World Wide Web 2 (1999) 29–45

Distributions of surfers' paths through the World Wide Web: Empirical characterizations

Peter L.T. Pirolli and James E. Pitkow

Xerox Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, CA 94304, USA
E-mail: pirolli@parc.xerox.com

Surfing the World Wide Web (WWW) involves traversing hyperlink connections among documents. The ability to predict surfing patterns could solve many problems facing producers and consumers of WWW content. We analyzed WWW server logs for a WWW site, collected over ten days, to compare different path reconstruction methods and to investigate how past surfing behavior predicts future surfing choices. Since log files do not explicitly contain user paths, various methods have evolved to reconstruct user paths. Session times, number of clicks per visit, and Levenshtein Distance analyses were performed to show the impact of various reconstruction methods. Different methods for measuring surfing patterns were also compared. Markov model approximations were used to model the probability of users choosing links conditional on past surfing paths. Information-theoretic (entropy) measurements suggest that information is gained by using longer paths to estimate the conditional probability of link choice given surf path. The improvements diminish, however, as one increases the length of path beyond one. Information-theoretic (total divergence to the average entropy) measurements suggest that the conditional probabilities of link choice given surf path are more stable over time for shorter paths than longer paths. Direct examination of the accuracy of the conditional probability models in predicting test data also suggests that shorter paths yield more stable models and can be estimated reliably with less data than longer paths.

1. Introduction

Surfing the World Wide Web (WWW) involves traversing the connections among hyperlinked documents. It is one of the most common ways of accessing WWW content. Theories and models are beginning to explain how observed patterns of surfing behavior emerge from fundamental human information foraging processes [Huberman *et al.* 1998; Pirolli and Card in press]. The ability to predict surfing patterns could be instrumental in solving many problems facing producers and consumers of WWW content. For instance, Web site designs could be evaluated and optimized by predicting how users will surf through their structures. WWW client and server applications could reduce user perceived network latency by pre-fetching content predicted to be on the surfing path of individual users or groups of like-minded users. Systems and user interfaces could be enhanced by the ability to recommend content of interest to users, or by displaying information in a way that best matches users' interests. Here, we present several analyses investigating how prior surfing behavior predicts future surfing choices. Figure 1 presents a conceptual model of the surfing process used in *spreading activation* models of the diffusion of surfers through a Web site [Huberman *et al.* 1998; Pirolli *et al.* 1996]. Other models [Brin and Page 1998; Cunha and Joccoud 1997; Kleinberg 1998; Padmanabhan and Mogul 1996] can be shown to be variants of this conceptual model. Figure 1 illustrates the elements of the spreading activation model:

(a) users begin surfing a Web site starting from different entry pages,

(b) as they surf the Web site, users arrive at specific Web site pages having traveled different surfing paths,

(c) surfers choose to traverse possible paths leading from pages they are currently visiting, and

(d) after surfing through some number of pages, surfers stop or go to another Web site.

Elsewhere [Huberman and Adamic 1998], models have been developed to address how users choose new sites and which pages they visit first. Models have also been developed [Huberman *et al.* 1998] to characterize the distribution of the number of pages visited by users at Web sites. Here we concern ourselves with how past surfing paths may contain information that predicts future surfing paths.

Surfing paths can be conceptualized as traversals of the graph representing the hyperlink structure of a Web site, where nodes represent WWW pages and edges represent hyperlinks among pages. A simple predictive model might assume that users visiting each page will randomly choose which links to follow, resulting in a uniform distribution of users traversing each link from a page. In this model the transition probabilities associated with each link, (e.g., p_1, p_2, p_3 in figure 1(c) are simply one divided by the number of links emanating from a page. Several predictive models [Brin and Page 1998; Cunha and Joccoud 1997; Huberman *et al.* 1998; Padmanabhan and Mogul 1996] make a Markov-like assumption that the choice of the next page to surf is dependent only on the last page. At least a few models [Brin and Page 1998] assume uniformly weighted transitions down links. The advantage of these models is that they can be constructed directly from the Web site's hyper-

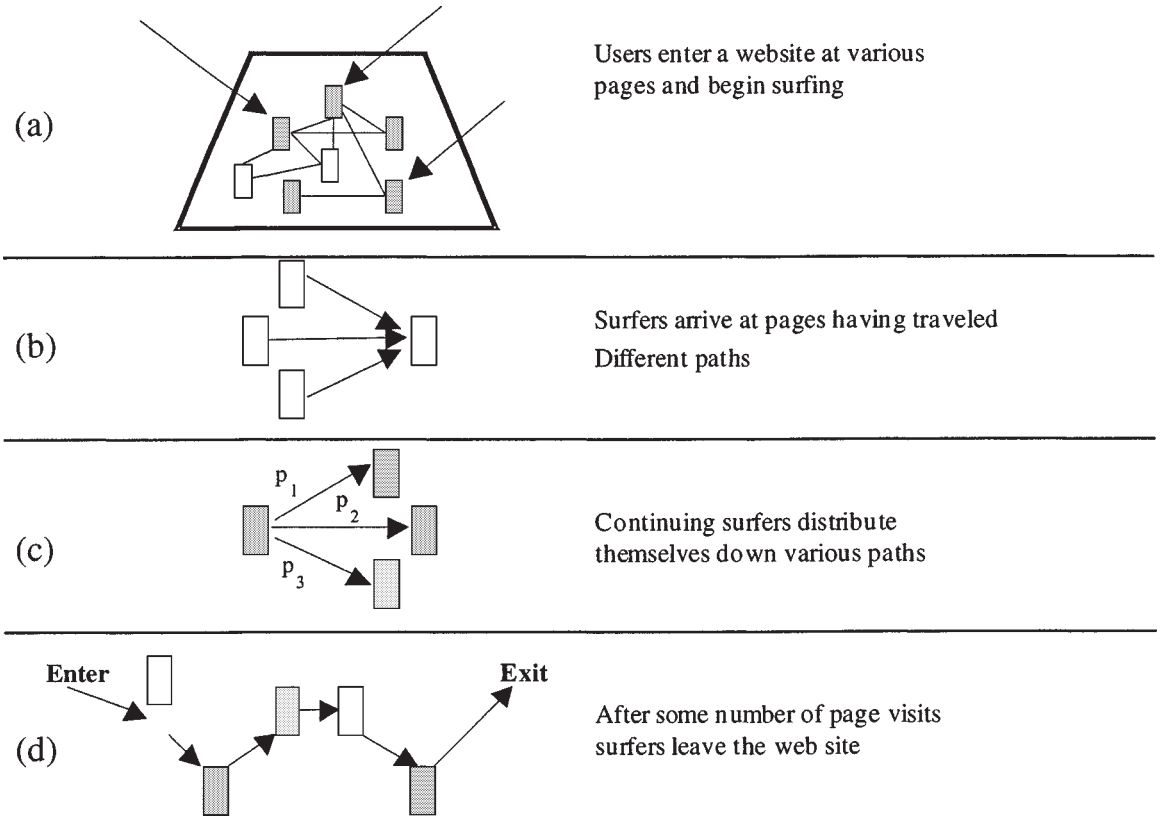

Figure 1. Model of WWW traversals. A conceptual model of surfers traversing a Web site.

link graph without collecting usage data, thereby keeping computational and storage requirements to a minimum.

However, given that it is possible to record the paths of users surfing through a Web site, non-uniform transition probabilities can be estimated for the links emanating from each page (e.g., p_1, p_2, and p_3 in figure 1(c)). With this slightly more complicated model, a user visiting a particular page is predicted to visit linked pages according to the estimated transition probabilities. This assumption was made in the spreading activation model presented in [Huberman *et al.* 1998]. One issue we investigate is the degree to which observed surfing transitions deviate from the assumption that surfers randomly choose linked pages.

Another assumption of Markov-like surfing models is that surfing paths leading up to the currently visited page do not influence transition probabilities. Kantor [Kantor 1997] challenges this notion by proposing a system that matches the observed surfing path of a user against the recorded paths of other users. Such a system would assume that future sequences of a user's surfing behavior would mirror paths observed by prior users. We present analyses that test whether there is predictive power to be gained by using longer prior surfing paths in prediction.

If predictive surfing models are based on user data to estimate transition probabilities, then we may also be concerned with the reliability and sensitivity of these estimates. We will present analyses concerning the impact of the size and span of the data sets used to construct transition estimates. We will also investigate how well estimates constructed over one span of time predict events at various distances into the future.

Before turning to these analyses, however, we first investigate the basic techniques that provide data on surfing patterns. A variety of methods have been used to extract surfing patterns from WWW logs, ranging from heuristics to the analysis of cookies. Very little is known about the quality of these measurement techniques. While descriptive statistics like the number of visits per page may not be affected by the choice of path determination algorithms, other statistics like the total visit time per user and the number of clicks per visit may be affected. Failure to accurately reconstruct user paths within Web sites makes accurate modeling of surfing problematic, if not erroneous. In the next section, we use several analytical methods to explore the reliability and impact of various path construction techniques.

2. Measurement of surfers and their paths

In order to examine the assumptions embedded within various models of surfing behaviors, the paths of users through Web sites have to be correctly identified. Despite nearly all Web servers being instrumented to record which Web pages are requested, the task of gathering reliable usage information from Web sites can be non-trivial, especially for path information [Pitkow 1997]. The presence of intermediary caches and proxies, the lack of client cookie compliance, the use of obfuscation and anonymizing tools,

and visits by robots all increase the difficulty in piecing together the sequence of user requests from server logs. Researchers attempting to characterize and model the Web employ a variety of methods and assumptions to reconstruct the paths of surfers, often times yielding different results. We hypothesized that some of the difference in these results may be attributed to different path construction methods.

This portion of the paper will quickly review the data generally recorded in server access logs and their limitations, followed by a description and comparison of various path construction methods. Specifically, we use the Xerox.com Web site to document the incidence rate of various assumptions, measure the impact of these methods on session times and clicks per session, and compute the similarity of generated paths using Levenshtein Distances [Levenshtein 1966].

Several limitations of this study are worth mentioning. First, the purpose of these results is to demonstrate the impact that various algorithms have on basic characterizations and, subsequently, on the models constructed of Web surfing. We are not attempting to make generalizations and recommendations that extend beyond the Web site used in our investigation. Rather, we motivate the need for careful consideration of the issues around path reconstruction and the techniques used to analyze paths. Second, we readily admit that other schemes exist for determining and analyzing user paths that are not included in our study, some of which may be better than the methods presented here. Again, our purpose is to illustrate that quality of path reconstruction matters, not to propose a specific path reconstruction methodology.

2.1. Recorded log file data

While Web servers have the capability to record vast amounts of information, relatively few fields are typically recorded. Several formats have evolved from the Common Logfile Format (CLF), including the Extended Logfile Format (ECLF) as well as a variety of customized formats. For the most part, the following fields are recorded by Web servers:

- the time of the request in seconds,
- the machine making the request recorded as either the domain name or IP address,
- the name of the requested URL as specified by the client,
- the size of the transferred URL,
- and various HTTP related information like version number, method, and return status.

Various Web servers also enable other fields to be recorded, the most common of which are:

- the URL of the previously viewed page (the "referrer" field),
- the identity of the software used to make the request (the "user agent" field),
- and a unique identifier issued by the server to each client (typically a "cookie").

While these fields are useful to analyze and provide reasonable characterizations, several enhancements would facilitate analysis that is more reliable and accurate as well as facilitate path reconstruction efforts. First, the unit of time recorded should encode a finer granularity like milliseconds or a site definable metric like ticks/second. This is especially important for heavily trafficked sites, where hundreds of requests per second can occur. Second, the URL recorded is the URL as requested by the user, not the location of the file returned by the server. This behavior can cause false tabulation for pages when the requested page contains relative hyperlinks, symbolic links, and/or hard coded expansion/translation rules, e.g., directories do not always translate to "index.html." It also can lead to two paths being considered different when in actuality they contain the same content. While both pieces of information are useful, the canonical file system-based URL returned by the server is arguably more useful as it removes the ambiguity of what resource was returned to the user.

The content of the information contained in the referrer field can be quite varied. Various browsers and proxies do not send this information to the server for privacy and other reasons. In addition, the value of the referrer field is undefined for cases in which the user requests a page by typing in the URL, selects a page from their Favorites/Bookmarks list, or uses other interface navigational aids like the history list. Furthermore, several browsers provide interesting values for the referrer field. To illustrate, suppose a user selects a listing for the Xerox Corporation on Yahoo. In requesting the Xerox splash page, the URL for the page on Yahoo is provided as the value for the referrer field. Now suppose the user clicks on the Products page, returns to the Xerox splash page, and reloads the splash page. In several popular browsers, the referrer field for Yahoo is included in the second request for the Xerox splash page although the last page viewed on the user's surfing path was the Product page in the Xerox site. If one chooses to reconstruct paths by relying upon the referrer field, the paths of two users may be identified instead of only one. Given these limitations, strong reliance upon the information in the referrer field may be more problematic than one would initially expect.

The user agent field also suffers from imprecise semantics, different implementations, and missing data. This can partially be attributed to the use of the field by browser vendors to perform content negotiation. Given that the rendering of HTML differs from browser to browser, servers have the ability of altering the HTML based upon which browser is on the other end. Consequently, the user agent field may contain the name of multiple browsers. Some proxies also append information to this field. As we shall later show, the value of the user agent field can vary for requests made by the same user using the same Web browser. Adding to the confusion, there is no standardized manner

to determine if requests are made by autonomous agents (e.g., robots), semi-autonomous agents acting on behalf of users (e.g., copying a set of pages for off-line reading), or humans following hyperlinks in real time. Clearly, it is important to be able to identify these classes of requests to construct accurate models of surfing behaviors.

Although cookies were initially implemented to facilitate shopping baskets, a common use of cookies is to uniquely identify users within a Web site. Cookies work in the following manner. When a person visits a cookie enabled Web site, the server replies with the content and a unique identifier called a cookie, which the browser stores on the user's machine. On subsequent requests to the same Web site, the browser software includes the value of the cookie with each request. Because the identifier is unique, all requests that were are with the same cookie are known to be from the same browser. Since multiple people may use the same browser, each cookie may not actually represent a single user, but most Web sites are willing to accept this limitation and treat each cookie as a single user. Recently, browser vendors have provided users with controls to select the cookie policy that maps to their privacy preferences. This enables users to choose various levels of awareness when visiting a server that issues cookies in addition to allowing users completely disable their browser from sending cookies. Consequently, unless a site requires people to use cookies to receive content, the cookie field may be null, which leaves the task of identifying user paths to relying upon the other recorded fields. We shall now explore various methods of reconstructing user paths.

2.2. Reconstructing user paths

Given the limitations of the information recorded in Web access logs, it is not surprising that sites require users to adhere to cookies or defeat caching to gain more accurate usage information. Still, numerous sites either do not use cookies or do not require users to accept cookie to gain access to content. In these cases, determining unique users and their paths through a Web site is typically done heuristically. Later, we provide an empirical analysis of the tradeoffs that exist using different methods of identifying users and their click streams. Clearly, without the accurate reconstruction of user paths, subsequent analyses and attempts to model surfing may be seriously flawed.

2.3. Usage data

All analyses reported in this paper were computed from ten days worth of usage data acquired from the Xerox.com Web site from May 10 through May 19, 1998. The site received between 220,026 and 651,640 requests per day during this period. Later in this paper, we explore various alternative methods for tabulating usage statistics. Over this period, there were 16,051 files on the Xerox.com Web site, of which 8,517 pages were HTML. The Web site issues cookies to users only upon entry to the Xerox splash page and records the user agent field for each request.

2.4. IP and domain name counting

The most simplistic assumption to make about users is that each IP address or domain name represents a unique user as in [Arlitt and Williamson 1996; Manley *et al.* 1997]. Using this method, all the requests made by the same host are treated as through from a single user. When a new host is detected, a new user profile is created and the corresponding requests are associated to the new user. We call this heuristic "IP."

Several methods that use additional information recorded in the access logs or other heuristics are also possible. One refinement is to use the user agent field. Using this method, new users are identified as above as well as when requests coming from the same machine have different user agents. We call this method "IP-Agent." Another refinement is to place session timeouts on requests made from the same machine. The intuition is that if a certain amount of time has elapsed, then the old user has left the site and a new user has entered. Empirically derived timeouts of 25 minutes (1 and $1/2$ standard deviates away from a mean of 9.2 minutes between user interface requests) were first used by [Catledge and Pitkow 1995]. Many commercial log file analysis programs use similar timeout periods between requests before starting a new user profile from the same host. We refer to this method as "IP-Timeout."

Given that the majority of users access the Web either through home via an ISP or via a firewall at work [Pitkow and Kehoe 1996], we hypothesized that these methods would not provide an accurate identification of users or accurate reconstruction of user paths. Another problem with using IP or domain addresses as user identifiers is that many ISPs load balance user requests through a number of proxies. Within one session, a user may rotate between several proxies, each with a different IP and domain name. An example of this occurs with America Online (AOL), where users are directed through prefix permuted proxy addresses. Typical entries in log files for such cases contain hosts like "ww-ta01.proxy.aol.com" and "ww-tl05.proxy.aol.com". On a randomly selected day during May 1998, the Xerox Web site observed over 230 different hosts within the "aol.com" domain. This problem also occurs in environments where IP addresses are assigned dynamically with short timeouts. One method for dealing with changing machine names is to chop the prefixes off domain names and, based upon the IP class (A, B or B), chop the suffix off IP addresses. We call this method "Host-munging."

When using these methods for identifying users, the following situations occur when sequentially processing access logs:

(1) a new IP address is encountered (assume this is a new user),

(2) an already processed IP address is encountered:

   (a) the user agent matches prior requests (assume this is the same user),

Table 1
Effects of various IP based counting methods as applied to all the requests and only page requests received by the Xerox Web site from May 10, 1998, through May 19, 1998. The data show the occurrence rate in the form of percentages for each case (same host, different host, and session timeout).

Method	Same host	New host	Session timeout
All file results			
IP	98.27%	1.73%	–
Host-Munging-IP	99.07%	0.93%	–
IP-Timeout	97.80%	1.72%	0.47%
Page results			
IP	89.62%	10.38%	–
Host-Munging-IP	94.37%	5.63%	–
Timeout-IP	88.45%	5.51%	6.04%
Host-Munging-Timeout-IP	91.25%	5.45%	3.30%

(b) the user agent field does not match any prior requests from the same IP (assume this is a new user),

(c) when a session is terminated due to a timeout, assume a new user has entered the site.

For this analysis, a list of IP addresses/domain names, user agents, and last access times are maintained while processing the log file.

Table 1 shows the incidence rate of these cases using the data from the Xerox.com May 1998 data set. Two sets of occurrence rates are presented in table 1. The first shows the average occurrence rate when all files (Web pages and all embedded images) are taken into account and the second shows the results when just Web pages are included in the analysis. Note that higher values do not translate to "better." Very little variation in the percentages reported occurred across the data set. As a percentage of all requests, between 0.93% to 1.73% are from new hosts with the remainder being repeat requests from these hosts. Using host-munging reduces the number of new hosts encountered by nearly a percentage point. Only 22,000 of the 4.8 million requests (0.47%) resulted in a session timeout using an inactivity period of 25 minutes.

When page views are used as the primary unit of analysis, the number of new hosts increases significantly to 10.38% of all page requests. That is, one out of every ten page requests were from new users. When host-munging is used, the number of new hosts decreases to 5.63%, with 94.37% of the remaining requests being issued from one of these hosts. When the combination of host-munging and timeouts are used, the number of timeouts decreases to 3.30%. These findings imply that half of the new hosts and timeouts were from hosts in the same domain/IP address space. From this we can infer that a large number of Xerox Web site users either connect to the Web via ISPs with load balancing proxies, or that a large number of different users access the site from within the same domain as would occur with a large company, or that some combination of both cases exists.

Regardless, a significant number of page requests resulted in ambiguous cases, where it is not possible to determine the existence of new users with certainty. While we expect the incidence rate to vary considerably from Web site to Web site, we find the results concerning since, as we shall see, these IP-based methods and other IP-based derivatives are used in cases where unique identifiers like cookies are not present.

2.5. Cookie counting

When processing a site that is capable of issuing cookies and logging user agents, several scenarios exist (see table 2). We denote this class of user identification scenarios "Cookie." For this analysis, we maintain a list of hosts, agents, and cookies while processing log files, so changes and new entities can be detected. While the processing of cookies may intuitively seem simple, it is actually a bit more complicated.

Several cases are possible when a previously encountered cookie is processed. If the request is coming from the same host regardless of the user agent, the request is being issued by the same user (case 1). This is true because a unique cookie is issued to only one browser. If the user agent field remains the same but the host changes, it is still the same user (case 2A) and some form of IP/domain name changing is occurring. This often occurs with users behind

Table 2
The set of possible scenarios for determining users when a site is capable of issuing cookies and recording the user agent field.

Case	Cookie	Host	Agent	Conclusion
1	Same Cookie	Same Host	Not significant	Same User
2A	Same Cookie	Different Host	Same Agent	Same User
2B	Same Cookie	Different Host	Different Agent	Error
3	No Cookie	Same Host	Not significant	Uncertain
4	No Cookie	Different Host	Not significant	Different User
5A	New Cookie prior cookie value "–" from host	Same Host	Same Agent	New User previously issued first cookie
5B	New Cookie prior cookie value from host not "–"	Same Host	Same Agent	Uncertain
5C	New Cookie, prior cookie value from host not "–"	Same Host	Agent changes	Different User
6	New Cookie	Different Host	Not significant	Different User

firewalls and ISPs that load-balance proxies. However, if we have the same cookie with a different user agent, then an error has most likely occurred as cookies are not shared across browsers (case 2B). If no cookies are present, we resort to the same set of heuristics used by the IP method. If the request comes from a known host, then we could have a new user or the same user (case 3), otherwise the request is from a different user (case 4). It is important to point out that these latter two cases could also be issued from non-cookie compliant crawling software.

An interesting set of cases occurs when a new cookie is encountered. If the request is from a host that has already been processed and the previous value of the cookie was "–" or "null" and the user agent is the same, it is fair to conclude that the request is from a new user that just received their first cookie from the server in the previous request (case 5A). If the client is not using cookie obfuscation software, one would expect the following requests from this user to all contain the same cookie (case 1). However, suppose the previous value from the same host and agent was a different cookie, it could be the same user obfuscating cookie requests, or a new user from the same ISP using the same browser version and platform as the user from the previous request. Barring any other piece of supporting evidence like the referrer field or consulting the site's topology, it is difficult to determine which is the correct scenario (case 5B). If the user agent is different from the previous request, but accompanies a new cookie from the same host, it is fair to assume that a new user has entered the site (case 5C). Of course, a new cookie from a new host regardless of the agent is a new user (case 6) and session timeouts can still occur (case 7).

Table 3 shows the results of processing the ten days of Xerox.com data. When all requested files and cookies are used, 86.56% of all requests originate from the same user on the same host (case 1). There was notable variation in this statistic over the course of the sample, with weekdays showing higher rates (86.65% to 89.31%) and weekends showing lower rates (82.03% to 83.32%). Inspection of the data revealed that this effect was due to the weekend users being less likely to be cookie compliant than their weekday counterparts. There were very few instances of users switching hosts (case 2A), and 0.05% of the cases in which the host and user agents changed though the cookie remained the same. While technically this should indicate an error condition, inspection of the log file showed cases in which a user's requests were being issued through two separate proxies, each running separate proxy software, and hence appending different user agent information to the request.

Slightly over 8% of the all requests did not send any cookie information. This number increased to 11% over the weekend. One percent of users were new to the site from the same host (case 5A and case 5C) and just over a percent appear to use some form of cookie obfuscation tool (case 5B). Roughly one percent of all requested files was from new users from new hosts (case 6). As one would expect when host-munging is used, all cases where different hosts are criteria showed a decrease. The number of timeouts that occurred was 0.08% or just 3,682 out of the 4.7 million number of files requested.

As with the IP methods, when only Web pages are considered, the influence of each individual's cookie policy increased. The number of requests being issued from cookie compliant users (cases 1, 2A, and 2B) decreased nearly twenty percentage points from 87.84% to 68.37%, further revealing the notable difference between hits and page view methods. The percentage of new cookies being issued is higher, with case 5B and case 6 posting the most gains. For case 5B, notable spikes were observed over the weekends, suggesting the weekend users are more likely to use cookie obfuscation technologies than weekday users. For case 6, the percentage of new users from different hosts remained stable across weekend/weekday transitions.

As with the hit analysis, the number of cases for page views did not change dramatically for host-munging or session timeouts. The exception was that the number of new cookies from new hosts (case 6) was lower for host-munging, with those cases being picked up by new cookies from a known host (case 5C). When all techniques are used – cookies, munging, and timeouts – the same host-munging driven effect occurs.

Table 3
Effects of various cookie based counting methods as applied to all the requests and only page requests received by the Xerox Web site from May 10, 1998, through May 19, 1998. The data shows the occurrence rate in the form of percentages for each of the ten possible cases.

Method	1	2A	2B	3	4	5A	5B	5C	6	7
	(Numbers express the percentage of all requests)									
All files results										
Cookie	86.56	1.23	0.05	7.99	0.56	0.45	1.29	0.52	1.37	–
Host-Munging-Cookie	87.67	0.14	0.02	8.23	0.32	0.44	1.26	1.15	0.78	–
Timeout-Cookie	86.49	1.23	0.05	7.98	0.55	0.45	1.29	0.51	1.37	0.08
Page-Cookie	66.88	1.45	0.04	6.17	2.96	1.89	8.72	3.21	8.68	–
Page results										
Host-Munging-Cookie	68.15	0.20	0.01	7.39	1.75	1.89	8.54	7.11	4.96	–
Timeout-Cookie	66.69	1.44	0.04	6.15	2.95	1.88	8.69	3.20	8.65	0.29
Host-Munging-Timeout-Cookie	67.72	0.20	0.01	7.34	1.74	1.88	8.48	7.07	4.93	0.64

2.6. Session length and number of clicks per session

In the above analyses, we measured the occurrence rate for users within a site for each method, providing a basis to understand how the construction of individual paths would be affected by each method. In this section, we examine the total time a user spends within a Web site, or "session" and the total number of clicks per session for each method (see table 4). It is important to note that both session time and number of clicks are right skewed distributions, and as such, the average case as reported by the mean is not the typical case encountered by most users. In order to measure the effect of each method, the following five groups were created and pair-wise Welch two-sample t-tests were performed to compare the statistical similarity of the resulting session time and number of click distributions for each method:

- IP, Munging-IP, and Timeout-IP,
- Page-IP, Page-Munging-IP, Page-Timeout-IP, Page-and Munging-Timeout-IP,
- Cookie, Host-Cookie, and Timeout-Cookie,
- Page-Cookie, Page-Munging-Cookie, Page-Timeout-Cookie, and Page-Munging-Timeout-Cookie,
- Page-IP, Page-Munging-Timeout-IP, and Page-Munging-Timeout-Cookie.

Except for the Page-Cookie and Page-Munging-Cookie clicks per session comparison, all distributions had unequal variances. The only cases that did not result in statistically significant different means were: (a) the Page-Timeout-Cookie and Page-Munging-Timeout-Cookie clicks per session, and (b) Page-IP and Page-Munging-Timeout-IP clicks per session.

Although the occurrence rate for the various IP methods using all requests was small, the impact of the various heuristics with respect to session times and the number of clicks per session is large. When host-munging is used, the median session time jumps from 4.15 minutes to 6.8 minutes. The increase can be attributed to more requests being incorrectly treated as a single user. When timeouts are used, again with the 25 minute default, more distinct users are detected, and the median session time decreases to 3.03 minutes per user. The number of clicks per session suffers the same effect of misidentifying users. When the pure IP-per-user metric is used, the median number of clicks is 38, but increases to 48 clicks for host-munging, and decreases to 30 clicks when timeouts are calculated.

The effect of falsely identifying users continues with page views, where the median session time is 5.95 minutes using just the IP of the requesting machine. The increase from page views can be attributed to the lack of images to inject noise into the inter-arrival time of requests. As one might expect, when the IP page view method is combined with host-munging and session timeouts, the typical session time drops to 5.08 minutes, which is longer than the timeout method (4.15 minutes), but shorter than the host-munging method (10.48 minutes). The impact of the various strategies is even more pronounced for the number of clicks per user, where the page view IP method users typically request 6 pages per session versus the 5 pages per session for host-munging, 1 page per session for timeouts, and 4 pages per session for the combination of all the methods. Even at the IP counting level, the impact of the various strategies is quite dramatic and can vastly sway the basic characterizations of session time and the number of clicks per session.

When cookies are used to measure the total number of files requested, the session time drops to 2.7 minutes, with users typically requesting 22 total items from the Web site. Although host-munging increases the number of clicks

Table 4
The number of clicks per session and the total time per session for all methods using the Xerox Web site May 1998 data.

Method	Clicks			Session time		
	Median	Mean	Stand. Dev.	Median	Mean	Stand. Dev.
All files results						
IP	38	59.68	289.57	249	4074	12635.22
Host-Munging-IP	48	106.10	782.20	412	8580	18098.17
IP-Timeout	30	46.74	251.78	182	605	3229.59
Cookie	22	33.89	68.86	162	1684	6744.78
Host-Munging-Cookie	23	34.89	70.18	178	2084	7541.36
Timeout-Cookie	21	31.35	65.20	137	707	3862.45
Page results						
IP	5	9.79	78.75	357	4358	12678.60
Host-Munging-IP	6	17.31	199.63	629	9491	18624.43
Timeout-IP	1	1.01	0.88	249	356	2460.345
Host-Munging-Timeout-IP	4	10.94	156.75	305	933	3903.87
Cookie	3	5.81	13.45	269	1899	6708.37
Host-Munging-Cookie	3	5.96	13.51	295	2368	7647.17
Timeout-Cookie	3	5.41	12.66	231	898	4311.14
Host-Munging-Timeout-Cookie	3	5.46	12.56	244	1016	4397.26

slightly, the variance remains stable with respect to the other methods. This suggests that the users are being identified more reliably and not lumping all requests from within an organization into one user.

In what appears to be the most stable group of methods, the results of the page view analysis using cookies are quite similar. While the typical paths for the Page-Cookie, Page-Host-Cookie, and Page-Timeout Cookie methods all yield the same 3 clicks per session, the means are statistically different, with the reading time showing more variance (4.48 minutes versus 4.92 minutes versus 3.85 minutes, respectively). This increase in variance can play a pivotal role in simulating Web traffic, where accurately modeling the heavy tail properties is very important. As noted previously, there was not a significant difference between the Page-Timeout-Cookie and the Page-Host-Timeout-Cookie methods with respect to the number of clicks per session, though the reading times were noticeably different (3.85 minutes versus 4.06 minutes, respectively). This finding underlies one of the weaknesses of comparing the metrics of clicks and total visit time: these metrics do not speak directly to the correctness of the paths generated by each method.

2.7. Levenshtein distance

We use the Levenshtein (or *edit*) Distance [Levenshtein 1966] to measure the similarity between the paths identified by the most promising path reconstruction methods. LD provides a quick method for judging the closeness of two arbitrary length strings based upon the number of insertions, deletions, and changes/reversals that are necessary to convert one string to another. For a string s, let $s(i)$ stand for its ith character. For two characters a and b, define

$$r(a,b) = \begin{cases} 0, & \text{if } a = b, \\ \textit{change}, & \text{otherwise,} \end{cases}$$

where *change* is a language-specific weighting parameter, typically set to one. Assume we are given two strings s and t of length n and m, respectively. We are going to fill an $(n+1)$ by $(m+1)$ array d with integers such that the low right corner element $d(n+1, m+1)$ will furnish the required values of the Levenshtein Distance $L(s,t)$. The definition of entries of d is recursive. First set $d(i,0) = i$, $i = 0, 1, \ldots, n$, and $d(0,j) = j$, $j = 0, 1, \ldots, m$. For other pairs i, j use

$$d(i,j) = \min\big(d(i-1,j) + \textit{deletion}, d(i,j-1) + \textit{addition}, d(i-1,j-1) + r\big(s(s), t(j)\big)\big).$$

Typically, *change*, *deletion* and *addition* are set to one to place equal importance upon insertions, deletions, and changes. We used one as the value for these weightings in our investigations.

One of the nice properties is that the numeric similarity produced by LD defines a metric space. A space X is metric if there is defined a real non-negative function of two variables $d(A,B)$. The function is known as the distance between the two points. It is characterized by the following properties. For $A, B, C \in X$:

(1) $d(A,B) = 0$ if and only if $A = B$ (the distance is 0 if and only if the points coincide).

(2) $d(A,B) = d(B,A)$ (the distance from A to B is the same as the distance from B to A).

(3) $d(A,B) + d(B,C) \geqslant d(A,C)$ (the sum of two sides of a triangle is never less than the third side).

In order to test the similarity of the paths generated by each method, the following comparisons were made (see table 5). "Standard" refers to counting methods that are typically employed by various logfile analysis programs and "Modified" refers to the paths generated by the most promising methods in this study. The LD was computed for each path generated by the standard method against all the paths generated by the modified method for the same host with replacement. One should note that this is a very forgiving method of comparison as it increases the likelihood that a standard path will match a modified path. A more conservative approach would match without replacement, i.e.,

Table 5
Levenshtein Distance comparisons of the most commonly used methods (Standard) against the most promising methods identified in this study (Modified). Only page views were used for the comparisons.

Standard	Modified	Mean	Max	Stand. Dev.	$\sum$
Host-Timeout-IP	Host-Timeout-Cookie				
	Changes per Path	0.8721	190	6.09	26415
	Deletions per Path	0.0092	22	0.23	646
	Insertions per Path	0.6479	170	5.41	45290
	LD Edits per Path	1.5370	190	8.61	107414
	Average LD per Page	0.2587	181	2.86	18085
Cookie	Host-Timeout-Cookie				
	Changes per Path	0.1691	92	1.40	6993
	Deletions per Path	0.0068	11	0.13	283
	Insertions per Path	0.1131	81	1.25	4676
	LD Edits per Path	0.2891	94	1.96	11952
	Average LD per Page	0.0319	11	0.16	1321

once a standard path matches a modified path, the modified path is removed from further comparisons for that host.

The comparison between treating each host-munged IP with 25 minute session timeouts against using cookie with host-munging and the same session timeouts resulted in an average of 1.54 insertions, deletions, or changes per every path considered. Incorrectly guessing a portion of the path was the most common form of modification and occurred in almost half of those cases (0.87). The average Levenshtein Distance per page calculation determines the likelihood that for each page in a path an edit of some sort will occur. It is not surprising given that most paths are short (median of 2 clicks), there is a one in four chance that using IP counting will result in an incorrect modification to a user's path.

When cookies are compared to the modified version that uses host-munging and session timeouts, the number of edits decreases significantly. As with the previous comparison, reversals occur with a higher frequency than additions or deletions, though we were unable to determine the exact reason. The average number of edits per page using the standard cookie-per-user algorithm was only 0.0319 edits/page, though for each path, this number increased to 0.29 edits per path. While the number of edits per path certainly decreases when the standard cookie algorithm is employed, the number of incorrect paths generated is still concerning, indicating that simple cookie-based path reconstruction is not as straightforward as one might initially think.

In the above sections, we presented empirical evidence that suggests the methods used to identify users and reconstruct paths have significant impact on basic characterizations of users' surfing behaviors as well as the reconstructed surfing paths. While the numbers presented for the Xerox.com Web site are by no means meant to be absolute with respect to other sites, it does provide an initial glimpse into the various cases associated with the dynamics of cookie and IP-based reconstruction methods. Having established the impact of various path reconstruction algorithms, we now turn our attention towards the assumptions being made about modeling surfing behaviors.

3. Distribution of users over links from a page

Imagine the users who visit a page on the WWW, who then decide to surf to other pages linked to that page. They process the content of the visited page and, based on some decision, click on a hyperlink that takes them to another page. On the one hand, it may be plausible to assume that every link emanating from that page would get chosen an equal number of times over the course of visits from many users. After all, although users may have different interests, their interests may be uniformly distributed over links when we aggregate over very large numbers of users. On the other hand, the population of users that visit a WWW page may have some systematic bias in their pattern of interest. Some links from a page may be generally more relevant than others, for that particular population of visitors. A similar bias results from systematic biases imposed by the structure of the interface to WWW content. For instance, it seems plausible to assume that users process displayed WWW pages in a relatively common and systematic manner (e.g., top-down and left-to-right). Such systematic interaction patterns might introduce biases in the patterns of observed link-following behavior. For instance, links encountered earlier in reading a page might have a higher likelihood of being selected than later ones, even when they are of equal relevance to the user.

Existing models make different assumptions about how users distribute themselves over links from a page. The algorithms used in Google [Brin and Page 1998] and Clever [Kleinberg 1998] assume that the links emanating from a page are equally weighted with respect to user interests, document relevance, or likelihood of being pursued. Spreading activation models [Huberman *et al.* 1998; Pirolli *et al.* 1996] allow for systematic biases. We now turn to an investigation of these assumptions.

3.1. Goodness-of-fit of uniform distribution model

Figure 2 presents a series of histograms of the observed proportions of users who choose links emanating from pages. We only include data for surfers who move from one page to the next (rather than leave the Web site). Each histogram displays a set of Web pages classified by their number of outlinks (links emanating from a page). All Web pages with two outlinks are characterized by one histogram, all with three outlinks another histogram, and so on up to pages with 16 outlinks as the number of pages within the Xerox site with more than 16 links was nominal. We have ignored pages with just one outlink since all continuing surfers will follow that link. Pages with greater than 16 outlinks are ignored. On each histogram, we have also marked, using a vertical line from top to bottom of each chart, the expected proportion of users who should follow links if they chose links with uniform weighting. For instance, for pages with two links, 0.5 of the users should choose each link, and in general, the expected proportion of users choosing links will be $1/$(number of links).

In figure 2, it seems that the modes of the observed distributions are close to the values expected by assuming a uniform distribution. However, the observed distributions also appear to be skewed, with a few large observed proportions and many small observed proportions. We computed χ^2 tests to determine the goodness-of-fit of the uniform distribution assumption. For each page i, we let n_i be the number of users observed to continue on to linked pages (for our data set), L_i the number of links emanating from i, and $p_{i\ell} = 1/L_i$ the expected proportion of users who will choose each of the $\ell = 1, 2, \ldots, L_i$ links. We let the expected frequency of users who travel any given link be

$$E_{i\ell} = n_i p_{i\ell} \tag{1}$$

and the corresponding observed frequencies of users traveling the same links, obtained from our data, are $O_{i\ell}$.

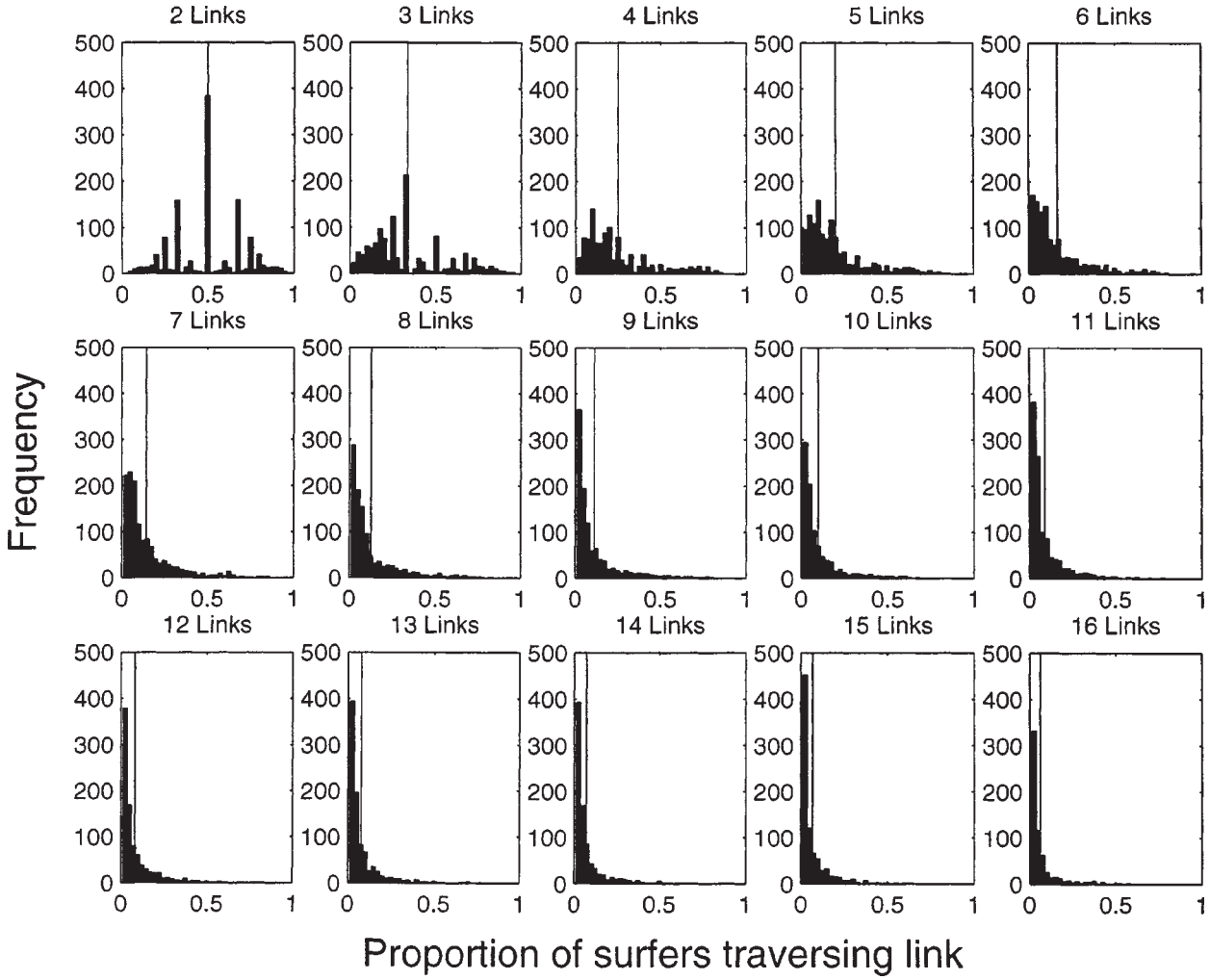

Figure 2. Estimated surfer transition probabilities. Histograms of estimated surfer transition probabilities for links emanating from pages at the Xerox.com Web site over the May 10,1998, through May 19, 1998 period. Pages are categorized by the number of outlinks. The vertical lines indicate the transition probabilities expected by a uniform distribution model.

Table 6
Goodness-of-fit tests of a model of surfers transitioning page outlinks with uniform probability. Note: $p < 0.001$ at values greater than $\chi^2(6) = 22.46$.

No. links, L_i	χ^2(df = 6)
$0 < L_i \leqslant 3$	25.21
$3 < L_i \leqslant 6$	25.21
$6 < L_i \leqslant 9$	172.99
$9 < L_i \leqslant 12$	59.70
$12 < L_i$	2666.30

Table 6 summarizes our χ^2 tests. Pages were categorized according to the number of links emanating from them, as indicated in table 6. From exploration of the expected frequency distributions, we chose to partition the observed and expected distributions into $k = 1, 2, \ldots, 7$ bins (to avoid bins with zero expected frequencies), and pooled the expected $E_{i\ell}$ and observed $O_{i\ell}$ into those bins to give us aggregate E_k and O_k for each $k = 1, 2, \ldots, 7$ bins. For each category of pages, we calculated the goodness-of-fit χ^2 statistics in table 1 using E_k and O_k.

All of the values of χ^2 are significant at $p < 0.001$, indicating a poor fit of the uniform distribution to the observed distribution of users over links. Moreover, the χ^2 values generally become larger with increasing number of links from a page, indicating greater deviations from the uniform distribution. Models that capture the non-uniform distribution of users surfing to linked pages should provide better fits to observations. Of course, the increased accuracy comes at the cost of more free parameters to be estimated from data.

4. Information contained in surfing paths

Our analysis of how surfers distribute themselves over outlinks considered the probability of transitioning down a link given only knowledge of the page currently visited. Longer sequences of surfers' previous page visits might provide more information about their next transition down a hyperlink. In this case, users can be thought of as building context for future page requests, or as part of some goal-

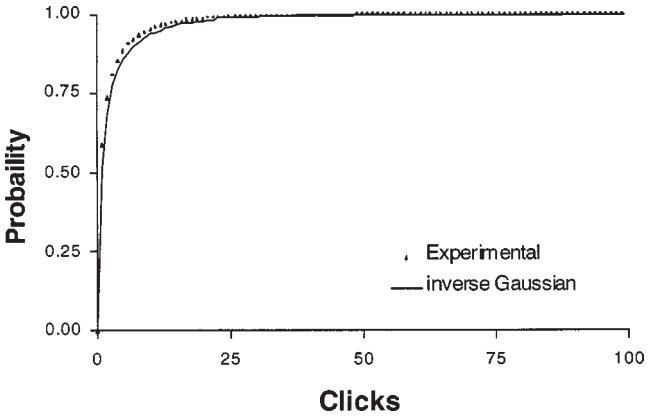

Figure 3. The law of surfing. The cumulative distribution function of AOL users as a function of the number of clicks surfing [Huberman *et al.* 1998]. The observed data were collected on December 5, 1997, from a representative sample of 23,692 AOL users who made 3,247,054 clicks. The fitted inverse Gaussian distribution has a mean of $\mu = 2.98$ and $\lambda = 6.24$.

directed behavior. For instance, if we saw that a surfer had visited a sequence of pages dealing with cars rather than a sequence dealing with books, we might predict that their future transitions are likely to deal with cars too.

We can think of these surfing paths as *n-grams*. Such n-grams can be represented as tuples of the form $\langle X_1, X_2, \ldots, X_n\rangle$ to indicate sequences of page clicks by a population of users visiting a Web site. Each of the components take on specific values $X_i = x_i$ for a specific surfing path taken by a specific user on a specific visit to the Web site. Our last analysis of the distribution of users over outlinks considered n-grams of the form $\langle X_1, X_2\rangle$ where X_1 was the current page visited by a surfer and X_2 was the page surfed to by transitioning down an outlink.

Users often surf over more than one page at a Web site. We may record surfing n-grams, $\langle X_1, X_2, \ldots, X_n\rangle$ of any length observable in practice. Assume we define these n-grams as corresponding to individual surfing sessions by individual users. That is, each surfing session is comprised of a sequence of visits made by a surfer, with no significantly long pauses. Over the course of a data collection period – say a day – one finds that the lengths, n, of surfing paths will be distributed as an inverse Gaussian function, as in figure 3. This appears to be a universal law that is predicted from general assumptions about the foraging decisions made by individual surfers [Huberman *et al.* 1998]. From figure 3 it is apparent that the bulk of recorded n-grams will be very short, although there will be a few very long surfing n-grams.

4.1. Entropy analysis: kth order Markov approximations

In our analysis of surfers distributing themselves over outlinks, we were concerned with n-grams of the form $\langle X_1, X_2\rangle$ and, more specifically, with the probability of transitioning to X_2 given that the surfer was visiting X_1. That is, we were interested in the conditional probabilities,

$$p(x_2 \mid x_1) = \Pr(X_2 = x_2 \mid X_1 = x_1). \tag{2}$$

We can generalize this to a concern with the conditional probability that a surfer transitions to an nth page given their previous $k = n - 1$ page visits:

$$p(x_n \mid x_{n-1}, \ldots, x_{n-k}) = \Pr(X_n = x_n \mid X_{n-1}, \ldots, X_{n-k}). \tag{3}$$

Such conditional probabilities are known as kth-order Markov approximations (or kth-order Markov models). The zeroth order Markov model is just the unconditional base rate probability:

$$p(x_n) = \Pr(X_n), \tag{4}$$

or, in our case, the simple probability of a page visit. This might be estimated as the proportion of times a page is visited over the course of some time period.

X_n is thought of as a random variable whose values, $X_n = x_n$, indicate which page will be visited by a surfer. For a given kth-order Markov model we may ask how much uncertainty there is about the values of X_n, and investigate how this uncertainty changes as we increase k, the length of the previous sequence of visits used to predict X_n in a conditional probability. This can be accomplished by analyzing the *entropy* or *conditional entropy* of the models.

The entropy $H(X)$ of a single random variable, X, is the expected (average) uncertainty of the random variable:

$$H(X) = E\left(\log_2 \frac{1}{p(X)}\right) = \sum_{x\in X} p(x) \log_2 \frac{1}{p(x)} = -\sum_{x\in X} p(x) \log_2 p(x) \tag{5}$$

which is measured in bits. One way to think of entropy is that it provides an indication of the minimal coding that would be required to represent the state of X_n. Higher bit values indicate higher uncertainty about the state of X_n and the fact that it would require more coding to represent the state of X_n. For instance, the values taken on by throws of two dice have higher entropy than the values taken on by throws of one die. It would take a code of more bits to minimally distinguish the states of two dice than the states of one die.

For our analysis of kth-order Markov models of surfing n-grams, we will also need to compute conditional entropy $H(X_n \mid X_{n-1}, \ldots, X_{n-k})$. In our case, this can be interpreted as the amount of uncertainty about X_n that remains after we know that a surfer has visited k previous pages. The conditional entropy can be calculated by

$$H(X_n \mid X_{n-1}, \ldots, X_{n-k}) = \sum_{x_n \in X_n} p(x_n) H(X_n \mid X_{n-1} = x_{n-1}, \ldots, X_{n-k} = x_{n-k}), \tag{6}$$

Table 7
Entropy and conditional entropy estimates (in bits) for kth order Markov models of the Xerox.com data from May 10, 1998, through May 19, 1998.

	kth order model									
Length n-gram	0th	1st	2nd	3rd	4th	5th	6th	7th	8th	9th
2	7.89	3.25								
3	8.01	3.28	2.40							
4	8.14	3.29	2.33	1.62						
5	8.18	3.29	2.24	1.46	0.79					
6	8.20	3.28	2.16	1.33	0.64	0.37				
7	8.21	3.28	2.10	1.24	0.55	0.29	0.14			
8	8.20	3.27	2.04	1.16	0.48	0.24	0.11	0.07		
9	8.20	3.27	2.01	1.11	0.44	0.22	0.09	0.05	0.03	
10	8.19	3.25	1.95	1.04	0.40	0.19	0.08	0.04	0.02	0.02

and by using the chain rule for entropy

$$H(X_1, \ldots, X_n) = H(X_1) + H(X_2 \mid X_1) + \cdots + H(X_n \mid X_1, \ldots, X_{n-1}), \quad (7)$$

which involves the joint entropy of variables,

$$H(X, Y) = -\sum_{x \in X} \sum_{y \in Y} p(x, y) \log_2 p(x, y), \quad (8)$$

where $p(x, y) = \Pr(X = x, Y = y)$ is the joint probability of the random variables X and Y.

If we model surfing paths by kth order Markov models, then we can use these entropy and conditional entropy measurements. To measure the amount of uncertainty that remains in predicting X_n if we know surfers' k previous page visits then:

zeroth order: $H(X_n)$ (based on the probability that page is visited regardless of path),

first order: $H(X_n \mid X_{n-1})$,

second order: $H(X_n \mid X_{n-1}, X_{n-2})$,

kth order: $H(X_n \mid X_{n-1}, \ldots, X_{n-k})$.

Table 7 presents results of an entropy and conditional entropy analysis of the Xerox.com data for May 10, 1998, through May 19, 1998. In table 7 we have stratified the data according to the length of the n-gram obtained from individual surfers' visits. For each length of n-gram, we then calculated the entropy and conditional-entropy for models of order zero up to length n. That is, we considered all models of order $0 \leqslant k \leqslant n - 1$ for each class of surfing n-gram of length $n = 2, \ldots, 10$ surfing transitions. Note, however, that the size of the reduction (in terms of bits) diminishes: The largest reduction in entropy is obtained by moving from a zeroth-order model to a first-order model, less reduction is obtained by moving from a first-order to second-order model, and so on.

As with the analysis of surfing distributions over outlinks, we find the ubiquitous trade-off in model complexity with model accuracy. If we model longer paths we can reduce uncertainty in predicting future visits, but the models become more complex and will require greater computing to estimate and greater storage to represent. The entropy measurement provided us with a characterization of the complexity of the model and its match to the data. By examining how this changes with the length of the path modeled, we may make practical decisions about the complexity of model to use, given constraints of data collection, storage costs, and computing power.

4.2. Stability of surfing distributions over time

In the analysis of kth-order Markov models of surfing paths, we computed the relevant estimators for our probabilities directly from the data (these were *maximum likelihood estimators*) for a particular day at a particular Web site. To use these models and estimators to predict activity on future days on the Web site requires assumptions that the stochastic processes generating the surfing paths are *stationary* and *ergodic*. Stationary stochastic processes are ones that do not change over time. Ergodic processes are ones in which states recur eventually.

It is rather unlikely that surfing patterns and the WWW fulfill these assumptions. Web sites change and the population of visitors and their interests probably change too. We can build upon the previous kind of analysis to investigate how much change occurs over days. The basic approach is to collect surfing path n-grams from one time period, estimate kth-order Markov models of the probabilities that surfers transition to page n given their k previous page visits, and then see how well the estimated probability distributions match those for future time periods.

There are a number of ways to compute the similarity (or more commonly, the dissimilarity) of two probability distributions. One common approach is the *Kullback–Leibler* formula for relative entropy, which characterizes the mutual information in two probability (mass) distributions. The Kullback–Leibler divergence (dissimilarity) of two distributions, p and q, is $D(p \parallel q)$, where

$$D(p \parallel q) = \sum_i p_i \log_2 \frac{p_i}{q_i}. \quad (9)$$

Comparing equation (9) to equation (5), one should recognize that this is another entropy measure.

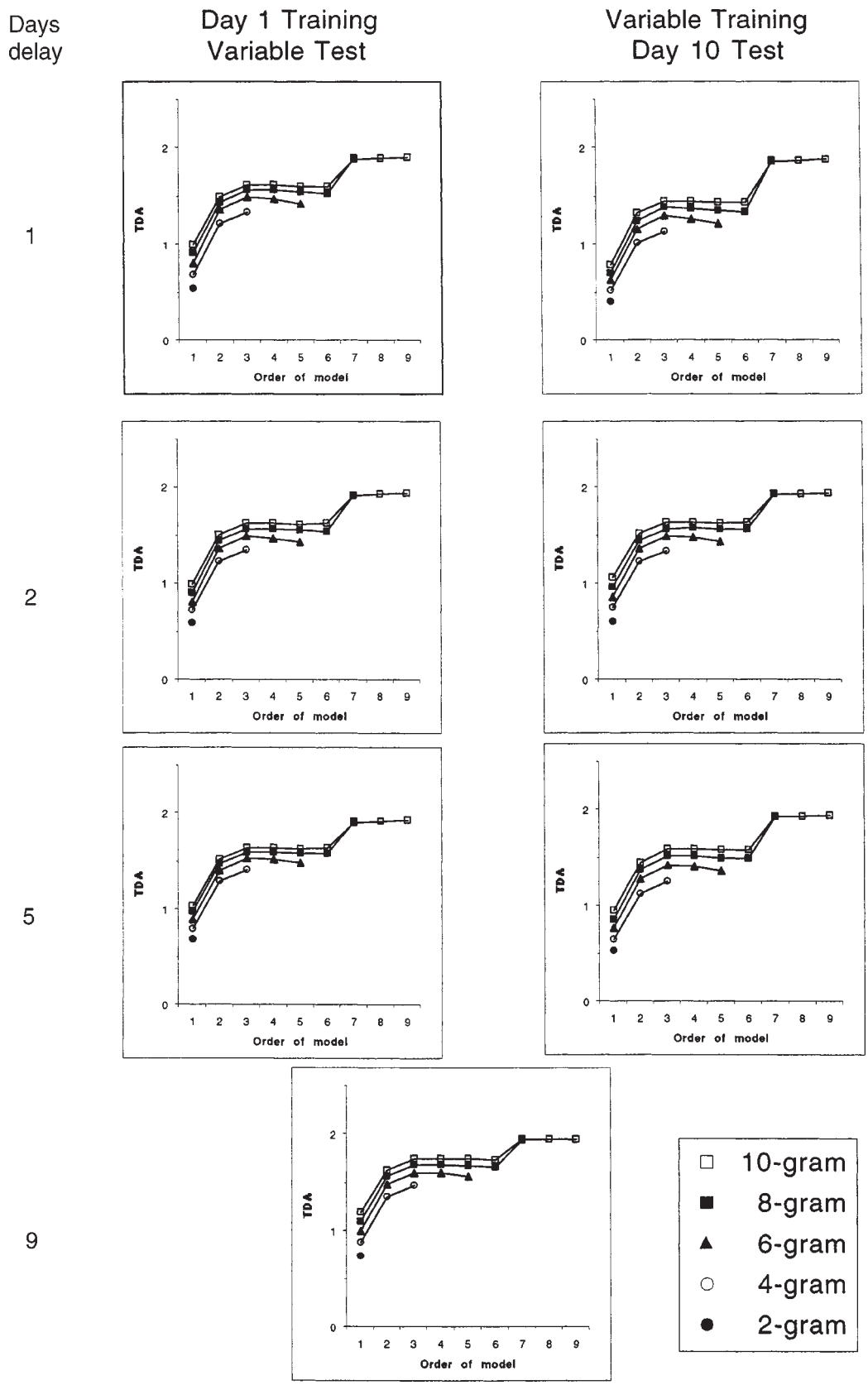

Figure 4. TDA analysis of divergence of training and test estimates of Markov approximations to surfing paths observed at the Xerox.com Web site 5/10/98 through 5/19/98. See text for details.

Unfortunately, application of this formula is problematic when there are zero probabilities for one of the distributions (i.e., q). This occurs often with our surfing data – for instance, pages that are visited on one day may not be visited the next. Instead of the Kullback–Leibler measure, we have used *total divergence to the average* (TDA). The divergence (dissimilarity) of two distributions is

$$\begin{aligned}\mathrm{TDA}(p,q) &= \mathrm{TDA}(q,p)\\ &= D\left(p \,\middle\|\, \frac{p+q}{2}\right) + D\left(q \,\middle\|\, \frac{p+q}{2}\right). \quad (10)\end{aligned}$$

This overcomes the problems of zero probabilities and, unlike Kullback–Lebler divergence, it is symmetric. TDA ranges from zero to $2\log 2$ for maximally different distributions.

We conducted TDA analyses on the set of data collected from Xerox.com for the dates 5/10/98 through 5/19/98. We designated the data sets as Day 1 (5/10/98) through Day 10 (5/19/98). We then estimated the kth-order Markov models on data from one day (the *training* set) and measured its TDA against kth-order Markov models estimated from another day (the *test* set). We did this at several levels of days of delay between the training set and test set: Delay $= 1$, 2, 5, or 10 days of delay (this is an approximately logarithmic sequence of delays). We did this in two ways: (a) by using Day 1 as the training set and various days (Days 2, 3, 6, and 10) as the test sets, and (b) by using Day 10 as the test set and various days (Day 9, 8, 5, and 1) as the training set (note that the Day 1 to Day 10 comparison is redundant).

Again, we stratified the data by surfing n-grams ranging from $n = 2$ to 10, and we examined kth order Markov models for $k = 1$ to 9. Figure 4 presents the TDAs for each of the training-test comparisons. Figure 5 presents the kth-order Markov models over days of delay between training and test. In figure 5, for the sake of clarity, we only present the data for the longest kth-order models estimated for each class of n-gram – that is, for n-grams of length n we present the $k = n - 1$ order model.

From figures 4 and 5 it is apparent that the first-order Markov models show the least divergence between training and test. Overall, the divergence between training and test appears to increase in the same way for all orders of model at all levels of delay with both variable training sets and variable test sets.

The TDAs for the 7th order, 8th order, and 9th order Markov models appear anomalous in comparison to the lower order models. It appears that the dissimilarity across days in these higher-order models is much greater than expected if we just extrapolated from the lower order models.

Interestingly, if we ignore the anomalous data, figure 4 suggests that on the longer n-grams, the worst divergences between training and test occur at the midranges of k. Recall that we are estimating the probability of a visit to page n given knowledge of the immediately prior k page visits of surfers, of $p(x_n \mid x_{n-1}, \ldots, x_{n-k})$. Figure 4 suggests that on longer n-grams, knowledge of longer lengths of prior visits (large k) or shorter lengths of prior visits (small k) shows less divergence over training to test than middle-sized lengths of prior visits (intermediate k). This might indicate that knowledge of the starting visits of surfers and knowledge of the visits made immediately prior to a transition are most stable and important to making accurate estimates of a visit to page n, for longer length n-grams.

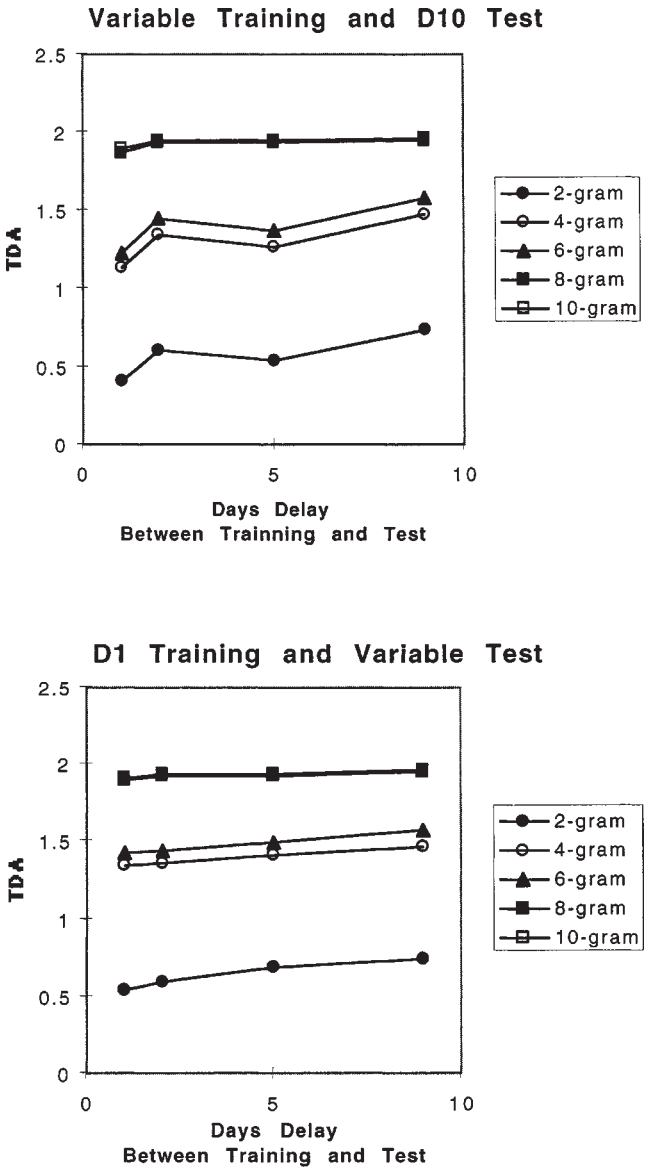

Figure 5. TDA analysis of divergence in distribution of surfing path probabilities over varying delays between training and test estimates. See text for details.

4.3. Predicting future visits using kth-order Markov models of past surfing paths

To provide a more concrete examination of these Markov models of surfing paths we consider a simple prediction scenario. Imagine that we estimate some kth-order Markov models of surfing transitions from training data and we want to use these to predict visits of surfers in the future. Suppose we have just observed a surfer make k page visits. In

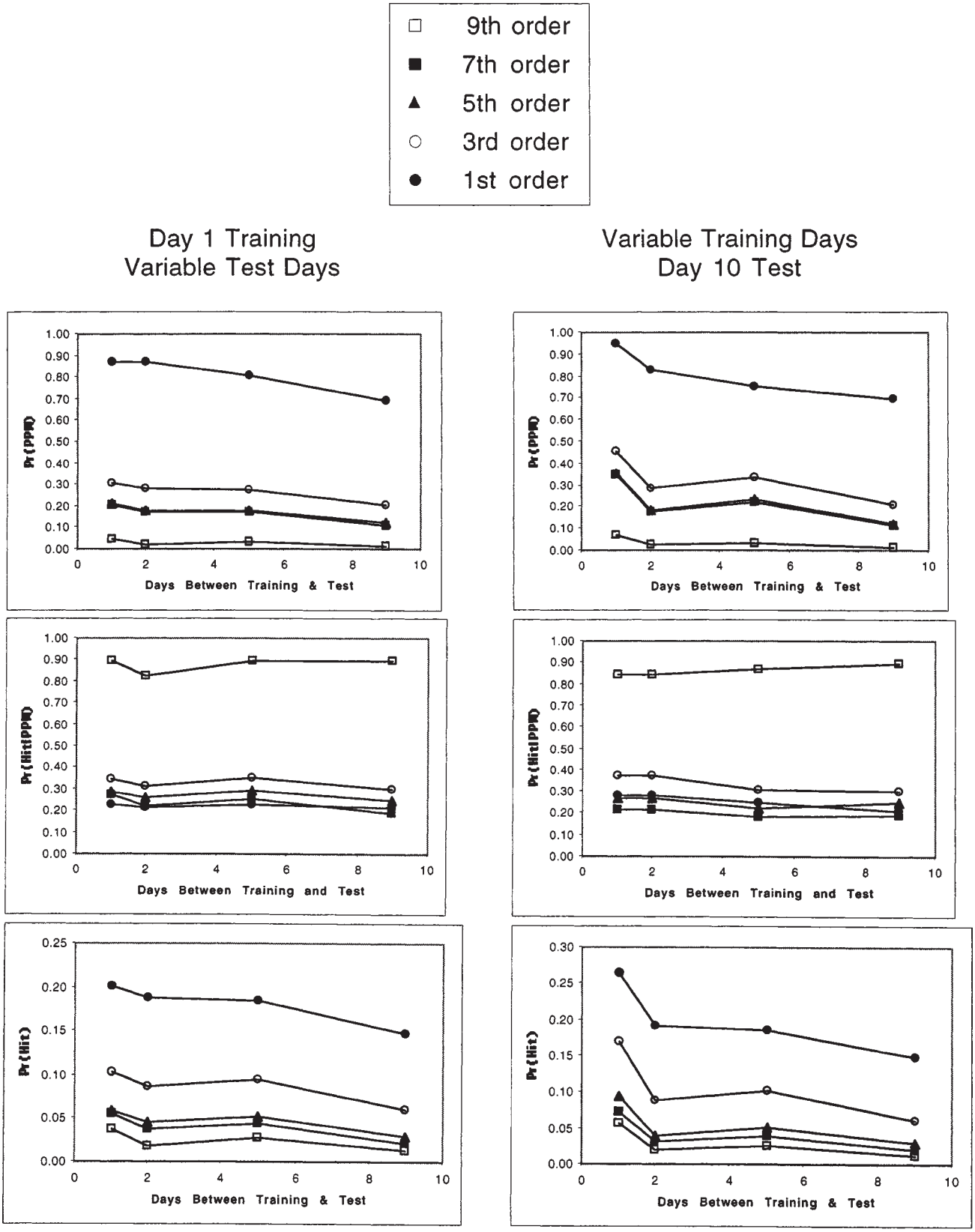

Figure 6. Analysis effects of delay between training and test data sets for a prediction scenario using kth order Markov models. See text for details.

order to make a prediction of the next page visit we want to have an estimate of $p(x_n \mid x_{n-1}, \ldots, x_{n-k})$ from our training data. This will require, however, that this particular path of k visits $\langle x_{n-1}, \ldots, x_{n-k} \rangle$ was observed in the training data. Let us call this sequence $\langle x_{n-1}, \ldots, x_{n-k} \rangle$ a *penultimate path*. Let us call the match of a penultimate path on a test day to the same penultimate path in training data a *penultimate path match* (PPM). Continuing our scenario, if we have a penultimate path match, we examine all the conditional probabilities $p(x_n \mid x_{n-1}, \ldots, x_{n-k})$ available for all pages x_n, and predict that the particular page having the highest conditional probability of occurring next will in fact be visited next. If we then observe that the surfer makes the predicted visit, then we say we have a *hit* (as opposed to a *miss*).

From training data we estimated $p(x_n \mid x_{n-1}, \ldots, x_{n-k})$ from all available n-grams of lengths $n = 1, \ldots, 10$. Against separate test data we estimated the following probabilities of interest:

- Pr(PPM) the probability that a penultimate path, $\langle x_{n-1}, \ldots, x_{n-k} \rangle$, observed in the test data was matched by the same penultimate path in the training data,
- Pr(Hit | PPM) the probability that page x_n is visited, given that $\langle x_{n-1}, \ldots, x_{n-k} \rangle$, is the penultimate path and the highest probability conditional on that path is $p(x_n \mid x_{n-1}, \ldots, x_{n-k})$,
- Pr(Hit) = Pr(Hit | PPM) · Pr(PPM), the probability that the page visited in the test set is the one estimated from the training as the most likely to occur (in accordance with the method in our scenario).

Figure 6 presents Pr(PPM), Pr(Hit | PPM), and Pr(Hit) for various training-test delays. As in our TDA analysis, we used Day 1 as a fixed training set and then tested our estimates at various delays (1, 2, 5, or 9 days), and we used Day 10 as a fixed test set and used various training sets to provide the same delays. All three probabilities drop as one increases the delay between the training set and test set, although the size of these reductions is not great. The size of the reductions generally diminishes with increasing delay. As indicated by the Pr(PPM), as one increases the length of the penultimate path (or equivalently, the order of the model), there is a marked decrease in the probability of finding a matching path in the training and test data sets. This is the major determinant in the superior Pr(Hit) estimates for lower-order models.

Figure 7 shows the improvements in prediction between training and test as a function of increasing the size of the training data set. Notice that the first order model does not improve as much as the 2nd–7th order models. This is because the probability of finding a penultimate path match, Pr(PPM), between training and test data is practically at ceiling for the first-order model with only one day of training data. As shown in figure 8, the gains in predicting hits in figure 7 are largely attributable to gains in finding matching surfing paths across the training and test data.

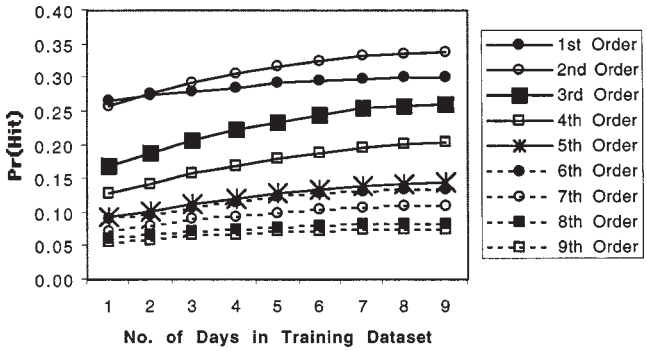

Figure 7. Effects of increasing the number of days of training on predicting visits based on surfing paths. See text for details.

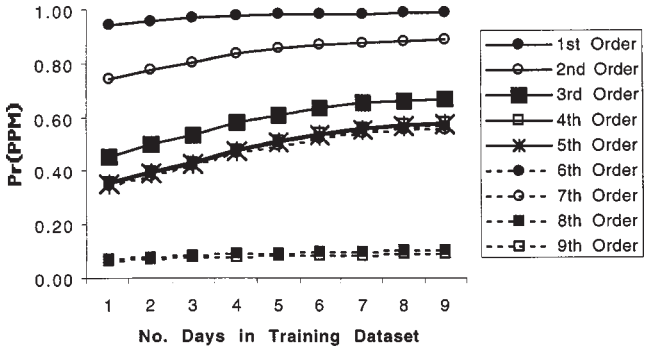

Figure 8. Effects of increasing the number of days of training on the probability of finding matching surfing paths. See text for details.

5. Conclusions

In this paper, we presented a number of studies that investigate various aspects of WWW user surfing paths. Several path reconstruction algorithms were demonstrated to have significant impact on basic characterizations like session times and the number of clicks per session. Levenshtein Distance was used to further understand the differences between the various approaches. While the purpose of these investigations was not to motivate a particular algorithm over another, our findings do reveal that careful choice must be taken when reconstructing user paths through WWW sites.

Researchers are beginning to develop models aimed at predicting the interests of surfers. Sometimes these models are based just on the hyperlink structure of the WWW [Brin and Page 1998] and sometimes they are based on statistics drawn from usage patterns [Huberman *et al.* 1998; Padmanabhan and Mogul 1996]. The assumptions for these models seem to be that surfers will follow WWW structure in similar ways, or that surfers will exhibit the same paths as earlier surfers. We presented a preliminary investigation of such assumptions by using a Markov model representation. Such models are well understood, but have strong (usually testable) assumptions. They provide a good initial basis for exploring the stochastic processes of surfing.

In the context of these models and their assumptions, we used entropy and conditional entropy as a way of measuring the uncertainty in predicting surfer visits, and the reduction

in uncertainty obtained by making our making predictions conditional on longer surfing paths. Measurements of divergence (TDA) provide a way of investigating the stability of surfing path distributions over time. A set of analyses and methods was also presented that began to uncover the impact of various path generation techniques on the overall integrity of paths collected.

Our information-theoretic measurements (entropy and TDA) suggest that information is gained by using longer paths to estimate the conditional probability of link choice given surf path. The improvements diminish, however, as one increases the length of path beyond one. Information-theoretic measurements suggest that the conditional probabilities of link choice given surf path are more stable over time for shorter paths than longer paths. Direct examination of the accuracy of the conditional probability models in predicting test data also suggested that shorter paths yield more stable models and can be estimated reliably with less data than longer paths.

It is important to note that we have used Markov models as a framework for stating empirical characterizations. We are not necessarily advocating their appropriateness as descriptive models of surfing behavior. Like human language, surfing activity may have a deeper structure ("grammar") or intention ("meaning" or "purpose") that can be derived from the simple statistics of surface behavior. We can still, however, use measurements like entropy to characterize the fit and complexity of these deeper models, if and when they are developed.

Acknowledgements

Parts of this research were supported by the Office of Naval Research grant N00014–96-C-0097 to P. Pirolli and S. Card. We would especially like to thank Hinrich Schuetze for his advice on statistics and information theory.

References

Arlitt, M. and C. Williamson (1996), "Web Server Workload Characterization: The Search for Invariants," In *ACM SIGMETRICS Conference*, Philadelphia, PA.

Brin, S. and L. Page (1998), "The Anatomy of a Large-Scale Hypertextual Web Search Engine," *World Wide Web 7*.

Catledge, L.D. and J.E. Pitkow (1995), "Characterizing Browsing Strategies in the World-Wide Web," *Computer Networks and ISDN Systems 26*, 6, 1065–1073.

Cunha, C. and C.F.B. Joccoud (1997), "Determining WWW User's Next Access and Its Application to Pre-Fetching," In *Proceedings of the International Symposium on Computers and Communication*, Alexandria, Egypt.

Huberman, B.A. and L.A. Adamic (1998), *Novelty and Social Search in the World Wide Web*, Xerox PARC, Palo Alto, CA.

Huberman, B.A., P. Pirolli, J. Pitkow, and R. Lukose (1998), "Strong Regularities in World Wide Web Surfing," *Science 280*, 95–97.

Kantor, P.B. (1997), *A Novel Approach to Information Finding in Networked Environments*, Rutgers, Piscataway, NJ.

Kleinberg, J. (1998), "Authoritative Sources in a Hyperlinked Environment," In *Proc. 9th ACM–SIAM Symposium on Discrete Algorithms.*

Levenshtein, V.I. (1966), "Binary Codes Capable of Correcting Deletions, Insertions and Reversals," *Soviet Phys. Dokl. 10*, 8, 707–710.

Manley, S., M. Courage, and M. Seltzer (1997), *A Self-Scaling and Self-Configuring Benchmark for Web Servers*, Harvard College, Boston, MA.

Padmanabhan, V.N. and J.C. Mogul (1996), "Using Predictive Pre-Fetching to Improve World Wide Web Latency," *Comput. Comm. Rev. 26.*

Pirolli, P. and S.K. Card (in press), "Information Foraging," *Psychol. Rev.*

Pirolli, P., J. Pitkow, and R. Rao (1996), "Silk From a Sow's Ear: Extracting Usable Structures From the Web," In *Proc. of Conference on Human Factors in Computing Systems, CHI '96*, Vancouver, Canada.

Pitkow, J.E. (1997), "In Search of Reliable Usage Data on the WWW," In *Proc. of The 6th International World Wide Web Conference*, Santa Clara, CA.

Pitkow, J.E. and C.M. Kehoe (1996), "GVU's 6th WWW User Survey," `http://www.gvu.gatech.edu/user_surveys.`