Un método de optimización lineal entera para el análisis de sesiones de usuarios web.

Pablo E. Román*
Juan D. Velásquez*
Robert F. Dell**

Resumen

"Web usage mining" es una nueva área de investigación que ha producido importantes avances en la industria del e-Business, mediante la entrega de patrones de comportamiento de compra y sugerencias de navegación que mejoran la experiencia del usuario web en el sitio. Una de las principales fuentes de datos usadas en web mining, son las sesiones (secuencias de páginas) de los usuarios web que deben ser reconstruidas a partir de los archivos de Log. El problema con los archivos de Logs es que incluyen una componente de ruido al no identificar explícitamente a los usuarios que generan los registros. Con este trabajo, se desarrolla una aplicación basada en modelos de optimización como el como el problema de "maximum cardinality matching" y programación entera, que comparamos con una heurística comúnmente usada. Se analizan variaciones de los modelos de optimización presentados para explorar la verosimilitud de sesiones específicas y características de las sesiones. Se obtiene como resultado sesiones de mejor calidad que las obtenidas con los métodos tradicionales, además de una metodología de análisis de ellas.

Palabras Clave: Web Usage Mining, Web User Session, Maximum Cardinality Matching, Network Flow Model, Integer Programming, Web Logs.

*Departamento Ingeniería Industrial, Universidad de Chile, Santiago, Chile
**Operations Research Department, Naval Postgraduate School, Monterey, California, USA

1. Introducción

Los archivos de Log de un servidor web contienen registros de las operaciones que realizan los usuarios al navegar por un sitio web, convirtiéndose en una potencial gran fuente de datos acerca de sus preferencias [23]. Un Log [2] es un gran archivo de texto donde cada línea (registro) contiene los siguientes campos: Tiempo de acceso al objeto web (Ej. página web), la dirección IP del usuario, el agente que es la identificación del navegador usado, y el objeto web. También contiene evidencia de las actividades de de los usuarios web y se le puede considerar como una gran encuesta sobre sus preferencias en relación a la información que aparece en el sitio web. Lo anterior ha motivado gran parte de la investigación que se realiza en web mining, y define un nuevo campo de investigación denominado Web Usage Mining [23].

Un archivo de Log por si mismo no necesariamente refleja las secuencias de páginas que acceden los usuarios web i.e., se registra cada acceso pero sin un único identificador que represente al cliente. Esto se debe a que muchos usuarios distintos pueden compartir la misma dirección IP y tipo de navegador (agente), generando la necesidad de reconstruir las sesiones de usuario usando los datos disponibles. En la actualidad se utilizan métodos heurísticos para reconstruir las sesiones desde los archivos de Logs, las que se basan principalmente en limitar la duración de las sesiones [3], [6] y [20]. Este trabajo se centra en proponer modelos de optimización para recuperar las sesiones y estudiar sus propiedades.

El presente artículo se organiza de la siguiente manera: La sección 2 resume el estado del arte en relación a la sesionización. La sección 3 presenta nuestro modelo de optimización. La sección 4 muestra variaciones del modelo de optimización para explorar la verosimilitud y propiedades específicas de las sesiones. La sección 5 describe los datos experimentales usados. La sección 6 presenta los resultados. La sección 7 concluye el trabajo y sugiere futuras investigaciones.

2. Estado del Arte

Las estrategias de sesionización, pueden ser clasificadas en reactivas y proactivas [20]. La sesionización proactiva captura todas las actividades realizadas por los usuarios durante su visita al sitio web, sin embargo, son invasivas y en general con poco resguardo a la privacidad de los usuarios. El uso de estas estrategias se encuentra regulado por ley en algunos países [20] de forma de proteger la privacidad de las personas [15]. Un ejemplo corresponde al uso

de cookie[1] [4] que registran las actividades del cliente de las cuales se puede extraer la sesión exacta del usuario. Otra técnica usada es la re-escritura de URL[2] [7], donde se incluye información en el URL que se envía al servidor que reconstruye la sesión. La forma mas invasiva de obtener sesiones es a través de los llamados Spyware, que son programas que registran cualquier actividad del usuario (Teclado, Mouse, etc.). Sin embargo son actualmente considerados como una actividad criminal en la mayoría de los países [16].

Las estrategias de sesionización reactivas tienen un alto nivel de resguardo a la privacidad de los usuarios ya que sólo usan los registros de Log y no manejan explícitamente los datos personales de los usuarios [20]. Sin embargo, los archivos de Log son una forma aproximada de obtener las sesiones por muchas razones. Los usuarios pueden tener el mismo IP debido a que los ISP comparten un limitado número de direcciones entre sus clientes. Los usuarios web pueden también hacer uso de los botones back y forward que en la mayoría de las veces no producen registros en los Logs del servidor. Otro factor que introduce ruido en los datos son los servidores Proxy[3] [9] que mantienen en cache un cierto numero de páginas frecuentemente visitadas para optimizar las velocidades de acceso, por lo cual nunca son registrados en los archivos de los del sitio web.

Los métodos que se manejan en la actualidad para reconstruir sesiones desde los archivos de Logs están basados en heurísticas que consideran un límite de tiempo para la duración de las sesiones (30 minutos) [20]. Otras heurísticas se basan en la estructura semántica del sitio y las sesiones se construyen de forma de seguir una semántica común [13].

Se han realizados estudios empíricos en relación al comportamiento estadístico de las sesiones. La función de probabilidad de distribución del largo n (número de saltos entre páginas) tiene un buen ajuste con un ley de potencia $(n^{\alpha}/\sum_k k^{\alpha})$ [10][22]. La distribución parece ajustarse a una variedad de sitios web, aunque con cambio en el parámetro α. Nosotros usamos esta propiedad que parece universal de las sesiones como una medida de calidad de éstas [18].

Existe una gran variedad de literatura para el minado de las sesiones una vez que estas han sido identificadas. Técnicas como análisis estadístico, reglas de asociación, clustering, clasificación, patrones secuenciales y modelamiento de dependencias han sido usados para descubrir patrones de comportamiento de usuarios web [12][14][21].

[1] Archivos que se almacenan en el computador del cliente que almacenan datos

[2] Dirección web de la página, e.g. http://www.dii.uchile.cl

[3] Servidor que almacena copias de páginas mas acezadas por los usuarios de una red, de forma distribuirlas en forma más rápida.

3. Modelos de optimización para la sesionización

Se presentan dos modelos de optimización para la sesionización, los cuales agrupan registros de Logs de un mismo IP y agente, así como también consideran la estructura de links del sitio web. A diferencia de la heurística, que construye las sesiones una por una, los algoritmos de optimización propuestos construyen en forma simultánea. Cada sesión así construida es una lista de registros de Logs, donde cada registro es usado una sola vez en una única sesión. En la misma sesión, un registro r_1 puede ser un predecesor inmediato de r_2 si: los dos registros poseen la misma IP y agente, un link existe desde la página asociada al registro r_1 hasta la página del registro r_2, y el registro r_2 se encuentra en una ventana de tiempo permitida según el registro r_1.

3.1. Bipartite Cardinality Matching

El primer modelo de optimización que presentamos está basado en el conocido problema "Bipartite Cardinality Matching" (BCM) (e.g. [1]), el cual consiste en encontrar en un grafo no dirigido el subconjunto de máxima cardinalidad que tenga la propiedad de "matching" (no hay 2 vértices que compartan la misma arista). Existen varios algoritmos especializados para resolver el problema BCM en un tiempo de computación $O(\sqrt{n}m)$ donde "n" es el número de vértices y "m" es el número de arcos (e.g. [1]). En nuestra red, cada registro es representado por dos nodos, unos que representan el predecesor inmediato y otros a los sucesores inmediatos. La figura 1 muestra un ejemplo con 6 registros.

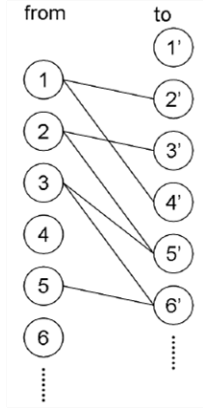

Figura 1: BCM: Cada registro es representado por dos nodos. Un arco existe si un nodo puede ser predecesor (from) de un vértice que puede ser su sucesor, según las restricciones del problema.

En cada lado, se ordenan los nodos (registros) en orden creciente de tiempo de acceso en los Logs del servidor web. Un arco existe de un nodo r_1 (from) a un nodo r_2 (to) si el registro correspondiente a r_1 puede ser un inmediato predecesor de r_2. Para el caso de la figura 1 asumimos que existen siete arcos.

Dada una solución, se construyen las sesiones de acuerdo al "matching" encontrado. Un vértice que no es sucesor de otro vértice, es el primer registro de la sesión. El resto de la secuencia de registros se reconstruye identificando los pares de vértices que corresponden a un mismo registro y siguiendo consecutivamente los sucesores habilitados por un arco. La figura 2 provee una solución factible de acuerdo a la figura 1. Los vértices 4 y 6 son los últimos en una sesión (no tienen sucesores), los vértices 1'y 4'van primero en sus respectivas sesiones ya que no tienen antecesores, las sesiones resultantes son entonces 1-2-3-5-6 y 4.

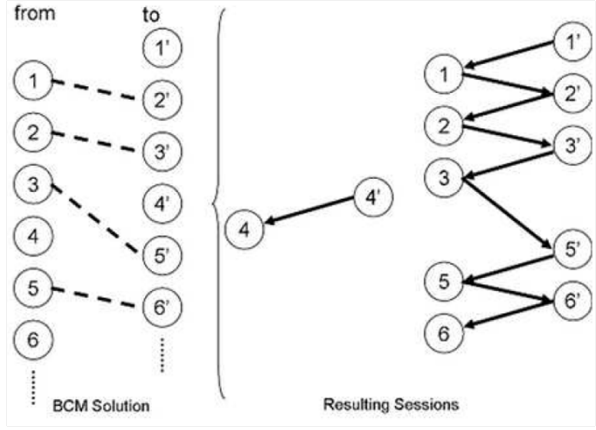

Figura 2: BCM tiene cuatro arcos. Se construyen dos sesiones desde el matching: 1-2-3-5-6 y una sesión con un sólo registro 4.

3.2. Un modelo de programación entera para la sesionización

Una solución óptima para el modelo BCM provee un límite inferior en el número de sesiones para un archivo de Log dado. Esta solución óptima tiene la propiedad que pocas sesiones tendrán largos pequeños (1 o 2). El limite superior en el número de sesiones para un archivo de Logs dado, es el numero de registros donde cada registro es una sesión por si mismo. De forma de construir otras sesiones con un límite superior en su cantidad, formulamos un modelo de programación entera.

El modelo de sesionización desarrollado, utiliza en su formulación programación entera (SIP) usa una variable binaria X_{ros} que tiene valor "1" si el registro de *Logr* es asignado en la posición o durante la sesión s y "0" en cualquier otro caso. Cada índice r identifica a un único registro, cada índice

s identifica a una única sesión de usuario, y el índice o es la posición de un registro en una sesión.

Presentamos a continuación la formulación del problema en formato NPS [5].

- **Índices**

  o Orden de un registro de Log durante una sesión (e.g. *o=1,2,...,20*). La cardinalidad de este conjunto define el máximo tamaño de una sesión.

  p,p' Numera a las páginas web.

  r,r' Numera a los registros de Logs.

  s Numera a las sesiones de usuarios.

- **Conjuntos de Índices**

  $r' \in bpage_r$ Es el conjunto de registros que pueden estar inmediatamente antes del registro r en la misma sesión. Basado en:

  - Páginas que tienen un link a la página del registro r.
  - La dirección IP que comparten r y r'.
  - El agente que comparten r y r'.
  - El tiempo del registro r y r', r debe ocurrir antes de r' dentro de una ventana de tiempo.

  $r \in first$ Es el conjunto de registros que puede ocurrir en primer lugar en una sesión.

- **Datos**

  Estos son usados para generar los conjuntos anteriores.

  $time_r$ El tiempo del registro r.

  ip_r La dirección IP del registro r.

  $agent_r$ El agente del registro r.

  $page_r$ La página del registro r.

  $\underline{mtp}$, mtp El tiempo mínimo y máximo entre páginas (segundos).

- **Variables Binarias**

  X_{ros}: "1" si el registro de Log r es asignado en el lugar o en la sesión s y "0" en otro caso.

- **Formulación**

Maximizar $\sum_{ros} C_{ro} X_{ros}$
s.a.

$$\sum_{os} X_{ros} = 1 \quad \forall r \tag{1}$$

$$\sum_{r} X_{ros} \leq 1 \quad \forall o, s \tag{2}$$

$$X_{r,o+1,s} \leq \sum_{r' \in bpage_r} X_{r'os} \quad \forall r, o, s \tag{3}$$

$$X_{ros} \in \{0,1\}, \quad \forall r, o, s. \quad X_{ros} = 0, \quad \forall r \in first, o > 1, s$$

La función objetivo expresa cuánto se considera a las sesiones de mayor largo donde$\sum_{r,o' \leq o} C_{ro'}$ es el beneficio por una sesión de largo o'. Por ejemplo, si fijamos $C_{ro'} = 1, \quad \forall r, o = 3$ y $C_{ro'} = 0, \quad \forall r, o \neq 3$ se tendrá una función objetivo que maximiza el número de sesiones de largo tres. La sección 5.2 muestra una variedad de resultados asociados a diferentes elecciones de parámetros C_{ro}.

El conjunto de restricciones (1) aseguran que cada registro sea usado una vez. Las restricciones (2) aseguran a cada sesión al menos un registro asignado a cada posición o. Las restricciones (3) aseguran un orden apropiado de registros en la misma sesión.

4. Otras variantes del modelo

Durante la investigación, se desarrollaron otras variantes del modelo para explorar la verosimilitud de sesiones específicas y características de las sesiones. Específicamente, se logra establecer una aproximación para el máximo número de copias de una respectiva sesión, el máximo número de sesiones de un mismo largo y el máximo número de sesiones con una determinada página web en un cierto lugar.

4.1. El máximo numero de copias de una sesión

Para encontrar el máximo número posible de copias de una sesión dada en un registro de Log, tenemos que analizar dos casos, según si se considera repetición de páginas:

1. Cuando cada página en la sesión es visitada solo una vez, esto puede ser modelado como un problema de máximo flujo (e.g. [1]). El problema

de máximo flujo busca una solución que envía el máximo flujo desde un vértice artificial llamado fuente hasta un vértice sumidero. Se construye el grafo con un vértice por cada registro de Log que corresponden a las páginas de la sesión escogida, se además incluye la fuente y el sumidero. Los arcos se construyen de la siguiente forma: los que parten de la fuente apuntan a registros que pueden ser registros iniciales de una sesión, se incluyen arcos desde cada registro hasta el sumidero y entre dos registros que puede ser inmediatos predecesor y sucesor. Los arcos relacionados con la fuente y sumidero tienen infinita capacidad, el resto tiene capacidad máxima de uno.

2. En el caso que las páginas en la sesión se repiten, se introducen restricciones adicionales para el problema de flujo máximo. En este caso, la red es similar al anterior pero se debe registrar el orden en que se acceden las páginas debido a la repetición. Entonces para páginas repetidas se repiten registros que correspondan a la página repetida manteniendo el orden y se restringe que el máximo flujo total que pueda salir de estos registros repetidos sea menor o igual a uno.

4.2. Maximizando el número de sesiones de un mismo largo

Usando el modelo SIP de la sección 3.2, se puede encontrar el máximo número de sesiones de un largo l ajustando la función objetivo a $C_{ro} = 0, \quad \forall r, o \neq l$y $C_{ro} = 1, \quad \forall r, o = l$. Con estos coeficientes, sólo las sesiones de largo mayor o igual que l tienen valor uno. Una solución optima puede incluir sesiones mayores que el que este largo, pero pueden ser separadas en dos sesiones donde una de ellas es de largo l.

4.3. Maximizando el número de sesiones con una página fija en cierta posición

Usando el modelo SIP, se puede encontrar el máximo número de sesiones que visitan una página fija en la posición o. Para ello, se fijan los coeficientes $C_{ro} = 1$ cuando el registro r corresponde a la página fija y/o a la posición escogida, en caso contrario se anula.

5. Los datos experimentales

Durante la presente investigación, se trabajó con un sitio web universitario (http://www.dii.uchile.cl) que mantiene el Departamento de Ingeniería Industrial de la Universidad de Chile. El cual esta compuesto por los sitios web corporativos, de proyectos, programas de diplomados y postgrado, páginas

personales, y webmail entre otras. Corresponde a un sitio web con alta diversidad y un tráfico de Internet suficientemente alto para satisfacer los objetivos de este estudio, aun sin considerar el sistema de webmail cuyo registro de Log no contiene sesiones muy variadas.

5.1. Estadísticas preliminares

Se obtuvieron 3.756.006 registros durante el mes de abril de 2008. Para observar las transiciones entre páginas, se deben filtrar los accesos a objetos multimedia y otros que no corresponden a páginas web. Se deben eliminar registros de spider[4] o robot, herramientas automáticas de monitoreo de seguridad[5], ataques de virus[6] que no son propiamente considerados como pertenecientes a sesiones realizadas por agentes humanos. Se obtuvo un total de 102.303 registros de páginas estáticas HTML con un total de 172 páginas, de estos 9.044 registros corresponden a accesos a la raíz del sitio web.

Se obtuvo que tan sólo unas pocas direcciones IP las que recogen la gran mayoría de los accesos a registros. Sobre un 98 % de todas las direcciones tiene menos de 50 registros por mes. De la misma forma, se constató que pocas direcciones IP tienen el acceso más diverso a páginas. Se almacenó la estructura de link del sitio web usando un web crawler[7] [17], obteniéndose 172 páginas con 1.228 link entre ellas considerando solo páginas registradas en los Logs.

5.2. Pre-procesamiento de datos

Como mencionamos existe una gran cantidad de registros triviales de analizar en el sentido que corresponden a sesiones de largo, ya sea por la baja cantidad de accesos o por la poca variedad de páginas accedidas. En este estudio nos enfocamos en el subconjunto de registros que muestren la mayor diversidad de patrones de acceso. Para ello, se consideran registros para nuestra sesionización mediante filtrado por numero IP. Para cada registro IP se propone una medida de diversidad de páginas visitadas basada en la entropía de distribución de páginas $S = \sum_p f_p Log_N (1/f_p)$, donde f_p es la frecuencia de accesos a la página p y N es el número total de páginas. La entropía S adquiere su valor máximo (igual a 1) cuando la distribución de acceso a páginas

[4]Programas automáticos que recorren la web visitando sitios, típicamente usados por los buscadores (e.g. Google) para mantener su base de datos de la web actualizada.

[5]Programas automáticos usados por los administradores de sistemas para chequear el estatus del sitio web.

[6]Existen virus que se propagan visitando sitios web e intentando ingresar y modificar las páginas para infectarlas, de forma que algún otro usuario que la visite se infecte.

[7]Un crawler es un programa automático que recorre las páginas web de un sitio almacenando su estructura y contenido.

es uniformemente distribuida, es decir existe diversidad de acceso. En cambio, cuando su valor es cercano a cero, solo unas pocas páginas son accedidas frecuentemente. En este caso agrupamos los registros por numero IP que tuviesen alta entropía ($S > 0,5$) y un alto numero de registros ($Log(N) > 3,8$) asegurando diversidad de comportamiento.

6. Resultados obtenidos

Se presenta los resultados de la sesionización y usamos una medida de la calidad de los resultados.

6.1. Medición de la calidad de las sesiones

Sin tener acceso a comparar con las verdaderas sesiones de usuarios, no es posible conocer en forma exacta cuán realista es el método desarrollado en el presente articulo, se propone comparar la distribución de largos de sesiones con la distribución empírica [10][22] observada en forma natural. Para ello se usa regresión lineal en el logaritmo del largo y logaritmo del número de sesiones. Se entregan entonces como resultado de la regresión, el coeficiente de correlación y el error estándar como medidas de la calidad de las sesiones.

6.2. Procesamiento

Es fácil construir instancias del modelo SIP propuesto que no puedan ser resultas por computador alguno debido al gran número de variables involucradas. Por ejemplo un servidor web (como el estudiado) con 100.000 registros, con un máximo de 5.000 sesiones y un largo máximo de sesiones de 20, genera 1010 variables binarias y un número aun mayor de restricciones. Un problema de esa envergadura requeriría cerca de 1Tb de memoria de acceso directo solo para almacenar las variables, las cuales deberían además modificarse en cada iteración. Afortunadamente se puede subdividir el problema separando el archivo de Log en unidades mas pequeñas que se agrupen por numero IP y agente; y fijando un limite máximo de tiempo entre registros (15 min.). Con esto se obtuvieron 403 unidades de registros fijando además que cada unidad disponga un mínimo de 50 registros para evitar por otro lado un número mayor de unidades. Todos los procesamientos fueron realizados en un PC de 1,6Ghz con 2Gb RAM. Se resolvieron las instancias usando el software GAMS [8] en los 403 programas lineales utilizando CPLEX [11] con la versión 10.1.0. El sistema generador de instancias fue realizado en PHP y los datos se almacenaban en una base de datos MySQL 5.0.27 [24].

6.3. Resultados del algoritmo BCM

Se resolvieron las 403 unidades usando CPLEX, donde la instancia más grande consistía en 1.500 variables y 200 restricciones. El tiempo total de resolución para las 403 unidades fue de menos de 5 minutos. Usando un algoritmo BCM especializado, se puede reducir mayormente el tiempo y además puede ser paralelizada resolviendo las 403 unidades concurrentemente. La solución entregó 12,366 sesiones. La sesión más larga tiene 41 registros, después vienen sesiones menores o iguales a 14 registros. Considerando todas las sesiones, obtuvimos un coeficiente de correlación de $R^2 = 0,88$ y un error estándar de 1,23. Considerando largos de sesiones hasta los 14 registros obtenemos un coeficiente de correlación $R^2 = 0,98$ y un error estándar de 0,39 (ver figura 3). Claramente la única sesión de 41 registros debe ser considerada especialmente ya que se encuentra en el rango fraccionario de la ley de distribución de potencias y el no considerarla para la regresión es una buena aproximación.

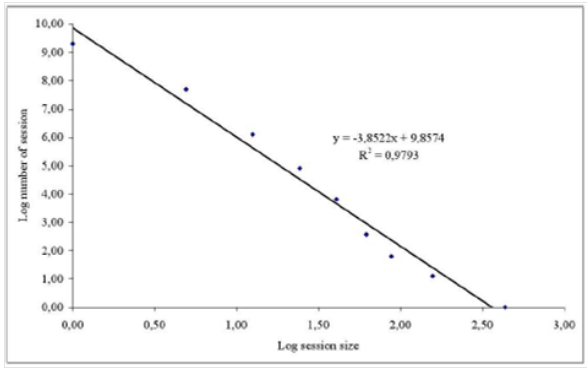

Figura 3: Regresión lineal de los resultados BCM.

6.4. Resultados SIP

Este modelo depende de la elección de los coeficientes lineales en la función objetivo. Por ejemplo si se escoge $C_{ro} = 0, \quad \forall r, o \neq 1$y $C_{ro} = 1, \quad \forall r$; entonces se busca maximizar el número de sesiones entregando como solución trivial sólo sesiones con un solo registro. Se requiere, por tanto, que los coeficientes C_{ro} sean una función monótona creciente en "o", con el fin de valorizar más a las sesiones de mayor largo en el problema de optimización. Para estos fines, se experimentó con varias funciones con distintas tazas de crecimiento obteniendo los resultados de la tabla 1:

El tercer conjunto mostró el mejor ajuste. Cabe señalar que corresponde a la elección de $C_o = Log(1/P_o)$ la cantidad de información siendo Po la probabilidad de tener una sesión de largo o. Como hemos mencionado, esta probabilidad tiene como aproximación una Gausiana inversa, cuyo valor se

	C_{ro}	R^2	StdErr	Total Sesions
1	$1/\sqrt{o}$	0.88	1.10	12.502
2	$Log(0)$	0.93	0.66	12.403
3	$3/2Log(o) + (o-3)^2/12o$	0.94	0.59	12.403
4	o	0.93	0.63	12.409
5	o^2	0.92	0.72	12.410

Tabla 1: Conjuntos de coeficientes para la función objetivo

ajusto en este caso. Consideramos como argumento de plausibilidad parea esta elección que la función objetivo se aproximaba a la entropía de la distribución de largos de sesiones, en cuyo óptimo se aproxima a la distribución deseada. Finalmente los tiempos totales de procesamiento fueron de 3 a 5 horas.

6.5. Comparación con la heurística comúnmente usada

Se evaluaron los resultados comparando con la heurística basada en el tiempo máximo de sesiones. Claramente la heurística consume mucho menos recursos computacionales resolviéndose en menos de 10 segundos para todos los registros y sin preprocesar. Pero obtiene el peor ajuste a la distribución estadística de largos de sesiones ($R^2 = 0,92$ y $std = 0,64$) teniendo aproximadamente el doble del error estándar obtenido con el método BCM (ver figura 4).

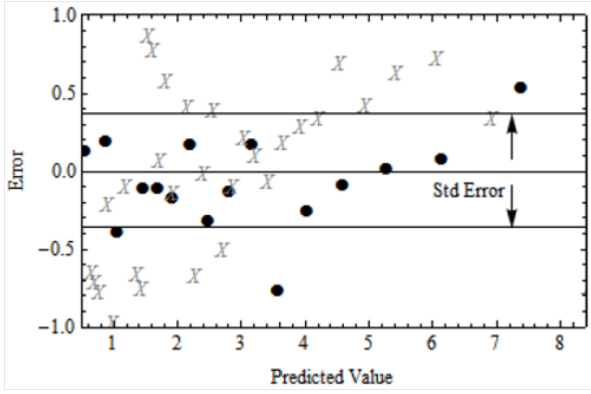

Figura 4: Error de ajuste a la ley de potencia del largo de sesiones.

Si se comparan los resultados logrados por SIP (mejor caso) y BCM en el número de sesiones logradas, tendríamos un GAP del 0,3 % entre ambos (SIP: cota superior, BCM: cota inferior) lo que indica la calidad de las sesiones obtenidas por este nuevo método.

6.6. El máximo número de copias de una sesión

Con el algoritmo propuesto, se rankea el máximo número de secuencias iguales de páginas obtenidas por todas las sesiones obtenidas con los métodos anteriores. Así, el procesamiento de este ranking de verosimilitud tardo menos de 9 horas, cuyos resultados son consignados en la tabla 2.

Session	BCM	SIP	max
1	41	41	186
2	5	3	43
3	4	5	39
4	7	4	34
5	4	4	34
6	1	0	22
7	1	0	22
8	7	0	19
9	1	0	16
10	1	0	16

Tabla 2: Las 10 sesiones más verosímiles y su comparación con su ocurrencia según el algoritmo.

6.7. El máximo número de sesiones según el tamaño

Como se indicó anteriormente, se ajusta la función objetivo en el modelo SIP para obtener el máximo número de sesiones de un mismo tamaño. Con ello se obtuvieron los resultados mostrados en la tabla 3.

Size	Num. Sessions
2	3000
3	1509
4	755
5	435
6	257
7	181
8	135
9	98
10	82

Tabla 3: Máximo número de sesiones dado un largo fijo.

6.8. Máximo número de sesiones que visitan una página en cierto orden

Se usó este algoritmo para rankear las páginas que aparecen en tercer lugar dado el máximo número de sesiones según este algoritmo. Para ello se proceso

para todas las páginas factibles de aparecer en tercer lugar (fueron 58) y se demoró aproximadamente de 29 horas.

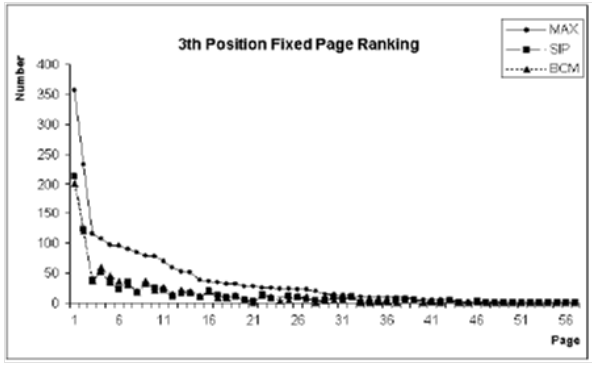

Figura 5: Ranking de páginas más verosímiles de aparecer en tercer lugar.

Este ranking entrega la noción de páginas más verosímiles de encontrar en tercer lugar en una sesión dado la posición examinada. La consistencia se compara con el número de apariciones en los otros algoritmos (figura 5).

7. Conclusiones

Se presenta un nuevo enfoque para la sesionización usando algoritmos de programación entera. Cuando se compara con los métodos heurísticos tradicionales, se reduce en un 50 % el error estándar de la distribución de largo de sesiones con respecto a la distribución esperada. Se dispone, además, de una versión altamente eficiente basada en el problema de "Bipartite Cardinality Matching". Esto nos permite obtener máximos sesiones con máxima probabilidad de ocurrencia dado largo fijo, secuencia de páginas fija y página fija en cierta posición. Los modelos no contemplan la posibilidad de uso de los botones back y forward del navegador, sin embargo dada la flexibilidad de los modelos de optimización en incorporar restricciones es posible lograr recuperar sesiones que incluyan este efecto.

Agradecimientos: Se agradece el apoyo del Instituto Milenio de Sistemas Complejos de Ingeniería y la Beca de Doctorado Nacional Conicyt.

Referencias

[1] Ahuja, T. Magnanti, J. Orlin. Network Flows: Theory, Algorithms, and Applications. *Prentice Hall*, 1993.

[2] Aulds, C. Linux Apache Web Server Administration. *Sybex.* 2002.

[3] Berendt, B., A. Hotho, G. Stumme. Data preparation for mining world wide web browsing patterns. *Journal of Knowlegde and Information Systems*, Vol.1 5-32. 1999.

[4] Berendt, B., B. Mobasher, M. Spiliopoulou, J. Wiltshire. Measuring the accuracy of sessionizers for web usage analysis. *Proc.of the Workshop on Web Mining*, First SIAM Internat.Conf. on Data Mining. 7-14. 2001.

[5] Brown, G., Dell, R. Formulating integer linear programs: A rogues'gallery. *Informs Transactions on Eduacation*, 7. 2007.

[6] Cooley, R., B. Mobasher, J. Srivastava. Formulating integer linear programs: A rogues'gallery. *Towards semantic web mining.* Proc. in First Int. Semantic Web Conference, 264-278. 2002.

[7] Facca, F., P. Lanzi. Recent developments in web usage mining research. *DaWaK*, 140-150. 2003.

[8] GAMS Development Corporation. General algebraic modeling system (gams). *www.gams.com*, Accessed December 2008.

[9] Glassman, S. A caching relay for the world wide web. *Computer Networks and ISDN Systems*, Vol.27 165-173. 1994.

[10] Huberman, B., P. Pirolli, J. Pitkow, R. Lukose. Strong regularities in world wide web surfing. *Science*, Vol.280 95-97. 1998.

[11] ILOG. 2008. Cplex 2008. Strong regularities in world wide web surfing. *www.ilog.com/products/cplex*, Accessed December 2008.

[12] Joshi, A., R. Krishnapuram. On mining web access Logs. *Proc. of the 2000 ACM SIGMOD Workshop on Research Issue in Data Mining and knowledge Discovery*, 63-69. 2000.

[13] Jung, J., Geun-Sik Jo. Semantic outlier analysis for sessionizing web Logs. *ECML/PKDD Conference*, 13-25. 2004.

[14] Kosala, R., H. Blockeel. Web mining research: A survey. *SIGKDD Explorations: Newsletters of the special Interest Group (SIG) on Knowledge Discovery and Data Mining*, 1 1-15. 2000.

[15] Langford, D. Internet Ethics. *MacMillan Press Ltd.* 2000.

[16] Mayer-Schonberger, V. Nutzliches vergessen. *Goodbye Privacy Grundrechte in der digitalen Welt (Ars Electronica)*, 253-265. 2008.

[17] Miller, R., K. Bharat. Sphinx: A framework for creating personal, site-specific web crawlers. *Proceedings of the Seventh International World Wide Web Conference (WWW7)*, 119-130. 1998.

[18] Román, P., R. Dell, J. Velásquez. Web user sesión reconstruction using integer programming. *Proc. of the 2008 Web Inteligence Conference*, Sidney, Australia. 2008.

[19] Román, P., J. Velásquez. Markov chain for modeling web user browsing behavior: Statistical inference. *XIV Latin Ibero-American Congress on Operations Research (CLAIO).* 2008.

[20] Spiliopoulou, M., B. Mobasher, B. Berendt, M. Nakagawa. MA framework for the evaluation of sesión reconstruction heuristics in web-usage analysis. *INFORMS Journal on Computing*, Vol.15 171-190. 2003.

[21] Srivastava, J., R. Cooley, M. Deshpande, P. Tan. Web usage mining: Discovery and applications of usage patterns from web data. *SIGKDD Explorations*, Vol.2 12-23. 2000.

[22] Vazquez, A., J. Oliveira, Z. Dezso, K. Goh, I. Kondor, A. Barabasi. Modeling bursts and heavy tails in human dynamics. *PHYSICAL REVIEW E 73*, 2006.

[23] Velásquez, J., V. Palade. Adaptive Web Sites: A Knowledge Extraction from Web Data Approach. *IOS Press*, Amsterdam, NL. 2008.

[24] Zawodny, J., D. Balling. High Performance MySQL. *O'Reilly*, 2004.