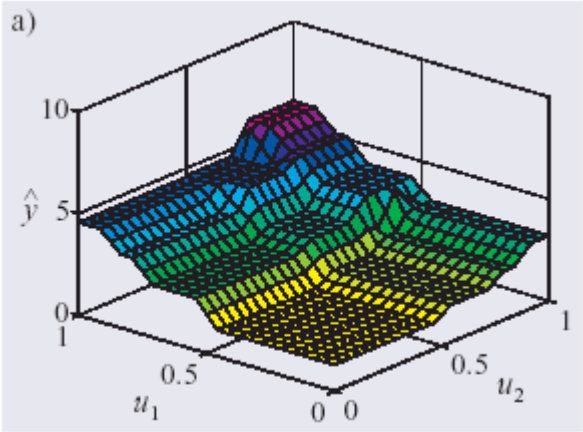


Figura 2.21 Función Energía de una Red Hopfield

Aprendizaje en Red Hopfield: Una diferenciación importante de esta red respecto a otras del tipo auto-organizadas es que el aprendizaje no corresponde tan solo al ajuste de los pesos sinápticos; en una red Hopfield los pesos de las conexiones sinápticas se pueden calcular. Estos pesos se mantienen fijos durante el proceso de aprendizaje *stan-alone* de los patrones, cambiando solamente el estado de las neuronas de cero a uno y viceversa.. El calculo del peso de una conexión cualquiera, *wij* y su conexión simétrica *wji*, esta expresado por la expresión siguiente:

$$w_{ij} = \sum_{q=1}^{Q} (2 * e_{qi} - 1) * (2 * e_{qj} - 1), \quad i <> j$$

en donde Q el número de patrones y e_{qt} la entrada a la neurona *Ni*.

Esta función es recomendable emplearla cuando los patrones que se han de aprender no son muy semejantes unos a otros; y en el caso supuesto que el número de ceros y unos son similares para todos los patrones presentándose un probabilidad de que la red entre en un estado indeterminado Con respecto al número de ceros y unos, el *umbral* de cada neurona puede utilizarse para regular este problema, distinguiéndose así dos casos posibles:

> *Si hay más ceros que unos, el umbral tiene que disminuirse, dado que existe una probabilidad más alta para la neuronas de adoptar el estado inactivo (0)*
>
> *Si la cantidad de unos en mucho mayor que ceros; el umbral tiene que incrementarse, porque las neuronas tienen una probabilidad más alta para hacerse activas (1).*

2.2.3 Conjuntos Optimizados Reducidos

Conjuntos Reducidos Optimizados, es una técnica que consiste en determinar qué subconjuntos de registros de datos provee la mejor caracterización para las entidades que están siendo evaluadas. Trabaja por descomposiciones sucesivas del conjunto de datos de entrenamiento en subconjuntos. A cada paso de descomposición un atributo es seleccionado y los registros que tienen el mismo valor que el atributo seleccionado, es extraído del set de entrenamiento, creándose un subconjunto nuevo. Esto está hecho recursivamente en los subconjuntos hasta que el criterio de terminación o meta sea encontrado. La predicción y la clasificación luego pueden estar hechas basadas en el valor medio de la variable dependiente en los subconjuntos resultantes.

Esta técnica puede ser mejor comprendida observando la Figura. 2.22; en esta se muestra parte de un modelo [b] para la predicción del esfuerzo de mantenimiento para el cual la confianza de los mantenedores en la tarea a ser realizada es ALTA. Del conjunto de datos de entrenamiento por medio de la variable lingüística "Confianza = alta", se extrae el suconjunto1, luego de este conjunto se extrae el tipo de tarea de mantenimiento usando la variable lingüística "Tipo de tarea = correctiva" formando el subconjunto2. El subconjunto2 se responsabiliza por el criterio de terminación y la predicción de esfuerzo queda determinada en base al contenido de este subconjunto.

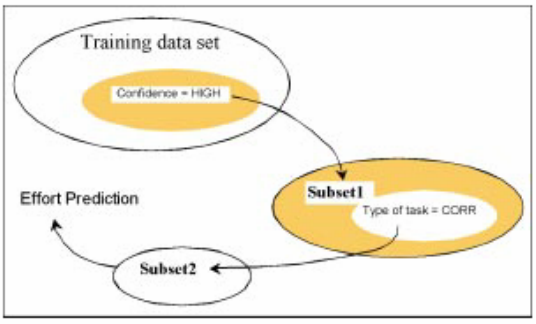


Figura 2.22 Método Jerárquico Aplicando Conjuntos Optimizados Reducidos

2.2.4 Redes Bayesianas de Creencias

Son redes gráficas que representan relaciones probabilística entre variables. Este método puede ser empleado por expertos del dominio para articular sus creencias acerca de las dependencias entre diferentes procesos y los atributos del un producto. Con estas creencias se puede propagar consistentemente el impacto del atributo conocido prediciendo adelante las probabilidades de resultados inciertos.

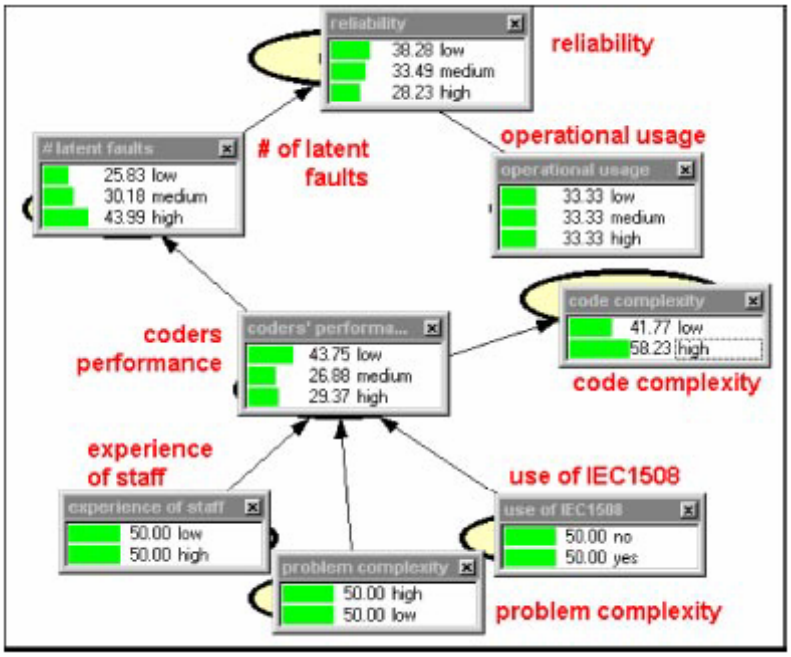


Figura 2.23 Ejemplo de una Red Bayesiana: Predicción de Fiabilidad de Software

En este ejemplo cada nodo tiene asignado una tabla de probabilidad para mapear las entradas y distribuirlas en las salidas, el arco corresponde a la representación de la influencia de un atributo

sobre otro y puede ser definido por un umbral; a su vez la tablas de probabilidad puede ser establecida de manera subjetiva por el experto o de manera objetiva por medio de la aplicación de métricas.

2.2.5 Minería de Datos Visual Redes o Visualización

Puede ser entendido como los métodos o técnicas empleados para mapear volúmenes de datos multidimensionales sobre la pantalla bidimensional de un computador. Una buena técnica de visualización apunta al usuario y tiene por objetivo presentar vistas tridimensionales, bidimensionales asignándoles colores u otras técnicas que permita una mejor interpretación de los patrones minados.

Minería de Datos Multimedia

Este tipo de minería de datos apunta al proceso de descubrimiento de conocimiento a partir de datos multimedia que son almacenados en bases de datos multimediales. Este tipo de datos corresponden a: videos, audio, imágenes, grafica y otras representaciones multimedia.

En general este tipo de minería de datos se emplean la mayoría de las técnicas anteriormente expuestas destacándose aquellas provenientes de la Computación Flexible la cual incluye las herramientas siguientes: Lógica Difusa, Redes Neuronales Artificiales, Algoritmos Genéticos y métodos específicos asociados al reconocimiento de patrones. Estos sistemas es posible clasificarlos en:

Sistemas de Procesamiento de Imágenes (PIQ, Web Seek- Meta Seek, Multimedia Miner.)

Sistemas para Aplicaciones Científicas (Diamond Eyes, Adam, Isis)

Sistemas de Búsquedas de Video (InforMedia, VideoQ, VideoClip, Multimedia Análisis.)

2.3 Data Mining Aplicado al Web

Esta tesis a definido su ámbito de trabajo en la evaluación de la información almacenada en los denominados web logs, fijando sus objetivos en el desarrollo de un nuevo algoritmo a ser utilizado en la etapa de preprocesamiento de datos de un proceso de descubrimiento de conocimientos orientado al web o Web Mining. Las diferencias entre el Data Mining y el Web

Mining planteadas y justificadas con anterioridad sugieren el desarrollo de nuevas herramientas de análisis o la modificación de las ampliamente utilizadas dentro de un proceso de Data Mining, herramientas de las cuales se han escogido las mas significativas, sus métodos o técnicas de análisis y los algoritmos empleados siendo presentadas en el desarrollo de este capítulo. El objetivo de conocer un proceso de Data Mining aceptando este concepto como sinónimo del proceso denominado descubrimiento de conocimiento en bases de datos o KDD: Knowledge Discovery in Database, corresponde a establecer las pequeñas o grandes diferencias entre el Web Mining y el Data Mining, diferencias que sugieren como hemos mencionado el desarrollo de nuevos algoritmos de análisis de datos con la finalidad de ser aplicados a un proceso de descubrimiento de conocimiento en el Web, motivación principal del desarrollo de esta tesis. Según Sankar, et. al [170] "El Web Mining puede ser definido en grandes rasgos como el descubrimiento y análisis de información útil desde el denominado Word Wide Web", y plantea que las tareas del Web Mining pueden ser ordenadas de acuerdo a la Figura 2.25.

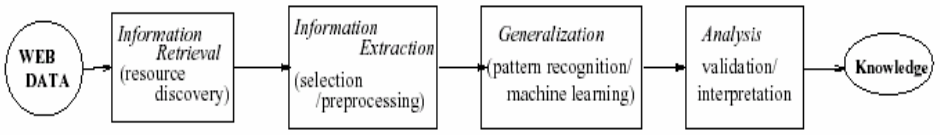


Figura 2.24 Principales Tareas del Web Mining

De acuerdo con estos planteamientos podemos decir que la minería de datos aplicada al web o *Web Mining* se ocupara de analizar o generalizar datos semiestructurados, no-textuales y otros provenientes del web con el objetivo de la extracción de patrones o características derivadas de un conjunto de datos o dominio de conocimientos previamente definido tarea de cualquier actividad de minería de datos como también la del web mining. La presencia de datos no estructurados determina que los patrones a determinar pueden estar contenidos en registros de datos que contienen información imprecisa o incompleta, como por ejemplo los *ficheros web logs,* este tipo de ficheros contienen conocimiento que esta mezclado con referencias a tiempo, direcciones ip, peticiones de recursos, datos que generalmente son imprecisos o incompletos, lo que sugiere de forma inmediata el empleo de herramientas como la *Lógica Difusa* la cual puede ser utilizada para la generación de conjuntos de datos depurados con el objetivo de la obtención de conocimiento, objetivo de este trabajo de tesis. Esta fuente primaria de información o dominio de conocimiento formada por los ficheros log a generado fuertes áreas de

investigación y de desarrollo de soluciones de software tendientes a dar solución a las necesidades de empresas virtuales y usuarios del web.

En esta parte del trabajo nos abocaremos a presentar las principales técnicas relacionadas al Web Mining que son útiles a los objetivos de preparar conjuntos de datos para que sean empleados en un proceso de descubrimiento de conocimiento, fijaremos nuestra atención en los conceptos más ampliamente aceptados por distintos grupos de trabajo que emplean el web como fuente de investigación; esta tarea será abordada sin seguir un orden cronológico o de influencia entre técnicas complementarias sino mas bien se presentaran las distintas visiones del mundo de la investigación asociada al web.

En primer lugar recordaremos de manera general la estructura de un fichero web logs como el de la Figura. 2.25 estos archivos registran las páginas web solicitadas, la fecha y hora de la petición y otros objetos que hayan sido dirigidos a una dirección ***ip***, dirección que a menudo corresponde a un usuario individual. Los "logs" poseen diferentes formatos el cual depende entre otros parámetros del sistema operativo del servidor; algunos de estos formatos típicos son: ***Common Log***, ***Extended Log Format*** o el formato generado por los servidores web de Microsoft denominado ***W3C Extended Log File***.

#	IP Address	Userid	Time	Method/ URL/ Protocol	Status	Size	Referrer	Agent
1	123.456.78.9	-	[25/Apr/1998:03:04:41 -0500]	"GET A.html HTTP/1.0"	200	3290	-	Mozilla/3.04 (Win95, I)
2	123.456.78.9	-	[25/Apr/1998:03:05:34 -0500]	"GET B.html HTTP/1.0"	200	2050	A.html	Mozilla/3.04 (Win95, I)
3	123.456.78.9	-	[25/Apr/1998:03:05:39 -0500]	"GET L.html HTTP/1.0"	200	4130	-	Mozilla/3.04 (Win95, I)
4	123.456.78.9	-	[25/Apr/1998:03:06:02 -0500]	"GET F.html HTTP/1.0"	200	5096	B.html	Mozilla/3.04 (Win95, I)
5	123.456.78.9	-	[25/Apr/1998:03:06:58 -0500]	"GET A.html HTTP/1.0"	200	3290	-	Mozilla/3.01 (X11, I, IRIX6.2, IP22)
6	123.456.78.9	-	[25/Apr/1998:03:07:42 -0500]	"GET B.html HTTP/1.0"	200	2050	A.html	Mozilla/3.01 (X11, I, IRIX6.2, IP22)
7	123.456.78.9	-	[25/Apr/1998:03:07:55 -0500]	"GET R.html HTTP/1.0"	200	8140	L.html	Mozilla/3.04 (Win95, I)
8	123.456.78.9	-	[25/Apr/1998:03:09:50 -0500]	"GET C.html HTTP/1.0"	200	1820	A.html	Mozilla/3.01 (X11, I, IRIX6.2, IP22)
9	123.456.78.9	-	[25/Apr/1998:03:10:02 -0500]	"GET O.html HTTP/1.0"	200	2270	F.html	Mozilla/3.04 (Win95, I)
10	123.456.78.9	-	[25/Apr/1998:03:10:45 -0500]	"GET J.html HTTP/1.0"	200	9430	C.html	Mozilla/3.01 (X11, I, IRIX6.2, IP22)
11	123.456.78.9	-	[25/Apr/1998:03:12:23 -0500]	"GET G.html HTTP/1.0"	200	7220	B.html	Mozilla/3.04 (Win95, I)
12	209.456.78.2	-	[25/Apr/1998:05:05:22 -0500]	"GET A.html HTTP/1.0"	200	3290	-	Mozilla/3.04 (Win95, I)
13	209.456.78.3	-	[25/Apr/1998:05:06:03 -0500]	"GET D.html HTTP/1.0"	200	1680	A.html	Mozilla/3.04 (Win95, I)

Figura 2.25 Fichero de Registro de Eventos Típico Ordenado o Web Server Log

Estos ficheros que almacenan eventos producidos por la navegación de usuarios sobre a las diversas fuentes primarias que almacenan objetos primitivos y contenedores, fuentes que constituyen el denominado *web data o dominio de conocimientos universal (virtual)*. Sobre este dominio

se pueden establecer una taxónomia de procesos asociado del web data. De acuerdo a Arotaritei D. et al [163] contenida en la Figura 2.26

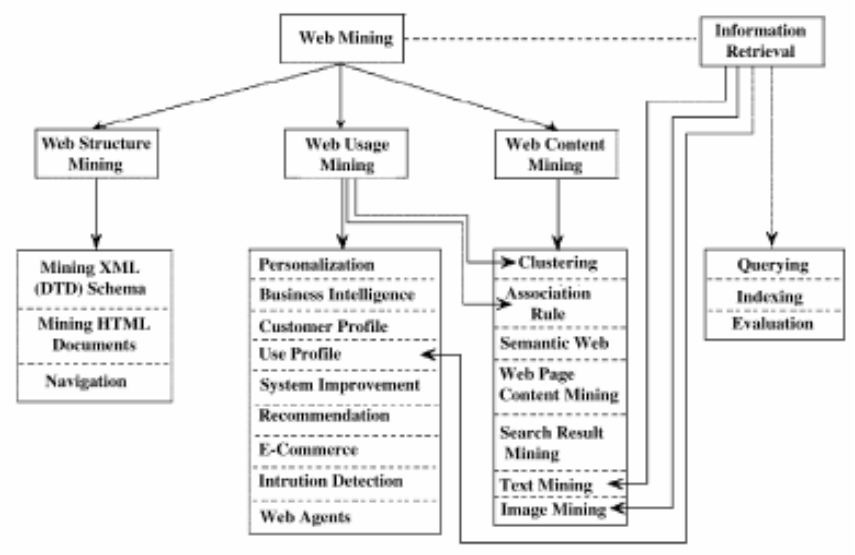


Figura 2.26 Taxónomia del Web Mining

A este árbol taxónomia web mining es posible agregarle una nueva rama definida como Multimedia Web Mining la cual estará dirigida al rescate e interpretación de información multimedia

2.3.1 Minería de Datos Web Multimedia o Multimedia Web Mining

Esta categoría apunta al proceso de descubrimiento de conocimiento a partir de datos multimedia que son almacenados en bases de datos multimediales contenidas en la Intranet o Internet. Este tipo de datos corresponden a: videos, audio, imágenes, gráfica y otras representaciones multimedia. En general este tipo de minería de datos emplean la mayoría de las técnicas anteriormente expuestas destacándose aquellas provenientes de la Computación Flexible la cual incluye las herramientas siguientes: Lógica Difusa, Redes Neuronales Artificiales, Algoritmos Genéticos, Sistemas Neurodifusos y métodos específicos asociados al reconocimiento de patrones. Estos sistemas es posible clasificarlos en:

1. *Sistemas para la extracción y recuperación de imágenes*
2. *Sistemas orientados a la Visualización, Cubos multimediales*
3. *Sistemas de búsquedas de video*
4. *Sistemas de apoyo a Investigación Científica: Astronomía, otros*

Este tipo de web mining emplea técnicas como el análisis automático de imágenes de video estableciendo por ejemplo las relaciones entre entidades de bajo nivel o estructuras de datos multimediales realizando el aprendizaje de componentes del dominio de conocimiento seleccionado (semántica visual). Este aprendizaje se concreta asignando componentes de imágenes o video a clases subjetivas. En el caso de un sistema orientado a aplicaciones científicas por ejemplo, los algoritmos implementados sobre este, permiten al experto (científico) como a los sistemas remotos (robots) buscar, analizar y catalogar objetos espaciales almacenados en grandes volúmenes de datos y eventos dinámicos o streams de imágenes en tiempo real.

2.3.2 Técnicas Asociadas al Web Mining

Según Zadeh L. [172] la lógica difusa puede transformarse en la columna vertebral para el desarrollo del paradigma web inteligente y del web semántico, en donde el rol de esta herramienta debiera orientarse al desarrollo de técnicas y algoritmos encaminados a resolver problemas como el clustering de usuarios y documentos, aprendizaje automático o manipulación de consultas fuzzy que encierren un lenguaje natural por medio de la utilización de términos o conceptos lingüísticos, como *alto*, *medio*, *mayor* (otros), hemos planteado anteriormente la importancia de la computación flexible en análisis de objetos no-textuales o multimedia. Datos habitualmente contenidos en nuestro dominio de estudios definido en este trabajo como el web. Dedicaremos un breve análisis a la descripción de algunas técnicas y algoritmos propios de el web mining.

Esta variedad de tipos de datos disponibles en el web permiten clasificar su explotación o Web Mining de acuerdo a lo planteado por Cooley R. [165], ordenando los dominios de datos en los siguientes:

> *Contenido (Content): Los datos reales que contienen las páginas fueron diseñados para satisfacer a los usuarios. Estos datos consisten generalmente en texto y gráficos., pero no se limitan tan solo a este tipo de información de datos.*
>
> *Estructura (Structure): Es la información de datos que describe la organización del contenido de la pagina web. La información de la estructura interna de una página incluye el arreglo de las varias etiquetas HTML o de XML dentro de esta. Esto se puede representar como una estructura de árbol en donde una etiqueta (HTML) se convierte en la raíz de este. La clase principal de información en la estructura interna de una página son los "hyper-links" que conectan una página con otra.*
>
> *Uso (Usage): Los datos que describen el patrón del uso de las paginas web, tales como direcciones IP, referencias a páginas, y la fecha y la hora de accesos.*

Perfil De Usuario (Usage): Datos que proporcionan la información demográfica sobre los usuarios del Sitio Web. Esto incluye datos del registro e información de perfil del cliente.

Figura 2.27 Taxónomia Básica de Web Mining

Sankar, et. al [170], Lee, et al [168], consideran que desde el punto de vista de los procesos de explotación minera de datos en el web la Taxónomia del web mining puede ser establecida en tres categorías generales denominadas Web Content Mining, Web Structure Mining y Web Usage Mining con una topología indicada en la Figura. 2.27 *El Web Content Mining,* asocia el concepto de descubrir información útil desde el contenido de los documentos en el web, el contenido de un documento no es solo texto, como es fácil instruir para cualquier persona que utilice Internet. En la red global se encuentran documentos cuyos contenidos son del más variado tipo, audio, video, símbolos, datos, meta-datos, link, hiperlink, textos y otros. ***Web*** *Structure* ***Mining*** en cambio explora la estructura de los hiperenlaces contenidos en una página web, estructuras similares a la indicada en el ejemplo de la *Figura. 2.28*. El denominado *Web Usage Mining*, extrae información de datos secundarios generados por los usuarios en su interacción por un sitio web o navegación. El Web Usage Mining incluye datos provenientes de accesos *a servidores web registrados en archivos del tipo logs*, *logs de servidores proxy*, *logs de motores de búsqueda*, *perfiles de usuario*, *archivos de enrolamiento o registro*, *sesiones de usuarios y transacciones*, *consultas de usuarios*, *carpetas marcadas*, *clic del ratón y desplazamientos (scroll)* y *otros tipos de información.*

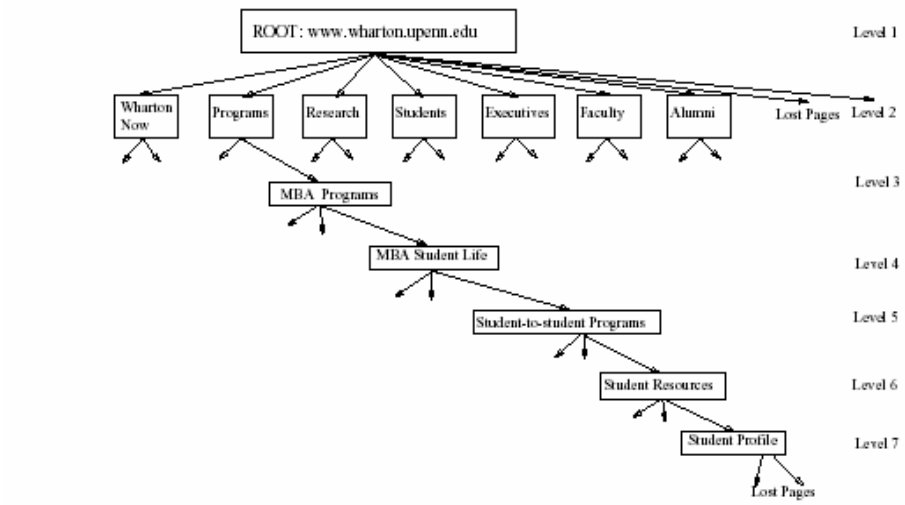


Figura 2.28 Ejemplo de una estructura de una página web

Tomando estas referencias ampliamente aceptadas, describiremos en el punto siguiente las técnicas y algoritmos relacionados con la computación flexible fijando nuestro objetivo en aquellos utilizados en Web Usage Mining

Clustering

El Clustering corresponde a una técnica del aprendizaje automático no supervisado, donde no están definidas las clases o en su defecto no están previamente asignadas. La clave de esta técnica es determinar una buena métrica de medición de la similaridad o semejanza entre instancias o patrones. El clustering o agrupamiento puede ser aplicado al web usage mining o al web content mining (de forma muy generalizada), con el objetivo de agrupar clases de documentos, clases de usuarios u otras de interés. En esencia esta técnica aplicada al web mining, difiere con respecto a la minería de datos estándar solo en la etapa de preprocesamiento de datos, etapa a la cual esta dedicada esta tesis y que considera la utilización de datos provenientes del web o datos no estructurados. La lógica difusa por tanto juega un rol importante en la metodología a ser aplicada con la finalidad de lograr agrupamientos de patrones que tengan como característica el ser ruidosos, pudiendo asignarlos a conjuntos difusos aplicando conceptos con grados de pertenencia basándose por ejemplo en cantidad o calidad.

Algoritmos de Clustering: Fuzzy c-Means

El objetivo de este algoritmo es minimizar una función del tipo

$$J(U,V) = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^{m} diss(v_i, x_j) \quad \text{subject to} \quad \sum_{i=1}^{c} u_{ij} = 1 \text{ for all } j,$$

en donde $diss(v_i,x_j)$, es la distancia del patrón x_j desde i_{th} al cluster, esta distancia puede ser expresada en términos de una distancia euclidiana, u_{ij} que es la función de pertenencia en el intervalo [0,1] del punto x_j **en** el i_esimo cluster tal que:

$$0 < \sum_{i=1}^{n} u_{ij} < n \ \forall i$$

U es la partición fuzzy o ***c-partición*** del conjunto de datos. ***V*** es el conjunto de prototipos ***c-prototipo*** y el exponente ***m>1*** es el fuzificador. Es destacable mencionar que este algoritmo tiene problemas para encontrar el cluster correcto ante la presencia de ruido o "outliers", dado que asume que cualquier punto del conjunto de datos es esencialmente clusterizable. El concepto de ruido [a0] puede ser empleado para agrupar todos aquellos patrones que tengan la presencia de ruido sobre un umbral y así agruparlos por ejemplo en "cluster ruidosos". Esta idea puede ser entendida de la siguiente forma: un punto espacial de ruido o con ruido supone que tiene una distancia $diss(v_i,x_j),=\delta$ de un prototipo *P,* esta medida δ pasa a ser la clave para establecer que patrón contiene ruido y es evidente que debe ser adaptada a cada tipo de problema, una buena solución es adaptar esta medida por ejemplo a un promedio estadístico como el siguiente:

$$\delta = \lambda * \left[\frac{\sum_{i=1}^{c} \sum_{k=1}^{n} diss(v_i, x_k)}{n(c-1)} \right],$$

donde λ corresponde a un valor experimental determinado como el factor de creencia (la certeza)que se ubica entre 0.1 y 100. (Nota: este factor de creencia puede ser abordado como aporte de tesis).

El problema de ruido en los datos es inherente al web y es quizás un tema de amplio interés, en la ecuación de la función tipo planteada la restricción probabilística de que el grado de pertenencia u_{ij} de un punto de datos x_j tenga la suma de uno, es relajada por medio del concepto posibilidad difusa de c-means, en donde se tiene que

max $x_j \{ u_{ij} \} > 0$ para todo j

Para evitar una solución trivial que es el caso en que $u_{ij} = 0$, a la función tipo es posible agregarle un término que penalice (la duda) a los grados de pertenencia bajos, agregándole este concepto a la función tipo, la función tipo toma la forma:

$$J_P(U,V) = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^m diss(x_j, v_i) + \sum_{i=1}^{c} \alpha_i \sum_{j=1}^{n}(1-u_{ij})^m,$$

Podemos destacar que el concepto de la lógica difusa "grado de pertenencia" que en este caso esta representado en la función tipo como u_{ij} es aplicado en este caso para representar la posibilidad de distribución de clusters U_i sobre el dominio de discurso.

Algoritmo de Clustering Robusto

La determinación de cluster a partir de un dominio de datos no estructurados que es el caso de los ficheros logs, es un problema no resuelto, si por ejemplo se marca el número de cluster a obtener en *C*, las técnicas tradicionales de clustering presentan resultados inciertos. El problema es particularmente serio con la presencia de ruido porque las posiciones de este pueden ser identificadas como patrones.

Joshi et. al [166] presentan un método denominado **Robust Clustering Methods** para resolver el problema de los outliers que presenta el algoritmo **FCM ,** método que puede ser resumido como sigue**:**

Sea $X = \{ x_j \mid j = 1...N \}$ *un conjunto de vectores de rasgo n-dimensional, y el conjunto* $B = (\beta_1, ..., \beta_X)$ *represente C-tuplas de prototipos cada uno de la cuales caracteriza uno de los grupos de cluster C. Cada* β_t *consiste en un conjunto de parámetros. Donde el termino* u_{ij} *representa el grado de pertenencia (membresía) del punto característico* x_j *en* β_i *y C X N matriz* $U=[u_{ij}]$ *es llamada la matriz de restricción fuzzy que satisface:*

$$u_{ij} \in [0,1] \text{ for all } i,j, \quad 0 < \sum_{j=1}^{N} u_{ij} < N \text{ for all } i, \quad \text{and} \sum_{i=1}^{C} u_{ij} = 1 \text{ for all } j.$$

El algoritmo Fuzzy c-Means divide los vectores en grupos o cluster C, basándose en una función objetivo J (**B,** U; X) de la forma

$$J(\mathbf{B},U;X)=\sum_{i=1}^{C}\sum_{j=1}^{N}u_{ij}^{m}d_{ij}^{2}.$$

en la ecuación anterior m ∈ [1,∞) y es el exponente o peso del fuzzificador, y dij^2 es la distancia del punto característico x al prototipo β_i. Es observable que la función J(**B**,U;X) esencialmente emplea un criterio dado por la suma de los cuadrados lo que es una aproximación débil o no es robusta en cuanto al criterio de clusterización. A partir de lo anterior existen variados planteamiento en cuanto a introducir mejoras al criterio de clusterización o a la función objetivo. Una reformulación de la función objetivo corresponde a la siguiente:

$$R(\mathbf{B};X)=\sum_{k=1}^{n}\left(\sum_{i=1}^{c}D_{ik}^{1/1-m}\right)^{1-m}=\sum_{k=1}^{n}H_{k},$$

donde Hk, corresponde a

$$H_k=\left(\sum_{i=1}^{c}D_{ik}^{1/1-m}\right)^{1-m}$$

Desde el Hk se desprende que los valores correspondiente a los valores atípicos(outliers) son grandes, luego la idea es diseñar la función objetivo con el fin de que su mínimo global sea logrado cuando un gran Hk sea determinado, siendo este ignorado o descontado del proceso. La función objetivo del algoritmo Robusto FCM es:

$$J_{RFCM}=\sum_{k=1}^{n}\rho(H_k).$$

Esta función objetivo aplica una función de perdida ρ(.) que es usada para reducir el efecto de los valores atípicos. La función de pérdida ρ(.) es típicamente lineal para distancias pequeñas y luego satura para mayores.

Estos planteamientos nos indican la necesidad de contar con "buenas métricas" que permitan resolver en parte el problema del agrupamiento, técnica que al ser aplicada a datos que no tienen marcas o de baja estructuración presenta problemas con los denominados "atípicos (outliers)". Una idea adicional tendiente a introducir nuevas mejoras es planteado por Frigui et. al [165] este

algoritmo se basa en una técnica denominada cluster competitivo, la función objetivo en este caso toma la forma:

$$J_R(B,U;X) = \sum_{i=1}^{C}\sum_{j=1}^{N} u_{ij}^2 \rho(d_{ij}^2) - \alpha \sum_{i=1}^{C}\left[\sum_{j=1}^{N} w_{ij} u_{ij}\right]^2 \quad \text{subject to} \quad \sum_{i=1}^{C} u_{ij} = 1 \ \text{ for all } j.$$

Pseudo código de un Algoritmo de Clustering

Un *pseudo código* de implementación de estos algoritmos considerando una función objetivo simplificada puede ser representado de la siguientes forma:

Sea $\mathbf{Xc = \{ x_i / i = 1, ..., n\}}$ *un conjunto de objetos y* $\mathbf{diss(x_i, x_j)}$ *la matriz de disimilaridad o no similitud entre los objetos* x_i *y* x_j *con el conjunto* $V_0 = \{ v_i / i = 1, ..., j\} \subseteq \mathbf{Xc}$ *siendo este un conjunto representativo de puntos (patrones). La función objetivo o tipo* E_m *que se minimiza sobre todo* V *en* X_c *esta dado como:*

$$E_m(V, X_c) = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^m diss(x_j, v_i),$$

en donde, μ_{ij}^m *, representa la membresía fuzzy posibilística de* $\mathbf{x_j}$ *de clusterizarse (agruparse) en* $\mathbf{i}$ *por medio de la determinación heurística de* $\mathbf{m}$*, parámetro que controla la condición de esta función de pertenencia. Los paso del algoritmos son:*

(i) Set the number of clusters c
(ii) Randomly pick up initial set of medoids V from X_c
(iii) Set $iter = 0$
(iv) For $i \leftarrow 1$ to c do // Compute the membership
  For $j \leftarrow 1$ to n do
   Compute u_{ij}^m, using eqn. (6), for $x_j \neq v_p$
  endfor
 endfor
 $V^{old} = V$ // Store the current medoids
 // Compute the new medoids
 For $i \leftarrow 1$ to c do

$$q = arg \min_{1 \leqslant p \leqslant n} \sum_{j=1}^{n} u_{ij}^m * diss(x_p, x_j)$$

$$v_i = v_q$$

(v) If $iter = MAX\_ITERATIONS$ or $V^{old} = V$, then stop else go to step 4

The membership function is expressed as

$$u_{ij}^{m} = \frac{\left(\frac{1}{diss(x_j, v_i)}\right)^{1/(m-1)}}{\sum_{p=1}^{c}\left(\frac{1}{diss(x_j, v_p)}\right)^{1/(m-1)}}.$$

Los algoritmos presentados anteriormente están basados en prototipos numéricos, por tanto no pueden ser usados si las características no son numéricas. Por otra parte, hay algoritmos de agrupamiento o clustering que no usan la idea de prototipos, sino que establecen reglas de asociación difusas que no emplean el concepto de prototipo. Estos tipo de algoritmos emplean matrices de conceptos o símbolos o matrices de similitud, matriz que puede ser entendida desde el punto de vista difuso, siendo destacable mencionar que este tipo de algoritmos de agrupamiento orientados al establecimiento de relaciones son generalmente más complicados de implementar. Fuzzy c-Means y sus herederos pueden ser utilizados para agrupar (cluster) datos desde documentos web en trocitos de estos (es decir frases ...) siendo adecuados para reducir el ruido característicos de estos documentos.

Otro método de clustering que emplea conjuntos difusos y funciones de membresía es el planteado por Runkler, et al [170], este método permite una generalización de fuzzy c-means y es denominado por los autores como Alternating Cluster Estimación (ACE), en este se emplean iteraciones alternativas sobre una arquitectura modelo, permitiendo al usuario seleccionar el grado de pertenencia y las funciones prototipos, pudiéndose escoger una partición U(t) y nuevos prototipos V(t) en cada iteración. Se consideran por este método dos dominios de datos: texto contenido en objetos primitivos almacenados en páginas web y secuencias de paginas web solicitadas por los usuarios; es decir Web Content Mining y Web Usage Mining. Este tipo de algoritmos pueden ser utilizados por ejemplo para determinar sesiones de usuarios, realizar seguimientos de clic y determinar los intereses del usuario.

2.3.3 Web Mining : Reglas de Asociación

Las reglas de asociación generalmente son entendidas como técnicas propias del Data Mining, siendo muy populares en el campo del marketing, la minería de reglas de asociación esta focalizada a encontrar reglas del tipo " *X → Y, con A% de soporte y B% de confianza",* en donde X e Y son conjuntos de ítems de una base de datos de transacciones" . Las reglas de asociación en

el contexto del web mining se refieren a la determinación de aquellas URLs que tienen tendencia a ser requeridas. Esto puede ser categorizado según la taxónomia indicada en la Figura. 2.27 bajo la caja del Web Usage Mining o Web Content Mining, empleándose como dominio de conocimiento los ficheros web logs.

Los denominados Mapas Cognitivos Difusos o Fuzzy Cognictive Maps (FCM), emplean reglas de asociación para modelar el conocimiento de un experto (humano) los cuales teóricamente forman mapas cognitivos dinámicos de sus experiencias expresadas bajo la forma de conocimiento. Este método pretende expresar ese conocimiento por medio de una técnica híbrida que emplea a la lógica difusa y la teoría de las redes neuronales. La ilustración grafica de un mapa cognitivo queda representado por un grafo de la forma indicada en la Fig. 3.31, grafo que contiene nodos con interconexiones ponderadas y retroalimentación. Los nodos son conceptos que se usan para describir el comportamiento de un sistema y ellos estos interconectados por arcos firmados y cargados con peso que representa las relaciones causales

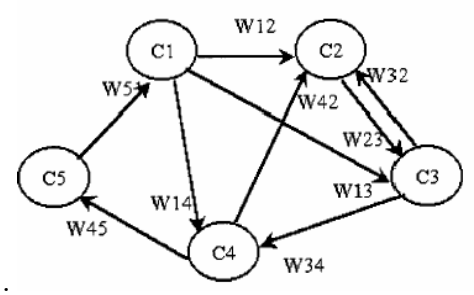


Figura 2.29 Mapa Cognitivo Difuso

Cada concepto representa una característica del sistema; y en general simboliza condiciones, variables, acontecimientos, acciones, metas, valores, tendencias del sistema que es modelado como un FCM. Cada concepto es caracterizado por un número *Ai,* lo cual representa su valor y eso resulta de la transformación del valor real de variable del sistema, para el cual este concepto está vigente, en el intervalo [0, l]. Todos los valores en la gráfica son difusos, y así los pesos de las interconexiones pertenecen al intervalo [-1, 1]. El modelo gráfico (grafo) es una representación de la conducta del sistema estudiado, lo cual permite dejar definido que concepto influencia a otros y en que medida o grado de pertenencia. entre los nodos concepto

Esta forma de representación permite la actualización fácil del modelo gráfico, como la adición o suprimiendo una interconexión o un concepto. El aporte más esencial de los mapas

cognitivos es la determinación de los mejores conceptos que describen el sistema, la dirección y el grado de causalidad entre los conceptos. La causalidad es otra parte importante en el diseño FCM; Indica si un cambio en una variable causa un cambio en otro, y debe incluir la causalidad escondida posible que podría existir entre varios conceptos. Hay tres tipos posibles de relaciones causales entre conceptos que expresan el tipo de influencia de un concepto sobre los demás. El peso de una conexión entre nodos Ci y Cj puede ser denotado como w_{ij} ,y este puede tomar los siguientes valores: ser positivos, para una causalidad positiva, ser negativo para una causalidad negativa o ser cero lo que significaría que no existe relación entre los conceptos Ci y Cj.

El valor de cada concepto es influenciado por los valores de los conceptos conectados con los pesos causales correspondientes y por su valor previo. Así es que el valor *Aj* para cada concepto *Cj* está calculado por la siguiente ecuación o regla:

$$A_j^s = f\left(\sum_{i=1\, i\neq j}^{n} W_{ij} A_i^{s-1} + A_j^{s-1}\right),$$

Donde $A^s j$ es el valor de concepto *Cj* en el paso *s*, Ai^{s-1} es el valor de concepto *Ci* en paso *s*- 1, Aj^{s-1} es el valor de concepto *Cj* en paso *s* -1, y *Wij* es el peso de la interconexión entre *Ci* y *Cj,* y *f* es un umbral basado en una función o función del umbral. Las funciones del umbral restringen el resultado de la multiplicación en el intervalo [0,l]. La ecuación incluye el valor previo de cada concepto, y así es que el FCM posee capacidades de memoria. El desarrollo y diseño del mapa cognitivo difuso es apropiado para la descripción de un sistema que requiere la contribución de conocimiento de un experto humano. Los expertos revelan mapas cognitivos difusos usando un procedimiento interactivo presentando este conocimiento en la operación y el comportamiento del sistema. El procedimiento para construir mapas cognitivos difusos es como sigue: El o los expertos definen los conceptos principales que representan el modelo del sistema, describen la estructura y las interconexiones de la red usando declaraciones condicionales difusas. Los expertos usan declaraciones o reglas IF – THEN para describir las relaciones causales entre los conceptos, y en base a ese conjunto de reglas, el FCM es estructurado y las interconexiones ponderadas son determinadas.

Las reglas difusas IF-THEN , usadas por el experto(s) para describir las relaciones entre conceptos y así construir el mapa cognitivo difuso, asumen la siguiente formalidad donde ***A*** y ***B*** son variables lingüísticas:

SI valor del concepto Ci es A ENTONCES el valor de concepto Cj es B.

El conjunto de variables lingüísticas para cada concepto puede tomar valores como los siguientes con las funciones de membresía correspondientes:

El valor de concepto Ci es muy muy bajo con la función de membresía U_ vvl

El valor de concepto Ci esta muy bajo con la función de membresía V_ vl.

El valor de concepto Ci esta casi dormido con la función de membresía ZZZ _ vgr .

Se puede observar que las reglas son simples y están orientadas a ser fácilmente interpretadas por el experto que las creo, es fácil deducir por tanto que de haber mas de un experto participando de la creación del mapa una regla podría tomar una forma distinta.

Lee K.C. et. al [168],construyen un sistema denominado Web Mining Inference Aplification (WMIA) basándose en los denominados mapas cognitivos difusos (FCM), sistema implementado en tres fases:

La primera fase extrae reglas de asociación con un algoritmo de prioridad desde un dominio de datos constituido por ficheros logs.

En la segunda fase o etapa se construye un mapa cognitivo fuzzy de conocimiento, lo que implica causalidad incorporando reglas positivas o negativas

En el estado final el sistema aplica "amplificación de la inferencia" en orden a enriquecer los resultados obtenidos.

El conocimiento causal es representado en este caso como una matriz de adyacencia que incluye la conectividad w_{ij} de los nodos

$$W^{\mathrm{T}} = \{w_{ij} \mid w_{ij} \in \{-1, +1\}\}.$$

en donde w_{ij} es el peso del arco desde el vértice / nodo C_i a C_j. La definición "*la causalidad de A incrementa B*", implica que un decremento o incremento de **A** provoca o tiene como causa un incremento / decremento de **B**. La relación fuzzy empleada por el sistema desarrollado incluye una etapa de amplificación de la inferencia o conocimiento causal dado por:

$$C_i \overset{w_{ij}}{\rightarrow} C_j;\ C_i \overset{\sim w_{ij}}{\rightarrow} \sim C_j;\ \sim C_i \overset{\sim w_{ij}}{\rightarrow} C_j;\ \text{and}\ \sim C_i \overset{w_{ij}}{\rightarrow} \sim C_j,$$

en donde w_{ij} es la función de pertenencia valorizada en R(Ci, Cj) que es la relación fuzzy entre el nodo concepto Ci y el Cj.

Es destacable mencionar que los autores adaptan el algoritmo a-priori de Agrawal & Srikant [163] para ser utilizado en la fase de prepocesamiento de los web logs y específicamente para inferir las reglas de asociación , este algoritmo tiene la forma siguiente:

Pseudo code of the apriori algorithm

C_k : Candidate transaction set of size k
L_k : frequent transaction set of size k
L_1 = {frequent items};
for ($k = 1$; $L_k != \phi$; $k++$) **do begin**
C_{k+1} = candidates generated from L_k;
for each transaction t in database do
    increment the count of all candidates in C_{k+1}
    that are contained in tL_{k+1} = candidates in C_{k+1} with min_support
endreturn L_k;

Figura 2.30 Algoritmo de Reglas de Asociación a-priori (Data Mining)

CAPÍTULO 3: *ANÁLISIS INTELIGENTE DE SITIOS WEB.*

Conceptualmente la información contenida Internet puede ser entendida con un conjunto de objetos agrupados en distintas fuentes primarias de información las cuales almacenan conocimiento, ya sea en los propios objetos o en las relaciones establecidas entre estos con la finalidad de implementar una solución web. Una fuente primaria de información o fuente primaria de conocimiento web, agrupa objetos como: páginas web, pendones (banners), fotos, objetos audio, objetos video, ficheros con texto, ficheros con índices de recursos u otros objetos; objetos que son requeridos por sujetos humanos o virtuales como robots, agentes inteligentes, agentes de indexación u otros. Estas fuentes construyen en su conjunto un dominio global de información (sitio web) en el cual subyace conocimiento; este conocimiento proviene de manera principal de la información almacenada en los objetos y de las relaciones establecidas entre objetos dispuestos en una fuente primaria de información o solución web.

Definición 1

> *"Sea B_Interntet un sistema formal o base de datos virtual B que consta de los siguientes elementos:*
>
> *Un conjunto numerable y finito de objetos primitivos, que contienen información no estructurada basada en símbolos interpretables para sistemas complejos sujetos reales(animales) o virtuales.*
>
> *Un conjunto (finito) de objetos contenedores que agrupan objetos primitivos y determinan bajo qué condiciones podemos afirmar que un conjunto de objetos primitivos es (o no) una fórmula (página web). El conjunto* ***S*** *de las fórmulas se denomina fuente primaria (o sitio web).*
>
> *Un conjunto(finito) de fuentes primarias* ***S ,*** *que definen reglas, métodos o sistemas de acceso a los distintos objetos de información que estas almacenan. Combinatorias que sirven para producir deducciones o interpretaciones formales sobre los sujetos que visualizan sobre una ventana virtual los objetos de información (i.e., determina qué secuencias de fórmulas o páginas web constituyen conocimiento bajo la forma de una deducción o percepción del objeto visualizado posible de obtener del sistema).*

Estas reglas normalmente incluyen la aceptación como verdaderas de un conjunto finito de sentencias (i.e. fórmulas sin variables libres) que reciben el nombre de principios del sistema.(consecuencia: lenguaje natural de acceso a objetos web)

Las sentencias del sistema . El conjunto de sentencias deducibles se llama Teoría Formalizada o Teoría del Conocimiento Web.

Definición 2

Dado el sistema formal B o base de datos distribuida que almacena objetos primitivos no-textuales los cuales contienen información visualizable γ es deducible del sistema la existencia de conocimiento, decimos que A es una consecuencia propia del sistema (semántica, visual, auditiva...etc) y se denota como sigue:

A $\rightarrow \gamma$ *. Si* γ *es una afirmación verdadera en cualquiera de las posibles interpretaciones del sistema formal, diremos que se trata de una consecuencia racional de* ***A*** *y lo denotaremos como* ***A*** $\models \gamma$ *o* ***verdad racional****.*

Si nos detenemos a observar la obtención de conocimiento del web a partir del paradigma de "web inteligente" será necesario considerar como hipótesis general la existencia de un Dominio Global de Conocimientos en el cual se pueden definir cierta reglas de acceso a las fuentes primarias que disponen de subconjuntos de objetos y datos ordenados o agrupados por características de similitud, subconjuntos desde los cuales se constituyen los siguientes sub-dominios de conocimientos a ser entendidos como:

1. Conocimiento Propio del Dominio: *Se aplica al conocimiento factible de obtener del dominio de datos seleccionados. Se aplica a los hechos, teorías y heurísticas deducibles o aplicables al dominio de datos seleccionado (sub-conjunto)*

2. Conocimiento de Control: *Este describe las estrategias, modelos y seguimiento de control y auditoria aplicables sobre el subconjunto de los objetos con la finalidad de dar solución a problemas específicos*

3. Conocimiento Explicativo o de Estructuras: *La base de conocimiento esta constituida por una serie de enlaces y punteros entre diversos servidores y organizaciones de datos al interior de estos servidores, este tipo de conocimiento podría por ejemplo permitir búsquedas de recursos inteligentes determinando rutas mínimas a los recursos requeridos.*

4. Conocimientos Semántico o Conocimiento del Contenido: *Concierne a la interpretación semántica y sintáctica de los contenidos de los diversos objetos y recursos almacenados en el Dominio Universal*

5. C*onocimiento Artificial: Corresponde al conocimiento obtenido a partir de relaciones entre los distintos dominios de conocimiento. Estas relaciones pueden ser interpretadas como verdades racionales obtenidas como un proceso de búsquedas o respuestas ante un problema en particular.*

Estas definiciones o contextos de trabajo permiten ser referidas al paradigma web inteligente, es decir la existencia de un dominio global de conocimientos representado por la información contenida en la red de redes internet., conocimiento que es posible de obtener por medio de una de un sistema formal que en el caso de esta tesis corresponde a la definición 1; luego una teoría del conocimiento web, se refiere a una cadena de procesos que tienen por objetivo obtener una verdad racional.

Las Fuentes Primarias de Conocimientos Web [55], almacenan objetos y datos crudos, estos objetos como también los datos crudos, poseen como principal característica el ser no estructurados y cambiantes en el tiempo, factores que desde el punto de vista de un observador externo se asemejan a la realidad cotidiana que un experto humano enfrenta con su dominio primitivo. A partir de lo anterior podemos suponer que el proceso más apropiado para el estudio de un dominio que replica al dominio contextual humano corresponde en analogía al: "proceso humano para el razonamiento" o en menor escala a "los procesos de razonamiento de organismos complejos animales u artificiales". El planteamiento anterior nos lleva a plantear una hipótesis de trabajo que se base en estas características, es decir un proceso de obtención de conocimiento que considere la existencia de un Dominio de Conocimientos Global (Internet u otro) como fuente de datos u origen de la información, debe necesariamente ser modelado empleando herramientas que se acerquen al razonamiento humano. Esto ultimo puede ser reforzado con el planteamiento de Maturana [168] que indica "*lo que un observador ve como conducta, es una dinámica de cambios que involucra a dos sistemas que operan en forma independientes: el sistema viviente y el medio*"

.

3.1 Descripción y Características Generales: Análisis de Ficheros Logs

Un dominio de datos o dominio de conocimiento ampliamente utilizado para la evaluación de un sitio web está constituido por los registros almacenados en los ficheros logs del o los servidores web que implementan un sitio web, este dominio de datos se construye de forma genérica por medio de procesos automáticos que van almacenando en ficheros logs los eventos que son consecuencia de las peticiones de recursos realizadas al servidor web por parte de los usuarios o clientes del sitio web. En la actualidad no existe un estándar que permita una única metodología y formato para el registro de los eventos producidos por la navegación de los usuarios en Internet, esta información que constituye la base de cualquier estudio que tenga como objetivo resolver problemas conceptuales como la **Medición de Audiencias de Internet** o el problema particular de la **Evaluación de Sitios Web** es dependiente del sistema operativo y de los procesos que permiten la construcción del sitio web y el registro de los eventos, lo cual implica una diversidad de formatos y calidades de la información almacenada por estos ficheros. Junto al método de "registro de eventos" o ficheros logs, han surgido otras metodologías como la selección de una muestra aleatoria de usuarios o panelistas a los cuales se les instala un software especializado en sus ordenadores, software que genera sus propios logs almacenando información de navegación de los usuarios seleccionados; estos software registran información como páginas visitadas, tiempos que están en cada una de ellas u otros parámetros de interés del fabricante o proveedor de la herramienta o programa. Otro método de relativa importancia corresponde al empleo de marcadores o etiquetas (código) implantadas sobre las páginas web, este método conocido como "registros por tags" o simplemente "tags" los cuales generan información basada en el empleo de contadores o por la navegación de un usuario (trackers), el sistema de tags en algunos casos utiliza cokies que se almacenan en el ordenador de los usuarios lo cual evidentemente es una desventaja, dado que necesariamente el usuario tiene que aceptar o tener habilitada la función de almacenamiento de cokies.

Independientemente del método empleado para registrar los eventos o sucesos de la navegación de los usuarios en Internet, la información capturada es almacenada en archivos o ficheros los cuales genéricamente son conocidos como Web Logs. Estos ficheros Web Logs contienen registros que almacenan la información de la navegación de los usuarios bajo la forma de eventos generados por las solicitudes o requerimientos de - objetos – almacenados en la fuente primaria (sitio web por ejemplo) y de los sujetos - que los solicitan usuarios, robots, agentes u otros.

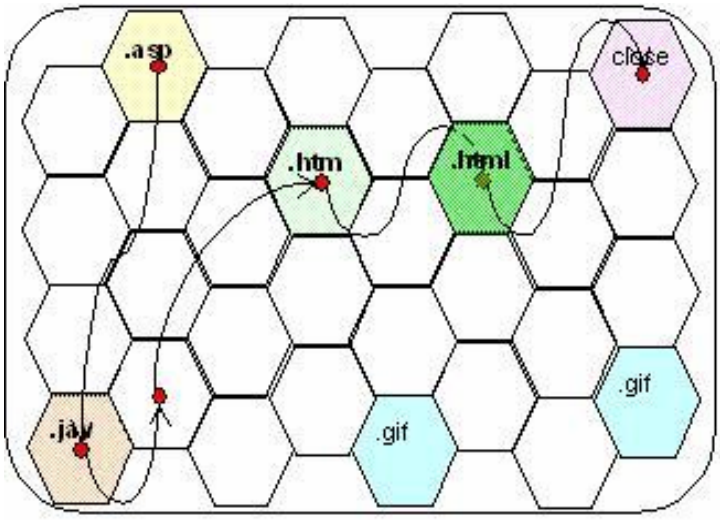


Figura 3.1 Patrón de Navegación Típica de un Usuario

Objetos disponibles en una Fuente Primaria de Información (Sitio Web vs. Páginas)

Sitio Web = {(**Objeto$_1$**: HomePage (H.html), **Objeto$_2$**: Producto(P.html), ...**Objeto$_n$**: Tipo$_n$(nombre.formato)}

Considerando el Sitio Web de la Figura 3.1, cuya ventana virtual corresponde a un conjunto de objetos del tipo página web enlazados de forma multidireccional entre "objetos-páginas" relacionados, una secuencia de navegación de un usuario generará eventos sobre aquellos puntos de la ventana virtual en los cuales se requiera un nuevo objeto página, necesariamente este objeto deberá ser alcanzable desde la página en la cual el usuario se encuentra en un momento dado. Es posible por tanto suponer que algunos objetos página solicitados no sean alcanzables desde la página que el usuario dispone, sino que el recurso sea alcanzable a través de un objeto-página mantenido en cache o en una ventana virtual adicional, condición que tendría por consecuencia la no-generación de un evento.

Todas las presunciones anteriores nos permiten deducir que los eventos o peticiones secuenciales de objetos, generados por la navegación de un usuario sobre una fuente primaria de información y almacenados en un fichero Web Logs, no reflejan de manera lineal el comportamiento del usuario, como tampoco siguen el diseño dispuesto por sus creadores; sino mas bien el registro de los eventos generados a partir de las solicitudes o peticiones de objetos corresponde a un modelo caótico o no lineal sobre una base de tiempo al cual se asocia el evento al momento que este es registrado.

Este factor necesariamente nos lleva a plantear que determinar el comportamiento de los usuarios a partir del análisis de los eventos generados por objetos solicitados al sitio web, corresponde a un proceso que se basa en la teoría de conocimiento web, nivel de abstracción que emplea conceptos propios del lenguaje humano, conceptos que se relacionan directamente con la justificación del paradigma "web inteligente" y por consecuencia definen la metodologías a emplear para el estudio, métodos o soluciones a aplicar para la obtención de conocimiento del web, conocimiento que puede ser utilizado de manera básica en la evaluación y personalización de sitios web y/o soluciones e-commerce u otras.

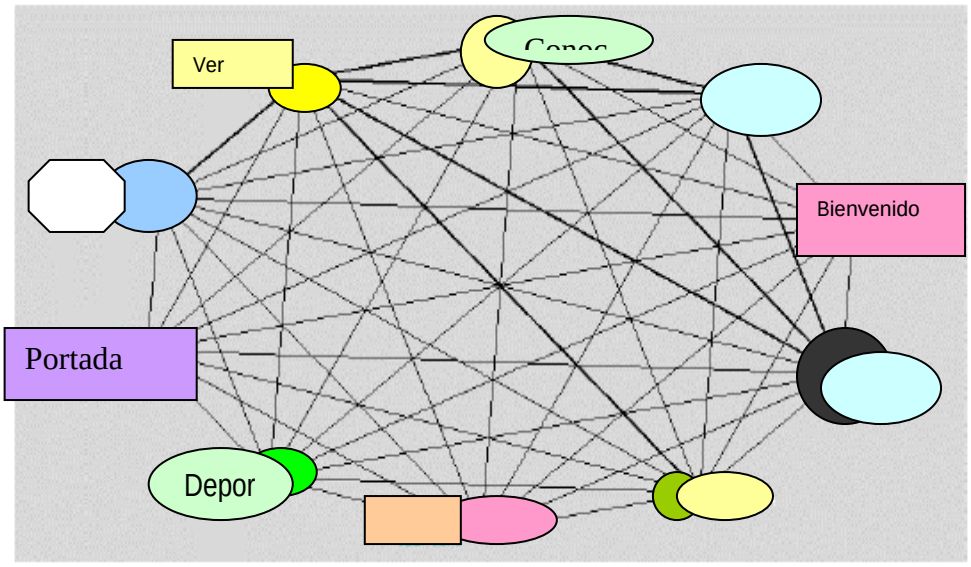


Figura 3.2 Navegación de Usuarios vs. Generación y Almacenamiento de Eventos

En la Figura 3.2 hemos querido representar el concepto de navegación y generación de eventos, es necesario precisar que los objetos páginas que son solicitados por los usuarios de un sitio web generan un evento (registrable) solo en el instante que este requerimiento es nuevo, es decir el objeto-página no esta almacenado en cache o en el proxy más cercano al usuario este factor queda representado en la Figura 3.2 como los vértices del poliedro, en donde la línea de tránsito o solicitud de un objeto ya solicitado desde uno que esta siendo visualizado; implica que la solicitud no es resuelta por el servidor que aloja el sitio, sino que esta página-objeto es rescatada desde el cache o desde el proxy teniendo como consecuencia la no generación del evento respectivo. Este factor adicional nos permite deducir la necesidad de disponer de un sistema

que maneje esta incertidumbre e inexactitud. Por otra parte si observamos el proceso de registro de eventos implementado en un típico servidor web y dado la característica caótica de peticiones de recursos por parte de los usuarios producto del comportamiento de navegación de los mismos, es habitual que los administradores restrinjan el tamaño de los ficheros logs a una cierta medida o en su defecto empleen métodos rotatorios para evitar que estos ficheros alcancen tamaños peligrosos para el funcionamiento de la solución web implementada, acción que necesariamente debe ser tomada en cuenta en cualquier proceso de web mining. Los eventos registrados en los ficheros logs son almacenados en tiempo real y restringidos a un tamaño, por tanto es posible introducir grandes sesgos en el análisis de la información contenida en estos de no considerarse esta característica. Supongamos por ejemplo que el día 1 del mes 3 el fichero alcanzó su tamaño máximo permitido y la política utilizada indica respaldo semanal, es claro que los eventos de los días 2 y siguientes hasta la fecha de la rotación del fichero logs y su respaldo se perdieron luego cualquier análisis a emprender tiene una muestra no representativa o sesgo que impediría llegar a conclusiones aceptables.

En la presente tesis se propone un modelo que considera los aspectos mencionados con anterioridad, es decir el hecho real que este tipo de ficheros almacenados en distintos servidores por medio de los cuales se implementa una solución web tienen las características mencionadas con anterioridad que pueden ser resumidas en: una misma solución web puede almacenar varios tipos de ficheros logs en distintos servidores, las peticiones de objetos realizadas por sujetos reales o virtuales pueden ser respondidas por distintos servidores o en el mejor de los casos por un único servidor, los fichero logs generalmente están restringidos en su tamaño.

Considerando las características anteriores estas pueden ser expresadas de la siguiente forma:

Definición 3

Sea una solución web S implementada por varios servidores y sean Fi los ficheros almacenados en cada uno de estos, el dominio de conocimiento genérico que reúne a estos esta dado por:

Dci = { F1, F2, F3,.....Fn} ; en donde

Fi = (Evento1, Evento2, ...Evento$_n$),

Cada evento como hemos visto de manera detallada en el capítulo dedicado a los datos queda almacenado en los ficheros log bajo la forma de un registro, un formato conocido y un matasellos o "timestamp" en tiempo real que se agrega al registro en el instante que se produce la petición de un objeto almacenado en ***S.*** Para nuestro trabajo los eventos quedan expresados de manera general de la siguiente forma:

Un evento Ei estará constituido por tres parámetros principales: una dirección ip que identifica al sujeto o cliente, un objeto solicitado y un matasellos que indica fecha y hora de la petición, todo lo anterior puede ser expresado como:

$E_i = [\ IP_i\ ,\ Objeto_c\ ,\ T_i\ ]$ ***en donde:***

IP = dirección de destino del objeto solicitado o sujeto que lo solicita

$Objeto_c$ ***= Objeto contenedor o página web***

T_i ***= fecha y hora de la petición***

3.2 Comportamiento de usuarios y Registro de Eventos a partir de sus solicitudes.

Los usuarios de un sitio web o solución e-commerce mantienen un comportamiento que se ve reflejado en parte en los objetos que solicitan al servidor que aloja la solución, este comportamiento puede ser analizado desde el punto de vista demográfico (crisp o tradicional) es decir cantidad de objetos-páginas solicitados, cantidad de usuarios que acceden al sitio, número de personas que se conectan al sitio, número de personas que se conectan a una página determinada, identificación de los usuarios que se conectan al sitio. Estos análisis mantienen fines como determinar secuencias de objetos solicitados por los usuarios, asociar sesiones de usuarios con objetos determinados, determinar link, servicios o productos que puedan ser percibidos como de preferencia de un usuario o de grupos de estos; estas tareas vienen acompañadas necesariamente con vincular una visita de usuario con un objeto determinado, objetos que son dispuestos en la ventana virtual que contiene la pagina visualizada por el usuario en un determinado tiempo.

Estas solicitudes o peticiones de objetos (GET, POST) y el respectivo almacenamiento del evento asociado a la solicitud en un archivo Web Logs, contiene información imprecisa propia

del método de registro empleado, e incertidumbre genérica que es producto del diseño dispuesto por los creadores de la solución web. Hemos visto anteriormente que un evento puede ser producido por medio del empleo de alguna metodología (tags, clickstream, registro de eventos, sniffers, etc.) y almacenado en distintos servidores y ficheros que forman parte de una solución web, este conjunto de datos definen un dominio de conocimiento que pueden ser analizarlo demográficamente, o con el objetivo y finalidad de responder preguntas como las siguientes:

¿ Cuál es el objeto-página más solicitado?

¿ Cuales son los principales dominios web de origen: países, comerciales, etc?

¿ Cuál es el número de páginas vistas de todo un sitio?

¿ Cuál es el número de personas que se conectan a un objeto página especifico?

¿ Qué tipo de usuarios se conectan a un objeto página?

¿ Que hora del día es la "hora punta" del sitio web?

¿ Cuales son los banners más vistos?

Las metodologías para la medición de audiencias de Internet (utilizadas por empresas especializadas) permiten responder preguntas como las planteadas anteriormente, estos métodos de análisis y medición pueden ser clasificados en: técnicas centradas (orientadas) en el usuario, centradas en el sitio o análisis de logs y orientadas a la publicidad. Para el caso de las técnicas basadas en el usuario, estas apoyan sus estudios en la definición de un panel o muestra de ínter-nautas a los cuales se les instala un software o programa que realiza el monitoreo y registro de sus actividades sobre Internet. El método de análisis centrado en el sitio en cambio basa sus estudios a partir de la información factible de ser capturada en el propio sitio, existiendo diversas tecnologías para llevar a cabo la captura de los datos los cuales finalmente se almacenan en ficheros del tipo logs. Por último las técnicas orientadas a la publicidad basan sus estudios en el tráfico generado por el despliegue de banners u otros tipos de publicidad; esta "contabilidad" de estos recursos publicitarios se lleva por medio de la inserción de un código html y cokies en el equipo del usuario, métodos por medio de los cuales se contabiliza cada vez que estos son objetos son desplegados, esta metodología es particularmente invasiva dado que

utilizan las cokies u otro tipo de programas para determinar perfiles de usuarios y de esa manera dirigir la publicidad.

Un parámetro básico de medición de audiencias de un sitio web corresponde al denominado *PageView* o *Páginas Vistas*, parámetro empleado por la mayor parte de los software de análisis de logs o de evaluación de Sitios Web, este parámetro recoge las estadísticas del empleo de las páginas del sitio clasificándolas por cantidad de requerimientos o "hits" de solicitudes, en términos más concretos los resultados obtenidos por medio del parámetro "PageView" se refieren a la cantidad de solicitudes de páginas y no se refieren a la efectividad de la información contenida en la página y el "impacto" de este contenido sobre los usuarios que la visitan o en palabras más simples a la percepción del contenido de la misma por parte de los usuarios. En muchos casos el *PageView* tiene la tendencia a transformarse en el dato y ser entendido como la información del contenido de una página y puede conducirnos a respuestas estadísticas (crisp) que no reflejen necesariamente el comportamiento de los usuarios de un sitio web; por ejemplo el solo hecho de solicitar un objeto página determinado (página web) por el conjunto de usuarios de un sitio web tiene por consecuencia inmediata un aumento del parámetro "página solicitada" o "número de personas que se conectan a una página web" (visitas), el dato "visitas" representa solo la cantidad de veces que el objeto a sido requerido, a modo de ejemplo tomemos el caso de un portal de noticias o periódico web:

Ejemplo 3.1 Hits de Navegación

Usuario 1 → visita página inicio (portada.html) → luego pasa a deportes (dep.xxx) → portada (cache)→ moda → portada (c)→fin de sesión

Conteo de Objetos: obj.-portada = 1 visita + 2 cache(supuestos)=3obj.-deportes = 1 visita

el resultado seria : 1 sesión (usuario), 5 pageview. :

Sitio Completo Page Views = 5, Total Usuario Único 1 →vobj.-moda = 1visita

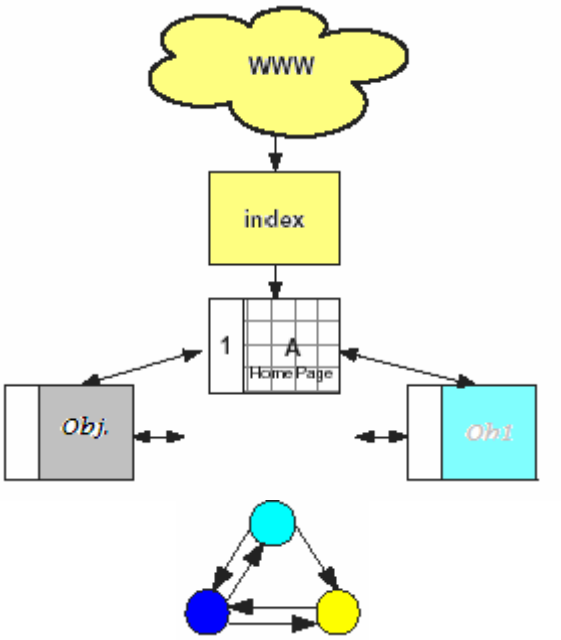


Figura 3.3 Estructura Básica de un Sitio Web y Grafo de Navegación

En la Figura 3.3 se observa que los objetos páginas que son solicitados por los usuarios de un sitio web generan un evento (registrable) solo en el instante que este requerimiento es nuevo representado en la figura por los vértices del triangulo o grafo, esto ocurre por que la petición del objeto-pagina es resuelta por el proxy más cercano o en la computadora del usuario; es posible que el usuario navegue entre esta estructura de manera aleatoria sin generar eventos o si el diseñador lo establece generando "siempre un evento" o hit sobre una pagina usada como puente para alcanzar otras, determinar por tanto el tiempo de permanencia en cada pagina nos puede dar independencia de la estructura de diseño al emprender una análisis de los datos y registros contenidos en los ficheros logs. La distancia o métrica entre objeto pasa ha ser un factor critico y esta queda representada por las transiciones entre un objeto que genera un evento registrable y otro.

Las estadísticas rescatables de un fichero logs esta asociadas a las peticiones de páginas realizadas vía http GET o POST, en el caso utilizado de ejemplo de una estructura de sitio web muy básica que considera dos niveles (el mínimo seria uno: home) de ejemplo, los eventos se producen en distintos equipos, sobre el servidor, sobre el proxy (cache), sobre el pc del usuario (cache). Es posible por tanto obtener respuestas como página mas visitada (que no sea home por supuesto) y con técnicas de data mining obtener relaciones como:

SI el usuario visita deportes, ENTONCES visita moda. ó si objeto1 ENTOCES objeto2

Se puede observar de la estructura del sitio web y adicionalmente del grafo de navegación que un sitio web puede ser diseñado con el objetivo de aumentar ficticiamente la medición de los accesos por parte de los usuarios o en su defecto para que estos siempre tengan que recurrir a un mismo camino para alcanzar un objeto determinado, junto a lo anterior no es deducible si el objeto fue visualizado o percibido por el usuario, podemos destacar que la identificación de usuario puede ser un problema complejo de determinar dado por ejemplo que todos los usuarios de un ISP (Internet Solutions Providers), realizaran sus peticiones http desde una única ip o en su defecto lo realizaran desde distintas producto de la asignación dinámica de estas.

3.3 Análisis inteligente de sitios web versus web mining

Un sitio web puede ser estudiado desde varios puntos de vista, el demográfico por ejemplo en el cual se refiere a la utilización de técnicas estadísticas que permitan medir audiencias en internet, o en otras palabras análisis descriptivos de usabilidad de los recursos almacenados en la solución web; la tarea de este tipo de método es un análisis de datos dirigido a la verificación, lo que en palabras más precisas se trata de análisis estadísticos de ficheros web logs, técnica que probablemente sea la más difundida en todo el planeta. Los análisis estadísticos de logs persiguen objetivos como determinar por medio del parámetro *pageview* u *otras medidas,* la cantidad de *páginas_vistas o hits* realizados por los usuarios del sitio, y determinar por ejemplo el uso de un recurso publicitario implementado en esta página. Una respuesta posible de obtener por medio de a herramientas estadísticas de análisis de ficheros logs puede ser la siguiente: *"el sitio de micropalta.com tiene un promedio de visitas o usuarios de 10^{57} hits sobre nuestro recurso publicitario contratado y alojado en el; en cambio en el sitio publiqueaqui.com la inversión se pierde dado que los pageview sobre la página que contiene nuestra publicidad están bajo el estándar".* Esta conclusión o respuesta se basa en las peticiones realizadas al servidor web que aloja la página web solicitando ese objeto, peticiones que se transforman eventos registrados en un fichero logs. El problema que presenta esta tecnología es que permite obtener respuestas muy elementales como la anterior, es decir *"la página fue requerida 10^{57} veces en el periodo de tiempo estudiado".* El problema se complica más aun al momento de necesitar relacionar *la página* de interés con grupos de usuarios o usuarios particulares, dado que una respuesta del tipo *todos los usuarios X visitaron Y*; se topa con el problema de la determinación efectiva del *usuario ip* que realizó la petición o la clasificación de estos; como hemos planteado con anterioridad en el capítulo dedicado al origen de los datos

contenidos en los ficheros logs esta ip puede ser la misma para muchos usuarios o distinta para los mismos, que es el caso cuando estos emplean direcciones dinámicas ip (dhcp) suministradas por su proveedor de servicios o utilizan servidores proxy corporativos. Esta pregunta solo es posible darle una respuesta razonable en el mundo del Web Mining o Minería de Datos aplicada a los ficheros logs metodología que es conocida ampliamente bajo el nombre de Web Usage Mining. Cualquier intento de dar respuesta a determinar relaciones como por ejemplo *usuarios X visitan Y* o *cliente_ ip_visita P y luego compra C,* necesariamente se enfrenta nuevamente al problema de la determinación de la ip cliente o usuario, problema que ha llevado a muchas abstracciones como por ejemplo la metodología de Web Usage Mining planteada en [165,169] referidas a la determinación de sesiones, determinación de episodios, determinación de caminos de búsquedas completos y definición de que es un "pageview", propuestas que tienen por objeto asimilar los registros contenidos en los ficheros web logs a una base de datos tradicional; para luego aplicar técnicas de minería de datos que permitan determinar *patrones* de búsqueda de recursos o comportamiento sobre el sitio web. Agrawal R. et. al [163] proponen en el trabajo referido dos algoritmos de minería de datos denominados *AprioriSome* y *AprioriAll* que son ampliamente utilizados en procesos de Web Usage Mining, estos eficientes algoritmos utilizados en base de datos transaccionales se enfrentan a la gran dificultad que es la alta dimensionalidad de los datos almacenados en los ficheros logs problema que deberá ser considerado como critico, luego el método de trabajo empleado por estos algoritmos es determinar grandes grupos de ítems o productos y asociarlos a un cliente perfectamente identificado, el *producto* puede en analogía ser asociado a una *página web* y el *cliente* a una *ip*, suponiendo que una correcta metodología pueda precisar que una ip es un cliente como por ejemplo el método *sesión-episodio-caminocompleto* [165], el problema en aplicar esta técnica puede ser focalizada en una observación importante, la cantidad de clientes como también los productos de un supermercado almacenados en una base de datos transaccionales bajo registros estructurados, son inmensamente menores que los registros de un fichero web logs, registros que se almacenan en forma dinámica y tienen un bajo grado de estructuración, a modo de ejemplo de la dimensionalidad Lycos genera alrededor de 4GB de ficheros logs todos los días a una tasa aproximada de 8000 hits/minutos, luego por tanto el coste del proceso informático asociado a la agrupación de clientes ip es demasiado elevado y no ajeno a fallas o resultados anómalos producto por ejemplo del ruido web. Las soluciones algorítmicas actuales basadas en métodos de minería de datos aplicados al web, conllevan complejos procesos que están orientados a reducir la dimensionalidad del problema y a aumentar la eficiencia del tiempo de procesamiento

para obtener resultados reflejado en patrones minados que permitan dar respuestas como *cliente_ ip_visita P y luego compra C.*

Otra dificultad generalmente omitida o no considerada en los procesos de Minería de Datos aplicada al Web o Web Mining corresponde al hecho real que internet en la actualidad es en esencia un dominio de datos visuales orientado a la percepción de estos por parte de los usuarios que navegan en los distintos sitios web, situaciones como determinar si un usuario realmente visito página web no tienen respuestas fáciles en los procesos de Web Mining, y en la medida que los problemas se acercan a preguntas habituales como la del ejemplo ¿nuestro banner de publicidad es efectivo?, pueden ser no resueltas por las herramientas del data mining aplicado al web. De ahí que nuestra propuesta de tesis se enfoca a un análisis de los datos contenidos en los ficheros logs *orientado a la obtención de conocimiento*., en donde las herramientas de la Computación Flexible y en especial la Lógica Difusa permiten ampliar los limites de estas preguntas al nivel de un lenguaje natural, permitiendo por ejemplo asociar conceptos como el de visita o sesión realizada a un sitio web por parte de un sujeto real o virtual concepto que es intrínsicamente impreciso, y asociarlo a términos lingüísticos como por ejemplo visita_corta, visita_media;. La lógica difusa y sus herramientas permiten el desarrollo del concepto de análisis inteligente de datos, dado que esta preparada para manejar conceptos vagos o imprecisos, problemas conceptuales usuales en el dominio internet, en donde los objetos contenidos en las distintas fuentes primarias distribuidas en la red global están orientados a la visualización y posterior percepción de los usuarios que los solicitan niveles de abstracción que pueden ser asociados a una simple pregunta como ¿la página fue vista o visualizada?. o ¿su contenido fue percibido? , las respuestas posibles a estas preguntas están en el domino del conocimiento factible de obtener desde las distintas fuentes primarias contenedoras de objetos, el concepto de percepción de un objeto por un sujeto real o virtual, el concepto de visita valorizada en términos de un grado de pertenencia a un conjunto difuso son ejes centrales del desarrollo de esta tesis, estos conceptos enmarcan sus respuestas en el denominado Web Inteligente concepto que permite extender las fronteras del Web Mining.

CAPÍTULO 4: *EL CONCEPTO DE PERCEPCION DE UNA PAGINA WEB*

Hemos visto que el parámetro "Page Views" o "número de páginas vistas de todo un sitio". (unidad de medida) tiene directa relación con los software y métodos demográficos para la medición de audiencias en Internet, métodos empleados por la casi todos los software de análisis de logs y muchas herramientas de data mining, o por soluciones nativas de las empresas del rubro dedicadas a los estudios de mercados o "Análisis y Medición de Audiencias en Internet". Este tipo de análisis o estudios de mercados están basados en modelos por medio de los cuales se definen ciertas métricas que representan por ejemplo la usabilidad de un sitio web. Una de las mediciones más habituales corresponde al parámetro PageViews el cual intenta medir el uso del sitio web o la: *cantidad de páginas vistas por los usuarios que han visitado el sitio o la suma de todas las visitas a las páginas de un sitio web.*

El PageViews tiene por objetivo representar una tendencia estadística del comportamiento de los usuarios del sitio web o de la solución e-commerce estudiada y por medio de esta establecer relaciones, agrupamiento u otras técnicas con el objeto de establecer conocimiento. El hecho real es que una página "visitada" o medida por medio de este parámetro no necesariamente es "vista" o "leída" en sus contenidos por los usuarios, esta simple deducción tiene por consecuencia la perdida de valor del concepto de visita a un sitio web asociado al parámetro PageView. Adicionalmente a lo anterior existe la tendencia de algunos desarrolladores de sitios a "inflar" o aumentar los hits de solicitudes al sitio introduciendo diseños que tiendan a reflejar un tráfico elevado en cuanto a las solicitudes de objetos del sitio, este tráfico por solicitud de objetos incluye a ficheros de sonido, banners, gráficos u otros objetos contenidos en las páginas. Lo anterior nos lleva a plantear algunas preguntas adicionales a las del punto anterior:

¿ El objeto-página fue visto realmente..? o ¿ la página fue vista?

¿Cual es el índice o grado de percepción del objeto página?

¿ El diseño del sitio guarda relación con el comportamiento real de un usuario?

¿ Que relación existe entre el tiempo de visita de un usuario y la solicitud de objetos del sitio?

¿ Cual es el uso promedio de un objeto-pagina? –

¿ Cual es el tiempo medio de visitante por objeto-página?

¿ Es posible clasificar los objetos-páginas por la utilización de las mismas?.

¿Cuantos visitantes realizaron visitas cortas, medianas, largas, indefinidas a objetos páginas determinados?

¿Cuál es la pagina "efectivamente" más visitada? (medida en valores crisp y difusos)

¿ Como determinar el perfil de un visitante a partir de los objetos que efectivamente solicita?

El nivel más fino de registro de información de los objetos solicitados de un sitio web se encuentra en el lado del servidor y puede ser obtenido por medio de programas *sniffers* (granularidad maximizada), los cuales revisan los paquetes TCP/IP almacenado el contenido de sus registros en ficheros del tipo logs, estos ficheros representan el comportamiento de los usuarios en su navegación por el sitio web o solución e-comercio, luego dependiendo de los objetivos del estudio web mining las fuentes de los datos pueden variar o ampliarse luego el Dominio de Conocimiento constituido por las diversa fuentes posibles para una misma solución, necesariamente tiene que ser reducido en su dimensionalidad con el objetivo de constituir dominios particulares en donde los datos puedan ser transformados con el objeto de construir distintos niveles de abstracción y por medio de estos datos depurados dar respuestas a algunas de las preguntas planteadas con anterioridad. Dependiendo de las preguntas seleccionadas y sus posibles respuestas estos dominios de datos depurados, son transformados y agregados a distintos niveles de abstracción con el objeto de obtener una verdad racional (paradigma web inteligente) aplicada como respuesta a una consulta sobre el dominio de datos seleccionado.

4.1 Niveles de abstracción: Sesión vs Visita Corta, Media, Larga

Los niveles de abstracción en torno a un dominio de conocimientos particular o dominio de datos restringido, se limitan en primer lugar al **conocimiento propio del dominio o subyacente en este**, el cual como hemos mencionado anteriormente se aplica a fijar los hechos, teorías, heurísticas (otros conceptos) deducibles o aplicables al dominio de datos seleccionado, un nivel básico de abstracción por ejemplo corresponde a establecer una visita de

usuario o sujeto que accede o visita un sitio web. Por medio de la determinación de una sesión de usuario o "visita" al servidor se podrá determinar por ejemplo la conducta o comportamiento de este en su interacción con el sitio web, pudiéndose precisar por ejemplo: secuencias de patrones que este emplea o secuencias de transacciones realizadas en una visita en particular. La noción de "sesión de usuario" puede ser obtenida a partir de los requerimientos que un visitante realiza sobre los objetos contenidos en el sitio web, objetos que son desplegados en las páginas contenidas sobre la ventana virtual que dispone el usuario en un instante de tiempo dado, su navegación corresponderá por tanto a secuencias de páginas que un usuario en particular solicita al servidor y el tiempo total desde que inicio una visita hasta que abandona el sitio, tiempo que este emplea en visitar o navegar sobre los contenidos de las distintas páginas que ha solicitado. Es destacable mencionar que en un web logs se almacena toda la actividad que los usuarios realizan sobre el sitio web, por tanto un registro de este tipo puede contener múltiples sesiones de un usuario o múltiples visitas; el concepto de sesión de usuario se refiere por tanto a la actividad de segmentar un web logs en tiempos de visitas para establecer una sesión real de un usuario en particular, el usuario puede ser identificado por todas aquellas secuencias de páginas-objetos requeridos en una visita, objetos que son asociados a una base de tiempo y una dirección **ip** al momento de ser requeridos al servidor web que los aloja. Algunos sitios web pueden poseer mecanismos de autenticación de usuarios con lo cual una sesión de usuario quedara delimitada por un inicio de sesión (login) y por cierre de sesión (logout).

El "concepto de visita o sesión de usuario" se relaciona directamente con otro nivel de abstracción básico el cual corresponde al o los objetos reclamados o solicitados al servidor por los usuarios, estos objetos solicitados pueden ser representados en un nivel de abstracción genérico por medio de la Página Web que los contiene o almacena, dependiendo del diseño de la página los objetos contenidos por la misma pueden ser agregados dinámicamente o estar dispuestos de manera estática al ser desplegados en la página que el usuario solicita en un momento dado. Una definición de sesión aceptada corresponde a la planteada por.[165], el cual define sesión como:

> *Una sesión de servidor (S) es un conjunto ordenado de páginas vistas (V) en el tiempo; para un único usuario durante una sesión única al Sitio Web en busca de un metadato (A) [3]*
> $S = [A: v_1, \ldots\ldots v_n]$
> $V = \{v_i.H^j, t^f, t^e.(d_1,\ldots\ldots d_m), c\}$
> $A = (a1, a2, a3 \ldots\ldots,an)$

Este concepto de sesión puede (tiene que ser) ser redefinido empleando su naturaleza, es decir un usuario no siempre establece sesiones en un tiempo determinado sino mas bien estas se establecen en rangos de tiempo. Estos rangos pueden ser expresados en términos de nuestro lenguaje natural empleando conjunto difusos asociados a términos lingüísticos que se definen como: ***Visita_corta, Visita_media, Visita_larga***, términos que serán explicados mas detalladamente en secciones siguientes.

Es claro que establecer sesiones desde un web log es una tarea compleja dado la diversidad de formatos y la variedad de fuentes posibles de logs, problemas que se agravan al consideran por ejemplo los logs de los servidores proxy o los provenientes de distintos servidores utilizados como respaldo a la gestión del sitio. Esto es de vital importancia dado que las distintas fuentes desde la cual se pueden integrar logs en un dominio de conocimiento poseen bases de tiempo distintas y a no ser de una sincronización perfecta puede nuevamente presentar sesgos en los resultados obtenidos en la determinación de sesiones de usuarios. Por ejemplo si un usuario realiza un petición al sitio web y esta es resuelta por el servidor proxy en primer lugar y luego por sucesivos servidores; los eventos almacenados en estos distintos servidores necesariamente tienen que ser transformados en una única sesión de usuario a partir de la ip que realizó el requerimiento luego la sesión pasa a depender de:

Algoritmo para el procesamiento de Múltiples Ficheros Logs

Pseudo Código

*Sea C el conjunto deducible de visitas o sesiones del Dominio_**C**$_c$ y sea **T** el reloj de tiempo real empleado para el registro de eventos, en donde **S** se implementa con servidores múltiples, tenemos que:*

***S** = {(servidor1, Ta), (servidor2, Tb), ...(servidorn, Tn), F_i (de formatos, granularidad)}*

Si Ta distinto Tb distinto Tc ENTONCES sesión indeterminada
Proceso de forma independiente cada F_i
Si Ta = Tb = Tc ENTONCES →procesar formato
Formato iguales ENTONCES → procesar granularidad
SI NO
Proceso cada/ formato de forma independiente
FIN

El algoritmo anterior solo opera sobre una base de tiempo idéntica, faltaría por tanto agregar conceptos de distancia en cuanto a que sesión efectivamente es una sesión real. La métrica en este caso es incierta por tanto los conjuntos difusos nos entregan la solución para el caso de bases de tiempos traslapadas, es decir si Ta < Tb o Ta>Tb +Delta o umbral.

4.2 Niveles de abstracción: Sesión de Usuario vs. PageView^2

Uno de los objetivos de la determinación de una "visita de usuario" corresponde a precisar el grupo de objetos o secuencias de objetos solicitados por un "usuario real" en una –visita efectiva o visita real – determinada a partir de una heurística similar a la presentada con anterioridad, heurística que permita segmentar y clasificar la actividad de un usuario en visitas particulares a partir de la información almacenada en un fichero del tipo logs o grupos de ficheros logs integrados a un dominio de conocimiento. Los objetos solicitados por las –visitas-de-usuarios - pueden ser estudiados a partir de la *teoría del conocimiento web* en cuanto a los grados de percepción (visualización) de los mismos, grados que pueden ser indicados por términos lingüísticos "*como no fue visto", "no fue percibido", "existe certeza de que fue leído"*, este grado de percepción no guarda relación con el grado de comprensión o entendimiento de los contenidos de un objeto por parte de un sujeto real o virtual sino mas bien debe de ser entendido en el contexto preferente de una visita a una fuente primaria ***S*** **;** por ejemplo un usuario que tiene un grafo de comportamiento como el indicado en la Figura 3.3 puede ser considerado como un "gran registrador de eventos", el total de estos podría haber sido realizado en un minuto o en una hora lo que marca una diferencia radical, luego por tanto será necesario considerar la variables como: *visita_corta, visita_normal o visita*_larga para que estas sean referenciadas a los objetos solicitados. Es muy distinto y diferente una visita larga sobre un objeto contenedor página web, que una visita corta o breve sobre el mismo, esto que es evidente puede ser ampliado al total del sitio y a la cantidad de objetos que este contenga.

El concepto de percepción del objeto por un sujeto que lo solicita a una fuente primaria, tiene relación con el tiempo empleado o visita, la relevancia subjetiva del objeto contenedor o página web y por la cantidad de objetos primitivos que este almacena. Para el usuario o sujeto es distinto visualizar una página que contenga solo un objeto primitivo el cual podría ser un texto, que una página que contenga tantos como puedan ser desplegados sobre la ventana virtual del usuario, considerando este hecho o el número de objetos primitivos contenidos en una pagina web un nuevo parámetro de medición o PageView^2 podría ser definido de la siguiente manera:

Definición 4

> *Una visualización de un objeto o percepción del mismo(PageView^2) puede ser entendida como una actividad imprecisa basada en el tiempo realizada por un sujeto sobre un objeto*

almacenado en una fuente primaria ***S.*** *Actividad que es registrada bajo la forma de un evento o suceso referido a una base de tiempo real o virtual*

PageView^2 → *P* { visita(difusa), Objeto (r), deltaT(conjunto difuso)}*

En donde la ***visita(D)*** queda representada por términos lingüísticos como corta, normal, alta o larga y conceptualmente esta puede ser asociada a conjuntos difusos, una función de creencia y un grado de pertenencia. Por otra parte el objeto contenedor ***Objeto(®)*** puede ser descrito por medio de la cantidad de objetos primitivos que este contiene y por la relevancia (difusa o crisp) de sus contenidos. En nuestra definición hemos considerado el parámetro ***deltaT(D),*** para representar el tiempo de captura de objetos primitivos desde el servidor con el objetivo de construir la pagina web que finalmente será visualizada o percibida.

La visualización efectiva de un objeto (*PageView*^{2}), conceptualmente representa un tipo especifico de actividad imprecisa de un usuario en una fuente primaria, esta actividad puede corresponder por ejemplo a la lectura de algún articulo de un periódico web (tiempo 1), agregar un producto al carrito de compras(tiempo 2) o simplemente navegar entre objetos y páginas que los contienen sin ningún patrón de comportamiento o clic (tiempo 3); cada una de estas actividades guarda cierta relación con el tiempo empleado en las mismas, por lo cual el concepto de visita esta directamente relacionado con el tiempo empleado en esta . Ahora desde el punto de vista de la medición de audiencias se acepta que el valor de la actividad de los usuarios sobre un sitio web puede ser medida por medio del parámetro pageview el cual es interpretado como la(s) página(s) que solicita un usuario en una única sesión en un único servidor, es decir el dominio queda absolutamente restringido al rescate de valores estadísticos, en suma el total de pageview se transforma en el dato y no en la información. Conceptualmente una visita a un sitio web o fuente primaria es realizada por medio de peticiones http empleando comandos o primitivas GET o POST las cuales solicitan objetos principalmente contenedores o páginas web, estas peticiones son respondidas por el servidor y trasladadas hasta la ventana virtual del usuario en donde esta la visualiza. Un concepto de visita y visualización efectiva necesariamente deberá ser establecido considerando los argumentos anteriores luego para nuestra propuesta hemos considerado redefinir este parámetro agregando los conceptos y argumentos expuestos anteriormente.

PageView^2 → *F{ visita(difusa), Objeto (difuso), Tiempo(conjunto difuso)}*

Las páginas visualizadas (requeridas no es lo mismo) son objetos contenedores que almacenan objetos primitivos, luego por tanto no existe la certeza que en un instante dado una página web contenga los mismos objetos. Esta particularidad en la actualidad se ve reflejada en que la mayoría de las soluciones web como e-commerce, e-learnig u otras, emplean bases de datos desde las cuales alimentan al objeto contenedor con ciertos objetos que cambian dinámicamente, que es el caso estudiado experimentalmente el cual corresponde a la solución e-learning o *UVirtual UTEM*. A cada objeto contenedor-página web (*Objeto_pg.)* solicitado por un usuario o visitante del sitio web (contenido en la solución web), es posible asociarle por parte del diseñador parámetros como importancia o peso del objeto, ubicación preferencial en la ventana virtual, contenido estático, contenido dinámico u otras cualidades que permitan ser orientadas al cliente, a la publicidad o al análisis de los contenidos del sitio.

Cada ***Objeto_páginaw*** y los objetos que la página contiene solicitado por un "visitante" del sitio, es registrado como un evento o suceso en un fichero logs en el lado del servidor y como hemos descrito anteriormente se almacena información restringida al formato escogido del fichero logs, ***evento***, que guarda por ejemplo el instante de tiempo en el cual son requeridos los objetos, identificación del usuario (dirección ip), que solicito el objeto, navegador utilizado, puerto solicitado, estado del requerimiento o éxito o fracaso de la operación solicitada.

Ejemplo 4.1 *Registro Típico Almacenado en un Archivo Logs*

[Wed Oct 11 14:32:52 2000] [error] [client 127.0.0.1] client denied by server configuration: /export/home/live/ap/htdocs/test

Definición 5

Un evento puede ser entendido como el resultado de una operación realizada por una fuente primaria ***S*** *como respuesta a una petición realizada por un sujeto real o virtual en un instante de tiempo T.*

Un evento es el resultado de una operación realizada por un servidor o fuente primaria ante una solicitud o requerimiento por parte de un sujeto (cliente, real o virtual), como por ejemplo la solicitud y posterior retorno de un documento, la ejecución de un script, la bajada de un archivo, el llenado del carrito de compras u otras solicitudes de servicios. Estos requerimientos de los usuarios hacia el servidor y las clases de respuestas que este entrega como resultado de las operaciones solicitadas pueden ser registradas empleando programas especializados que capturan el resultado de la operación, almacenando los distintos componentes asociados al tipo

de solicitud y las respuesta entregadas por el servidor en un archivo de registro de eventos; este proceso automático de captura y registro de eventos es normalmente conocido como el "loggins de eventos" del servidor. Considerando la arquitectura de Web más sencilla, cada vez que un cliente realiza un requerimiento a un servidor de red, es enviado un paquete HTTP (Hypertext Transfer Protocol) sobre la red, desde el cliente hacia el servidor nombrado en el campo URL (Universal Resource Locator) de la solicitud. Luego el servidor retorna uno o más paquetes conteniendo o bien la respuesta, o bien un código de error. Desde el punto de vista de la topología de redes, existen cuatro lugares posibles desde donde capturar los eventos o transacciones sobre el servidor de red, sobre los servidores proxy, sobre los clientes o sobre la red. A partir de la información almacenada en los archivos logs, es posible establecer relaciones u asociaciones posteriores como por ejemplo: la hora y fecha cuando fue requerida una página web por parte de un "cliente" del servidor web, el total de páginas requeridas por un usuario en particular, la página más visitada o establecer relaciones más sofisticadas a partir del conocimiento subyacente en la información de datos contenidas en los diversos archivos que registran los eventos de un servidor web. La información contenida en los registros almacenados en los ficheros logs es dependiente del *método de registro* utilizado, como por ejemplo la utilización de un programa sniffers, una aplicación del sistema operativo u otros programas de software, el conocimiento almacenado por tanto dependerá de la *granularidad* proporcionada por el método de registro de eventos y factible de ser almacenada por medio del formato en el cual son dispuestos los eventos generados al momento que un "visitante" requiere o solicita los distintos elementos o sistemas que intervienen en la interacción de un usuario con el sitio o solución web, estos elementos son:

1. *La estructura o topología del sitio, topología constituida por caminos de navegación dispuestos por el diseñador. Estos caminos se forman por medio de páginas enlazadas entre si (diseño del sitio)*

2. *los objetos dinámicos y estáticos que son visualizados por medio de un objeto-página (página web). Estos objetos tienen su origen en distintas fuentes de información o repositorios de objetos*

3. *la navegación por el sitio, la cual puede ser caracterizada como la búsqueda de páginas web (definición Obpg) que contienen objetos los cuales son visualizados sobre la ventana virtual que un usuario posee en un instante de tiempo*

4. *un sistema de registro de eventos basado en tiempo real*
5. *un fichero logs que almacena los eventos (bajo la forma de un registro) generados por peticiones de objetos al servidor*
6. *un dominio de conocimientos o partición de un fichero logs (o conjunto de particiones)*
7. *una sesión o secuencia de navegación determinada por medio de una heurística*
8. *Un conjunto de objetos requeridos por una sesión real o visita (carrito de compras)*

Es indudable que una visita a un sitio web en particular no responde a algún patrón de comportamiento fijo por parte de un usuario sino mas bien las visitas quedan sujetas al libre albedrío del visitante al momento de interactuar con los objetos-páginas enlazados entre sí de acuerdo a una estructura o topología dispuesta por el diseñador, la navegación o comportamiento del visitante puede ser analizada desde el punto de vista de la obtención del Conocimiento desde el dominio definido por el Conjunto de Ficheros Web Logs Depurados. Por ejemplo si consideramos nuestra pregunta base ¿la página fue vista?, se requerirán métodos, algoritmos o heurísticas orientadas a dar respuesta a esta pregunta o problema, en primer lugar será necesario determinar el tipo de visita realizada por el usuario y luego en base a esta visita determinar si el objeto fue visto o percibido. En el caso de determinar por ejemplo el comportamiento de un usuario (conocimiento propio del dominio) a partir de las secuencia de patrones empleados por un visitante en un segmento de fichero logs o dominio de conocimiento será necesario orientar la metodología nuevamente a la determinación en primer lugar de las visitas efectivas constituidas por el total de veces que el visitante estableció una sesión real y el conjunto de objetos requeridos en cada una de estas sesiones.

De acuerdo con lo anteriormente expuesto es posible definir una hipótesis de trabajo general basándose en el paradigma web inteligente con el objetivo de establecer relaciones entre sesiones reales de visitantes difusas o con cierto grado de incertidumbre de una solución web y los objetos dinámicos, estáticos o borrosos que esta contiene, con la finalidad de determinar hechos, teorías, o verdades racionales que permitan obtener "conocimiento artificial"

Hipótesis General:

Sea R el conjunto de todas las visitas factibles de ser determinadas de un fichero logs F_i por medio de una heurística ***H*** *definida a partir del dominio de datos* $\mathbf{D_i}$*. El objetivo de la heurística* ***H*** *es reconstruir una visita de usuario a partir de la información almacenada en el fichero web log y de esta visita o conjunto de sesiones reales de usuario determinar relaciones, vínculos, usabilidad, percepción de los objetos solicitados con la finalidad de establecer conocimiento propio del dominio de datos seleccionado.*

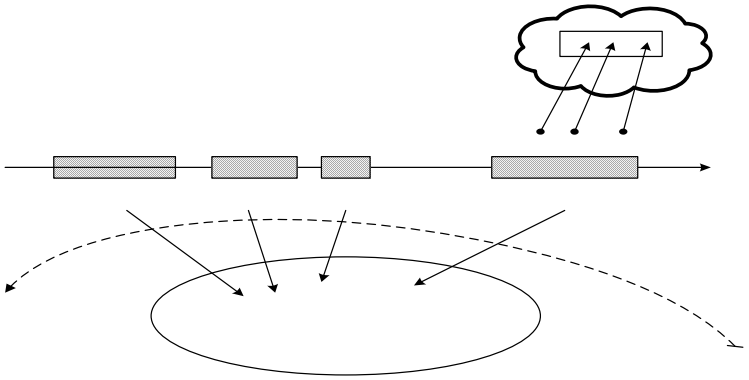


Figura 4.1 Relación entre una Visita Real y Objetos Web

4.3 Descripción de una Página Web vs. PageView[2]

Una página web podría ser descrita (*o definida*) como un objeto contenedor el cual es capaz de almacenar (*o contener*) un número arbitrario y finito de otras entidades (*u objetos*) tales como: vínculos (*URLs*) con otros objetos contenedores o colecciones de estos, ficheros de diverso tipo *(*.doc, *. pdf, *. aplicación*), imágenes (*fotos, dibujos, iconos*), animaciones *(banners, ficheros Flash, Objetos Java, etc*), vídeos (*bajo la forma de un archivo o en tiempo real*), audio *(bajo la forma de un archivo o en tiempo real*), y otros elementos no clasificados (*por desarrollar*) . Hemos empleado el termino "objeto página web" para referirnos al hecho de que un fichero de este tipo tiene sus limites bien definidos establecidos sobre la ventana virtual que un usuario posee en un instante dado, ventana delimitada como la interfaz gráfica (pantalla, teléfono, televisor, palm, etc) que posee el usuario-visitante para interactuar con la solución web (Sito Web) y los objetos contenidos en

esta. Desde el punto de vista de su visualización o percepción una página web es un objeto que al ser requerido representa un tipo especifico de actividad que un usuario realiza sobre el sitio web, actividad que puede ser ejemplificada en acciones como la lectura de un párrafo de información, la búsqueda de un recurso, la agregación de un producto al carrito de compras o simplemente el solicitar páginas web sin ningún tipo de intencionalidad. ("hojear" el sitio).

Definición 6

Definición de Objeto Página Web

*Objeto Contenedor Página Web (Objeto_pg), será todo archivo visualizable mediante un navegador que encapsula un número arbitrario y finito de otras entidades y objetos cerrados los cuales almacenan información con grados de incertidumbre. El termino contención hace referencia a una contención lógica de enlaces o ligaduras sobre otros objetos contenedores, de escenarios visuales y de escenarios virtuales por medio de los cuales se implementa la ventana virtual sobre la cual "una sesión real de usuario" interactúa en un instante de tiempo o visita sobre una solución web **S** o sitio web.*

Como es evidente el contenido de una página web ***(Obpg)*** es modificable ya sea agregando información a la misma o agregando nuevos objetos a la página, esta característica nos indica que la información contenida en un instante de tiempo dado puede ser distinta a otro tomado como referencia para un mismo sitio web o solución web, estas diferencias de información pueden producirse ya sea modificando dinámicamente los contenidos de los objetos cerrados que implementan las páginas o simplemente reemplazando los ya existentes. Esta particularidad nos indica que el objeto contenedor "pagina web" tiene una característica bien definida: la información contenida en el, posee grados de ***incertidumbre e imprecisión***. Esta información imperfecta puede encontrarse tanto en los contenidos dinámicos y estáticos factibles de ser almacenados en un ***Obpg*** como en los eventos generados por solicitudes o requerimientos por parte de los usuarios reales o usuarios virtuales tales como robots, spiders, agentes inteligentes u otros. Entenderemos que un ***Obpg*** posee *incertidumbre e imprecisión* en la información que este almacena, dado a que no existe la suficiente información en los datos almacenados o contenidos en estos objetos, para evaluar la certeza o falsedad en la percepción de los contenidos del objeto por parte de los usuarios que los requieren .(***PageView***^{2})

4.4 Planteamiento de un Algoritmo de Preprocesamiento de Ficheros Logs

Como hemos podidos apreciar a lo largo de este documento la tarea de preparación del conjunto de datos que será sometido a análisis con el objetivo de obtener conocimiento es una tarea critica, en este proceso se encuentran actividades como la política de almacenamiento de datos en los ficheros logs, la integración de ficheros provenientes de distintas fuentes en un dominio de datos o conocimiento, la limpieza del ruido web, la restricción de los objetos de los ficheros a estudiar, la transformación de los datos a un modelo entendible por la herramienta u algoritmos a emplear o como en el caso de un data mining aplicado al web la determinación de pageview, la determinación de sesiones, la determinación episodios, la determinación de caminos válidos. En la mayoría de los casos todos estos procesos están referidos a niveles de abstracción como el concepto "pageview" o "sesionización " o "episodio" abstracciones planteadas por [169] por ejemplo, niveles o abstracciones que determinan fuertemente el proceso de preparación de los datos y sus resultados. Hemos planteado en esta tesis que uno de sus objetivos es el desarrollo de una herramienta y una metodología que se oriente a un proceso de obtención de conocimiento a partir de la información almacenada en los ficheros logs, luego por tanto nuestra visión se acerca al entendimiento de los conceptos asociados al conocimiento desde el punto de vista de como una persona real lo establece sobre el medio virtual con el cual interactúa, lo cual justifica el modelamiento abstracto del problema aun ejercicio efectivo o experimento teórico por una parte y su posterior implementación que sea una demostración de la argumentación planteada. Este ejercicio teórico ha sido desarrollado a lo largo de los distintos capítulos que forman parte de este documento, luego por tanto traduciremos todos los conceptos planteados con anterioridad en un modelo de solución para el análisis de los fichero logs de cualquier tipo, formato o procedencia. Nuestro objetivo de generalización y formalización planteado como definiciones e hipótesis de trabajo apuntan al entendimiento en escala humana de los diversos problemas asociados con la obtención de conocimiento, dado que los resultados buscados están asociados al paradigma web inteligente.

4.4.1 Hipótesis y Definiciones necesarias para la construcción de (Minero P*)

Hipótesis 1.

Sea el dominio de conocimiento genérico ***Dg*** *constituido por distintos ficheros logs que almacenan eventos restringidos a una lista exhaustiva* ***L*** *conocida y definida. Dominio en donde es posible identificar dirección ip, hora y fecha y objeto.*

Esta definición nos permite restringir la alta dimensionalidad de los datos a solo aquellos de interés por ejemplo nuestra lista de objetos negativos y positivos esta constituida por objetos cuyas extensiones son las siguientes:

Tabla de Objetos Negativos **Ln** = [objeto$_1$, objeto$_2$,.....objeto$_n$]

Fichero F_1	Objeto	Tipo de Fichero	id
	Fotos	*.gif, *.jpeg, *.jpg, *.jpe, *.jfif, *.otros.	1
	Dibujos	*.bmp, *.dib, *.tiff, *.tif; *.otros	2
	Imágenes	*.png	3
	Videos	*.avi	4
	Audio	*.mp3, *.mp4	5
	Animaciones	Flash, Java, JavaScrip, otras	6
	robots	lista	7
	agentes	lista	8
	.		
	Otros		n

Tabla de Objetos Positivos **Lp** = [[objeto$_1$, objeto$_2$,.....objeto$_n$]

Fichero F_1	Objeto	Tipo de Fichero	id
	Página web	*.html, *.php, *.htm, *.nuevos	1
	ficheros	*.doc, *.pdf, *.xls, *.mail , *.otros	2
	Base de datos	Referencias,	3
	Videos	*.avi	4

Figura 4.2 Referencias a Datos no estructurados contenidos en Dominio *Dg*

El dominio genérico esta constituido por referencias a datos no estructurados y por otra parte el propio fichero esta formado por datos numéricos y texto, lo cual nos da un indicativo de su tratamiento en la etapa de preprocesamiento deben considerarse esto problemas con el objetivo

de reducir la dimensionalidad de los datos contenidos. La etapa de limpieza de los datos o fase inicial contempla las tareas siguientes:

1. *Creación de la lista de objetos positivos o de estudio*
2. *Creación de lista de objetos negativos como la del ejemplo*
3. *Eliminación del ruido web, robots u objetos de poco interés.*
4. *Otros de acuerdo al problema.*

Los objetos anteriores son definidos como objetos primitivos, dentro de estos encontramos ficheros de diverso tipo, los que en conjunto forma un objeto contenedor o pagina web. Es claro que este objeto siempre estará referido a una ventana virtual implementada sobre algún dispositivo electrónico de visualización en donde por defecto se supone la pantalla de una computadora. Esta ventana restringe a una matriz A x B de tamaño fijo el almacenamiento de objetos, luego por tanto cada objeto a ser visualizado tendrá una ubicación predeterminada o mapa de la página web.

Definición 6 *extensión*

Definición de Objeto Página Web (definición imprecisa)

Objeto Contenedor Página Web (Objeto_pg), será todo archivo vísualizable mediante un navegador que encapsula un número arbitrario y finito de otras entidades y objetos cerrados los cuales almacenan información con grados de incertidumbre. El termino contención hace referencia a una contención lógica de enlaces o ligaduras sobre otros objetos contenedores, de escenarios visuales y de escenarios virtuales por medio de los cuales se implementa la ventana virtual sobre la cual "una sesión real de usuario" interactúa en un instante de tiempo o visita sobre una solución web S o sitio web.

Sea P_w el conjunto de todos los objetos contenedores del tipo ***Obpg (página web)*** , objetos-páginas por medio de los cuales se implementa una solución web la cual define un dominio de discurso cerrado ***S*** en donde:

$S \rightarrow$ ***{ Sitio Web o Solución Web }***

$S = \{Pw_1, Pw_2, Pw_3, \ldots Pw_{n-1}, Pw_n\}$

El objeto contenedor P_w tiene sus limites bien definidos sobre la ventana virtual que implementa es decir P_w es directamente proporcional a la ventana virtual que es implementada por medio de este objeto y almacena un número finito de objetos cerrados cuyas clases se definen como:

P_w = [URL_s , *ficheros* $_{(tipo)}$, *otras clases por definir*] o

$P_w = \{ obj_{clase\_0}, obj_{clase\_1}, obj_{clase\_2}, ...obj_{clase\_Z} \}$

en donde el objeto contenedor página web P_w quedará definido por :

$$Pw = \sum_{i=1}^{a} obj_{(i,a)} \qquad (1)$$

la suma total de objetos cuyas zonas están bien delimitas sobre el Dominio Ventana Virtual
En donde a es el área total que ocupa el objeto en la ventana al cual esta referido

Se emplea el termino (1) para indicar que el total de zonas o espacios ocupados por cada uno de los objetos dentro de la pagina web u objeto contenedor, tiene por consecuencia una ventana virtual definida en dos ejes o un plano.

En términos más precisos un objeto contenedor del tipo página web (P_w) será todo archivo o fichero visualizable sobre una ventana virtual por medio de la cual un usuario mediante el empleo de un navegador centra su atención en los contenidos de la misma, contenidos almacenados en los distintas clases de objetos cerrados contenidos en el "contenedor" P_w .

El archivo o fichero página web tiene sus limites bien definidos en la memoria disponible para una ventana por Ej. La pantalla estándar de un PC tiene limites en dos dimensiones (X x Y), a partir de estos limites la cantidad de objetos a incluir en el contenedor están "limitados" o se reducen a particiones sucesivas del concepto ventana o matriz. Luego un objeto contenedor P_w u ***Objeto Contenedor Página Web*** corresponderá a una función compuesta por objetos cerrados cuyas clases se definen de forma básica en dos: URL y ficheros. Desde el punto de vista de un análisis inteligente es posible definir estos objetos como un patrón cuya ubicación en la ventana virtual corresponde a un vector o conjunto de estos, una matriz , una zona fija o difusa o simplemente un mapa o topología centrada en el dominio ventana virtual.

El concepto ventana virtual se puede estudiar en forma real en dos y tres dimensiones; teóricamente en más de tres. Adicionalmente se amplia la posibilidad de definir $\mathbf{Vv} = P_w + P_g$ en grandes ventanas de dos dimensiones o tres, luego la percepción de contenidos dependerá de la ubicación en el espacio del objeto mas el tiempo empleado en su visualización teórica, tiempo de visita: corta, mediana, larga, otras métricas.

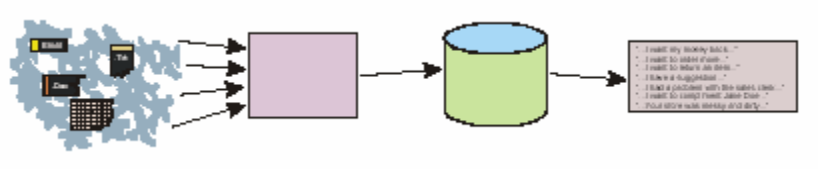


Figura 4.3 Relación de Objeto con Ventana Virtual

La ventana virtual queda definida por tanto de la siguiente manera

V.Virtual ={ (x, y) / pertenecen a A / Objeto(a_i*) =* $[x_0, x_1] \cup [y_0, y_1]$*} en donde*

A es el área del dispositivo de visualización *y* ***Objeto (a)*** esta asociado a *conjuntos restringido*s

Si *Objeto(a*) es máximo es decir *A = a* entonces *V.Virtual =Pw = Objeto (a*)

Pw → {a contener un solo objeto} *percepción asociada a* ***visita*** *es máxim*a.

Si Objeto (a) es minino o tiende a cero, es decir un píxel *V.Virtual =Pw = Objeto (0*) = 0

Pw → {a vacío} percepción asociada a visita es nula o vacía

El objeto no es posible de ser visualizado, las razones pueden ser variadas, el objeto fue diseñado para una ventana inapropiada o esta no existe.

La percepción de la página web dependerá por tanto a la cantidad de objetos de la misma o de acuerdo con $Pw = \sum_{i=1}^{a} obj_{(i,a)}$ y al tiempo de visualización del objeto que se obtiene de la visita a ***S***

De acuerdo con Definición 3 tenemos que:

PageView^2 → *P** { visita(difusa), Objeto (r), deltaT(conjunto difuso)}

En donde ***deltaT*** o tiempo de captura de objetos, es el tiempo que se produce en la construcción de *P**, factor que afecta la percepción por causa que todo navegador y diseñador conoce y generalmente es relacionado con el peso de la página. Este peso o factor de corrección tiene nuevamente características de incertidumbre e imprecisión luego puede ser ajustado a un grado de pertenencia sobre un conjunto difuso el cual es definido de manera experimental sobre distintas plataformas de software y de hardware.

4.4.2 Algoritmo *Minero P** para la determinación de Visitas y Visualización de Objetos

Un proceso de obtención de conocimientos aplicado al web consiste generalmente en varios procesos o etapas que pasan en primer lugar por la preparación de los datos con el objeto de aplicar técnicas de web mining, lamentablemente los ficheros logs de cualquier tipo tienen un bajo grado de estructuración lo cual lleva asociado altas tareas de computo y de procesamiento; para evaluar un sitio web por ejemplo es necesario construir en primer lugar el dominio de conocimientos genérico constituido por diversos ficheros logs, en donde la identificación de los usuarios y objetos es el objetivo con la finalidad de determinar relaciones de objetos con

usuarios de manera similar a un proceso de minería de datos aplicado por ejemplo a la base de datos de retail de un supermercado. La dificultad principal que lleva este tipo de razonamiento al ser aplicado a una fuente primaria que aloja una solución web ***S***, es que en este caso el dominio de datos contiene principalmente referencias de tiempo asociadas a una serie de registros que indican una petición de recursos, a diferencia de una base de datos de retail en el cual nuevamente existe una transacción que queda reflejada como un registro completo en donde el tiempo no es un parámetro que importe al momento de su análisis, dado que el comportamiento del usuario queda determinado por los productos que compra o carrito de compras y que son almacenados en el registro referido; en otras palabras siempre será posible revisar una lista con los productos más vendidos, para luego poder establecer relaciones entre productos que estén ocultas en los registros que es la misión de la minería de datos. En nuestro caso los ficheros logs almacenan peticiones de objetos las cuales no siempre pueden ser asociadas a una transacción que es el caso del retail. Es posible por tanto sostener que la etapa de preprocesamiento de datos en si es una etapa que debe considerar factores asociados con el "análisis inteligente" de los datos con la finalidad de prepararlos para que sean sometidos a un nivel de abstracción mas alto.

El núcleo principal de cualquier proceso de web mining orientado a la personalización de un sitio web corresponde a la etapa de preparación de datos, etapa en que se fijan los objetivos del estudio y se restringe este a un dominio de conocimientos genérico. La personalización de un sitio web por ejemplo tiene por objetivo adaptarlo a las necesidades de un usuario en particular o de grupos de estos, luego la preparación de datos se orienta a detectar las necesidades de estos e inferir conocimiento en cuanto al uso de recursos, el uso o petición no indica si la página fue *percibida* o en un nivel de abstracción mas bajo *visitada*. En nuestro planteamiento consideramos que la percepción tiene una relación gradual con el grado de pertenencia a un conjunto difuso asociado a la visita, conjuntos definidos para realizar una estimación del tiempo que ocupo el usuario en su tránsito de un objeto a otro dentro de un mismo sitio; es destacable mencionar nuevamente el concepto de teoría del conocimiento web planteada como base de nuestro sistema formal para el análisis de los ficheros logs, el cual sostiene que el conocimiento esta relacionado con la percepción del objeto observado, conocimiento que finalmente es establecido por el sujeto que observa, por lo cual nuestra estimación corresponde a un grado de relevancia determinado a partir de la cantidad de objetos primitivos que son almacenados en la página web .y el tiempo de visualización

Mencionábamos en el capítulo anterior que los algoritmos de minería datos se enfrentan en el momento de determinar patrones desde los ficheros logs con el problema de la dimensión de los datos, situación que en la medida que estos crecen, decrece la eficiencia del algoritmo. Otro hecho no considerado por el web usage mining tradicional en términos generales, es el factor o característica de que los datos almacenados en los ficheros logs ***son dinámicos,*** en cuanto a su almacenamiento como también en cuanto a su registro, si tomamos por ejemplo el caso de un servidor web que sea visitado por múltiples robots de manera periódica, el resultado para ese servidor será el denominado ruido web que estará presente de manera aleatoria almacenado en los ficheros logs impidiendo determinar visitas, problema que se estima resuelve el algoritmo planteado en esta propuesta mediante un método de análisis estadístico dinámico de los tiempos de visita fijando estos en rangos medios de los valores visita corta, visita media y visita larga como valores reales, valores a los cuales convergerían los resultados de las ip o clientes u objetos analizados cuando la tendencia de almacenamiento de eventos en un fichero logs tienda a ser grande

En nuestra propuesta en primer lugar se basa en el concepto de que las visitas pueden ser asociadas a ciertos rangos de tiempo que hemos denominado visitas cortas, medianas o largas, fijándose estos tiempos tomando como referencia un estándar ampliamente aceptado en el mundo de la Medición de Audiencias en cuanto a fijar la estimación media de una sesión en tormo a los 25 minutos. Estos rangos de tiempo visita corta, media y larga y sus respectivas medias y valores de restricción que nos permiten descartar valores estimados como muy pequeños (mw) o muy grandes (MW), tienden a ser los valores aglutinadores por los cuales los objetos de estudio indicados en tablas convergen. El objetivo por tanto es aglutinar en torno a las medias de cada rango de tiempo fijado al objeto de estudio sea este un ip, un tipo de fichero o cualquier otro elemento que forma parte de la solución web como por ejemplo un índice que refiere a un objeto dinámico almacenado en una base de datos. , rangos que están asociados al nivel de abstracción o concepto visita corta, visita media o normal y visita larga o alta, los resultado obtenidos dinámicamente para los objetos de estudio.

A partir de los resultados obtenidos para el grupo de objetos de estudio contenidos en una tabla de restricción se puede estimar el tipo de visita, el grado de percepción de una página, y la relevancia de esta empleando la definición del nuevo parámetro "inteligente" que hemos denominado *PageView²* o :

PageView² → *P** { visita(difusa), Objeto (r), deltaT(conjunto difuso)}

Los objetos clasificados por el algoritmo Minero P*, pueden ser asociados a distintos conjuntos difusos o métodos de inferencia para dar respuesta a las preguntas bases de esta tesis, como por ejemplo determinar la página "efectivamente" más visitada, dado que esta será aquélla que tenga la máxima cardinalidad en el conjunto a la cual fue referida por el concepto visita corta, media o larga calculado por nuestro Minero P*.

1. *Limpieza de datos, eliminación de ruido web*
2. *Integración de ficheros logs a un Dominio de Conocimientos Genérico (rojo)*
3. *Creación de Listas Lp y Ln (Objetos Positivos y Objetos Negativos) (azul)*
4. *Aplicación de Algoritmo para determinación de visitas: corta, normal, alta (verde)*
5. *Método de prueba: ¿la página fue vista?*

4.4.3 Definición de Algoritmo Minero P*

Definición 7 Algoritmo *Minero P**

1. ***Hipótesis 1:***
   Se supone que existe un fichero logs Fi del formato ECL restringido en tamaño f y rotatorio en el tiempo (dinámico, semanal)
2. ***Hipótesis 2:***
   Nuestro analizador de Fi conoce lista de restricción de objetos Lp y Ln

Enunciado 1

Visualización de Objeto Pagina Web

Será asociado a la aparición de un evento del tipo $\mathbf{V= (W_i , IPi , Ti)}$ *referido al Objeto Contenedor Página Web W (Objeto_pg), visualizable mediante un navegador que encapsula un número arbitrario y finito de otras entidades y objetos cerrados los cuales almacenan información con grados de incertidumbre.*
El termino $\mathbf{W_i}$ *se refiere a la lista exhaustiva o restringida de objetos página web que forman parte del dominio de conocimiento*
El termino ***IPi*** *hace referencia a una contención lógica de enlaces o ligaduras sobre un escenario visual en donde interactúa un usuario real.(o usuario)*
El termino ***Ti*** *corresponde al instante de petición del objeto_pw, tiempo basado en la referencia de la fuente o time of year. (reloj de tiempo real).*

Enunciado 2

Duración de la Visualización de Objeto Pagina Web

Será la diferencia ***d(V)*** *entre el instante de la siguiente aparición de un evento* ***V1*** *referido a otro* ***Objeto_pw1*** *a partir del* ***Objeto_pw0*** *o referencia. Ambos objetos pertenecen al dominio o lista de restricción Lp.*
El valor resultante será referido a duración mínima o m_w y duración máxima o M_W.
Entendiendo que el usuario que visite por debajo m no le ha podido visualizar y por encima de M ha perdido la conexión con la fuente primaria ***S.***

En caso de que la medida d(v) < 2 entonces d(v) =0 y si M_W > 28 entonces asumiremos que d(V) = M_W lo que se puede interpretar como visita efectiva. Todos los valores en minutos

Enunciado 3

*Duración **Media** de Visualización de un Objeto Página Web*

*Será la media o **dm(W)** de los **d(V)** para todos los **V=(W, IP, T)** o media de visualización de un **Objeto_pw** . Este parámetro puede ser calculado aplicando métricas apropiadas al problema, es destacable mencionar que corresponde a un parámetro dinámico.*

Enunciado 4

*Duración Media de una Visualización **Larga** de un Objeto Página Web*

*Será la media o **dl (W)** de los **d(V)** para todos los **V=(W, *,*)** visualización de un Objeto_pw , con d(v) mayor que dm(W). Es decir es la duración media de los usuarios que visitan **S** durante un tiempo mayor que la media.*

Enunciado 5

*Duración Media de una Visualización **Corta** de un Objeto Página Web*

*Será la media la media o **dc(W)** de los d(V) para todos los V=(W, *,*) visualización de un Objeto_pw, d(v) menor que dm(W). Es decir es la duración media de los visitantes de **S** que permanecen un tiempo menor que la media*

4.4.4 Seudo Código de Algoritmo Minero P* para el calculo dinámico de dm, dc, dl

1. Sea una tasa de ajuste pequeña en torno a 0.01 o 0.1
2. Para cada objeto página web W Inicializar:
   Dm (W) = (M_W + m_w)/2 ; media del rango de visualización normal (30 minutos)

   dc (W) = (m_w + Dm)/2

   dl (W) = (M_W + Dm)/2
3. Recorrer el Dominio de datos, ficheros log integrados F_i = { F1, F2.....FN} y para cada entrada **V**
4. SI ***W*** pertenece a ***Lp*** es objeto_página web valida ENTONCES calcular ***d(V)***

   i. Si dm (W) < d(V),

      dl(W) ← dl(W) + e [d(V) – dl(W)] y

      Dm ← Dm(W) + e[d(V) – Dm(W)]

   ii. Si dm (W) > d(V),

      dc(W) ← dc(W) – e [dc(W) – d(V)] y

      Dm (W) ← Dm(W) – e [dm (W) – dm (V).

   SI NO

   i. No hacer nada

   FIN.

El algoritmo procesa las referencias de tiempo contenidas en los ficheros logs, para los objetos de interés contenidos en las listas de restricción negativas y positivas de manera dinámica, calculando la distancia de estas a la media. Cuando el número de entradas en el dominio de conocimiento tiende a infinito es decir las entradas del conjunto de ficheros logs o un fichero en el mejor de los casos, y considerando el dominio genérico de objetos restringido por medio de lista $\boldsymbol{L}$, se tiene que el calculo tiende a los valores correctos para dc, dm y dl para cada W o objeto de estudio. Este resultado tiene algunas implicancias significativas, independiza la fuente de origen de los ficheros logs permitiendo además su mezcla por medio del algoritmo definido par el procesamiento múltiple de ficheros log o de construcción del dominio de conocimiento genérico.

4.5 Estudio de Comportamiento de Usuarios

4.5.1 Aplicación Teórica de los Conceptos PageView2 y Minero P*

Plantearemos a continuación una arquitectura o modelo que permite utilizar las herramientas PageView^2 y Minero P* planteadas en la sección anterior, arquitectura orientada a determinar aquellas páginas de mayor uso por parte de los usuarios de un sitio web a fin de disponerlas en un almacenamiento secundario o cache de páginas. Es posible por medio de esta solución establecer adicionalmente comportamientos de usuarios con algunas pequeñas modificaciones a la arquitectura planteada..

Esta solución resuelve el problema de la complejidad al emplear conjuntos difusos definidos como navegación Baja o N-Baja, navegación Media o N-Media, navegación alta o N-Alta, usuario descarta sitio o No-se-Sabe, o N-Sabe, usuario se queda en sitio o Cliente-VIP o C-Vip, asociándolos al comportamiento de un usuario en el sitio web, se considera por ejemplo que el comportamiento de un usuario tienen que ver con el tiempo de conexión y con aquellas páginas visitadas. Se emplea el concepto de máxima cardinalidad para obtener el patrón lingüístico relacionado con la página, es decir la extensión del archivo (asp, htm u otra), patrón que es "minado" (extraído). La solución planteada presenta varias ventajas que pueden ser descritas como:

contempla análisis simultaneo (spool) en línea de los web logs, es decir es dinámica

se definen conjuntos que permitan establecer si un usuario descarta una página por algún motivo

dado que los archivos son voluminosos el tiempo de pre y post procesamiento *queda reducido a listas de restricción previamente establecidas.*

Permite establecer índices de calidad de páginas requeridas

4.5.2 Seudo Código Aplicación Minero P* y PageView[2]

A.1 Construcción de Dominio de Conocimientos Genérico

Aplicar Algoritmo de Procesamientos Múltiple de Ficheros Log (APMF.),

A.2 Construir Lista de Restricciones

Sea Lp =[*.asp, *.htm, *.html,] y Sea Ln =[Todos los demas]
Definir mw = 2 minutos, y MW=35 minutos

A.3 Inicio de Lectura de web logs y se aplica Minero P*

Minero P*

A.4 Etapa Fuzzificación: Se definen los siguientes conjuntos difusos

- *navegación Baja o N-Baja,*
- *navegación Media o N-Media*
- *navegación alta o N-Alta*
- *usuario descarta sitio o No-se-Sabe, o N-Sabe*
- *usuario se queda en sitio o Cliente-VIP o C-Vip*

El universo de discurso ***D*** de estos conjuntos corresponderá al campo "tiempo de navegación" o rango máximo almacenado en el archivo web Log, la pertenencia por tanto de un elemento genérico ***x*** perteneciente al dominio o contexto referido estará dada por los conjuntos:

N-Baja ={ x / x se encuentra entre 5 y 15 minutos, grado de pertenencia / x pertenece a D }

N-Media ={ x / x se encuentra entre 10 y 30 minutos, grado de pertenencia / x pertenece a D }

N-Alta ={ x / x se encuentra entre 25 y 40 minutos, grado de pertenencia / x pertenece a D }

N-Sabe ={ x / x se encuentra entre 0y 10 minutos, grado de pertenencia / x pertenece a D }

C-VIP ={ x / x se encuentra entre 35 y $n \in R_{estringido}$ *minutos, grado de pertenencia / x pertenece a R* $\subset$ *D }*

Las funciones de pertenecía a aplicar son del tipo triangulares, a fin de establecer una interpretación por medida de creencia

Como el dominio o contexto ***D*** es continuo los conjuntos difusos puede ser expresados en términos del total de los elementos que forman parte de este:

$$C_{difuso} = \int_{do\min io} G_p(x)/x$$

En donde en signo integral denota el conjunto de elementos ***x*** pertenecientes al dominio y su grado de pertenencia al conjunto difuso. Esta medida o cardinalidad será empleada para construir un ***Índice de Relevancia***, el cual esta asociado a aquel conjunto con máxima cardinalidad, permitiendo establecer un ranking de páginas.

El grado de pertenencia será establecido por medio de una función y será interpretado como una medida de creencia de que un elemento del dominio cumpla con el atributo.

A.5 Definición de Reglas de Inferencia

1. Si (patrón es Lp) y tiempo navegación CORTO Entonces salida es *N-Baja*
   A = {TIEMPO, nombre pagina / grado de pertenencia}
2. **Si** (patrón es **Lp**) y tiempo navegación es MEDIO **Entonces** salida es *N-Media*
   B = {TIEMPO, nombre página / grado de pertenencia}
3. **Si** (patrón es **.Lp**) y tiempo navegación ALTO es **Entonces** salida es *N-Alta*
   C = {TIEMPO, nombre pagina / grado de pertenencia}
4. **Si** (patrón es **.Lp**) y tiempo navegación es menor que mw **Entonces** salida es *N-Sabe*

   D = { TIEMPO, nombre pagina / grado de pertenencia}
5. **Si** (patrón es **Lp**) y tiempo navegación es mayor que MW **Entonces** salida es *C-VIP*

   E = {TIEMPO, nombre pagina / grado de pertenencia}

A.6 Definición de Reglas de Inferencia

Aplicamos máxima cardinalidad a los distintos conjuntos difusos, obteniendo como resultado. Pagina mas visitada, menos visitada y paginas de acceso normal

. $C_{difuso} = \int_{do\min io} G_p(x)/x$

A.7 Definición de Base de Conocimiento

Se asigna página de máxima cardinalidad a servidor proxy cache

A.8 Repetir ciclo

4.5.3 Aplicaciones Posibles Personalización de Sitios Web

El sistema planteado con anterioridad en donde se destaca el algoritmo Minero P¨* asociado a conjuntos difusos, puede ser empleado para dar respuesta a una variada cantidad de preguntas habituales en el ámbito de la personalización de una solución web, tanto en sus aspectos de administración productiva eficiente, como en el contexto de la empresa que desea saber el comportamiento de sus usuarios para una correcta toma de decisiones. Hemos mencionado que las características de la información contenida en los ficheros logs, es la incertidumbre y la imprecisión, luego la metodología que hemos escogido para su estudio, se relaciona a resolver los problemas que plantea su análisis de manera similar a como los seres humanos resuelven problemas similares.

De manera habitual la resolución de problemas del mundo real, se refiere a encontrar un método en el cual a partir de una condición inicial se obtenga un objetivo, este objetivo puede ser representado por ejemplo como un proceso de aprendizaje o la obtención de conocimiento artificial a partir de un dominio de datos o conocimiento. Nuestro dominio de conocimientos constituido por los ficheros logs crudos contienen variada información y son de difícil análisis para un experto humano dado el gran volumen de datos que estos contienen, a modo de ejemplo podemos indicar que el tamaño de estos varia desde varios megas diarios, hasta cantidades inmanejables para periodos mensuales incluso para una computadora personal de mediano tamaño. Un procedimiento habitual por parte de los administradores de los servidores web es fijar el tamaño máximo de estos archivos, a fin de no consumir recursos indispensables

de almacenamiento poniendo en peligro la estabilidad del sistema operativo; esta práctica normal agrega nuevas dificultades al proceso de análisis de los datos crudos contenidos en los archivos de registros, problema que se refleja en la perdida de información vital para determinar el comportamiento de los usuarios de un sitio web en un momento dado, que se produce al caducar la cuota de espacio en disco asignada y los eventos posteriores que no son grabados. Es en este punto en donde se justifica plenamente el análisis automatizado de estos archivos por medio del un motor de pre-procesamiento el cual ordene la información de interés eliminando aquellas referencias sin importancia, de un formato de mejor calidad que permita establecer reglas de asociación entre los patrones dependiendo de la clase de conocimiento que se desea obtener, y sincronice la captura de la información desde distintas fuentes manteniendo la calidad de los datos capturados.

El ejemplo anterior y su seudo código puede ser aplicado por ejemplo para la clasificación de páginas en un ranking de las "efectivamente más visitadas", dado que estas se determinan a partir de la definición de visita corta, media, larga.

Con unas pequeñas modificaciones el algoritmo Minero P* puede determinar el peso de una página, el cual dependerá entre otras cosas del stress del servidor y de la cantidad de objeto que esta contiene transformándose en este caso en *deltaT(difuso),* permitiendo la aplicación del parámetro *PageView²* de acuerdo a reglas como las siguientes:

1. *Sea PageView²* → *P** { visita(difusa), Objeto (r), deltaT(conjunto difuso)}
2. Vt = visita(difusa),← Minero P* corta, mediana, larga
3. Pt = deltaT(conjunto difuso)}← Minero P*, pocos , muchos, demasiados
4. Luego *PageView²* ←*(Vt, Pt) ENTONCES*

   Si Vt es CORTA y Pt es ALTO entonces *PageView² es bajo, lo que implica que pagina no fue vista*

   Si Vt es ALTA y Pt es BAJO entonces PageView² es alto, lo que implica que página fue vista y el grado de percepción es alto.

Emplearemos la metodología desarrollada a lo largo de esta tesis para ser aplicada en un proceso de web mining orientado a determinar característica derivadas de los ficheros log,

características genéricas como por ejemplo: el tiempo promedio de visita a un objeto página web corresponde a X y su grado de percepción es Y. Es sistema planteado con anterioridad y pude ser empleado para dar respuestas a las preguntas siguientes:

¿Cuantos visitantes realizaron visitas cortas, medianas o largas a la solución web?

¿Cuáles son las paginas que suele visitar un usuarios con visita media?

¿Cuál es la pagina que tiene mas visitas medias...?

¿la página fue vista o percibida

Las preguntas anteriores tienen una respuesta genérica común: la verdad o certeza, conocer la verdad o tener certeza de algo no es una tarea fácil, sino mas bien se puede tener ciertos grados de credibilidad respecto a las respuestas posibles por tanto estas tienen directa relación con los grados de pertenencia de una variable a un conjunto difuso.

4.5.4 Un pequeño ejemplo ilustrativo

La metodología de anterior ha sido aplicada al sitio web de la revista Mathware & Soft Computing, (http://docto-si.ugr.es/Mathware/ENG/mathware.html), a fín de revisar el índice de audiencia de cada una de las páginas del sitio, y analizar si están cumpliendo su función.

Éste sitio web dispone de página principal con una información muy escueta sobre la revista, y con enlaces a distintas secciones o subpáginas, entre las que hemos estudiado:

- Editorial.
- Información de carácter general.
- Información para autores.
- Contenidos de los números publicados.

Los resultados se han obtenido estudiando el fichero log del sitio desde primeros del año 2006, y eliminando los accesos propios del mantenimiento del sitio, y con los filtros típicos de eliminación de ruido. Los resultados obtenidos quedan reflejados en la siguiente tabla:

	Page View	Visita Corta	Visita Media	Visita Larga	PageView2 (corta,media,larga)
Homepage	910	(0,1,1.6,1.8)	(1.6,1.8,2.2,2.4)	(2.2,2.4,-,-)	853.9 (78.6,797.9,.33.5)
Editorial	73	(0,2.1,2.7,3.4)	(2.7,3.4,5.8,6.2)	(5.8,6.2,-,-)	72,0 (8.7,34.8,29.5)
Informacion general	526	(0,2.3,2.9,3.1)	(2.9,3.1,3.9,4.2)	(3.9,4.2,-,-)	373,3 (198.4, 302.6,25)
Información autores	113	(0,2.1,2.8,3.1)	(2.8,3.1,4.1,4.4)	(4.1,4.4,-,-)	61,1 (67.7,34.1,11.2)
Contenidos números	692	(0,1.3,1.7,1.9)	(1.7,1.9,2.2,2.4)	(2.2,2.4,-,-)	648,38 (81.1,504.6,106.3)

El análisis semántico completo de estos resultados corresponderá a los gestores del sitio, pero aquí si podemos extraer algunas conclusiones:

- El orden de relevancia del índice de percepción (pageview2) puede resultar significativamente distinto del índice de visitas (pageview).
- Los conceptos de visita corta, media y larga resultan distintos según la página, y podemos intuir que dependerá fundamentalmente de los contenidos de la misma.
- Existen páginas donde número de visitas e índice de percepción son bastante similares, y páginas donde son muy distintos.

En definitiva, pensamos que este ejemplo ilustra la utilidad de los conceptos y algoritmos desarrollados, puesto que añaden información que entendemos resulta de utilidad con

respectos a los sistemas clásicos de índices de audiencias. Al menos, así ha resultado para los encargados del sitio web analizado en este ejemplo.

CAPÍTULO 5: *UN MODELO LINGÜÍSTICO DE ANÁLISIS DE WEB LOGS*

En esta tesis ha considerado dentro de sus objetivos establecer o definir un sistema formal basado en términos lingüísticos que permitan asociarlos a procesos automatizados de análisis de datos no estructurados contenidos en los denominados ficheros logs. Con este objetivo se han definido una serie de términos, conceptos e hipótesis de trabajo los cuales serán resumidos en el presente capítulo con la finalidad de ser aplicados a un nuevo modelo de análisis de web logs basado en el paradigma web inteligente con el objetivo de implementar un proceso orientado al descubrimiento de patrones lingüísticos y otras características derivadas.

Este modelo corresponde esencialmente al desarrollo de una metodología que permita manejar el concepto de *imprecisión e incertidumbre* en base a *términos lingüísticos*; metodología que puede ser resumida en las siguientes etapas en un proceso de Obtención de Conocimiento:

1. *Construcción del Dominio de Conocimientos Genérico*
2. *Fuzzy Web Mining o Análisis Inteligente de Sitios Web.*
3. *Descubrimiento de Conocimiento Racional*
4. *Sistemas de Interpretación del Conocimiento o Visualización*

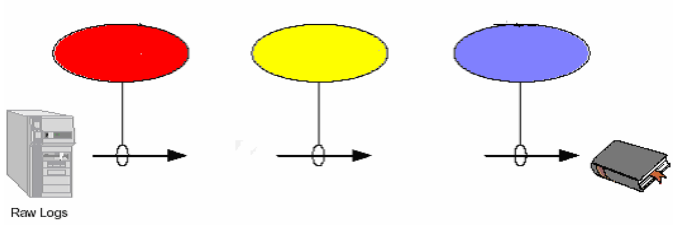


Figura 5.1 Etapas de Un Proceso de Obtención de Conocimiento

5.1 Definiciones para la Obtención de Conocimiento Racional

Un sistema basado en *inteligencia*, establecerá su entorno en el análisis de un dominio de conocimientos genérico para la obtención autónoma de conocimiento racional; es decir en construir relaciones entre conceptos y símbolos por medio del empleo de términos y conceptos lingüísticos con elementos de ese dominio en particular y de los dominios restringidos de datos que puedan ser establecidos como subconjuntos del principal; este sistema formal permitirá el análisis inteligente de la información o datos perteneciente al dominio y con estos elementos establecer semejanzas o patrones entre objetos diferentes, y por medio de estas semejanzas se pueda establecer una clasificación y construir clases distintas de objetos, sujetos, géneros o especies; por medio del empleo de procedimientos heurísticos o inferenciales para la resolución de problemas del mundo real. El sistema formal planteado se basa en las siguientes definiciones, de las cuales es posible establecer teorías, heurísticas, algoritmos, y otros métodos con el objetivo de descubrir conocimiento del dominio datos genérico seleccionado:

Definición 1

> *"Sea B_Interntet un sistema formal o base de datos virtual B que consta de los siguientes elementos:*
>
> *Un conjunto numerable y finito de objetos primitivos, que contienen información no estructurada basada en símbolos interpretables para sistemas complejos sujetos reales(animales) o virtuales.*
>
> *Un conjunto (finito) de objetos contenedores que agrupan objetos primitivos y determinan bajo qué condiciones podemos afirmar que un conjunto de objetos primitivos es (o no) una fórmula (página web). El conjunto* ***S*** *de las fórmulas se denomina fuente primaria (o sitio web).*
>
> *Un conjunto(finito) de fuentes primarias* ***S ,*** *que definen reglas, métodos o sistemas de acceso a los distintos objetos de información que estas almacenan. Combinatorias que sirven para producir deducciones o interpretaciones formales sobre los sujetos que visualizan sobre una ventana virtual los objetos de información (i.e., determina qué secuencias de fórmulas o páginas web constituyen conocimiento bajo la forma de una deducción o percepción del objeto visualizado posible de obtener del sistema).*

Estas reglas normalmente incluyen la aceptación como verdaderas de un conjunto finito de sentencias (i.e. fórmulas sin variables libres) que reciben el nombre de principios del sistema.(consecuencia: lenguaje natural de acceso a objetos web)

Las sentencias del sistema . El conjunto de sentencias deducibles se llama Teoría Formalizada o Teoría del Conocimiento Web.

Definición 2

Dado el sistema formal B o base de datos distribuida que almacena objetos primitivos no-textuales los cuales contienen información visualizable γ es deducible del sistema la existencia de conocimiento, decimos que A es una consecuencia propia del sistema (semántica, visual, auditiva...etc) y se denota como sigue:

A $\rightarrow \gamma$ *. Si* γ *es una afirmación verdadera en cualquiera de las posibles interpretaciones del sistema formal, diremos que se trata de una consecuencia racional de* ***A*** *y lo denotaremos como* ***A*** $\models \gamma$ *o* ***verdad racional****.*

Definición 3

Sea una solución web S implementada por varios servidores y sean Fi los ficheros almacenados en cada uno de estos, el dominio de conocimiento genérico que reúne a estos esta dado por:

Dci = { F1, F2, F3,.....Fn} ; en donde

Fi = (Evento1, Evento2, ...Evento$_n$), y se determina por medio de

Algoritmo 1 : Algoritmo para Construcción de Dominio de Conocimiento

Pseudo Código : Procesa Múltiples Ficheros Logs y Salida Dominio $D_{c\_generico}$

*Sea C el conjunto deducible de visitas o sesiones del Dominio_* ***C_c*** *y sea* ***T*** *el reloj de tiempo real empleado para el registro de eventos, en donde* ***S*** *se implementa con servidores múltiples, tenemos que:*

S *= {(servidor1, Ta), (servidor2, Tb), ...(servidorn, Tn),* F_i *(de formatos, granularidad)}*

Si Ta distinto Tb distinto Tc ENTONCES sesión indeterminada
Proceso de forma independiente cada F
i
Si Ta = Tb = Tc ENTONCES →procesar formato

Formato iguales ENTONCES → procesar granularidad
SI NO

Proceso cada/ formato de forma independiente

FIN

Definición 4

Una visualización de un objeto o percepción del mismo(PageView^2) puede ser entendida como una actividad imprecisa basada en el tiempo realizada por un sujeto sobre un objeto almacenado en una fuente primaria ***S.*** *Actividad que es registrada bajo la forma de un evento o suceso referido a una base de tiempo real o virtual*

PageView^2 → P { visita(difusa), Objeto (r), deltaT(conjunto difuso)}*

Definición 5

Un evento puede ser entendido como el resultado de una operación realizada por una fuente primaria ***S*** *como respuesta a una petición realizada por un sujeto real o virtual en un instante de tiempo* ***T****.*

Definición 6

Definición de Objeto Página Web y Sitio Web

Objeto Contenedor Página Web (Objeto_pg), será todo archivo vísualizable mediante un navegador que encapsula un número arbitrario y finito de otras entidades y objetos cerrados los cuales almacenan información con grados de incertidumbre. El termino contención hace referencia a una contención lógica de enlaces o ligaduras sobre otros objetos contenedores, de escenarios visuales y de escenarios virtuales por medio de los cuales se implementa la ventana virtual sobre la cual "una sesión real de usuario" interactúa en un instante de tiempo o visita sobre una solución web ***S*** *o sitio web.*

Definición de Objeto Página Web (definición imprecisa)

Objeto Contenedor Página Web (Objeto_pg), será todo archivo vísualizable mediante un navegador que encapsula un número arbitrario y finito de otras entidades y objetos cerrados los cuales almacenan información con grados de incertidumbre. El termino contención hace referencia a una contención lógica de enlaces o ligaduras sobre otros objetos contenedores, de escenarios visuales y de escenarios virtuales por medio de los cuales se implementa la ventana virtual sobre la cual "una sesión real de usuario" interactúa en un instante de tiempo o visita sobre una solución web S o sitio web.

Sea P_w el conjunto de todos los objetos contenedores del tipo ***Obpg (página web)*** , objetos-páginas por medio de los cuales se implementa una solución web la cual define un dominio de discurso cerrado ***S*** en donde:

S → ***{ Sitio Web o Solución Web }***

$S = \{Pw_1, Pw_2, Pw_3, \ldots Pw_{n-1}, Pw_n\}$

El objeto contenedor $\boldsymbol{P_w}$ tiene sus limites bien definidos sobre la ventana virtual que implementa es decir $\boldsymbol{P_w}$ es directamente proporcional a la ventana virtual que es implementada por medio de este objeto y almacena un número finito de objetos cerrados cuyas clases se definen como:

P_w = [URL_s , *ficheros* $_{(tipo)}$, *otras clases por definir*] o

$P_w = \{ obj_{clase\_0}, obj_{clase\_1}, obj_{clase\_2}, \ldots\ldots obj_{clase\_Z} \}$

en donde el objeto contenedor página web $\boldsymbol{P_w}$ quedará definido por :

$$Pw = \sum_{i=1}^{a} obj_{(i,a)} \qquad (1$$

5.1.1 Hipótesis General:

Sea R el conjunto de todas las visitas factibles de ser determinadas de un fichero logs F_i *por medio de una heurística* **H** *definida a partir del dominio de datos* $\boldsymbol{D_i}$ *. El objetivo de la heurística* **H** *es reconstruir una visita de usuario a partir de la información almacenada en el fichero web log y de esta visita o conjunto de sesiones reales de usuario determinar relaciones, vínculos, usabilidad, percepción de los objetos solicitados con la finalidad de establecer conocimiento propio del dominio de datos seleccionado.*

Definición 7 Algoritmo *Minero P* para el Procesamiento de datos de un Dominio de datos.*

Algoritmo2 : Algoritmo para la determinación de visitas de un sitio web

3. *Hipótesis 1:*
   Se supone que existe un fichero logs Fi del formato ECL restringido en tamaño f y rotatorio en el tiempo (dinámico, semanal)

4. *Hipótesis 2:*
   Nuestro analizador de Fi conoce lista de restricción de objetos Lp y Ln

Enunciado 1

Visualización de Objeto Pagina Web

Será asociado a la aparición de un evento del tipo $\boldsymbol{V = (W_i, IPi, Ti)}$ *referido al Objeto Contenedor Página Web W (Objeto_pg), vísualizable mediante un navegador que encapsula un número arbitrario y finito de otras entidades y objetos cerrados los cuales almacenan información con grados de incertidumbre.*
El termino $\boldsymbol{W_i}$ *se refiere a la lista exhaustiva o restringida de objetos página web que forman parte del dominio de conocimiento*
El termino **IPi** *hace referencia a una contención lógica de enlaces o ligaduras sobre un escenario visual en donde interactúa un usuario real.(o usuario)*
El termino **Ti** *corresponde al instante de petición del objeto_pw, tiempo basado en la referencia de la fuente o time of year. (reloj de tiempo real).*

Enunciado 2

Duración de la Visualización de Objeto Pagina Web

*Será la diferencia **d(V)** entre el instante de la siguiente aparición de un evento **V1** referido a otro **Objeto_pw1** a partir del **Objeto_pw0** o referencia. Ambos objetos pertenecen al dominio o lista de restricción Lp.*
El valor resultante será referido a duración mínima o m_w y duración máxima o M_W.
*Entendiendo que el usuario que visite por debajo m no le ha podido visualizar y por encima de M ha perdido la conexión con la fuente primaria **S.***

En caso de que la medida d(v) < 2 entonces d(v) =0 y si M_W > 28 entonces asumiremos que d(V) = M_W lo que se puede interpretar como visita efectiva. Todos los valores en minutos

Enunciado 3

*Duración **Media** de Visualización de un Objeto Página Web*

*Será la media o **dm(W)** de los **d(V)** para todos los **V=(W, IP, T)** o media de visualización de un **Objeto_pw** . Este parámetro puede ser calculado aplicando métricas apropiadas al problema, es destacable mencionar que corresponde a un parámetro dinámico.*

Enunciado 4

*Duración Media de una Visualización **Larga** de un Objeto Página Web*

*Será la media o **dl (W)** de los **d(V)** para todos los **V=(W, *,*)** visualización de un Objeto_pw , con d(v) mayor que dm(W). Es decir es la duración media de los usuarios que visitan **S** durante un tiempo mayor que la media.*

Enunciado 5

*Duración Media de una Visualización **Corta** de un Objeto Página Web*

*Será la media la media o **dc(W)** de los d(V) para todos los V=(W, *,*) visualización de un Objeto_pw, d(v) menor que dm(W). Es decir es la duración media de los visitantes de **S** que permanecen un tiempo menor que la media*

5.1.2 Seudo Código de Algoritmo Minero P* para el calculo dinámico de *dm, dc, dl*

5. Sea una tasa de ajuste pequeña en torno a 0.01 o 0.1

6. Para cada objeto página web W Inicializar:
Dm (W) = (M_W + m_w)/2 ; media del rango de visualización normal (30 minutos)

dc (W) = (m_w + Dm)/2

dl (W) = (M_W + Dm)/2

7. Recorrer el Dominio de datos, ficheros log integrados F_i = { F1, F2.....FN} y para cada entrada **V**
8. SI ***W*** pertenece a ***Lp*** es objeto_página web valida ENTONCES calcular ***d(V)***

   i. Si dm (W) < d(V),

      dl(W) ← dl(W) + e [d(V) – dl(W)] y

      Dm ← Dm(W) + e[d(V) – Dm(W)]

ii. Si dm (W) > d(V),

dc(W) ← dc(W) – e [dc(W) – d(V)] y

Dm (W) ← Dm(W) – e [dm (W) – dm (V).

SI NO

ii. No hacer nada

FIN.

5.2 Descripción de las Distintas Etapas o Procesos: Un ejemplo simple.

Se aplicara la metodología expuesta en el punto 5.1 del presente capítulo, en un ejercicio de Fuzzy Web Mining con el objetivo de determinar u obtener patrones lingüísticos de una solución web básica o simple similar a la descrita en la Figura 5.2, la solución web esta constituida por tres objetos contenedores, los cuales almacenas una serie de objetos primitivos.

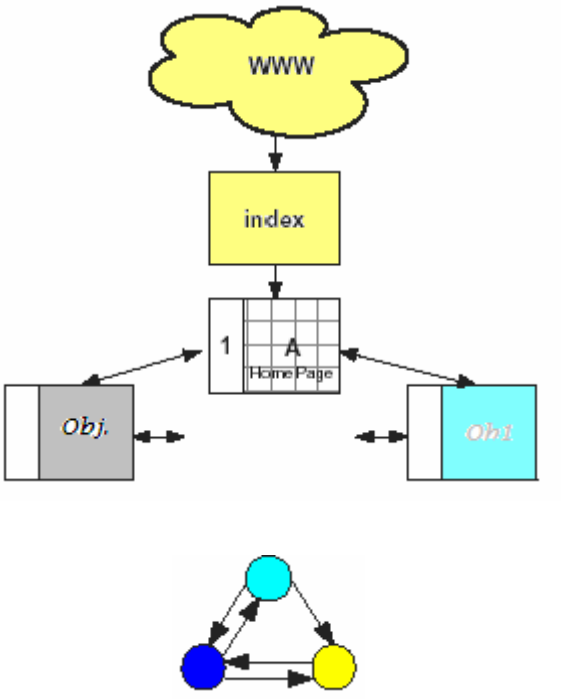


Figura 5.2 Solución Web Básica o Simple

5.2.1 Etapa Construcción del Dominio de Conocimientos Genérico (*Dominio_Cg*)

La etapa de Construcción del Dominio de Conocimientos Genérico es el foco principal para el desarrollo de esta tesis y esta podría se resumida en dos grandes tareas Integración de Datos provenientes de distintas fuentes y Transformación de estos con el objetivo de construir el

dominio de conocimientos genérico, sobre el cual se aplicaran procedimientos de descubrimiento de conocimiento. Esta etapa analizada en las secciones anteriores esta basada en el *Algoritmo 1* de reunión e integración de datos en donde se aplica la *Definición 3*. Esta definición permite restringir el dominio universal o global solo a la información de interés para el descubrimiento de conocimiento asociado al problema a resolver. Esta tarea o proceso corresponde principalmente a la selección, importación e integración de los ficheros logs mediante un programa o motor de preprocesamiento determinado por el *Algoritmo 1*, las tareas de este motor se refieren a empaquetar logs y después inferir reglas o métricas sobre los datos contenidos en estos archivos; este proceso de recorte reduce el tiempo de procesamiento refinado y permite por ejemplo fusionar varios tipos de logs relativos a la actividad que un usuario a tenido en el sitio web, permitiendo incluso relacionar múltiples servidores web. Dado que la integración se realiza con datos provenientes de servidores múltiples, y de acuerdo a definición de evento *Definición 3,* es necesario referir estos a una base de tiempo única con el objetivo que todos los eventos relacionados con las fuentes que resuelven la petición de un sujeto tengan la misma referencia. El tiempo de agrupación de y limpieza de datos bajo un dominio de conocimientos es:

Dcgenérico = { F1, F2, F3,.....Fn} → bajo y tratándose del ejemplo

Para valores reales obtenidos de un Sitio de Referencia, con política fijada para los ficheros logs de ser rotatorios con frecuencia semanal el tamaño de los ficheros logs estudiados toman un valor cercano a 20Megas (Fuente UVirtual UTEM) y para un política rotatoria mensual alcanzan un tamaño máximo de 340Megas con datos crudos, esta diferencia en el tamaño es dependiente del comportamiento de los usuarios del sitio, dado que el valor máximo se obtiene en Julio del 2006 fecha peak o fin de semestre académico

Dcgenérico = { F1} → 0, para un rango de 20 Megas ; luego construimos tablas de restricción Lp y Ln, y para nuestro ejemplo

Sea Lp = {Objeto1, objeto2, objeto3, objetos contenedores] y Ln ={robots}

Una vez construido el dominio de conocimientos genéricos o *Dominio_Cg* y *tablas de restricción* aplicamos el algoritmo indicado en la *Definición 7* al cual hemos denominado *Minero P** este nos permite seleccionar objetos, sujetos y relacionarlos posteriormente sobre una serie de conjunto difusos que permitan responder l preguntas como:

¿Cuál es la pagina más visitada?

¿La página fue percibida o vista?

¿Cómo es el comportamiento de los usuarios relacionados con los contenidos, o cuales son las características de las visitas que estos establecen con el sitio. Es decir tipo de visita: corta, media o larga?. Es destacable mencionar que este algoritmo esta orientado a resolver dos problemas de manera principal: la alta dimensionalidad de los datos contenidos en los ficheros logs o su tamaño y el hecho de que el almacenamiento de los eventos es dinámico o cambiante en el tiempo, situación considerada por el algoritmo.

Si la tendencia del número de entradas del *Dominio_Cg* es crecer hacia infinito el calculo obtenido por medio de Minero P* , se ajustan a los valores correctos de las medias ***dm(W)***, ***dl(W)***, ***dc(W)*** indicadas en los *Enunciados 3,4,5* de *Minero P**

A modo de ejemplo para el tamaño fijado en 20MG el tiempo de procesamientos es bajo. Dado que la cantidad de registros de eventos almacenados es "reducida" en el caso de selecciona a un cliente ip.con el objetivo de determinar visitas que este realiza A modo de referencia para uno de los ficheros estudiados el valor de los registros de eventos para una ip alcanza 391305 como se indica en la Figura 5.

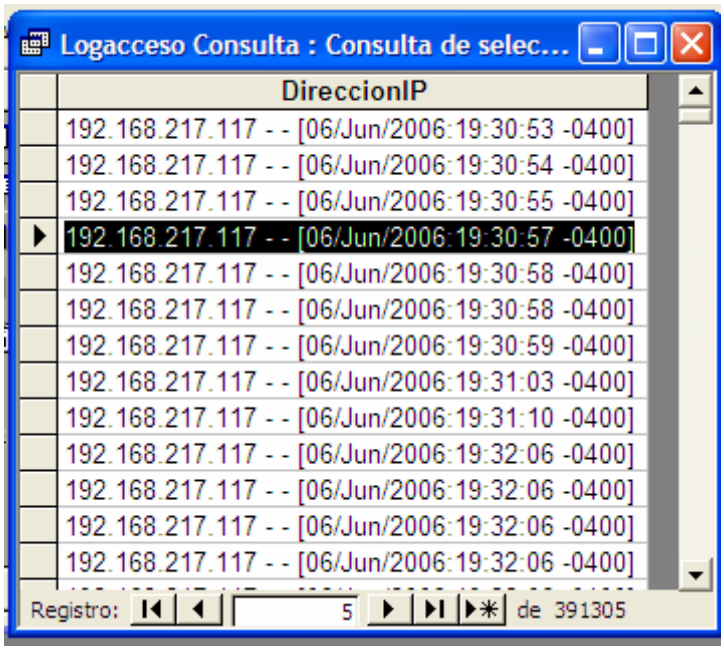


Figura 5.3 Cantidad de Registros Almacenados para una IP (ip →visita corta)

El tiempo empleado por Minero P en determinar los valores de visita corta, media y larga valores considerando el tamaño de referencia para un Dominio genérico de datos de 20Megas → a ser bajo o despreciable*

5.2.2 Fuzzy Web Mining, Determinación de Patrones Lingüísticos del Sitio Web Ejemplo Básico

En nuestro ejemplo esta t etapa se refiere a la determinación de *características derivadas* del *Dominio_Cg* , como por ejemplo el objeto de mayor uso de la solución web ***S** es **W**$_1$ y la mayoría de la visitas sobre este objeto es breve*. En resumen el objeto de esta etapa es determinar patrones lingüísticos asociados a los objetos; patrones que quedan restringidos a las listas L_p y L_n utilizadas por Minero P* .

Se consideran para este análisis aquellos ficheros considerados como importantes para la solución web como por ejemplo ***.asp, .htm, .html, .jva, .cgi*** u otros de importancia que entreguen información del comportamiento de los usuarios los cuales pasan a forma la lista $\boldsymbol{L_p}$. Archivo como los .jpg o .gif, se consideran dentro de las páginas requeridas pudiendo ser omitidos o recortados del archivo de salida en el proceso de reunión y pasan a integrar la lista $\boldsymbol{L_n}$. El criterio de selección de los archivos evidentemente se refiere al conocimiento que deseamos obtener, para nuestro caso de estudio el objetivo es determinar aquellas páginas relacionadas con visitas cortas, normales o altas al sitio web con la finalidad de establecer mejoras en o personalización del sitio web. Como mencionábamos anteriormente en la etapa de pre-procesamiento o reunión de datos, es posible analizar más de un logs a fin de construir mejores reglas de inferencia del tipo ***Si x es A$_i$ Entonces y será C$_j$***, basándose para su construcción en el tipo de conocimientos deseado. Las reglas pueden ser del tipo asociativo, reglas de clasificación, reglas de agrupamiento, o reglas útiles para la asociación de patrones o secuencias de estos. Estas reglas de inferencia se basan en conjuntos difusos asociados a los términos lingüísticos como por ejemplo: *visita _ corta*, *visita _ media* o normal y *visita _ larga* o alta, términos que pueden ser definidos para establecer ciertos grados de depuración o fineza de los resultados o de las reglas de inferencia consideradas como respuesta al problema planteado. Para nuestro ejemplo definiremos tres conjunto difusos asociados a una base de tiempo referida como dominio de discurso. Este valor se fijo en tres horas para nuestro sitio de prueba, dado que corresponde a una solución de e-learning.:

1. Visita corta será aquella perteneciente al rango [0, 70] minutos

2. Visita media será aquella perteneciente al rango [10, 120] minutos

3. Visita media será aquella perteneciente al rango [80, sin definir] minutos

4. mw = 10 minutos y MW = 120 minutos.

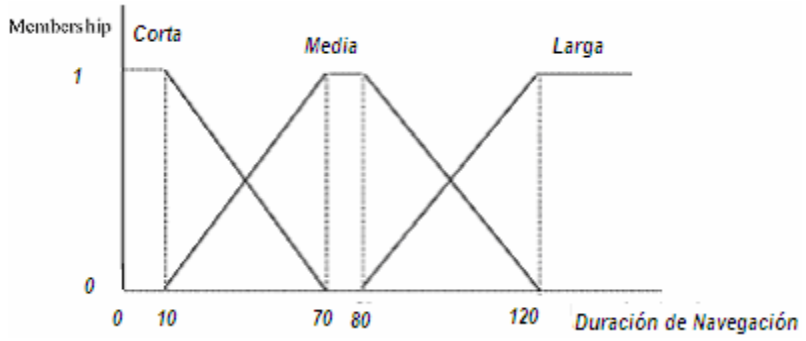


Figura 5.4. Conjunto Difusos Visita Corta, Visita Media o Normal, Visita Larga

Refiriéndonos al problema de determinación de el uso de los recursos de un sitio web estos conjuntos podrían tomar valores más depurados estableciendo un grado de fineza para el tiempo considerado como visita normal el cual puede ser representado por visita normal: pequeña, mediana, larga.

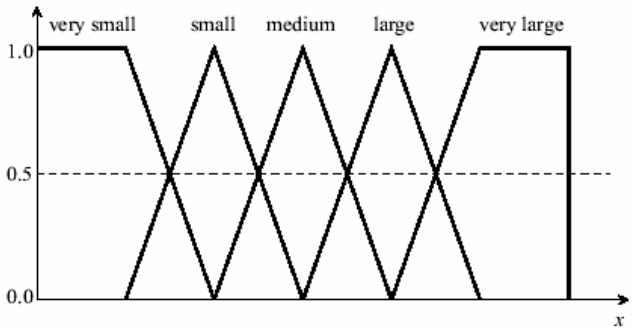


Figura 5.5.Depuración de Conjuntos Difusos orientándolos a Percepción

Esta depuración tiene por objetivo determinar la correcta percepción del objeto visualizado, estableciendo reglas como las siguientes:

1. SI *visita normal es muy pequeña* ENTONCES *objeto página_w no fue visto* y percepción es nula o PageView^2 → 0

2. SI *visita normal es pequeña* ENTONCES *objeto pagina_w fue visto* y percepción del mismo es baja o PageView^2 → dc(W)

3. SI *visita normal es media* ENTONCES *objeto pagina_w fue visto y percepción del mismo es normal.* PageView^2 → dm(W)

4. SI *visita normal es alta* ENTONCES *objeto pagina_w fue visto y percepción del mismo es normal.* PageView^2 → dl(W)

5. SI *visita normal es muy alta* ENTONCES *objeto pagina_w fue visto y percepción del mismo es normal. y usuario perdió la conexión, luego PageView^2 → outliers.*

5.3 Extensión a otros conceptos y cuestiones

Los conjuntos difusos han sido utilizados para definir conceptos simples o primitivos. Estos conjuntos difusos le asocian a cada elemento del universo del dominio de conocimientos genérico construido previamente un grado de pertenencia que es un número entre 0 y 1 de ese conjunto al objeto de estudio. Lo cual es una correspondencia (o función) que a cada elemento del universo le asocia su grado de pertenencia a los grupos de conjunto definidos. Luego estos conjuntos fueron establecidos con el objetivo de determinar el comportamiento de los usuarios de una solución web con el objetivo de adecuarla o personalizarla a sus pautas de comportamiento. La definición de estos grupos de conjuntos se basan en dos conceptos: el *concepto de visita* y el *concepto de percepción* de los objetos requeridos por parte del sujeto visitante. Identificando a todos aquellos elementos o términos que se aproximan en cierto grado a pertenecer a un conjunto y que cumplen la condición de *visita* al sitio web y la condición de *percepción / visualización* de los objetos requeridos, representado en este caso por el grado de pertenencia G_p del elemento al conjunto por medio del cual expresamos la posibilidad de asignar más valores de verdad que *Falso* o *Verdadero*. Este G_p puede ser expresado por medio de una función de referencia acotada al dominio, función que refiere a un valor en el intervalo [0,1]; y el conjunto de los infinitos pares ordenados { x, G_p (x) } indican el grado de similitud o de pertenencia de un elemento genérico x perteneciente al dominio o contexto referido. Como el dominio o contexto *Dominio_Cg* es continuo en nuestro caso, el conjunto difuso puede ser expresado en términos del total de los elementos que forman parte de este o:

$$C_{difuso} = \int_{do\min io} G_p(x)/x$$

En donde en signo integral denota el conjunto de elementos ***x*** pertenecientes al dominio y su grado de pertenencia al conjunto difuso y nos permite responder preguntas como: ¿Cuántos visitantes realizaron una visita corta, mediana o normal o larga?.

El grado de pertenencia a los conjunto difusos definidos puede ser interpretado dependiendo del contexto o problema a resolver y algunas posibles interpretaciones pueden hacer referencia a un atributo de la *"tupla_3"* por medio de una *medida de creencia* en cuanto a la verdad este atributo seleccionado, una *proporción* en la que se posee dicho atributo, una *probabilidad* de que un elemento del dominio cumpla con el atributo:

Proporción: Si visita es corta, ENTONCES para el sujeto_ip o **Xip** la expresión indicada por 100 • $\mathbf{G_p}$ (x) es el porcentaje con el que **Xip** a realizado visitas cortas al sitio web **.** *Es decir la proporción en la que el sujeto identificado por la ip posee el atributo de pertenecer al conjunto visita corta.* Es posible una interpretación adicional considerando operaciones como el *Complemento*, la *Intersección* o la *Unión* operaciones en donde los grados de pertenencia se interpretan como proporciones. El *complemento* de un conjunto difuso ***D****(visita)* asigna a cada objeto **x** el grado complementario: $g_{\overline{D}}(x) = 1 - g_D(x)$. La *intersección*. dos conjuntos difusos **D, E** asocia el mínimo de los grados de pertenencia, es decir para cada objeto x: $g_{\overline{D} \cap E}(x) = Min[g_D(x), g_E(x)]$ y por ultimo la *unión* de dos conjuntos difusos **D, E** asocia el máximo de los grados de pertenencia, es decir, para cada objeto x $g_{\overline{D} \cup E}(x) = Max[g_D(x), g_E(x)]$

Si consideramos dar respuesta a la pregunta *¿Cuáles son las páginas que suele visitar un usuarios?,* el contexto de la respuesta cambia a determinar el comportamiento de un sujeto real; este comportamiento puede ser interpretado para el caso de una solución web como e_commerce una tendencia de comportamiento o de compra en donde se pueden establecer respuestas como:

SI sujeto_ip visita normal(Wpg) ENTONCES es probable que compre objeto_C

Probabilidad : *Considerando que P es un evento probabilístico o variable aleatoria en el dominio, entonces cada par ordenado { x,* $\boldsymbol{G_p}$ *(x) }, es la probabilidad de que* ***x*** *ocurra en el evento* ***P****; en resumen* $\boldsymbol{G_p}$ *(x) = Probabilidad(x ∈ P)*

Observando los grados de pertenencia como probabilidades, se tiene para las operaciones básicas de Unión. Complemento e Intersección interpretaciones más refinadas que apuntan a establecer mejorías en la interpretación de los conjuntos difusos definidos, considerando por ejemplo el *complemento* de uno de estos, la probabilidad del complemento de un conjunto difuso *D(visita),* es la probabilidad complementaria: $g_{\overline{D}}(x) = 1 - g_D(x)$. La *Intersección***,** en cambio es la probabilidad de ocurrencia simultánea de dos eventos; la intersección esta muy ligada al concepto de probabilidad condicional. Así pues, teniendo una función **d** que a dos eventos cualesquiera **A, B** les asocia una "densidad de probabilidad condicional **d(A | B)** tal que a cada objeto **x** le asocia un valor **d(A | B)(x)** de manera que: $d(A \mid B)(x) \bullet g_B(x) = d(B \mid A) \bullet g_A(x)$. Entonces para dos conjuntos difusos cualesquiera D, E se puede definir el grado de pertenencia a la intersección como: $g_{D \cap E}(x) = d(D \mid E) \bullet g_E(x)$

Para definir la operación de intersección, basta tener un operador de "probabilidad condicional". De manera recíproca, si se tiene definida de alguna manera al operador de intersección, entonces siguiendo el *teorema de Bayes* se puede definir un operador de "probabilidad condicional. Por tanto, las nociones de intersección (probabilista) de conjuntos difusos y la de la probabilidad condicional son reducibles una a la otra. La *Unión* en cambio**,** es la es la probabilidad de uno, mas la probabilidad del otro, menos la probabilidad de que ocurran ambos eventos: $g_{D \cup E}(x) = g_D(x) + g_E(x) - g_{D \cap E}(x)$.

Una medida de creencia en cambio, esta asociada a las reglas de inferencia determinadas por un experto, estas medidas o métricas permiten obtener ciertas conclusiones a partir de premisas conocidas. Lo anterior tiene por consecuencia la posibilidad de establecer una gran variedad de problemas de deducción y de inferencia dado que es posible formular hipótesis y además probarlas sobre un contexto de vaguedad o dominio de conocimiento. La lógica difusa permite manejar proposiciones o premisas difusas o vagas y obtener por medio de un proceso de inferencia, conclusiones sobre conjuntos difusos enmarcados en un cierto universo de discurso ***U.*** En este caso la pregunta *¿la página fue vista o percibida? ,* tiene implicancias en cuanto a las respuestas posibles, dado que estas dependen de la subjetividad del experto que establece las reglas de inferencia. Y la medida de creencia que fijan las respuestas posibles. Por ejemplo es ampliamente aceptado por las empresas del rubro Medición de Audiencias de Internet que la visita promedio de usuarios a un sitio web gira en torno a los 25 minutos, conocimiento determinado a partir de pruebas experimentales realizadas sobre diversos grupos de usuarios y diversas fuentes primarias. de usuarios. Para nuestro planteamiento hemos considerado

responder la pregunta anterior por medio del parámetro PageView[2] definido con anterioridad tiene una definición (basada en medida de creencia) que se relaciona con el tiempo de visita, la cantidad de objetos contenidos en una pagina web, la ventana virtual y la relación que establece un sujeto con el objeto percibido.

En este caso al considerar al atributo A, entonces para cada elemento x del dominio, $G_p(x)$ es un grado con el que se cree que x posee el atributo A

Las funciones Unión, Complemento e Intersección en una interpretación por grados de creencia, que nos permitirán responder a cuestiones sobre diversos objetos, o a cuestiones compuestas por conectivos lógicos, pueden ser definidas de manera arbitraria basándose en la experiencia del experto o en su defecto asociadas a las respuestas que se esperan del domino de conocimientos genérico que ciertamente en nuestro caso es un domino. El *complemento* como $g_{\overline{D}}(x) = 1 - g_D(x)$. La *intersección* de dos conjuntos difusos *A(tupla_3), B(tupla_3)* con sendos grados de pertenencia g_A y g_B , si para un punto dado **x**, la suma $g_A + g_B$ es menor que **1** entonces descartamos que ese punto sea común a ambos conjuntos, es decir, no debe estar "en la intersección". En otro caso, se toma como grado de pertenencia, a la intersección, a la razón de la diferencia $[g_A + g_B] - 1$ entre el máximo de g_A y g_B. Que es el caso de las restricciones supuestas para una sesión ultra larga o extremadamente corta. Una expresión matemática basada en símbolos que refleja lo anterior esta dada por :

$$g_{A\cap B}(x) = \begin{cases} 0 & \text{si } g_A(x) + g_B(x) < 1 \\ \frac{g_A(x)+g_B(x)-1}{\max(g_A(x),g_B(x))} & \text{si } g_A(x) + g_B(x) \geq 1 \end{cases}$$

En el caso de la *unión* dado dos conjuntos difusos *A(tupla_3), B(tupla_3)* con sendos grados de pertenencia g_A y g_B, si para un punto dado $\boldsymbol{x}$ la suma $g_A + g_B$ es mayor que 1 entonces convenimos en que ese punto "está en la unión". En otro caso, se toma como grado de pertenencia, a la unión, al máximo de las razones $g_A(x)/(1-g_B(x))$ y $g_B(x)/(1-g_A(x))$. En símbolos:

$$g_{A\cup B}(x) = \begin{cases} 1 & \text{si } g_A(x) + g_B(x) \geq 1 \\ \max\left(\frac{g_A(x)}{(1-g_B(x))}, \frac{g_B(x)}{(1-g_A(x))}\right) & \text{si } g_A(x) + g_B(x) < 1 \end{cases}$$

Estos operadores básicos nos dan un marco adecuado para gestionar expresiones complejas con los conceptos lingüísticos introducidos, mediante el empleo de estos operadores o conectivos lógicos. Esto, unido al modelo de inferencia de la lógica difusa, hace que nuestro modelo sea capaz de generar conocimiento acerca nuevos conceptos a partir de los conceptos ya desarrollados en la tesis, incluso mediante el uso de reglas expresadas lingüísticamente y que trasladen el conocimiento del sentido común.

A modo de ejemplo, como una primera aproximación muy simple, y solamente para ilustrar como puede desarrollarse esos nuevos conceptos derivados, podríamos plantear una forma inicial del concepto "interés de los usuarios por los contenidos de una página", mediante el empleo de reglas como las siguientes:

Ejemplos arbitrarios de Reglas que emplean concepto de visita corta, media y larga con parámetro PageView^2

- Si (visita es corta o media) y (PageView^2 es bajo) , entonces (interés_de_usuario U en_contenido_página _X es bajo)

- Si (visita es larga o media) y (PageView^2 es medio), entonces (interés_de_usuario U en_contenido_página _X es normal)

- Si (visita_a_página_X es alto), y (PageView^2 es bajo), entonces (interés_de_usuario U en_contenido_página _X es normal)

Por supuesto, la definición y cálculo de cada concepto podrá utilizar, además de los conectivos lógicos, las ideas de proporción, probabilidad y cardinalidad tal y como se han definido. Lo que nos abre un amplio abanico de posibilidades para definir, estudiar y calcular, conceptos interesantes en el ámbito del análisis de ficheros logs.

En definitiva, nuestro modelo supone una metodología que permite extraer conocimiento e información de los ficheros logs, que se suele expresar de forma lingüística, que se pueda

mediante reglas simples, y que sin embargo es mas complicado tratarlos con los modelos clásicos.

CONCLUSIONES

Desde el punto de vista de una empresa u organización es posible definir *conocimiento* como la información que posee valor, es decir aquella que representa algún particular punto de interés y que es potencialmente útil para la empresa. Este "conocimiento" construido en grandes cantidades por los miembros de la organización, generalmente esta *almacenado y distribuido* en diversas fuentes tanto al interior de la intranet como también en el web; siendo estas fuentes de acopio de datos variadas, como también variados son sus formatos de almacenamiento. El problema por tanto consiste principalmente en rescatar el conocimiento alojado en la información almacenada sobre el contexto en que opera la organización, empleando algunas técnicas de análisis que consideren estas características: datos distribuidos en distintas fuentes, diversos formatos, baja estructuración, contenidos multimedia, logs de registros de eventos, logs de transacciones, mail y de forma resumida datos no estructurados.

Los mecanismos más habitualmente empleados para la obtención de conocimiento a partir de grandes volúmenes de información consisten en el empleo de herramientas de software o sistemas constituidos por varios programas que analizan de manera automática un universo o dominio de conocimientos, buscando tendencias, desviaciones, agrupaciones o relaciones, para posteriormente ser particularizados a casos o situaciones especificas las cuales pueden ser visualizadas a *posteriori*, procesos que en el caso de ser particularizados al web es denominado *Web Mining*, que en principio es una fase dentro del denominado *Descubrimiento de Conocimiento Web*.

Un modelo formal para el análisis inteligente de web logs

Es importante destacar que es muy común confundir el proceso de *minería de datos* como también el *web mining*, con un análisis estadístico de estos, la diferencia fundamental entre ambas técnicas radica en que una evaluación estadística de un dominio de conocimientos definido a partir de *b_internet,* por medio de funciones como por ejemplo *"Chi square test", "Normal", "Student"* u otras determinan el valor que toma una variable dentro de un dominio de variabilidad o en otras palabras su clase (o verdad), es conocida a priori, faltando por establecer

las relaciones con otras clases lo cual en definitiva permitirá extraer generalidades desde un gran conjunto de datos.

Un sistema de inteligencia artificial establecerá su entorno en la obtención autónoma de *conocimiento racional*; es decir en construir relaciones entre conceptos y símbolos con elementos de un dominio en particular, al manejo de información o datos perteneciente a dominios restringidos y con estos elementos establecer semejanzas o patrones entre objetos diferentes, y que por medio de estas semejanzas se pueda establecer una clasificación y construir clases distintas de objetos, géneros o especies; por medio del empleo de procedimientos heurísticos o inferenciales para la resolución de problemas del mundo real. *Un objetivo de esta tesis ha sido relacionar las ideas y conceptos planteados anteriormente con modelos artificiales para la obtención de conocimiento a partir de grandes volúmenes de datos.*

Así, en esta tesis se ha mantenido como objetivo fundamental establecer o definir un sistema formal para el análisis inteligente de ficheros logs, de forma que permita expresar con la imprecisión propia del lenguaje natural los resultados del análisis de los ficheros logs, ampliando de forma considerable el modelo de análisis de estos ficheros, pues amplia notablemente el conjunto de preguntas y respuestas que se pueden obtener como resultado de ese análisis.

De manera habitual la resolución de problemas del mundo real, se refiere a encontrar un método en el cual partir de una condición inicial se obtenga un objetivo, este objetivo puede ser representado por ejemplo como un proceso de aprendizaje o la obtención de conocimiento artificial a partir de un volumen de información de datos.

El concepto de percepción de una página web

Cualquier intento de describir fenómenos observados de la realidad o de la realidad virtual basada en el web, nos lleva implícitamente a modelos matemáticos, sencillos en algunos casos y muy complejos en otros, por ejemplo la geometría euclidiana introduce ciertos conceptos puramente teóricos como rectas, puntos, figuras y otros que pueden ser interpretados como una representación concreta de la realidad, podemos concluir por tanto que esta geometría constituye un modelo matemático que nos permite determinar ciertas regularidades como por ejemplo que la suma de los ángulos interiores de un triángulo es π. El que dicha teoría

matemática pueda considerarse como un modelo satisfactorio de los fenómenos observados es una cuestión que solo puede ser resuelta con la experiencia o sabiduría. Al mencionar la *experiencia* estamos precisando que es necesario establecer experimentos aleatorios que puedan repetirse sobre distintas poblaciones o conjuntos de datos u objetos, con el objetivo de obtener resultados aislados para cada prueba y a partir de estos resultados concluir en el concepto de similitud o clase de acuerdo al modelo establecido. Cuando mencionamos *sabiduría* lo hacemos desde la perspectiva humana del conocimiento el cual surge de la observación de los objetos ya clasificados u ordenados; en donde los resultados de estas observaciones o pruebas siempre tendrán una variabilidad dado que los sujetos que observan formaran sus propias deducciones. Un punto de vista basado en *la teoría del conocimiento web* planteada en los capítulos anteriores y brevemente expuesta en el párrafo anterior, tiene su origen en el "análisis inteligente" de la información que en nuestro caso de estudio esta contenida en *b_internet,* información que tiene su génesis en *objetos* almacenados en alguna fuente primaria o por causa del empleo de los mismos por parte de *sujetos* que los solicitan. Esta relación ***[sujeto, objeto]*** puede ser modelada de múltiples formas o maneras, como por ejemplo considerando los atributos de los objetos, de acuerdo a las necesidades de los sujetos reales o virtuales u orientando los resultados a la toma de decisiones o a determinar el comportamiento de los usuarios de una solución web.

Nuestro planteamiento o método para dar respuestas a cuestiones como las anteriores que pueden ser expresadas como la *obtención de conocimiento del web*, problema que en esta tesis se ha representado como una respuesta a la pregunta ¿la página fue vista? , se basa en el hecho objetivo de que la respuesta esta dirigida a un ser humano o a un sistema basado en inteligencia artificial que pueda interpretar posibles respuestas como las siguientes:

No lo creo porque la visita fue muy breve

Si es posible

Si y No (+ o -)

Basándonos en la certeza o certidumbre del contenido de verdad de cualquier respuesta posible nos ha permitido sistematizar un modelo de análisis de los datos contenidos en el web y particularmente desarrollar los distintos conceptos y términos definidos en los capítulos antecesores, los cuales apuntan a un objetivo: establecer un contexto (modelo) para ser aplicado al desarrollo de métodos o arquitecturas que permitan dar respuestas a cuestiones

como nuestra pregunta base en donde estas respuestas puedan ser interpretadas ya sea por *sujetos reales* o *sistemas artificiales* desarrollados para la obtención de conocimiento

Los niveles de abstracción definidos como: *dominio de conocimiento genérico*, *sujeto real*, *objetos primitivos y contenedores*, *fuente primitiva* y otros como: *duda*, *escepticismo* o *verdad racional* tienen su sustento en el hecho de que los seres humanos a diferencia de las computadoras tienen la capacidad de manejar conceptos vagos que no encajan en valores absolutos como 1 o 0, verdadero o falso o en una función matemática que exprese un resultado.

Percepción, Visualización, Visita son términos que han sido utilizados con el objetivo de destinarlos a la interpretación de seres humanos, siguiendo los lineamientos del paradigma web inteligente con la finalidad de reducir la dimensionalidad de las respuestas e interpretaciones posibles de estas, o de respuestas que no tienen conocimiento asociado como por ejemplo: los "hits" sobre la página "mensaje del director" fueron 12345; estos hits no indican si este mensaje fue percibido, visto y en una escala superior leído o comprendido. El parámetro *PageView*^{2} ha sido planteado como una alternativa y contiene *conceptos* que califican la visita de un usuario a un sitio web relacionándolo con términos lingüísticos como visita corta, visita normal, o visita larga, términos que permiten tener un grado de certeza o verdad sobre la percepción del objeto visualizado. Estos conceptos, al contrario del de PageView clásico. son conceptos dependiente de cada página, pues intenta recoger, al menos parcialmente el hecho de que la página fue percibida, y por tanto dependerá de los contenidos y estructura de la misma.

En este sentido, además de la introducción del concepto, se ha planteado un algoritmo capaz de calcular y adaptar de forma dinámica el concepto de visito corta, media, o larga, culminando con una aproximación algorítmica al concepto de percepción de la página.

Un modelo lingüístico de análisis de web logs

Tras la formulación del modelo formal para el análisis inteligente de web logs, y la incorporación y cálculo de conceptos lingüísticos relacionados la percepción y hábitos en la visualización de cada página web, como son los conceptos de visita corta, media y larga, esta tesis ha proporcionado una metodología para el análisis de web logs, capaz de responder a cuestiones que previamente sería difícil simplemente formular. En los modelos previos, estas

cuestiones nuevas que operan con estos conceptos precisarían de un modelo particular diseñado ad hoc para responder a esa cuestión. Por el contrario, en el nuevo modelo planteado, se trataría simplemente de aplicar la metodología propuesta, consistente en incorporar las técnicas clásicas de web mining a este nuevo contexto de datos.

Así, la tesis concluye con una metodología global capaz de extraer un conocimiento mas elaborado que los modelos anteriores de web mining.

Trabajos Futuros

Este trabajo de tesis ha tenido como objetivo construir un nuevo modelo general para el análisis inteligente de sitios web. Esta metodología general tan solo ha sido aplicada en profundidad al problema de cálculo de audiencias, e incorporándose un simple esbozo de cómo extender la metodología a otras cuestiones.

La continuación natural de la tesis debe ser la aplicación de esa metodología para abordar en profundidad el estudio de otros tópicos del análisis de ficheros logs:

- Perfiles de usuario
- Personalización
- Mejora del diseño del sitio web
- Patrones de navegación

etc ….

Otra de las líneas futuras de trabajo es el análisis de la estructura y los contenidos de una solución web, incorporando estas técnicas, algo especialmente interesante en el caso de sitios web dedicados a contenidos educativos, como es el caso del sitio web de la Utem virtual que es donde se han probado los algoritmos de esta tesis. En esta línea, resulta clara la posibilidad de aplicar de algoritmos de clustering de páginas web basados en los conceptos de visita corta, larga y media, a fin de identificar las estructuras y contenidos de las páginas con mayor índice de audiencia.

Finalmente, y como resultado tangible de la tesis, se realizará un producto software para análisis de ficheros logs, que incorpore el modelo de la tesis, especialmente en el aspecto de medición de audiencia

BIBLIOGRAFÍA

1. Access Log Analyzers, http://www.uu.se/Software/Analyzers/Access-analyzers.html.

2. Agrawal, R., Gehrke, J., Gunopulos, D. y Raghavan, P. (1998). Automatic subspace clustering of high dimensional data for data mining applications. En Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data (págs. 94–105). Seattle, WA,

3. Agrawal, R., Imielinski, T. y Swami, A. (1993). Mining association rules between sets of items in large databases. En Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data (págs. 207–216). Washington, D.C., Estados Unidos.

4. Agrawal, R., y Srikant, R. (1994). Fast algorithms for mining association rules. En Proceedings of the 20th International Conference on Very Large Data Bases (págs. 487–499). Santiago, Chile.

5. Araya, S.; Silva, M.; Weber, R. A Methodology for Web Usage Mining and Its Application to Target Group Identification. *Fuzzy Sets and Systems* 2004, *148,* 139-152.

6. J. Arjona, R. Corchuelo, J. Pena, D. Ruiz, Coping with web knowledge, in: Advances in Web Intelligence, Springer-Verlag, Berlin, Heidelberg, 2003, pp. 165–178.

7. Arotatei D. y MitraS; Web mining: a survey in the fuzzy framework, Fuzzy Sets and Systems 148 (2004) 5–19

8. Axelrod, R.: Structure of Decision: The Cognitive Maps of Political Elites, Princeton Univ.

9. M. Balabanovic and Y. Shoham. Learning information retrieval agents: Experiments with automated Web browsing. In *On-line Working Notes of the AAAI Spring Symposium Series on Information Gathering from Distributed, Heterogeneous Environments*, 1995.

10. A. Bargiela, W. Pedrycz, Granular Computing—An Introduction, Kluwer Academic Publishers, Boston, Dordrecht, London, 2003.

11. P. Batista, M. Silva, Prospecção dos Dados de Acesso a um Servidor de Notícias na Web, *2ª Conferência sobre Redes de Computadores*, Évora, Portugal, Outubro 1999.

12. H.-U. Bauer, T. Villmann, Growing a hypercubical output space in a self-organising feature map, IEEE Trans. Neural Networks 8 (2) (1997) 218–226.

13. Berendt, B.; Hotho, A.; Stumme, G. *Towards Semantic Web Mining*; 2002; pp 264-278.

14. Berendt, B. Detail and Context in Web Usage Mining: Coarsening and Visualizing Sequences. *Webkdd 2001 - Mining Web Log Data Across All Customers Touch Points* 2002, *2356,* 1-24.

15. T. Berners-Lee, J. Hendler, O. Lassila, The Semantic Web, Scienti.c American, 2001, pp. 34–49.

16. M. Berry, G. Linoff, Data Mining Techniques – For Marketing, Sales and Customer Support, John Wiley & Sons, 1997.

17. Borges, J.; Levene, M. Ranking Pages by Topology and Popularity Within Web Sites. *World Wide Web-Internet and Web Information Systems* 2006, *9,* 301-316.

18. Borzemski, L.; Zatwarnicki, K. Performance Evaluation of Fuzzy-Neural HTTP Request Distribution for Web Clusters. *Artificial Intelligence and Soft Computing - Icaisc 2006, Proceedings* 2006, *4029,* 192-201.

19. Borzemski, L. The Use of Data Mining to Predict Web Performance. *Cybernetics and Systems* 2006, *37,* 587-608.

20. P. S. Bradley, U. M. Fayyad, C. A. Reina, Scaling Clustering Algoritms to Large Databases, *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining (KDD98)*, pp. 9-15, AAAI Press, 1998.

21. T. Bray,J. Paoli, and C. M. Sperberg-McQueen. Extensible markup language XML) 1.0 W3C recommendation. Technical report,W 3C, 1998.

22. L. Breslau, P. Cao, L. Fan, G. Phillips, S. Shenker, Web Caching and Zipf-like Distributions: Evidence and Implications, *Proceedings of the IEEE INFOCOM'99*, New York, March 1999.

23. Brian F. J. Manly, Multivariate Statistical Methods – A Primer, Chapman & Hall, 1986.

24. S. Brin, L. Page, The anatomy ofa large-scale hypertextual web search engine, Comput. Networks ISDN Syst. 30(1–7) (1998) 107–117.

25. G. Buchner, S. S. Anand, M. D. Mulvenna, J. G. Hughes, Discovering Internet Marketing Intelligence through Web Log Mining, *Proc. of Unicom99 Data Mining and Data Warehousing*, pp. 127-138, 1999.

26. L. Catledge and J. Pitkow. Characterizing browsing behaviors on the World Wide Web. *Computer Networks and ISDN Systems*,27(6), 1995.

27. R. Cooley,B. Mobasher,an d J. Srivastava. Web mining: Information and pattern discovery on the World Wide Web. In *International Conference on Tools with Artificial Intelligence*,p ages 558–567, Newport Beach, CA,1997.

28. S. Chakrabarti, Mining the Web: Analysis ofHypertext and Semi Structured Data, Morgan Kaufmann Publishers, Los Angeles, 2002.

29. Chen, Y. L.; Ye, C. H.; Wu, S. Y. Mining Predecessor-Successor Rules From DAG Data. *International Journal of Intelligent Systems* 2006, *21,* 621-637.

30. M.S. Chen,J .S. Park,a nd P.S. Yu. Data mining for path traversal patterns in aWeb environment. In *Proceedings of the 16th International Conference on Distributed Computing Systems*,p ages 385–392, 1996.

31. Chen, J.; Li, Q.; Jia, W. J. Automatically Generating an E-Textbook on the Web. *World Wide Web-Internet and Web Information Systems* 2005, *8,* 377-394.

32. Cho, Y. H.; Kim, J. K.; Kim, S. H. A Personalized Recommender System Based on Web Usage Mining and Decision Tree Induction. *Expert Systems with Applications* 2002, *23,* 329-342.

33. S.-B. Cho, Neural-network classi/ers for recognizing totally unconstrained handwritten numerals, IEEE Trans. Neural Networks 8 (1) (1997) 43–53.

34. S.-B. Cho, Self-organizing map with dynamical node splitting: application to handwritten digit recognition, Neural Comput. 9 (6) (1997) 1343–1353.

35. S.-B. Cho, Ensemble of structure-adaptive self-organizing maps for high performance classi/cation, Inform. Sci. 123 (1–2) (2000) 103–114.

36. S.-B. Cho, J.-H. Kim, Combining multiple neural networks by fuzzy integral for robust classi/cation, IEEE Trans. Syst. Man Cybern. 25 (2) (1995) 380–384.

37. S.-B. Cho, J.-H. Kim, Multiple network fusion using fuzzy logic, IEEE Trans. Neural Networks 6 (2) (1995) 497–501.

38. Clementine User Guide, Version 5, Integral Solutions Limited, 1998.

39. Cooley R.; Web Usage Mining: Discovery and Applications of Interesting Patterns from Web Data. Thesis PhD University of Minnesota /2000

40. G. F. Cooper and E. Herskovitz. A Bayesian Method for the Induction of Probabilistic Networks from Data. *Machine Learning*, 9, pp. 309-347, 1992.

41. M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, S. Slattery, Learning to construct knowledge bases from the World Wide Web, Arti.cial Intelligence 118 (1–2) (2000) 69–113.

42. M.J. Cresswell, Logic and Languages, Methuen, London, UK, 1973.

43. Dae-Young Choi pag. 31; Enhancing the power of Web search engines by means of fuzzy query. Elsevier .Decision Support Systems 35 (2003) 31– 44

44. De, S. K.; Krishna, P. R. Clustering Web Transactions Using Rough Approximation. *Fuzzy Sets and Systems* 2004, *148,* 131-138.

45. B. Efron. Estimating the Error Rate of a Prediction Rule: Improvement on Cross-Validation. Journal of the American Statistical Association, 78(382), pages 316-331, June 1983

46. S. G. Eick, A. Mockus, T. L. Graves, A. F. Karr. A Web Laboratory for Software Data Analysis.*World Wide Web*, 12, pp. 55-60, 1998.

47. S. G. Eick, J. L. Steffen, and E. E. Summer, Jr. SeeSoft – A Tool for Visualizing Line-Oriented Software Statistics. *IEEE Trans. on Soft. Eng.*, (18)11, pp. 957-968. November 1992.

48. S. Elo-Dean and M. Viveros. Data mining the IBM o.cial 1996 Olympics Web site. Technical report,I BM T.J. Watson Research Center,1997.

49. Facca, F. M.; Lanzi, P. L. Recent Developments in Web Usage Mining Research. *Data Warehousing and Knowledge Discovery, Proceedings* 2003, *2737,* 140-150.

50. U. Fayyad and G. Piatetsky-Shapiro and P. Smyth. The KDD process for extracting useful knowledge from volumes of data, *Communications of the ACM*, 39(11), pages 27-34, November 1996.

51. U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy (eds), *Advances in Knowledge Discovery and Data Mining*, AAAI/MIT Press, 1996.

52. U. Fayyad and R Uthurusamy. Data mining and knowledge discovery in databases,*Communications of the ACM*, 39(11), pages 24-26, November 1996.

53. N. E. Fenton. Bayesian Belief Networks – An Overview Web Article. In WWW: http://www.agena.co.uk/bbn_article/bbns.html. Agena Ltd, 1999.

54. N. E. Fenton. Software measurement: A necessary scientific basis. *IEEE Transactions on Software Engineering*, 20(3), March 1994

55. N. E. Fenton. *Software Metrics: A Rigorous Approach*. Chapman Hall, 1991.

56. Flesca, S.; Greco, S.; Tagarelli, A.; Zumpano, E. Mining User Preferences, Page Content and Usage to Personalize Website Navigation. *World Wide Web-Internet and Web Information Systems* 2005, *8,* 317-345.

57. W. J. Frawley, G. Piatetsky-Shapiro, and C. J. Mattheus. Knowledge discovery in databases:An overview. *AI Magazine*, pages 57-70, Fall 1992.

58. H. Frigui and R. Krishnapuram, "A Robust Clustering Algorithm Based on Competitive Agglomeration and Soft Rejection of Outliers," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, June 1996, pp. 550-555.

59. T.F. Gamat, Language, Logic and Linguistics, University of Chicago Press, 1996.

60. Galindo G. J.; "Conjuntos y Sistemas Difusos (Lógica Difusa y Aplicaciones)"; Departamento de Lenguajes y Ciencias de la Computación; Universidad de Málaga, España.

61. D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Company, 1989.

62. R.C. Gonzalez, R.E. Woods, Digital Image Processing, Addison-Wesley, Reading, MA, 1992.

63. G. Graefe, U. Fayyad, S. Chaudhuri, On the Efficient Gathering, of Sufficient Statistics for Classification from Large SQL Databases, *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining (KDD98)*, pp. 204-208, AAAI Press, 1998.

64. .J. Gray,A . Bosworth, A. Layman,an d H. Pirahesh. Data cube: A relational aggregation operator generalizing group-by,cross- tab, and sub-totals. In *IEEE 12th*, *International Conference on Data Engineering*,p ages 152–159, 1996.

65. Guo, J.; Keselj, V.; Gao, Q. Integrating Web Content Clustering into Web Log Association Rule Mining. *Advances in Artificial Intelligence, Proceedings* 2005, *3501,* 182-193.

66. R.M. Haralick, K. Shanmugam, I. Dinstein, Textural features for image classi8cation, IEEE Trans. Systems Man Cybernet. 3 (1973) 610–621.

67. Hebb D.; "The Organization of Behavior", Wile New York, NY 1949

68. Hessen J. Teoría del Conocimiento [1926]

69. S. Hettich, S.D. Bay, The UCI KDD Archive, http://kdd.ics.uci.edu.

70. G. E. Hinton. Connectionist Learning Procedures. In Kodratoff and Michalski [35], pages 555-610.

71. J. H. Holland, K. J. Holyoak, R. E. Nisbett, and P. R. Thagard. *Induction: Processes of Inference, Learning, and Discovery*. MIT Press, Cambridge, MA, 1986

72. M. Holsheimer and A. P. J. Siebes. Data mining: the search of knowledge in databases. Technical Report CS-R9406, CWI – Department of Algorithms and Architecture, Amsterdam, The Netherlands, 1994.

73. T.-P. Hong, K.-Y. Lin, S.-L. Wang, Mining linguistic browsing patterns in the World Wide Web, Soft Comput. 6 (2002) 329–336.

74. Hu, J.; Zhong, N. Organizing Multiple Data Sources for Developing Intelligent E-Business Portals. *Data Mining and Knowledge Discovery* 2006, *12,* 127-150.

75. Y.-P. Huang, Y.-C. Lee, L. Lin, An intelligent approach to mining the related websites, in: Proc. 20th NAFIPS Conf. Vancouver, Canada, July 2001, pp. 435–440.

76. T. Joachims,D . Freitag, and T. Mitchell. Webwatcher: A tour guide for the World Wide Web. In *Proc. of the 15th International Conference on Artificial Intelligence*, pages 770 – 775,N agoya, Japan,1997.

77. M. Jørgensen. Experience With the Accuracy of Software Maintenance Task Effort Prediction Models. IEEE TSE, 21(8), pp. 674-681, August 1995

78. A. Joshi, R. Krishnapuram, Robust fuzzy clustering methods to support web mining, in: Proc. ACM-SIGMOD Workshop on Data Mining and Knowledge Discovery, August 1998.

79. A. Joshi, R. Krishnapuram, On mining web access logs, in: Proc. ACM-SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, 2000, pp. 63–69.

80. Jung, J. J. Semantic Preprocessing of Web Request Streams for Web Usage Mining. *Journal of Universal Computer Science* 2005, *11,* 1383-1396.

81. T. M. Khoshgoftaar and E. B. Allen. Modeling Software Quality with Classification Trees. In *Recent Advances in Reliability and Quality Engineering*, Hoang Pham Editor. World Scientific, Singapore, 1999.

82. T. M. Khoshgoftaar, E. B. Allen, J. P. Hudepohl, and S. J. Aud. Neural Networks for Software Quality Modeling of a Very Large Telecommunications System. *IEEE Trans. On Neural Networks*, (8)4, pp. 902-909, July, 1997

83. T. M. Khoshgoftaar and D. L. Lanning. A Neural Network Approach for Early Detection of Program Modules Having High Risk in the Maintenance Phase*. J. Systems Software*, 29(1),pp. 85-91, 1995.

84. K.-J. Kim, S.-B. Cho, A personalized web search engine using fuzzy concept network with link structure, in: Proc.Joint Nineth IFSA World Congress and 20th NAFIPS Internat. Conf. 2001, pp. 1:81–1:86.

85. E. Kim, W. Kim, Y. Lee, Combination ofmultiple classi/ers for the customer's purchase behavior prediction, Dec.Support Syst. 34 (2) (2003) 167–175.

86. W. Klösgen and J. M. Zytkow. Knowledge discovery in databases terminology. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smith, and R. Uthurusamy, editors*, Advances in Knowledge Discoveryand Data Mining*. AAAI Press/ The MIT Press, Cambridge, MA, 1996

87. Y. Kodratoff and R. S. Michalski, editors. *Machine Learning, an Artificial Intelligence Approach, Volume 3*. Morgan Kaufmann, San Mateo, California, 1990.

88. T. Kohonen, The self-organizing map, Proc. IEEE 78 (9) (1990) 1464–1480.

89. Kohonen T.; "Self –Organized formation of topologically correct feature maps" Biological Cybernetics 1982

90. T. Kohonen, S. Kaski, K. Lagus, J. Salojarvi, J. Honkela, V. Paatero, A. Saarela, Selforganization of a massive document collection, IEEE Trans. Neural Networks 11 (2000) 574–585.

91. Kolari, P.; Joshi, A. Web Mining: Research and Practice. *Computing in Science & Engineering* 2004, *6,* 49-53.

92. R. Kosala, H. Blockeel, Web mining research: a survey, SIGKDD Explorations 2 (1) (2000) 1–15.

93. Kraaij W,; Variations on Language Modeling for Information Retrieval Taaluitgeverij Neslia Paniculata / CTIT Ph.D. -thesis series No. 04-62 Copyright c 2004, Wessel Kraaij, Rotterdam. ISBN 90-75296-09-6 ISSN 1381-3617; No. 04-62 (CTIT Ph.D -2004.)

94. R. Krishnapuram, A. Joshi, L. Yi, A fuzzy relative of the k-medoids algorithm with application to document and snippet clustering, in: Proceedings of IEEE International Conference on Fuzzy Systems (FUZZ IEEE'99), Korea, August 1999, pp. 3:1281–3:1286.

95. R. Krishnapuram, A. Joshi, O. Nasraoui, L. Yi, Low complexity fuzzy relational clustering algorithms for web mining, IEEE Trans. Fuzzy Systems 9 (2001) 595–607.

96. R. Krishnapuram, J.M. Keller, A possibilistic approach to clustering, IEEE Trans. Fuzzy Systems 1 (1993) 98–110.

97. U. Krohn and C. Boldyreff. Application of Cluster Algorithms for Batching of Proposed Software Changes. *J. Softw. Maint: Res. Pract.* 11, 151-165. May-June 1999

98. Krulwich, B. (1997). Lifestyle Finder: Intelligent User Profiling Using Large- Scale Demographic Data. Artificial Intelligence Magazine, 18(2), 37– 45.

99. R. Kruse, A. Klose, Information mining with fuzzy methods: Trends and current challenges, in: Methods and Models in Automation and Robotics, Szczecin, Poland, 2002, pp. 117–120.

100. A.S. Kumar, S.K. Basu, K.L. Majumdar, Robust classi/cation ofmultispectral data using multiple neural networks and fuzzy integral, IEEE Trans. Geosci. Remote Sensing 35 (3) (1997) 787–790.

101. R. Kumar, P. Raghavan, S. Rajagopalan, A. Tomkins, Trawling the web for emerging cyber-communities, Comput. Networks 31 (11–16) (1999) 1481–1493.

102. Langville, A. N.; Meyer, C. D. A Survey of Eigenvector Methods for Web Information Retrieval. *Siam Review* 2005, *47,* 135-161.

103. F. Lanubile and G. Visaggio, Evaluating predictive quality models derived from software measures lessons learned", *The Journal of Systems and Software*, 38:225-234, 1997.

104. Lazzerini, B.; Marcelloni, F. A Hierarchical Fuzzy Clustering-Based System to Create User Profiles. *Soft Computing* 2007, *11,* 157-168.

105. H.-M. Lee, S.-K. Lin, C.-W. Huang, Interactive query expansion based on fuzzy association thesaurus for web information retrieval, in: Proc. the 10th IEEE Internat. Conf. on Fuzzy Systems, 2001, pp. 2:724–2:727.

106. K.C. Lee, J.S. Kim, N.H. Chung, S.J. Kwon, Fuzzy cognitive map approach to web mining inference ampli8cation, Expert Systems Appl. 22 (2002) 197–211.

107. D.B. Lenat, A large-scale investment in knowledge infrastructure, Communications of the ACM 38 (11) (1995) 32–38.

108. R. P. Lippmann. An Introduction to Computing with Neural Nets*. IEEE Acoustical, Speech, and Signal Processing Magazine*, 4, pp. 4-22, 1987. Reprinted in *Neural Networks: Theoretical Foundations and Analysis*, Edited by Clifford Lau, IEEE Press, 1992. Also reprinted in *Optical Neural Networks*, Edited by S. Jutamulia, SPIE Optical Engineering Press, 1994

109. Liu, J. M.; Zhang, S. W.; Yang, J. Characterizing Web Usage Regularities With Information Foraging Agents. *Ieee Transactions on Knowledge and Data Engineering* 2004, *16,* 566-584.

110. H. J. Loether and D. G. McTavish. *Descriptive and Inferential Statistics: An Introduction. Part VI - Inferential Statistics*. Allyn and Bacon, Inc. Needham Heights, MA, 1988

111. Lopes, C. T.; David, G. Higher Education Web Information System Usage Analysis With a Data Webhouse. *Computational Science and Its Applications - Iccsa 2006, Pt 4* 2006, *3983,* 78-87.

112. McCulloch, Pitts W.; "A Logical Calculus of the Ideas Immanent in Nervous Activity"; Bulletin of Mathematical Biophysics (1943)

113. J. B. MacQueen. Some Methods For Classification and Analysis of Multivariate Observations. In L. M. LeCam and J. Neyman, editors*, Proceedings of the Fifth Berkeley Symposium on Mathematical Statistic and Probability*, pages 281-297, University of California Press, Berkley,CA, 1967

114. S. Madria, S. S. Bhowmick, W.-K Ng, E. P. Lim, Research Issues in Web Data Mining, *Proc. of the First International Conference on Data Warehousing and Knowledge Discovery* (DaWaK99), pp. 303-312, 1999.

115. N. Maria, P. Gaspar, N. Grilo, A. Ferreira. M. Silva, ARIADNE – Digital Library Architecture, *Proceedings of the Second European Conference on Research and Advanced Technology for Digital Libraries*, Springer,1998.

116. M.J. Martin-Bautista, H.L. Larsen, M.A. Vila, A fuzzy genetic algorithm approach to an adaptive information retrieval agent, J. Amer. Soc. Inform. Sci. 50 (1999) 760–771.

117. Maturana H., Varela F. ;El Árbol del Conocimiento; (1984)

118. Maturana H.; El Origen de lo Humano en la Biología de la Intimidad

119. Maturana H., Varela F. ;El Árbol del Conocimiento; (1984)

120. G. Meghabghab, Mining user's web searching skills through fuzzy cognitive state map, in: Proc. Joint 9th IFSA World Congress and 20th NAFIPS Internat. Conf. 2001, pp. 1:429–1:434.

121. D. Michie, D. J. Spiegelhalter, C. C. Taylor (eds), Machine Learning, Neural and Statistical Classification, Ellis Horwood, 1994.

122. S. Mitra, S.K. Pal, P. Mitra, Data mining in soft computing framework: a survey, IEEE Trans. Neural Networks 13 (2002) 3–14.

123. G. Miller. The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. Psychological Review, 101(2), pp. 343-352, April 1994

124. G. W. Milligan. An Examination of the Effect of Six Types of Error Perturbation of Fifteen Clustering Algorithms. Psychometrika, 45(3), pp. 325-342, September 1980

125. Mobasher B.; Web Usage Mining and Personalization, 2004 CRC Press LLC

126. Mobasher M, Cooley R., Srivastava J. ;Automatic Personalization's Based on Web Usage Mining Communications of the ACM August 2000/Vol 43

127. O. Nasraoui, H. Frigui, R. Krishnapuram, A. Joshi, Extracting web user pro8les using relational competitive fuzzy clustering, Internat. J. Arti8cial Intelligence Tools 9 (2000) 509–526.

128. M. Neil and N. E. Fenton. Predicting software quality using Bayesian belief networks. *Proc 21st Annual Software Eng Workshop*, NASA Goddard Space Flight Centre, pp. 217-230, Dec, 1996

129. M. Neil, B. Little wood, and N. E. Fenton. Applying Bayesian belief networks to systems dependability assessment, *in Proceedings of 4th Safety Critical Systems Symposium*, Springer Verlag, pp. 71-93, 1996

130. Norguet, J. P.; Zimanyi, E.; Steinberger, R. Improving Web Sites With Web Usage Mining, Web Content Mining, and Semantic Analysis. *Sofsem 2006: Theory and Practice of Computer Science, Proceedings* 2006, *3831,* 430-439.

131. V. Novak, I. Per.lieva (Eds.), Discovering the World with Fuzzy Logic. Studies in Fuzziness and Soft Computing,Physica-Verlag, New York, Heidelberg, 2000.

132. A. NVurnberger, A. Klose, Improving clustering and visualization of multimedia data using interactive user feedback, in: Internat. Conf. on Information Processing and Management of Uncertainty in Knowledge Based Systems, Annecy, France, 2002.

133. Open Market Inc. Open Market Web reporter. http://www.openmarket.com,1996.

134. Pazzani, M. J. (1999). A Framework for Collaborative, content-based and Demographic Filtering. Artificial Intelligence Review, 13(5–6), 393– 408.

135. W. Pedrycz, Conditional fuzzy c-means, Pattern Recognition Lett. 17 (1996) 625–632.

136. W. Pedrycz, F. Gomide, Introduction to Fuzzy Sets, MIT Press, Cambridge, MA, 1998.

137. J. Pei, J. Han, B. Martazavi-asl, H. Zhu, Mining Access Patterns Efficiently from Web Logs, *Proceedings of the Pacif-Asia Conference on Knowledge Discovery and Data Mining* (PAKDD00), 2000.

138. Pierrakos, D.; Paliouras, G.; Papatheodorou, C.; Spyropoulos, C. D. Web Usage Mining As a Tool for Personalization: A Survey. *User Modeling and User-Adapted Interaction* 2003, *13,* 311-372.

139. A. Podgurski, W. Masri, Y. McCleese, and F. G. Wolff. Estimation of Software Reliability by Stratified Sampling. *ACM Trans. on Soft. Eng. and Methodology*, (8)3, pp. 263-283, July 1999.

140. A. A. Porter and R. W. Selby. Empirically Guided Software Development Using Metric-Based Classification Trees. *IEEE Software*, 7(2), pp. 46-54, March 1990.

141. A. A. Porter and R. W. Selby. Evaluating Techniques for Generating Metric-based Classification Trees. *J. Systems Software*, pp. 209-218, December 1990.

142. Público On-Line, http://www.publico.pt.

143. J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufman, 1992

144. J. R. Quinlan. Induction of Decision Trees. *Machine Learning*, 1(1), pages 81-106, 1986.

145. T.A. Runkler, J. Bezdek, Web mining with relational clustering, Internat. J. Approx. Reason. 32 (2003) 217–236.

146. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P. y Riedl, J. (1994). Grouplens: An open architecture for collaborative filtering of netnews. En Proceedings of the Conference on Computer Supported Cooperative Work (págs. 175–186). Chapel Hill, NC, Estados Unidos.

147. Resnick, P., y Varian, H. R. (1997). Recommender systems. Communications of the ACM, 40(3), 56–58.

148. Rich, E. (1979). User Modeling via Stereotypes. Cognitive Science, 3,

149. Ripley, B. D. (1996). Pattern recognition and neural networks. Cambridge University Press. Rosenstein, M., y Lochbaum, C. (2000). Recommending from Content: Preliminary Results from an E-Commerce Experiment. En Proceedings of CHI'00: Conference on Human Factors in Computing. La Haya, Holanda.

150. T.A. Runkler, J.C. Bezdek, Alternating cluster estimation: a new tool for clustering and function approximation, IEEE Trans. Fuzzy Systems 7 (1999) 377–393.

151. T.A. Runkler, J.C. Bezdek, Relational clustering for the analysis of internet newsgroups, in: O. Opitz, M. Schwaiger, (Eds.), Exploratory Data Analysis in Empirical Research, Proceedings ofthe 25th Annual Conference of the German Classi8cation Society, Studies in Classi8cation, Data Analysis, and Knowledge Organization, Springer, Berlin, 2002, pp. 291–299.

152. Salton G.; Automatic Information Organization and Retrieval McGraw-Hill New York 1968

153. Sankar K. Pal, Sushmita Mitra,Data Mining in Soft Computing Framework: A Survey, IEEE TRANSACTIONS ON NEURAL NETWORKS, VOL. 13, NO. 1, JANUARY 2002

154. Saridis, G.: Analytic formulation of the principle of increasing precision with decreasing intelligence for intelligent machines, Automatica 25 (1989), 461–467.

155. J. C. Schilimmer and P. Langley. Machine Learning. In S. C. Shapiro, editor, *Encyclopedia of Artificial Intelligence*, volume 1, pages 785-805. John Wiley & Sons, 1992.

156. Schneider,M., Shneider, E., Kandel, A., and Chew, G.: Automatic construction of FCMs, Fuzzy Sets and Systems 93 (1998), 161–172.

157. Schafer, J. B., Konstan, J. y Riedl, J. (1994). Recommender Systems in ECommerce. En Proceedings of the Conference on Computer Supported Cooperative Work (págs. 175–186). Chapel Hill, NC, Estados Unidos.

158. Schwab, I., Kobsa, A. y Koychev, I. (2001). Learning User Interests through Positive Examples Using Content Analysis and Collaborative Filtering (Internal Memo). St. Augustin, Alemania: GMD.

159. R. W. Selby and A. A. Porter. Learning from Examples: Generation and Evaluation of Decision Trees for Software Resource Analysis. *IEEE Trans. on Soft. Eng.*, 14(12), pp. 1743-1757,December 1988

160. Shahabi, C.; Banaei-Kashani, F. Efficient and Anonymous Web-Usage Mining for Web Personalization. *Informs Journal on Computing* 2003, *15,* 123-147.

161. Shardanand, U., y Maes, P. (1995). Social Information Filtering: Algorithms for Automating "word of Mout". En CHI'95: Conference Proceedings on Human Factors in Computing Systems (págs. 210–217). Denver, CO, Estados Unidos.

162. Shavlik, J. W., y Dietterich, T. G. (1990). Readings in machine learning. Morgan Kaufmann.

163. Sheikholeslami, S., Chatterjee, S. y Zhang, A. (1998). A multi-resolution clustering approach for very large spatial databases. En Proceedings of the 24th International Conference on Very Large Data Sets (págs. 428–439). Nueva York, NY, Estados Unidos.

164. Shaw M.J. Shaw ; Chandrasekar Subramaniam a, Gek Woo Tan a, Michael E. Welge bŽ . Decision Support Systems 31 2001 127–137

165. Smith B, Welty C. What is Ontology? Ontology: Towards a new synthesis, in: Proceedings of the Second International Conference on Formal Ontology in Information Systems, 2002.

166. M.K. Smith, C. Welty, D. McGuinness (Eds.), 2003. OWL Web Ontology Language Guide. W3C Working Draft31..

167. Song, Q. B.; Shepperd, M. Mining Web Browsing Patterns for E-Commerce. *Computers in Industry* 2006, *57,* 622-630.

168. JF. Sowa, Ontological categories, in: L. Albertazzi (Ed.), Shapes of Forms: From Gestalt Psychology and Phenomenology to Ontology and Mathematics, Kluwer Academic Publishers, Dordrecht, 1999, pp. 307–340.

169. K. Srinivasan and D. Fisher. Machine Learning Approaches to Estimating Software Development Effort. *IEEE Trans. On Soft. Eng.*, 21(2), pp. 126-137, February 1995

170. J. Srivastava, R. Cooley, M. Deshpande, P.-N. Tan, Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data, *SIGKKD Explorations*, (1)2, Jan 2000.

171. Strang, G. (1988). Linear Algebra and Its Applications. Harcourt Brace.

172. Stylios, C. D. and Groumpos, P. P.: The challenge of modeling supervisory systems using fuzzy cognitive maps, J. Intelligent Manufacturing 9 (1998), 339–345.

173. Stylios, C. D. and Groumpos, P. P.: Using fuzzy cognitive maps to achieve intelligence in

174. Su, Z., Ye-Lu, Q. Y. y Zhang, H. J. (2000). WhatNext: A prediction system for web requests using N-gram sequence models. En WISE 2000 Proceedings: 1st International Conference on Web Information Systems Engineering (págs. 214–221). Hong Kong, China. Technology, Plenum, New York, 1991.

175. K. Swingler. *Applying Neural Networks: A Practical Guide*. Academic Press, London, 1996.

176. Taber, R.: Knowledge processing with fuzzy cognitive maps, Expert Systems Appl. 2 (1991),

177. Terveen, L., y Hill, W. (2001). Human Computer Collaboration in Recommender Systems. En J. Carroll, editor, human Computer Interaction in the New Millenium (págs. 487–509). Nueva York, NY, Estados Unidos: Addison-Wesley.

178. Thelwall, M.; Vaughan, L.; Bjorneborn, L. Webometrics. *Annual Review of Information Science and Technology* 2005, *39,* 81-135.

179. Theodoridis, S., y Koutroumbas, K. (1999). Pattern recognition. Academic Press.

180. J. Tian. Integrating Time Domain and Input Domain Analyses of Software Reliability UsingTree-Based Models. *IEEE Trans. on Soft. Eng.*, 21(12), pp. 945-958, December 1995

181. J. Tian and J. Palma. Analyzing and Improving Reliability: A Tree-based Approach. *IEEE Software*, pp. 97-104, 15(2), March-April 1998.

182. Towle, B., y Quinn, C. (2000). Knowledge-Based Recommender Systems Using Explicit User Models. En Knowledge-Based Electronic Markets, Papers from the AAAI Workshop, AAAI Technical Report WS-00-04 (págs. 74–77). Menlo Park, CA, Estados Unidos.

183. Using the Intelligent Miner for Data, IBM Corporation, 1998.

184. Vachtsevanos, G. and Kim, S.: The role of the human in intelligent control practices, in: Proc. with fuzzy cognitive maps, Internat. J. Comput. Intelligence Org. 1 (1996), 120–123.

185. Wang, W., Yang, J. y Muntz, R. (1997). STING: A statistical information grid approach to spatial data mining. En Proceedings of the 23rd International Conference on Very Large Data Bases (págs. 186–195). Atenas, Grecia.

186. Widrow B., Hoff M.; "Adaptative Switching Circuits" 1960

187. WWW Committee Web Usage Characterization Activity, http://www.w3.org/WCA, Web Characterization Terminology & Definitions Sheet, W3C Working Draft, May 1999.

188. Yager (eds), Advances in Fuzzy Set Theory and Applications, North-Holland, Amsterdam

189. Yager, R. R. Fuzzy Logic Methods in Recommender Systems. *Fuzzy Sets and Systems* 2003, *136,* 133-149.

190. Yang, Q.; Zhang, H. H. Web-Log Mining for Predictive Web Caching. *Ieee Transactions on Knowledge and Data Engineering* 2003, *15,* 1050-1053.

191. Yates, R. D., y Goodman, D. J. (1999). Probability and stochastic processes. John Wiley & Sons.

192. Zadeh L..A.; A note on web intelligence, world knowledge and fuzzy logic Data & Knowledge Engineering 50 (2004) 291–304

193. L.A. Zadeh, A new direction in AI—toward a computational theory of perceptions, AI Magazine 22 (1) (2001) 73–84.

194. L.A. Zadeh, From computing with numbers to computing with words from manipulation of measurements to manipulation of perceptions, IEEE Transactions on Circuits and Systems 45 (1) (1999) 105–119.

195. L.A. Zadeh, Fuzzy logic, neural networks, and soft computing, Communications of the ACM—AI 37 (1994) 77–84.

196. L.A. Zadeh, A fuzzy-algorithmic approach to the de.nition of complex or imprecise concepts, I nternational Journal of Man-Machine Studies 8 (1976) 249–291.

197. L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning, Part I, Information Sciences 8 (1975) 199–249;

198. L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning, Part II, linformation Sciences 8 (1975) 301–357;

199. L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning, Part III, Information Sciences 9 (1975) 43–80.

200. O. R. Zaiane, M. Xin, J. Han, Discovering Web Access Patterns and Trends by Applying OLAP and Data Mining Technology on Web Logs*, Proceedings of Advances in Digital Libraries Conference (ADL98)*, Santa Barbara, CA, April 1998.

201. Zaki, M. J.; Hsiao, C. J. Efficient Algorithms for Mining Closed Itemsets and Their Lattice Structure. *Ieee Transactions on Knowledge and Data Engineering* 2005, *17,* 462-478

202. Zhang, W. R., Chen, S. S., and Besdek, J. C.: Pool2: A generic system for cognitive map

203. Zheng, Z. Q.; Padmanabhan, B.; Kimbrough, S. O. On the Existence and Significance of Data Pre-Processing Biases in Web-Usage Mining. *Informs Journal on Computing* 2003, *15,* 148-170.

LISTA DE ILUSTRACIONES

INDICE DETALLADO