Large-scale Mining, Discovery and Visualization of WWW User Clickpaths

Brendan Kitts, Kevin Hetherington, Martin Vrieze

Vignette Corporation
230 Third Avenue, Waltham, MA. 02451. USA
Email: bkitts@vignette.com
Ph: 781 487-2478
Fax: 781 487-2801

Abstract

Analysis of user clickstreams on the world wide web is made challenging by the volume of data and the difficulty of visualizing millions of different navigation paths. We present a method for identifying user clickpaths which scales well on large amounts of data, and provides an intuitive and insightful visual representation of user activity. Our technique borrows from the data mining literature on association rules and the computer graphics literature on graph layout optimization. The method is demonstrated with data from two commercial sources and paints a fascinating picture of web activity.

Keywords: click, clickpath, clicktrail, clickstream, graph, visualization, web, usage mining, navigation

Introduction

The world wide web has only been in existence for a short time, yet has already changed the way we work, shop and play. Yet, despite the proliferation of activity, analysis of user navigation has remained rudimentary. This is in stark contrast with the off-line world where elaborate studies have been conducted in order to understand shopper behavior (Underhill, 2000).

The mystery as to "where customers are going" and "what causes a purchase" is especially problematic for commercial sites, which face a desperate battle to improve browser-to-buyer conversion rates, increase revenue from transactions, and deliver meaningful content before the user departs.

Making sense of user clickstreams has therefore become a key problem area for data mining. Clickstream analysis or what some have termed "Web usage mining" presents a series of difficulties ranging from the manipulation of unprecedented amounts of data - hundreds of times the volume of off-line sources - to analysis techniques and finally visualization.

In this paper we propose a new method to address these difficulties. Our method is designed to scale and provide insightful visual analysis of user navigation. We demonstrate the method using commercial data sets.

Previous work

The mining of associations from data was originally pioneered by Agrawal et. al. (1996) who was interested in mining retail Point of Sales data. Association mining was subsequently applied to web pages by many authors including Mannila, et. al. (1995), Zaki (2000) and Buchner et. al. (1999). Borges and Levene (1999) recognized that the above association patterns could be formalized in a language they called a Hypertext Probabilistic Grammar, a subclass of probabilistic regular grammars. Web log miners have been developed by Spiliopoulou, Faulstich et. al., (1999) and Wu et. al. (2001). Pitkow and Bharat (1994) is one of the few previous attempts we know of to generate graphs of web usage activity, and we extend their work by using significance thresholds for showing or hiding edges, and improving the graph layout by using force-directed methods.

Although the idea of using graphs to represent clicktrails has been recognized in previous work, it has rarely been attempted except for illustrative purposes, and has not been attempted on a large scale, for example, to view a site. Our primary contributions are showing how clickpaths can be efficiently extracted, tested for statistical significance, and displayed with the aid of force-directed graphing techniques. The combination of these methods allows us to reveal dramatic insights into customer navigation behavior that would otherwise have remained hidden in the data.

Preliminaries

Let $W=<e_1,e_2,e_3,\ldots,e_g>$ be a web log where e_i is an event tuple comprising $e_i=(s,c,t,p)$. s is the session identifier, c is the customer, t is the time, and p is the page or content requested. After we process the web log to recover click sequences for each customer (essentially the sequence of pages p visited by each customer c,session s, listed in time order), we can talk in terms of simplified sequences of pages called clickstreams. Let $A=<a_1,a_2,\ldots,a_q>$ and $B=<b_1,b_2,\ldots,b_r>$ be sequences of pages or clickstreams, $X_s=<arrivesite,x_1,x_2,\ldots,x_o,leavesite>$ be the clickstream for session s, and $X=\{X_1,X_2,\ldots,X_n\}$ be the set of all clickstreams in W. <.> defines a structure known as a sequence in which all elements are ordered and duplicates allowed. $time(x \in X)$ is the ordinal position of x in the sequence X. # denotes the cardinality operator.

The *Traffic* of a page sequence is the number of times the sequence was accessed in different sessions.

$$Traffic(A) = \sum_{s} \left(A \subseteq X_s \right)$$

The probability of a page sequence occurring in a session is equal to the number of sessions in which that page occurred, divided by the number of sessions.

$$\Pr(A) = \frac{Traffic(A)}{\#X}$$

In trying to understand user clickstreams, we will be particularly interested in page sequences which co-occur in the same user sessions. Pages which co-occur often will be said to have an "affinity" with each other (Agrawal et. al., 1996). We will define an affinity as follows:

$$Affinity(A,B) = \sum_{s} \left(A,B \subseteq X_s : \psi \right)$$

where ψ is a predicate that determines if the sequence contributes to the count. We will categorize web affinities into six major types (Figure 1):

1. Full paths: complete paths beginning at an arrival page, ending at a departure page, eg. <*arrivesite*,*a*,*b*,*c*,*d*,*a*,*b*,*leavesite*>
2. Forward paths: paths through the site traveled without backing up. Eg. <*a*,*b*,*c*,*d*>
3. Path fragments: commonly traversed sub-paths through the site, eg. <*b*,*c*,*d*>
4. Sequential affinities: next-click sequences through the site, eg. <*a*,*b*>
5. Temporal affinities: anticipation of what page a customer is trying to reach several clicks ahead, eg. <*a*,*c*>
6. Non-temporal affinities: pages that appear in the same sessions, eg. <*b*,*a*>

We will also describe two additional analyses relating to single pages:

7. Real-estate analysis: proximity of each page to the site's "front door", and the number of hits and revenue that each page attracts compared to its expected performance.
8. Simple page statistics: number of hits, sessions, and revenue generated.

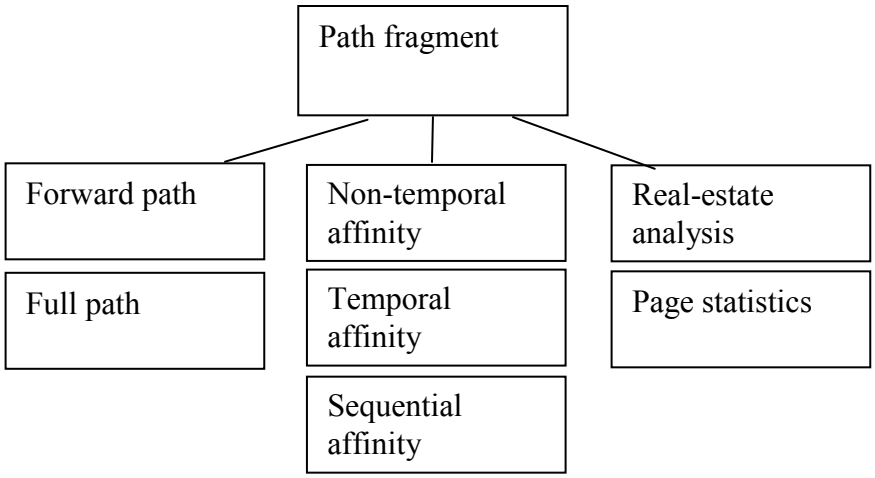

Figure 1: Taxonomy of clickstream analyses

For the remainder of this paper we will show how affinities can be efficiently computed, we will introduce Non-temporal, Temporal, and Sequential affinities, we will describe how they can be graphed, and show examples of their application to commercial web sites. We will also introduce Real-estate analysis, which is a novel method for measuring the performance of pages.

Generating Affinities from Very Large Data Sources

Calculating affinities requires the analysis of vast amounts of data. This presents a problem. Large amounts of data generally cannot be copied into computer Random Access Memory (RAM). Instead, algorithms have to retrieve small chunks of data into memory from mass storage devices such as disks or relational databases. Because of the sequential read from these devices, the algorithm for traversing this data set must be carefully designed.

To calculate affinities, an algorithm must count the number of times that page *a* appeared with page *b* in the same customer session *s*. We might start with the first record of web log data, read the session identifier *s* and the page *a*. Next we might scan down the web log data for every other instance of the same session *s*, and increment a counter (*a*,*b*) for each page *b* encountered. This algorithm has a time complexity of $O(\#W^2)$.

However, if the same data could be contiguously ordered by session, the time complexity of this algorithm could be reduced substantially.

Definition 1: Contiguous re-ordering

A contiguous re-ordering by a variable s is a function $f_s: W \rightarrow W$ such that $\neg\exists(<(s_i,c_x,t_y,p_z),(s_j,c_{x'},t_{y'},p_{z'})> \cup \varphi \cup <(s_i,c_{x''},t_{y''},p_{z''})> \subseteq (f_s^{\circ} W) : s_i \neq s_j)$ In other words, after encountering a session s_i followed by s_j, we will never encounter s_i in any later records.

Lemma 1: After contiguously re-ordering W by session, the time to compute pair affinities for one session X_s in W can be bounded from above by $O(\#X_s^2)$.

Proof

The re-ordering operation ensures that when we encounter a new session s during a sequential scan of W, all subsequent pages in the same session will be found immediately following. Therefore, to find all pages b that are associated with a, we need only step through $\#X_s$ pages, rather than $\#W$ pages. The number of combinations of (a,b) in X_s is equal to $\#X_s$ combinations for the first pair, multiplied by $\#X_s$-1 combinations for the second. Thus the number of pairs to traverse is bounded from above by $O(\#X_s^2)$.

℘

Lemma 2: After contiguously re-ordering W by session, if P is the maximum length of any session in W, then the worst-case time complexity for counting pair affinities will be realized when all sessions (with the possible exception of the last one) are equal to P.

Proof

The processing time of each session adds together to give the total processing time. Because of this, we need only compare the time to process a single session Y of size P (1) against the time to process h sessions $X_{1..h}$ such that $\sum_{i=1}^{h} \#X_i = P$ (2). In other words, (1) and (2) have the same number of constituent pages being processed; the difference is that we have broken (2) into h sessions. Comparing the time to process (1) against time to process (2), we have

$$\sum_{i=1}^{h} ComputationTime(X_i) \leq ComputationTime(Y)$$

$$\sum_{i=1}^{h} \#X_i^2 \leq P^2$$

$$\sum_{i=1}^{h} \#X_i^2 \leq \left(\sum_{i=1}^{h} \#X_i\right)^2$$

which is true by triangle inequality. Therefore the time to process a session of size P will always be greater than or equal to the time to process a series of smaller sessions which together add to give the same number of pages. Thus a web log of sessions of length P will incur the slowest running time for the algorithm.

℘

Lemma 3: After a contiguous re-ordering, the worst-case time complexity to count all pair affinities in W can be bounded from above by $O\left(P^2 \cdot \left\lfloor \#W / P \right\rfloor + (P-1)^2\right)$.

Proof

Since in the slowest time case, each session is P in length, the number of sessions to be processed would be $\left\lfloor \#W / P \right\rfloor$ plus one remainder session of size < P, or 0 if P divides without remainder into #W. Therefore, time to process all sessions (except for the remainder session) will be $P^2 \cdot \left\lfloor \#W / P \right\rfloor$ (3). The remainder session must always have a length which is less than or equal to P-1, and the time to process that session will therefore be $(P\text{-}1)^2$ (4). Thus the addition of (3) and (4) gives us the upper bound for the total processing time.

℘

Discussion

The upshot of these results, is that if a contiguous re-ordering is performed, the time complexity of the affinity counting algorithm can scale as a quadratic function of the maximum session length P, rather than as a quadratic function of the web log length #W. Therefore, P becomes the primary bottleneck for controlling the speed of processing.

If a session is encountered with more than a user-defined threshold of P_{MAX} pages, we can even take the additional step of either truncating or removing that session. This strategy is advantageous for several reasons. Many long paths are generated by robots, site programs that test if pages are accessable, or other autonomous processes that are unlikely to elicit meaningful clicktrails. Thus overly long sessions should probably be removed. But we just saw that algorithm speed is critically dependent upon P_{MAX}. By setting P_{MAX} ahead of time, we can dramatically speed up the performance of the algorithm, and improve the predictability of the running time. This simple improvement makes the algorithm fast, scalable and robust to vast amounts of real-world data.

One detail has yet to be worked out – how can the web log W be efficiently re-ordered?

Lemma 4: Assuming the existence of a hash function with zero collisions, W can be contiguously ordered in O(#W) time using mass disk storage.

Proof

The basic idea is to use a disk-based hashing algorithm to store all data pertaining to the same session together on disk, and then traversing the stored data using this hash index. If the hashing function does not cause any collisions the hashing step will require one pass of the data.

Discussion

In conclusion, for the expense of an O(#*W*) re-ordering, our total time for computing affinities can be reduced from O(#W^2) to O(P_{MAX}^2). This is a dramatic difference (eg. consider P_{MAX}=100 and #*W*=1x10^7). This strategy enables us to process vast amounts of data very quickly. The method achieves its performance without relying upon RAM, and as a result, is scalable on vast amounts of data.

Generating Clickstream Graphs

After computing affinities, the results could be viewed by reading through text reports like that shown in Figure 2. Unfortunately, such reports have the disadvantage of being long and not providing much insight into broader patterns of user navigation.

Top lift affinities for ParenthoodWeb

SiteB	Lift(A,B)	Pr(B|A)	Pr(A|B)	Pr(A)	Pr(B)
BabyCenter	35.7	9.7%	2.2%	0.1%	0.3%
ParentTime	34.9	1.8%	2.2%	0.1%	0.1%
ParentsPlace.com	32.4	2.1%	2.0%	0.1%	0.1%
Parent Soup	22.8	5.4%	1.4%	0.1%	0.2%
Burstware	12.9	4.1%	0.8%	0.1%	0.3%
IVillage	8.8	4.5%	0.6%	0.1%	0.5%
Family.com	5.7	2.1%	0.4%	0.1%	0.4%
About.com	5.4	3.1%	0.3%	0.1%	0.6%
Women.com	4.1	2.4%	0.3%	0.1%	0.6%
CoolSavings	4.0	1.6%	0.3%	0.1%	0.4%
Mining Company, The	3.9	11.3%	0.2%	0.1%	2.9%
HomeArts	3.7	4.6%	0.2%	0.1%	1.2%
WWWomen.Com	3.4	2.5%	0.2%	0.1%	0.7%
Women's Wire	3.4	5.5%	0.2%	0.1%	1.6%
Oprah	2.6	1.8%	0.2%	0.1%	0.7%

Figure 2: Affinities for SiteA=ParenthoodWeb from Experiment 1, sorted by lift score

Recognizing that *Affinity(a,b)* is an *I*x*I* matrix where *I* is the number of pages, we should be able to represent these affinities using a graph. Each node of the graph may represent a page, and each edge between two nodes will represent an affinity statistic.

However, there may be thousands of page sequences in a large site and displaying this many arcs in graphical form would be unwieldy. This is where *Graph Drawing* methods can help (Battista, et. al. 1999). Graph Drawing algorithms attempt to optimize graph layouts. They have attracted a great deal of interest because of their application in industrial circuit design (Quinn and Breur, 1979). A new design needs to be layed out on a 2D wafer, so that the total number of wire crossings are kept to a minimum. Also, the

surface area of the circuit should be kept as low as possible to again minimize material use. The same techniques can be applied to make graphs easier to read.

A popular approach to optimizing graph layout is the use of force-directed methods (Tutte, 1963; Kamada and Kawai, 1989; Davidson and Harel, 1996). These use the idea of attraction and repulsion between nodes and edges to optimize graph layout; for example, edges can be represented as springs, and nodes as charged particles which repel each other. The force in the x-dimension on a vertex can then be calculated as:

$$F_X(v) = \sum_{(u,v)\in E} k_{uv}^{(1)}(d(p_u,p_v) - l_{uv})\frac{x_v - x_u}{d(p_u,p_v)} + \sum_{(u,v)\in V} \frac{k_{uv}^{(2)}}{d(p_u,p_v)^2}\frac{x_v - x_u}{d(p_u,p_v)}$$

where l_{uv} is the natural or zero energy length of the spring between u and v, $k_{uv}^{(1)}$ is the stiffness of the spring (the larger this value, the closer the spring should be to its ideal distance), $p_v=(x_v,y_v)$ is the position of node v, $k_{uv}^{(2)}$ gives the strength of repulsion between nodes u and v. (Battista, Eades, Tamassia, Tollis, 1999). Many variants on this energy equation are possible and have been tested. For instance, Davidson and Harel (1996) include additional terms to penalize edge crossings and other aesthetic criteria. Finding the minimum energy configuration of the graph falls to numerical algorithms.

Other graph drawing methods are also possible, the most noteable among these being hierarchical layouts which also use numerical techniques (as well as some clever top-down and left-right ordering algorithms) to arrange the graph into a downward flow (Battista et. al., 1999).

Link removal by significance testing

Unfortunately, even with the best numerical methods, there may be no way to neatly show very densely connected graphs. Euler proved that an I vertex planar graph (a graph with no edge crossings) must have fewer than or equal to $3I$-6 edges in total (Bondy and Murty, 1976). In practice the number of web affinities tends to approach the worst case of $O(I^2)$ very quickly, and so we are often faced with a graph that is "chronically ungraphable".

A further step is needed. In simple terms, we need to delete some links.

We will make the assumption that we mostly care about "interesting" links. If two pages are being clicked together at the random rate for two pages, we will say that is "not interesting" and simplify our graphing problem by just not showing the link. Our measure for interestingness is the number of times higher than random, that two pages are clicked together. This can be expressed with the following metric we call *Lift*:

$$Lift(a,b) = \frac{\Pr(a \wedge b)}{\Pr(a)\cdot\Pr(b)}$$

In this form, we can see that *Lift* divides the observed probability of *A* and *B*, by the expected random rate of occurrence between the two pages. Thus, *Lift* is equal to the number of times higher (or lower) than random, that the two pages are visited. *Lift* is closely related to the Chi-Square significance test (Pearson, 1900) and the Kullback-Leibler disparity (Kullback and Leibler, 1951) which also divide observed by expected.

We will use *Lift* by limiting the graph to only show the highest-scoring lift links. As a result, most of the random links will be removed from the graph entirely.

This strategy for finding significant links is remarkably effective. However, there is one remaining problem. As the number of observations decreases, the lift score becomes less reliable. Let's say that we have 1 million sessions, observed *a* and *b* once each, and they both occurred in the same session. The lift of this event is 1 million times higher than random. However, if *a* and *b* did not occur in that session the lift would drop to 0 - quite a swing based on a single observation.

In statistics, the number of observations and degrees of freedom are incorporated into a final score called a *p*-value, which is the probability of observing a value as or more extreme than the one observed, assuming that the null hypothesis is true (Pearson, 1900). However, when analyzing weblog data, we have typically found *p* values to be tiny because of the large number of observations. Therefore, we address the problem of reliability by adding a second parameter - the minimum number-of-observations for an edge or node to meet, in order for it to be included in the graph. If an edge fails to meet either the minimum observations or the lift criteria, it is removed.

Clickstream Analysis

We have now described a way for mining navigation sequences that is fast and scalable. We have also described a way for visualizing affinities using force-directed graphs, and using lift and observation thresholding to restrict graphs to only show statistically important links. We will now apply these techniques to mining web data from a variety of sources. In the following sections we describe four navigation analyses, and illustrate their use on commercial web sites.

Non-temporal Affinities

Non-temporal affinities *N* attempt to quantify the degree to which two pages tend to be accessed together in the same session. *N* can be defined as:

$$N(a,b) = \sum_{s} a,b \in X_s$$

We will also define a quantity we call click distance *C*. This measures how far on average customers have to travel from *a* to reach *b*, measured in clicks.

$$C(a,b) = {}^{1}\!/_{N(a,b)} \sum_{s} \left[\min \left| time(b) - time(a) \right| \right] : a,b \in X_s$$

Experiment 1: Web usage map with Non-temporal affinities

Our first experiment utilizes data from an unnamed company. The data comprises 719,532,990 page hits made over the course of a month by customers who opted into allowing their behavior to be observed as they browsed the world wide web. Analysis of this data required 36.5 hours on a 4 CPU, 500MHz Intel PC.

Using non-temporal affinities, we constructed a map of the resulting patterns of navigation across the world wide web. Figure 3 shows the graph, restricted to the highest 500 lift non-temporal page pairs, and pages having more than 100,000 hits per month. As the weak links drop away, a fascinating map of the web is produced.

WWW Map

We see that the web is partitioned into several regions with similar content. In the Western part of the map, there is a large group of connected pages orientated around News, Sports, AOL, Games, Stock quotes, and Search engines. In the Eastern area, we see a variety of isolated groups, including a Travel group, Music, Auto sales, French language, and Parenting/Kids group.

Figure 4 shows a close-up of the parenting area. The graph reveals that Baby center, burstware, Parent soup, and iVillage are strongly related. Elsewhere on Figure 3 these parenting areas are connected to women, gardening and kids sites. Such a grouping of interests may not have been apparent to content providers.

Figure 5 shows a close-up of a news area showing unusually high movement between the New York Times, Washington Post and Time.

Figure 6 shows another close-up revealing that customers often move between rival travel sites. For example, theTrip, Travelocity, PreviewTravel, iTravel, and Internet Travel Network are (incompletely) connected. This might suggest customers defecting between competitor travel sites online.

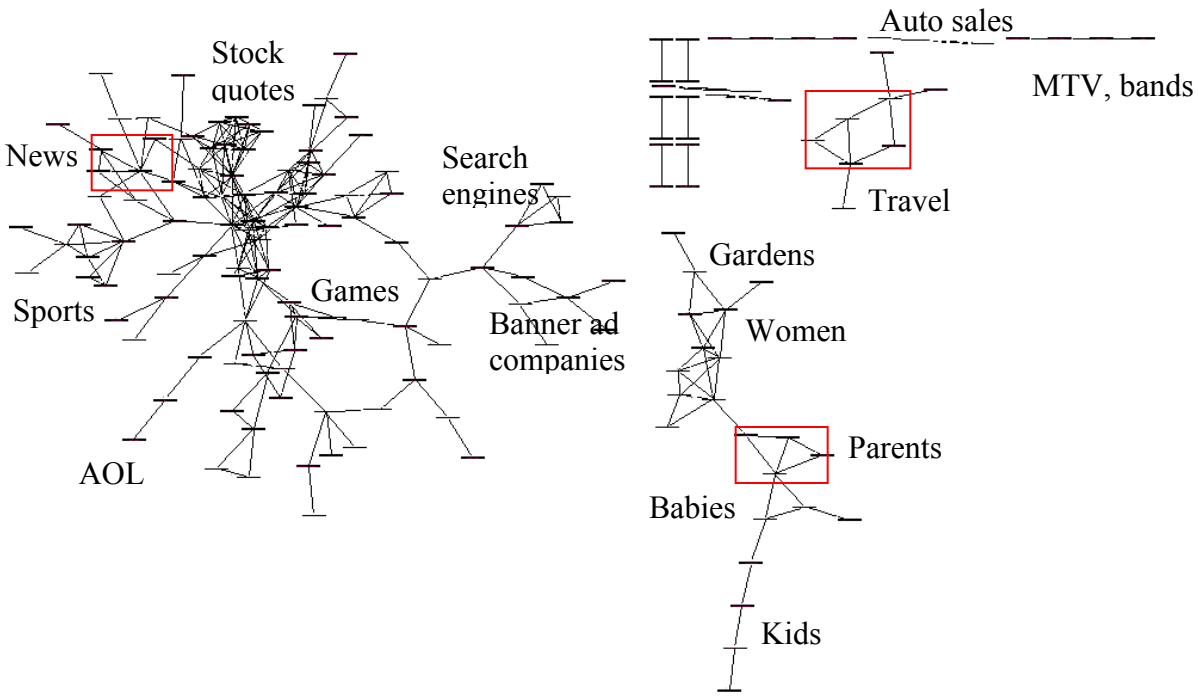

Figure 3: WWW Map optimized using force-directed methods and limited to the top 500 lift edges with greater than 100,000 hits per month. This shows how users move across the web.

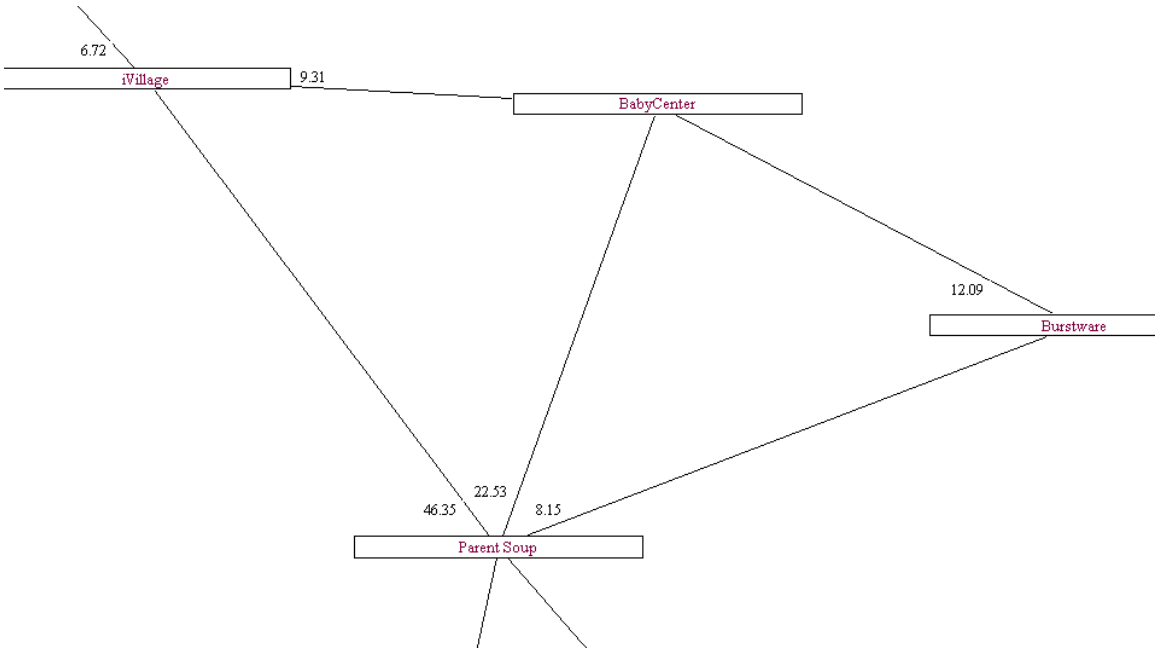

Figure 4: Parents region

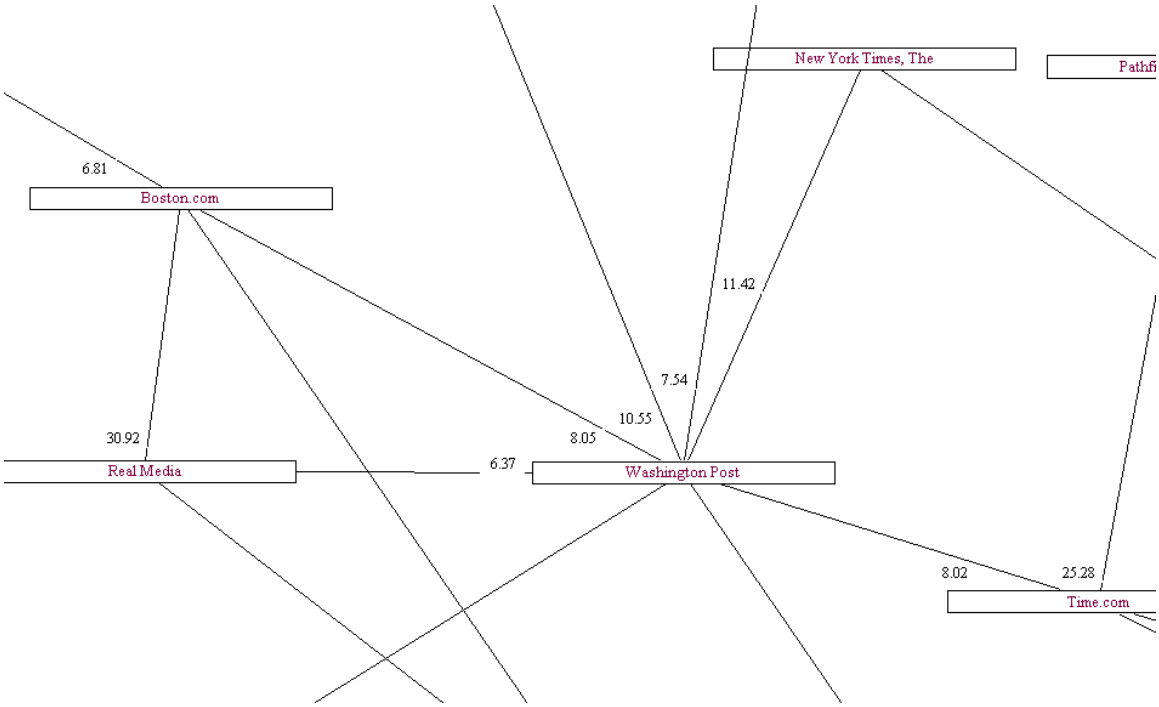

Figure 5: News region

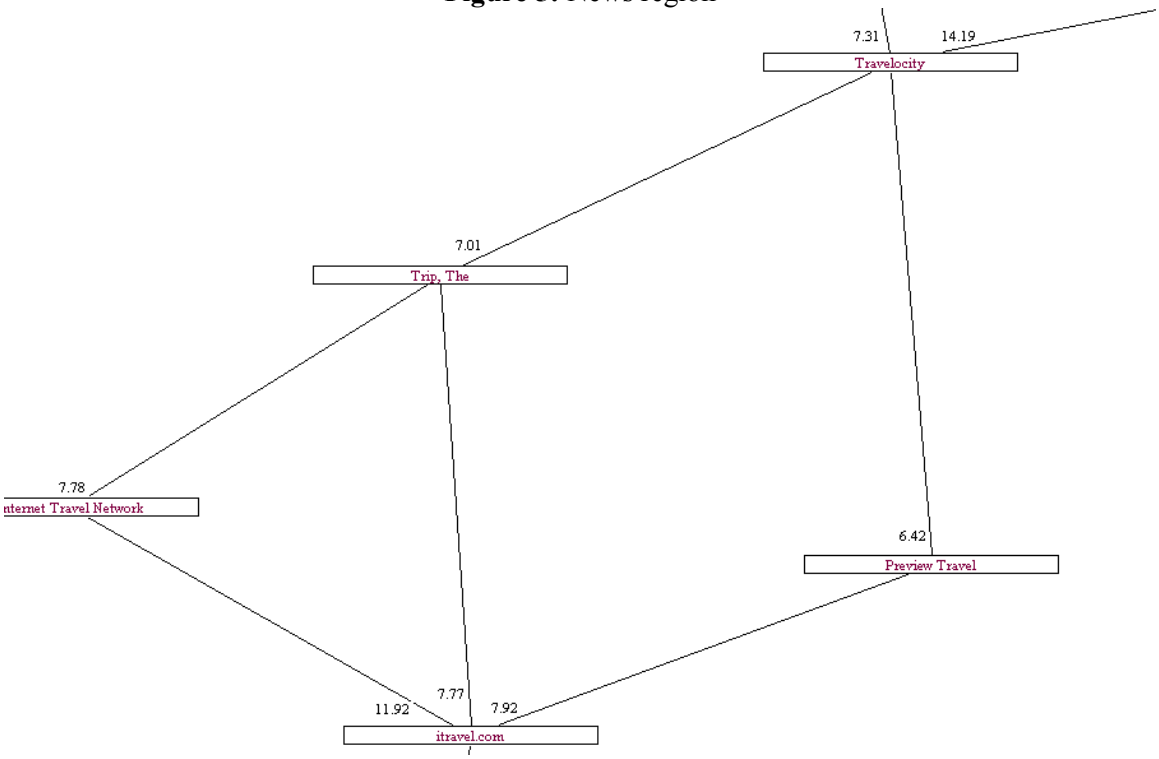

Figure 6: Travel region

Sequential affinity

A sequential affinity *S* exists between two pages if page *b* occurs exactly one click after page *a*. For example, if the Clearance Sale screen is visited 1000 times, and 80 of those customers proceed on their next click to click onto the Clearance Sale Handtools screen, then there is a sequential affinity between the two pages. A Sequential affinity $S(a,b)$ is defined as

$$S(a,b) = \sum_{s} a,b \in X_s : time(b) = time(a) + 1$$

Sequential affinities are useful for showing the page-by-page click behaviour of users as they navigate through the site. A sequential affinity is in fact a Markov model of site usage (Borges and Levene, 1999) where each edge is labeled with a probability of moving between the two. Sequential affinities can also infer the site's hyperlink structure, since edges are only shown if customers are able to click between them in one click.

Experiment 2: Retailing Site Sequential affinities

Experiment 2 uses data from a large and successful retailer of hand tools and home repair equipment. This retailer allowed us to analyze 1 week of web log data from October 1999, comprising 617,510 page hits. We will call this retailer Home Improvement Corp or "HIC".

Landing pages

The most likely page for a customer to "land" on can be found by looking at the arrival probabilities Pr(*a*|*arrivesite*). Figure 17 shows these pages. Most customers "land" on the index page (2.4%), followed by an item page (0.92%), followed by various other pages. The large number of customers arriving at item pages might seem unexpected, since site designers would think of these pages as being "furthest away" from the index page. However, HIC products are often advertised on third party sites or archived by site engines, and furthermore, HIC has more item pages than any other, resulting in the probability of a customer wandering onto HIC's site being more likely to happen through an item page.

This phenomenon means that basic navigation information needs to be displayed on item pages, to ensure newly landed customers can find their way to the main index page.

Killer pages

The departure probability Pr(*leavesite*|*a*) reveals what proportion of hits that the page received were exit hits. Pages with high departure probabilities will be provocatively called "killer pages", since they seem to bring user sessions to an end[1].

Figure 19 shows that the highest Departure probability pages at HIC are "catrequest form actionarg=2" and "payment order". The first is the screen where customers fill out their personal address to order a mail catalogue, in which case 44% of those customers depart on the next click. The second is the screen a customer reaches after completion of an order, in which 31% of customers depart on the next click. Customers reaching these pages have completed their transactions.

Other high Departure pages include the "cs.main", "cs.taf", and "retailstores" pages. The "cs" pages are customer service pages and bring up a telephone number for the customer to call if they need assistance. The "retailstores" page contains the locations of HIC's 80 retail stores. Customers who reach each of these pages defect heavily, most likely indicating that they are switching to HIC's telemarketing or retail store channels to complete their business.

Hotlinks is also a high defection area of the site. Because of this high rate of defection, perhaps site designers could alter the HIC site so that the new site launches in another window, so that the HIC session is at least retained. This might reduce the loss of customers who leave the site and can never find their way back.

Sticky pages

Pages with low Departure probabilities will be called "sticky" pages. Figure 19 shows that one of the stickiest pages is the Frequent Buyer page. The Frequent Buyer page is a page for "members" who pay $9.99 per year for special access to special deals on this page. Customers who visit the Frequent Buyer Club (FBC) have only a 3% chance of defection, compared to 7% for customers alighting from the index page. FBC is also associated with the longest browsing behavior of any page. If FBC is viewed, on average 18 additional clicks are made before a customer departs, compared to just 11.1 clicks after arriving at the site (Figure 18).

Site map

Figure 9 shows a hierarchical graph of the HIC site with $S(a,b) > 100$ and $Lift(a,b)>10$. Figure 10 shows the same data but with higher thresholds $S(a,b) > 200$ and $Lift(a,b)>5$, and force-directed layout. This reveals that the site is partitioned into roughly seven regions. We have labeled these Auction, Frequent Buyer Club, Free offers, Clearance sale pages, Inventory pages, Search pages, and Basket add/delete/order pages.

[1] Throughout this paper we will often loosely talk about X "causing" Y to occur. However, it should be understood that inferring true causation is usually impossible from off-line analysis. As a result, all interpretations are only hypotheses.

Search effectiveness

The three distinctive triangular recurrent processes on the Eastern side of the *Inventory hub* in Figure 10 are search processes (…for searching products, hence their connectivity to Inventory). There are three searches available on the HIC site, *displaycat*, *headsearch* and *displayitem*. The triangle is due to a recurrent link, meaning that the next site a customer visits after a search is often another search.

Figure 15 shows a screenshot of the main search page. Figure 16 shows the same page in numbers - with departure probabilities. It appears that customers using *displayitem* search – which involves typing in a 5 digit SKU number[2] - leave faster (8% probability), than customers who are browsing the *displaysubcats* entries (3% probability). This could be because customers typing in an SKU know what they are looking for and reach it more quickly. *displayitem* customers spend another 8 clicks on-site, where-as *headsearch* and *displaysubcat* customers spend over 11-12, indicating that they are more likely to be browsing (Figure 18).

Ordering process

The process of adding/deleting/ordering is shown in Figure 10-Figure 13. Figure 13 shows that after adding to their basket and electing to "checkout", customers are taken to a screen where they need to fill in their address for shipping called actionarg=1. There is a 54% chance of the customer filling in that information and moving to the actionarg=2 screen, which asks the user to enter their visa number and confirm the order. However, there is only an 11% chance of clicking "confirm order". Further, 5% of the time customer's will refresh on the confirm order screen (represented by a recurrent link).

Figure 13 also shows that after visiting help, there is a 29% chance of the customer adding something to their basket. This seems very high. We will examine the role of help later in our discussion on temporal affinities.

Temporal affinities

A temporal affinity *T* between two pages exists if page *a* occurs before page *b* during a customer session. For example, if 1000 customers visitited the product-specs page, and 40 clicked from there to the order page at some future time in their session, there would be a temporal affinity between product-specs and ordering. The definition is below.

$$T(a,b) = \sum_{s} a,b \in X_s : time(b) > time(a)$$

$$C(a,b) = {}^{1}\!\!\diagup\!\!{}_{T(a,b)} \sum_{s} \left[\min\left(time(b) - time(a)\right)\right] : a,b \in X_s \wedge time(b) > time(a)$$

[2] SKU or "Stock Keeping Unit" is a code used to uniquely identify products.

Temporal affinities are used to predict where customers go, after viewing a page. After visiting one page, they might only have a limited number of links. However, temporal affinities will reveal that customers will tend to visit another page that doesn't even have a link to it – perhaps 5 clicks in the future. *This shows where customers are going regardless of the link structure that they need to traverse*. In addition, *C* quantifies how many clicks it takes them to reach their destination.

Experiment 3: Retailing Site Temporal affinities

The role of help

We used sequential affinities to discover that most baskets were being abandoned at the confirm order screen. We also saw that customers who visited help, 29% of the time on their next click, added something to their basket.

Figure 14-top shows a view of the same ordering process, but with $Lift(a,b)>20$, temporal affinities, and force-directed layout. We learn that: If a customer clicks on help, their probability of completing their order jumps from 0.4% to 31% (Figure 22). Assuming the customer reaches help, order completion comes to fruition on average 10 clicks later. However customers are only able to find their way to help 5% of the time, after viewing their basket. (Figure 20, Figure 21). There are no help tabs on any of the major screens.

Effectiveness of free offers

Figure 14-bottom shows more detail with $Lift(a,b) > 10$. We now uncover two new relationships. Freeoffers appears to be good at causing completion of orders – and keeping customers on site. If a customer views a free offer they have a 6% chance of completing an order (compared to 0.4% chance when they first arrive), which occurs after 22 clicks – indicating a lot of browsing.

Real-estate analysis

The real-estate location *R* of a page is its traversal order in the clickstream of the average customer. For example, if the average customer first encounters the search page on their 19th click of their clickstream, the search page would have a real-estate location of 19.

$$R(a) = {}^{1}\!/_{Traffic(a)} \sum_{s} \min\left(time(a)\right) : a \in X_s$$

The *real estate* of HIC's site is shown in Figure 8. "Freeoffers" and "clearance sales" are usually viewed at 11 clicks. "Leavesite" occurs 14 clicks later, whilst "search" is viewed after 19 clicks. Thus, most customers leave before using the search engine.

How is knowing a page's real-estate location useful? We should expect pages that are buried in the site will receive few page hits – a customer is likely to leave before finding them. Pages that are landing sites or main pages tend to attract a lot of hits. Because of this relationship, the real-estate at different values of *R* has a different value to the site. Real-estate near landing pages is extremely valuable, where-as the real-estate of high numbers is not as valuable. We will now formalize this relationship as the hypothesis below.

Hypothesis 1: *As the real-estate location of a page increases, the Traffic that page receives will decrease.* ℘

Hypothesis 1 is supported by other authors such as Huberman, et. al. (1998) and Levene, Borges, et. al. (2001) who noted that the probability of a surfer remaining on site appears to decline with each additional click. The relationship between *Real-estate* and *Traffic* observed at HIC.com data is also consistent with hypothesis 1 (Figure 7). We will now show how this can be used for site design.

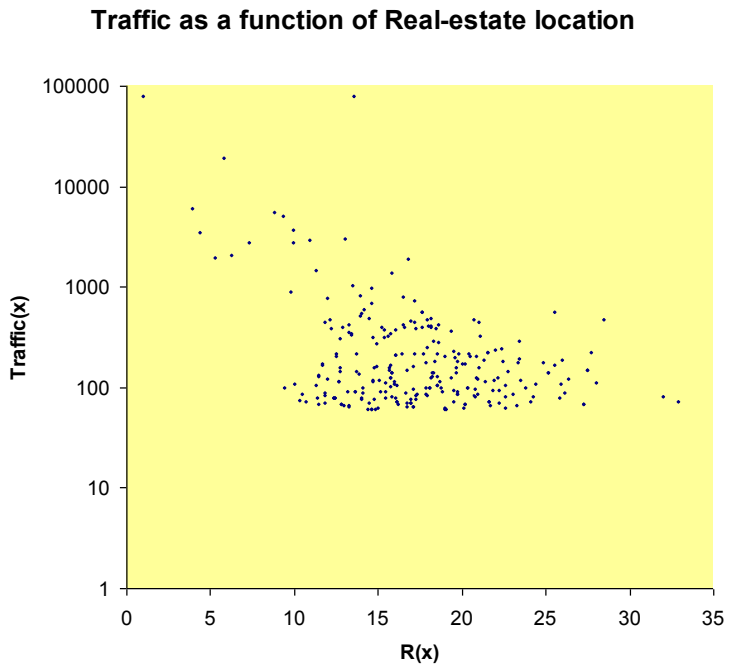

Figure 7: Scatterplot of *Real-estate location* of page versus *Traffic* of page at HIC.com

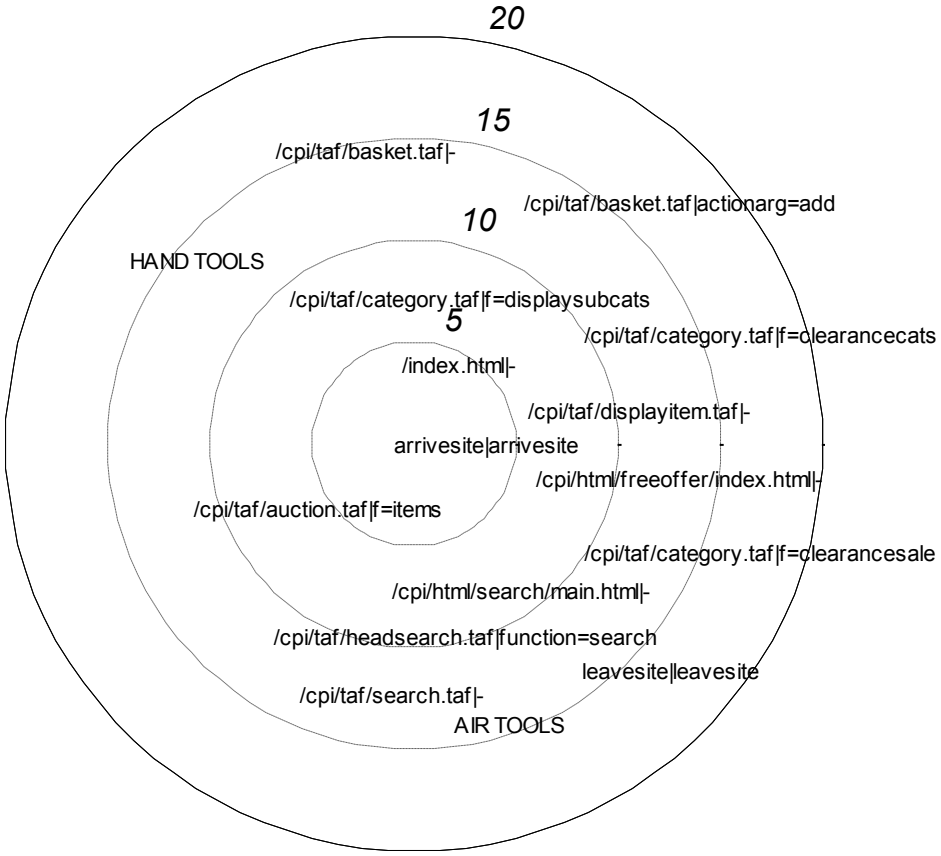

Figure 8: Real-estate locations for HIC.com (measured from center to bottom-left corner of each label). Concentric circles show the number of clicks it will take for the customer to reach the first occurrence of each page type above.

Using web-real estate to measure page performance

For every page on the site, let us measure *performance* as the actual pageviews a page receives, divided by the expected number of page views at this location, or the location's "real-estate value appraisal". The real-estate appraisal can be estimated in many ways; below we have chosen a thin-plate spline which is a close cousin of Radial Basis Function estimators (Moody and Darken, 1989), but uses a logarithmic basis function rather than a gaussian since this is parameter-less and performs well on log-scaled data.

$$Trafficpredicted\,(a) = \sum_{d=1}^{D} G\left(\left|r_d - R(a)\right|\right) \cdot t_d$$

$$G(a) = a^2 \log(a)$$

$$SSE = \sum_{a}\left[\left(\sum_{d=1}^{D} G\left(\left|r_d - R(a)\right|\right) \cdot t_d\right) - Traffic(a)\right]^2$$

where D are the number of prototypes used for the approximation, r_d are prototypical real-estate values and t_d are traffic values which are chosen to minimize the sum of squared errors *SSE*

Now that we have an expected number of hits, let performance be defined as follows:

$$performance(page) = \frac{Traffic(page)}{Trafficpredicted\,(page)}$$

We will now use this to improve site design. If we find bad pages that are occupying high-value land, it would be logical to swap them for better performers. Similarly, star-performers that are buried in the site should be elevated in the site hierarchy so they receive more exposure.

Experiment 4: Retail site real-estate

Figure 23 shows the result of this analysis. Frequent Buyer pages have excellent performance given their location. FBC pages are reached on average 15 clicks into a session, and should be receiving around 200 views. Instead they are receiving 400 views. We have shown elsewhere that FBC pages have very low probabilities of defection, and are correlated with unusually long browsing behaviour.

Reconed tools, outdoor products, and garden equipment all appear to be over-performers. Reconed tools are only reached after 17 clicks – they seem to be buried in the site - and yet are attracting 1.9 times the number of hits that other pages in this order bracket are. We believe these pages could be moved up the hierarchy, and could be displayed closer

to the main page, perhaps in the same way as second-hand and clearance sale items are displayed from the main page.

The worst performers are the auction screens. Auction login appears in pristine real-estate, only 8.9 clicks from the customer entry point. However, they are clicked only 75 times. Perhaps only a small number of customers actually have an auction account, and can access these pages.

Conclusion

We have presented a method for mining clickstreams from very large data sources, finding significant associations, and interactively displaying them. We have also presented applications of this work, including Sequential analysis (next click probabilities), Temporal analysis (where people are going) and Real-estate analysis (which pages are performing well/poorly). These methods enable us to understand how users are behaving, where they are bailing, and what factors seem to be leading them to purchase or sign up for further services.

There are many future directions in which to take this work. Spiliopoulou, Faulstich, and Winkler (1999) have developed a SQL-based query language to retrieve paths matching regular expression criteria. However, these authors have yet to develop an effective method for visualizing those paths. The methods described in this paper could be applied to full path queries.

Acknowledgements

Graphs were drawn with the aid of software by Tom Sawyer Inc. The authors would also like to thank the clients and Vignette for giving permission for the work to be published.

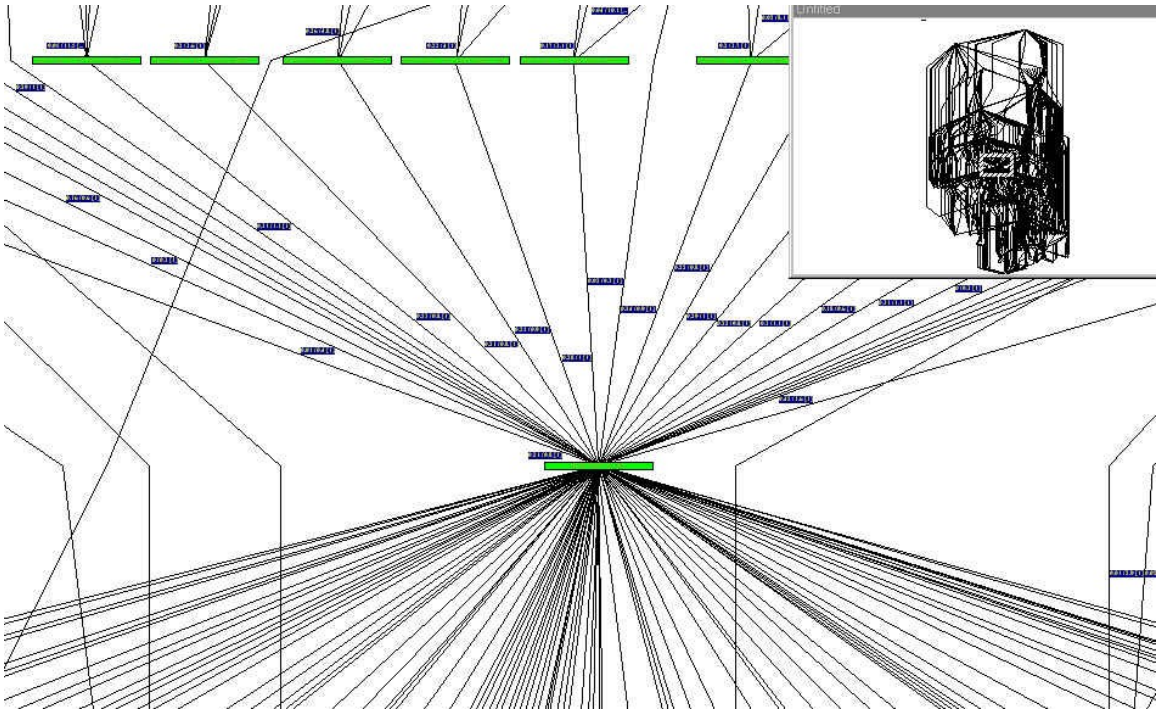

Figure 9: Hierarchical graph of sequential clickpaths at HIC.com, with all links traversed more than 100 times shown. The inset shows what the overall graph looks like. The very top of the graph is the "arrive site" node, and the bottom of the graph is the "leave site" node.

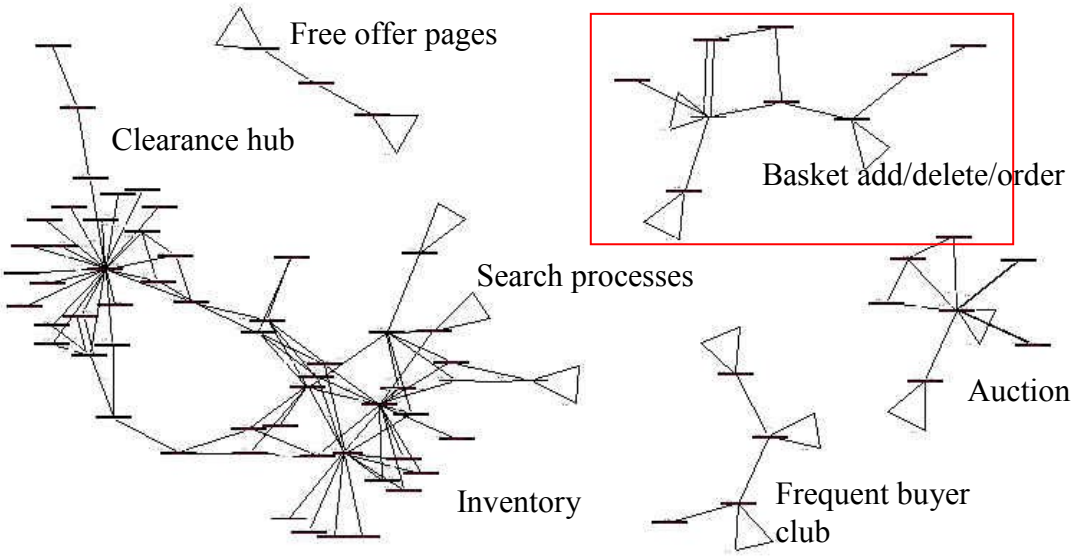

Figure 10: Overview of HIC.com sequential clicktrails using force-directed layout. Graph generated by showing only the strongest links, those with *Lift*(*a*,*b*)>5 and *S*(*a*,*b*)>200. Seven identifiable areas on the site can be seen.

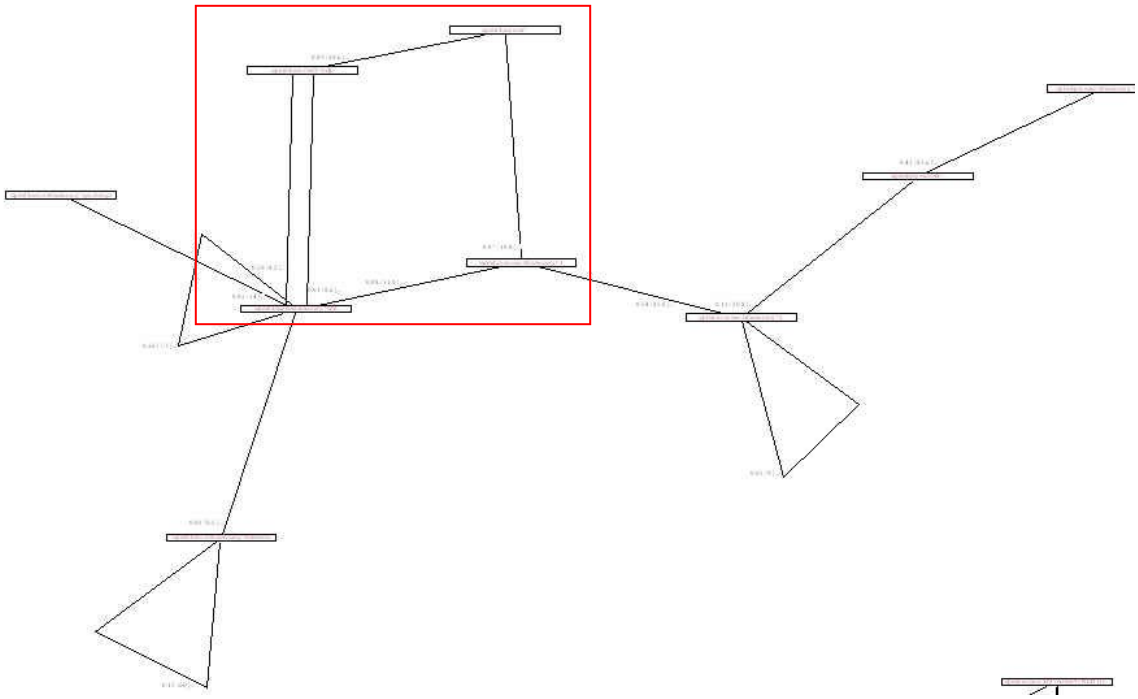

Figure 11: Basket process zoom.

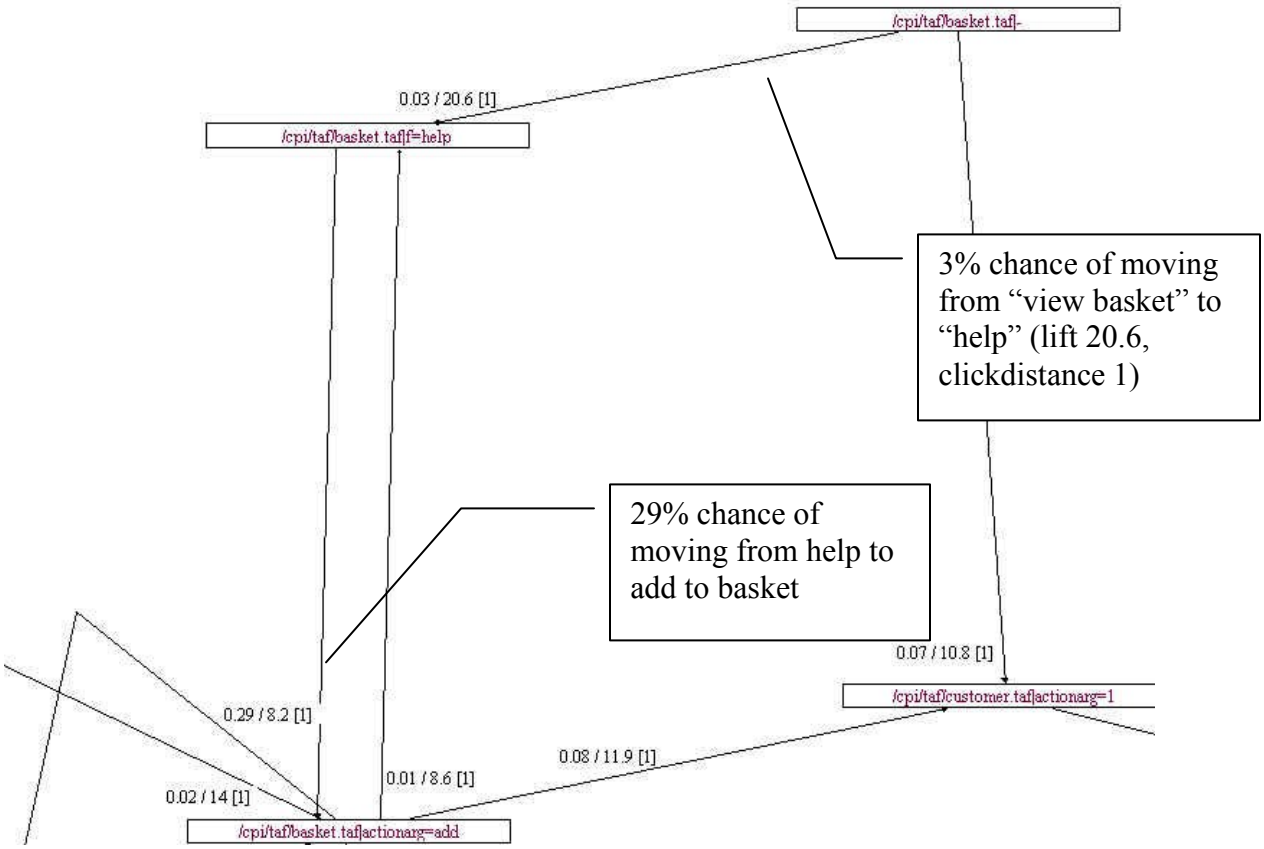

Figure 12: Basket process zoom x2. Edges are labeled with the following statistics: Pr($b|a$) / *Lift*(a,b) [C(a,b)]. This graph shows sequential affinities, which are movements from one page to another (and so all clickdistances equal 1). We see a clear sequential link between help and adding-to-basket. Temporal analysis will also indicate that viewing help results in a very high probability of order completion.

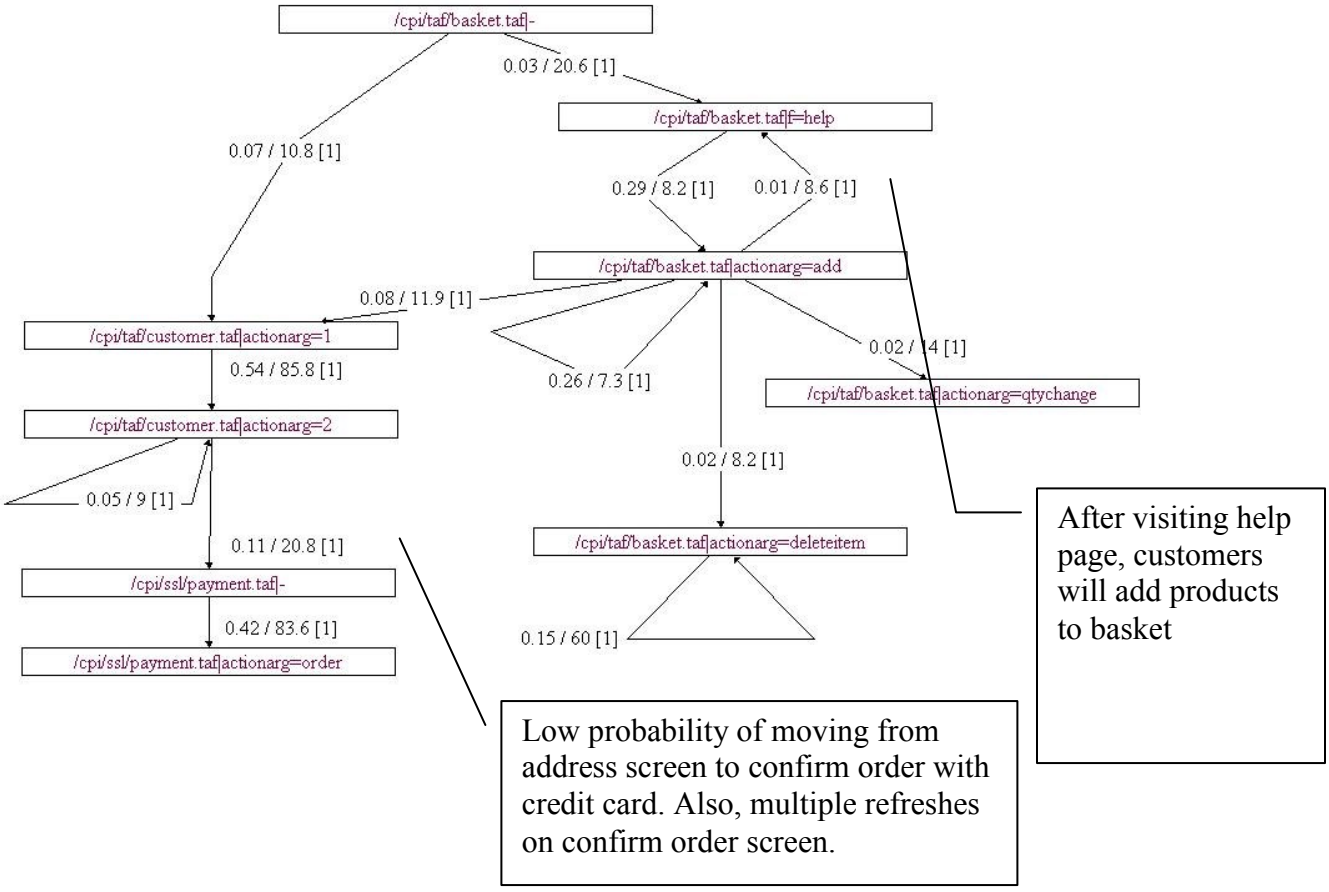

Figure 13: The same basket process using a hierarchical layout.

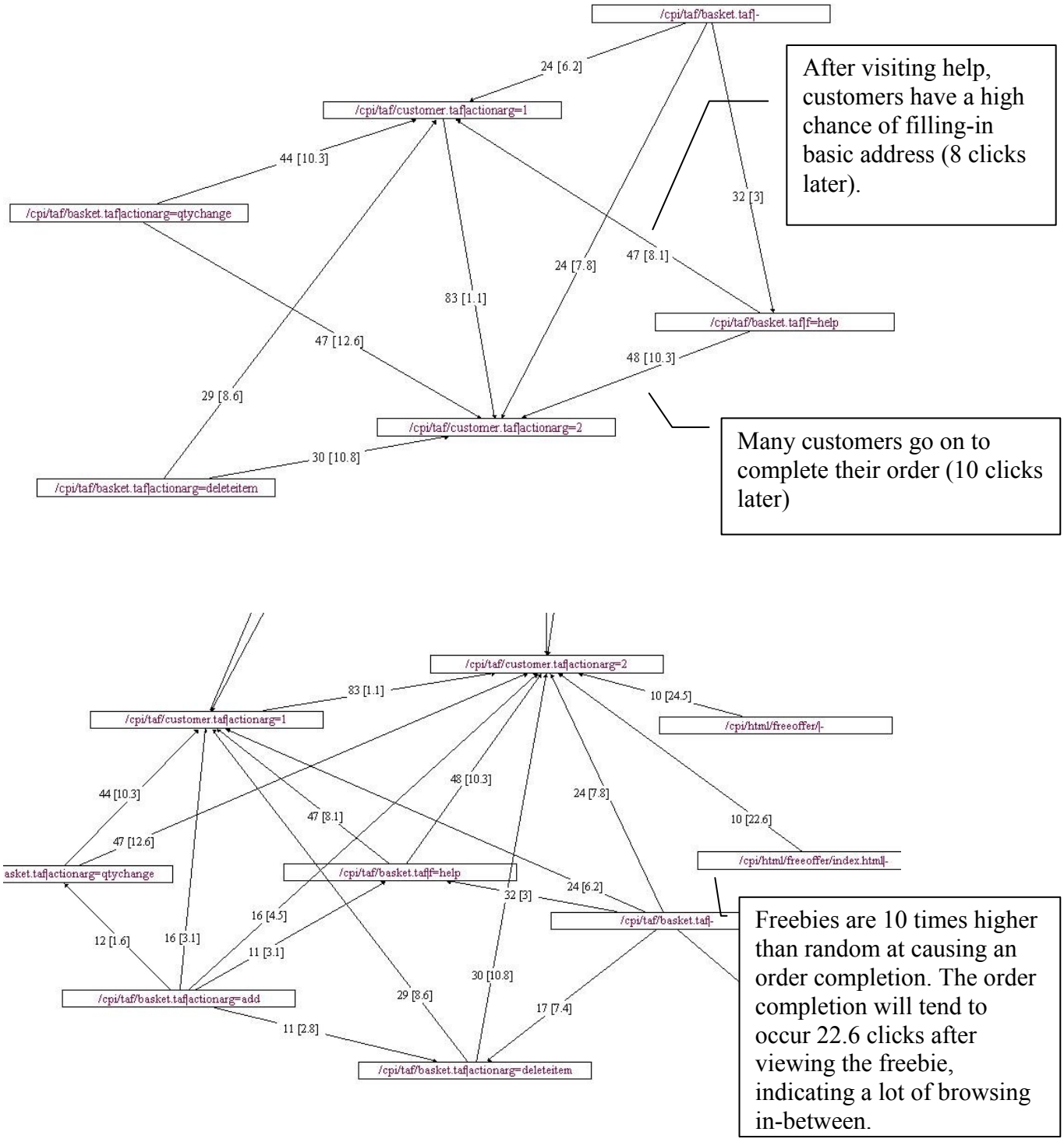

Figure 14: Basket process in terms of temporal affinities, displayed using force-directed layout, $T(a,b)>200$, with $Lift(a,b)>20$ (top graph) and $Lift(a,b)>10$ (bottom graph). The edge statistics in this case are $Lift(a,b)$ [$C(a,b)$]

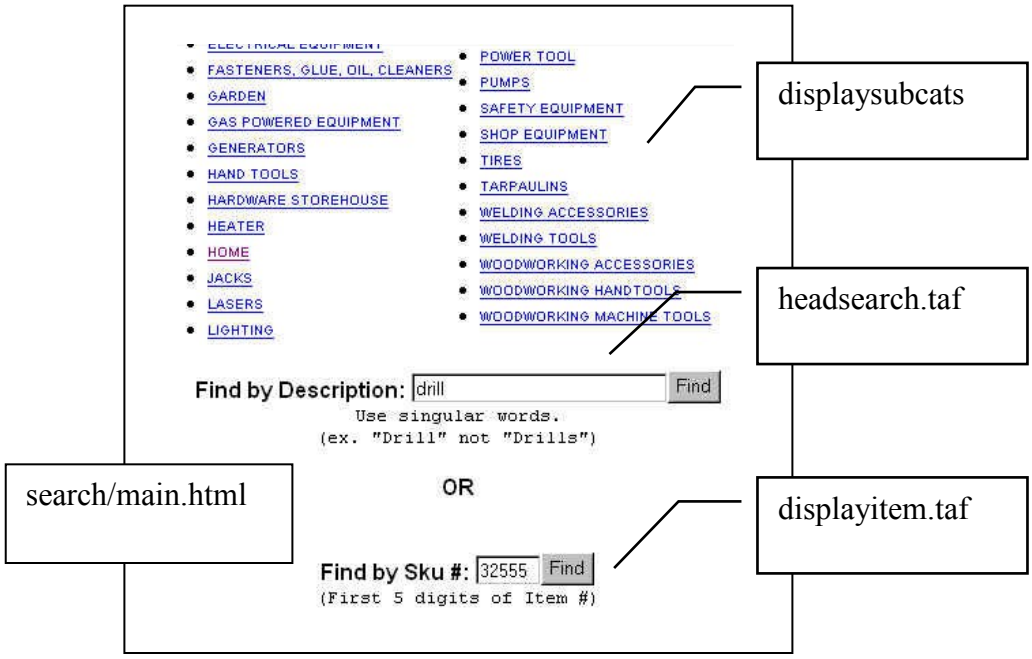

Figure 15: The main search screen at HIC.com.

Probability of lost customer given use of search method

PageA	PageB	SuppA	SuppB	SuppAB	P(B|A)	Description of search
/cpi/taf/displayitem.taf|function=search	leavesite	6959	617510	567	8.15%	Type in a 5 digit SKU number
/cpi/taf/headsearch.taf|function=search	leavesite	22305	617510	1574	7.06%	Type in a text description of a product
/cpi/taf/category.taf|f=displaysubcats	leavesite	34371	617510	1529	4.45%	Click on one of 20 subcategories in order to locate a product
/cpi/html/search/main.html|-	leavesite	21464	617510	248	1.16%	Main search screen, contains the three types of searches above

Figure 16: "Stickiness" for different search methods

Landing pages

Rank	PageA	PageB	Pr(B|A) (probability of arriving on page)
1	arrivesite	/index.html|-	2.44%
2	arrivesite	/cpi/taf/displayitem.taf|-	0.92%
3	arrivesite	/cpi/html/email1/index.html|-	0.53%
4	arrivesite	/cpi/t/h.taf|-	0.40%
5	arrivesite	/cpi/taf/category.taf|-	0.37%
6	arrivesite	/cpi/taf/category.taf|f=clearancesale	0.23%
7	arrivesite	/cpi/taf/category.taf|f=clearancecats	0.22%
8	arrivesite	/cpi/html/search/main.html|-	0.17%
9	arrivesite	/cpi/html/freeoffer/|-	0.14%
10	arrivesite	/cpi/taf/auction.taf|f=list	0.13%

Figure 17: Landing pages

Number of clicks before exiting site

PageA	PageB	SuppA	SuppB	P(B|A)	ClickAB
/cpi/taf/fbc.taf|f=logon	leavesite	1912	617510	100%	18.95751
/cpi/taf/category.taf|f=clearancesale|AIR COMP	leavesite	1787	617510	100%	17.70081
/cpi/taf/category.taf|f=salecats	leavesite	1580	617510	100%	17.685
/cpi/taf/category.taf|f=clearancesale|AIR TOOLS	leavesite	2816	617510	100%	17.67804
/cpi/html/newitems/index.html|-	leavesite	3130	617510	100%	16.06644
/cpi/html/greatoutdoors/index.html|-	leavesite	1297	617510	100%	15.71638
/cpi/photos/|-	leavesite	2678	617510	100%	15.53992
/cpi/html/woodshop/index.html|-	leavesite	2191	617510	100%	15.45828
/cpi/html/bestof/index.html|-	leavesite	3184	617510	100%	15.33431
/cpi/html/powertools/index.html|-	leavesite	2898	617510	100%	15.21241
/cpi/html/autoshop/index.html|-	leavesite	2339	617510	100%	14.90323
/cpi/html/grabbag/index.html|-	leavesite	2566	617510	100%	14.79087
/cpi/html/homeelec/index.html|-	leavesite	1343	617510	100%	14.57127
/cpi/taf/category.taf|-|RECONED	leavesite	4938	617510	100%	14.50833
/cpi/html/reconcntr/index.html|-	leavesite	4087	617510	100%	14.36224
/cpi/taf/search.taf|-	leavesite	9478	617510	100%	14.2889
/cpi/taf/auction.taf|f=login	leavesite	1008	617510	100%	13.55517
/cpi/html/freeoffer/|-	leavesite	4831	617510	100%	12.01344
/cpi/html/search/main.html|-	leavesite	21464	617510	100%	11.95002
/cpi/taf/displayitem.taf|function=search	leavesite	6959	617510	100%	11.24634
arrivesite|arrivesite	leavesite	617510	617510	100%	11.19845
/cpi/taf/cs.taf|-	leavesite	1306	617510	100%	10.66705
/cpi/taf/auction.taf|f=bid	leavesite	1705	617510	100%	9.142857
/cpi/taf/headsearch.taf|function=search	leavesite	22305	617510	100%	8.802101
/cpi/ssl/payment.taf|-	leavesite	3546	617510	100%	8.108245
/cpi/taf/catreq.taf|f=catform	leavesite	6449	617510	100%	7.248707
/cpi/ssl/payment.taf|actionarg=order	leavesite	3173	617510	100%	7.200871
/cpi/taf/catreq.taf|actionarg=2	leavesite	5536	617510	100%	5.037371

Figure 18: Tenporal affinities for Pr(*leavesite*|*x*). Frequent Buyer Club pages are the stickiest pages at HIC. After logging on, customers will on average spend 18 clicks before leaving. The greatoutdoors and auction site are also high performers in terms of stickiness. On the other hand, customers who fill out a catalogue request form are generally finished with the site 7 clicks later, and customers who complete orders tend to be done with the site 5 clicks later. Customers leaving after order completion are perhaps not surprising. But why is the great outdoors site doing so well? Perhaps users would be interested in more such content.

Killer pages / Sticky pages

PageA	PageB	SuppA	SuppB	SuppAB	P(B|A)
/cpi/taf/catreq.taf|actionarg=2	leavesite	5536	617510	2415	43.62%
/cpi/ssl/payment.taf|actionarg=order	leavesite	3173	617510	968	30.51%
/cpi/taf/cs.taf|-	leavesite	1306	617510	331	25.34%
/cpi/taf/hotlinks.taf|-	leavesite	2674	617510	621	23.22%
/cpi/taf/customer.taf|actionarg=2	leavesite	3901	617510	897	22.99%
/cpi/retailstores/index.html|-	leavesite	4400	617510	693	15.75%
/cpi/taf/catreq.taf|f=catform	leavesite	6449	617510	840	13.03%
/cpi/html/freeoffer/index.html|-	leavesite	5262	617510	633	12.03%
/cpi/taf/fbc.taf|-	leavesite	2719	617510	265	9.75%
/cpi/taf/displayitem.taf|function=search	leavesite	6959	617510	567	8.15%
/cpi/taf/displayitem.taf|-	leavesite	166634	617510	13063	7.84%
/index.html|-	leavesite	47798	617510	3543	7.41%
/cpi/taf/basket.taf|-	leavesite	8054	617510	572	7.10%
/cpi/taf/headsearch.taf|function=search	leavesite	22305	617510	1574	7.06%
/cpi/taf/category.taf|-|RECONED	leavesite	4938	617510	267	5.41%
/cpi/taf/category.taf|f=clearancesale	leavesite	33837	617510	1676	4.95%
/cpi/html/reconcntr/index.html|-	leavesite	4087	617510	200	4.89%
SHOP EQP	leavesite	6134	617510	281	4.58%
AIR TOOLS	leavesite	5996	617510	274	4.57%
/cpi/taf/auction.taf|f=items	leavesite	15079	617510	681	4.52%
/cpi/taf/category.taf|f=displaysubcats	leavesite	34371	617510	1529	4.45%
/cpi/taf/category.taf|-	leavesite	49378	617510	1766	3.58%
/cpi/taf/fbc.taf|f=list	leavesite	6143	617510	204	3.32%
/cpi/taf/basket.taf|actionarg=add	leavesite	21845	617510	273	1.25%
/cpi/html/search/main.html|-	leavesite	21464	617510	248	1.16%

Figure 19: Sequential affinities for leaving the site. Again we see that the Frequent Buyer Club pages have the distinction of being amongst the stickiest pages on HIC (3rd from the bottom), with only a 3% chance of a customer on those pages leaving on their next click. The main search screen has the lowest probability of departure, meaning customers tend to proceed on to actually perform a search. Catalog request and payment screens have the least sticky footprint, with 43% and 30% chances of customers leaving immediately after visiting each page.

Probability of reaching help on next click

PageA	PageB	SuppA	suppB	SuppAB	Pr(B|A)	Lift
/cpi/taf/basket.taf|-	/cpi/taf/basket.taf|f=help	8054	1006	271	3.36%	20.65398
/cpi/taf/basket.taf|actionarg=add	/cpi/taf/basket.taf|f=help	21845	1006	308	1.41%	8.654554

Figure 20: Probability of reaching help on next click is very low. This could be because links to help are not shown on any of the pages.

Probability of reaching help at some point in the future

PageA	PageB	SuppA	SuppB	SuppAB	Pr(B|A)	Lift	Click
/cpi/taf/basket.taf|-	/cpi/taf/basket.taf|f=help	8054	1006	423	5.25%	32.23849	3.002364
/cpi/taf/basket.taf|actionarg=add	/cpi/taf/basket.taf|f=help	21845	1006	402	1.84%	11.29588	3.144279
/cpi/html/search/main.html|-	/cpi/taf/basket.taf|f=help	21464	1006	236	1.10%	6.749123	15.05508
/cpi/taf/category.taf|f=displaysubcats	/cpi/taf/basket.taf|f=help	34371	1006	216	0.63%	3.857515	11.44907
/cpi/taf/category.taf|f=clearancesale	/cpi/taf/basket.taf|f=help	33837	1006	209	0.62%	3.791407	11.67943
/index.html|-	/cpi/taf/basket.taf|f=help	47798	1006	267	0.56%	3.428843	11.94007
/cpi/taf/category.taf|-	/cpi/taf/basket.taf|f=help	49378	1006	228	0.46%	2.83431	10.01316
/cpi/taf/displayitem.taf|-	/cpi/taf/basket.taf|f=help	166634	1006	362	0.22%	1.333494	5.569061
arrivesite|arrivesite	/cpi/taf/basket.taf|f=help	617510	1006	637	0.10%	0.633201	17.36892

Figure 21: Probability of reaching help is very low. Compare with figure 15 for the probability of reaching help on next click (sequential affinity).

Probability of page view resulting in order completion at some point in the session

PageA	PageB	SuppA	SuppB	SuppAB	P(B|A)	Lift	ClickAB
/cpi/taf/customer.taf|actionarg=1	/cpi/taf/customer.taf|actionarg=2	4344	3901	2298	52.90%	83.7391	1.16232
/cpi/taf/basket.taf|f=help	/cpi/taf/customer.taf|actionarg=2	1006	3901	307	30.52%	48.3068	10.3518
/cpi/taf/basket.taf|actionarg=qtychange	/cpi/taf/customer.taf|actionarg=2	1051	3901	317	30.16%	47.7446	12.6498
/cpi/taf/basket.taf|actionarg=deleteitem	/cpi/taf/customer.taf|actionarg=2	1584	3901	306	19.32%	30.5798	10.8595
/cpi/taf/basket.taf|-	/cpi/taf/customer.taf|actionarg=2	8054	3901	1261	15.66%	24.784	7.89056
/cpi/taf/basket.taf|actionarg=add	/cpi/taf/customer.taf|actionarg=2	21845	3901	2277	10.42%	16.4998	4.59157
/cpi/taf/customer.taf|actionarg=2	/cpi/taf/customer.taf|actionarg=2	3901	3901	376	9.64%	15.2574	7.60106
/cpi/taf/category.taf|f=clearancesale|HAND TOOLS	/cpi/taf/customer.taf|actionarg=2	3312	3901	281	8.48%	13.4302	23.4057
POWER ACCY	/cpi/taf/customer.taf|actionarg=2	3980	3901	337	8.47%	13.4034	21.3976
HOME PROD	/cpi/taf/customer.taf|actionarg=2	3000	3901	216	7.20%	11.3973	23.3056
HAND TOOLS	/cpi/taf/customer.taf|actionarg=2	6295	3901	448	7.12%	11.2655	21.7679
AUTOMOTIVE	/cpi/taf/customer.taf|actionarg=2	3305	3901	230	6.96%	11.016	23.3174
OUTDOOR PROD	/cpi/taf/customer.taf|actionarg=2	3372	3901	231	6.85%	10.8441	23.4589
WOOD ACCYS	/cpi/taf/customer.taf|actionarg=2	4197	3901	279	6.65%	10.5228	22.2115
GARDEN PROD	/cpi/taf/customer.taf|actionarg=2	3142	3901	208	6.62%	10.4791	22.5144

/cpi/taf/category.taf|f=clearancesale|SHOP EQP	/cpi/taf/customer.taf|actionarg=2	3730	3901	242	6.49%	10.2701	24.1983
/cpi/taf/email.taf|f=insert	/cpi/taf/customer.taf|actionarg=2	3417	3901	221	6.47%	10.238	22.3846
/cpi/html/freeoffer/|-	/cpi/taf/customer.taf|actionarg=2	4831	3901	307	6.35%	10.0593	24.5537
/cpi/html/freeoffer/index.html|-	/cpi/taf/customer.taf|actionarg=2	5262	3901	334	6.35%	10.0476	22.6467
ELEC EQUIP	/cpi/taf/customer.taf|actionarg=2	3381	3901	211	6.24%	9.87883	24.1185
/cpi/taf/manuals.taf|f=form	/cpi/taf/customer.taf|actionarg=2	3436	3901	210	6.11%	9.67463	21.7286
/cpi/taf/search.taf|-	/cpi/taf/customer.taf|actionarg=2	9478	3901	574	6.06%	9.58657	18.7561
/cpi/html/search/main.html|-	/cpi/taf/customer.taf|actionarg=2	21464	3901	1293	6.02%	9.53577	19.604
METAL ACCY	/cpi/taf/customer.taf|actionarg=2	3406	3901	204	5.99%	9.48099	23.0392
AIR TOOLS	/cpi/taf/customer.taf|actionarg=2	5996	3901	354	5.90%	9.34565	22.6186
SHOP EQP	/cpi/taf/customer.taf|actionarg=2	6134	3901	360	5.87%	9.29024	23.1694
/cpi/taf/displayitem.taf|function=search	/cpi/taf/customer.taf|actionarg=2	6959	3901	402	5.78%	9.14423	12.1169
POWER TOOL	/cpi/taf/customer.taf|actionarg=2	3837	3901	221	5.76%	9.11735	23.3213
CRDLS TOOLS	/cpi/taf/customer.taf|actionarg=2	5415	3901	297	5.48%	8.68212	22.8283
RECONED	/cpi/taf/customer.taf|actionarg=2	4941	3901	252	5.10%	8.07335	22.4524
/cpi/taf/category.taf|-|RECONED	/cpi/taf/customer.taf|actionarg=2	4938	3901	250	5.06%	8.01414	22.576
/cpi/taf/category.taf|f=clearancecats	/cpi/taf/customer.taf|actionarg=2	20935	3901	805	3.85%	6.08683	20.0149
/cpi/taf/headsearch.taf|function=search	/cpi/taf/customer.taf|actionarg=2	22305	3901	783	3.51%	5.55684	13.7458
/cpi/t/h.taf|-	/cpi/taf/customer.taf|actionarg=2	8343	3901	287	3.44%	5.44537	22.784
/cpi/taf/category.taf|f=displaysubcats	/cpi/taf/customer.taf|actionarg=2	34371	3901	1111	3.23%	5.1167	13.0432
/cpi/taf/category.taf|function=list	/cpi/taf/customer.taf|actionarg=2	17734	3901	550	3.10%	4.90935	16.2655
/cpi/taf/category.taf|f=clearancesale	/cpi/taf/customer.taf|actionarg=2	33837	3901	809	2.39%	3.78464	17.1916
/cpi/taf/category.taf|-	/cpi/taf/customer.taf|actionarg=2	49378	3901	1145	2.32%	3.67063	10.7921
/index.html|-	/cpi/taf/customer.taf|actionarg=2	47798	3901	976	2.04%	3.23227	20.5912
/cpi/taf/displayitem.taf|-	/cpi/taf/customer.taf|actionarg=2	166634	3901	1774	1.06%	1.68523	5.86246
arrivesite|arrivesite	/cpi/taf/customer.taf|actionarg=2	617510	3901	2323	0.38%	0.59549	26.1128

Figure 22: Probability of pageview resulting in a purchase at some future point in the session

Over-performers given Real-Estate address (candidates for elevation in site hierarchy)

Rank	Page	hits	Real-estate location	hits/E[hits]
1	/cpi/taf/basket.taf|actionarg=add	2990	12.99775	9.603032
2	/cpi/taf/headsearch.taf|function=search	2898	10.88959	9.307554
3	/cpi/taf/displayitem.taf|function=search	969	14.63426	4.437626
4	/cpi/taf/customer.taf|actionarg=1	559	25.50502	3.845625
5	/cpi/taf/displayitem.taf|-	19496	5.810787	3.500004
6	HAND TOOLS	805	16.51225	3.363135
7	/cpi/taf/fbc.taf|f=list	685	14.6	3.137021
8	/cpi/taf/customer.taf|actionarg=2	472	28.43243	3.03212
9	POWER ACCY	472	20.71705	2.844744
10	HOME PROD	369	19.35498	2.717231
11	/cpi/ssl/payment.taf|actionarg=order	446	20.96667	2.688042
12	/cpi/taf/catreq.taf|f=catform	775	11.95198	2.48908
13	/cpi/ssl/payment.taf|-	477	17.93048	2.121131
14	/cpi/taf/category.taf|-|RECONED	564	17.59316	1.875748
15	RECONED	564	17.59316	1.875748
16	/cpi/taf/fbc.taf|-	404	15.19626	1.850156
17	OUTDOOR PROD	413	17.99197	1.836535
18	/cpi/taf/hotlinks.taf|-	413	18.11429	1.836535
19	GARDEN PROD	390	18.41204	1.734258

Underperformers given Real-Estate address (candidates for demotion in hierarchy)

rank	page	hits	Real-estate location	hits/E[hits]
1	/cpi/taf/auction.taf|f=loginform	75	10.28	0.013464
2	/cpi/taf/auction.taf|f=bid	100	9.395349	0.017952
3	/cpi/taf/auction.taf|f=login	108	9.978261	0.019389
4	/cpi/html/aboutus/main.html|-	90	13.18	0.025914
5	/cpi/taf/fbc.taf|f=logon	216	13.72727	0.062194
6	/cpi/html/grabbag/index.html|-	334	13.42268	0.09617
7	/cpi/html/newitems/index.html|-	422	13.232	0.121509
8	/cpi/retailstores/index.html|-	514	13.95286	0.147999
9	/cpi/html/reconcntr/index.html|-	544	13.98754	0.156637
10	/cpi/taf/auction.taf|f=list	897	9.77957	0.161033
11	/cpi/html/associate/home.taf|-	66	12.97059	0.211973
12	/cpi/taf/auction.taf|f=mybids	69	11.47826	0.221608
13	/cpi/html/paintsupp/index.html|-	85	17.33333	0.282693
14	/cpi/taf/search.taf|-	1028	13.50424	0.295998

Figure 23: Over-performers and under-performers for HIC

References

Agrawal, R., Mannila, H., Srikant, R. et. Al. (1996), Fast Discovery of Association rules, in Fayyad, U. et. Al. (eds) *Advances in Knowledge Discovery and Data Mining*, pp. 307-328. AAAI Press, Menlo Park, CA.

Battista, G., Eades, P., Tamassia, R. and Tolis, I. (1999) *Graph Drawing: Algorithms for the visualization of graphs*, Prentice Hall, NJ.

Bondy, J. and Murty, U. (1976), *Graph Theory with Applications*, Macmillan, London.

Borges, J. and Levene, M. (1999) Data Mining of User Navigation Patterns, *Workshop on Web Usage Analysis and User Profiling, Fifth ACM SIGKDD International Conference on Knowedge Discovery and Data Mining*, August 15, San Diego

Buchner, A., Baumgarten, M., Anand, S. et. al. (1999) Navigation pattern discovery from internet data, *Workshop on Web Usage Analysis and User Profiling, Fifth ACM SIGKDD International Conference on Knowedge Discovery and Data Mining*, San Diego, CA

Catledge, L. and Pitkow, J. (1995), Characterizing browsing strategies in the World Wide Web, *Computer Networks and ISDN Systems*, Vol. 26, No. 6, pp. 1065-1073.

Davidson, R. and Harel, D. (1996) Drawing Graphics Nicely using Simulated Annealing, *ACM Transactions on Graphics*, Vol. 15, No. 4, pp. 301-331.

Huberman, B., Pirolli, P., Pitkow, J. and Lukose, R. (1998), Strong Regularities in World Wide Web Surfing, *Science*, Vol. 280, No. 3, April.

Kamada, T. and Kawai, S. (1989) An Algorithm for Drawing General Undirected Graphs, *Information Processing Letters*, Vol. 31, pp. 7-15.

Kullback, S. and Leibler, R. (1951). On information and sufficiency, *Annals of Mathematical Statistics*, Vol. 22, pp. 79-86.

Levene, M., Borges, J. and Loizou, G. (2001), Zipf's law for web surfers, *Knowledge and Information Systems an International Journal*, Vol. 3, No. 1.

Levene, M. and Loizou, G. (1999), A probabilistic approach to navigation in hypertext, *Information Sciences*, Vol. 114, pp. 165-186.

Mannila, H., Toivonen, H. and Verkamo, I. (1995), Discovering frequent episodes in sequences, *First International Conference on Knowledge Discovery and Data Mining*. AAAI Press, Montreal, Canada.

Moody, J. and Darken, C. (1989), Fast learning in networks of locally tuned processing units, *Neural Computation*, Vol. 1, pp. 281-294.

Mukherjea, S. and Foley, J.D. (1995), Visualizing the World-Wide Web with the Navigational View Builder, *Computer Networks and ISDN Systems*, Special Issue on the Third International Conference on the World Wide Web, Darmstadt, Germany. http://www.igd.fhg.de/www /www95/proceedings/papers/44/mukh/mukh.html

Pearson, K. (1900), On the Criterion that a Given System of Deviations from the Probable in the Case of a Correlated System of Variables is Such that it Can reasonably supposed to have arisen from random sampling, *Philosophical Transactions of the Royal Society of London*, Vol. 50, pp. 157-175.

Pitkow, J. and Bharat, K. (1994), WebViz: A Tool for WWW Access Log Analysis, Graphics, Visualization and Usability Center, College of Computing, Georgia Institute of Technology, Atlanta, GA.

Quinn, N. and Breuer, M. (1979), A force directed component placement procedure for printedcirucuit boards, *IEEE Transaction on Circuits and Systems*, CAS 26, no. 6, pp. 377-388.

Spiliopoulou, Faulstich, and Winkler (1999), A Data Miner analyzing the Navigational Behaviour of Web users, Technical Report, Institut fur Wirtschaftsinformatik, Humboldt-Universitat zu Berlin.

Spiliopoulou, M. and Faulstich, L. (1999), WUM: A Tool for Web Utilization Analysis, *Proceedings of the EDBT Workshop*, pp. 184-203. Berlin, Springer.

Tutte, W. (1963), How to Draw a graph, *Proceedings of the London Mathemtical Society*, Vol. 13, No. 3, pp. 743-768.

Underhill, P. (2000), *Why we buy: The science of shopping*, Touchstone Books.

Webb, G. (2000), Efficient search for association rules, *Proceedings of the Sixth International Conference on Knowledge Discovery and Data Mining*, pp. 99-107, Association for Computiong Machinery, Boston, MA.

Wu, K., Yu, P. and Ballman, A. (2001), SpeedTracer: A Web usage mining and analysis tool, *IBM Systems Journal*, Vol. 37, no. 1.
http://www.research.ibm.com/journal/sj/371/wu.html

Zaki, M. (2000), SPADE: An efficient algorithm for mining frequent sequences, *Machine Learning Journal*, Vol. 42.