Chapter 7

Crawling the Infinite Web

We have seen in Chapter **??** several scheduling policies for ordering pages during Web crawling. The objective of these policies is to retrieve "good" pages early in the crawl. We have considered that the Web is bounded, but a large amount of publicly available Web pages are generated dynamically upon request, and contain links to other dynamically generated pages. This usually produces Web sites which can create arbitrarily many pages.

This poses a problem to Web crawling, as it must be done such a way that it stops downloading pages from each Web site at some point. But how deep must the crawler go?

In this chapter:

- We propose models for random surfing inside a Web site when the number of pages is **unbounded**. For that, we take the tree induced by the Web graph of a site, and study it by levels.
- We analyze these models, focusing in the question of how "deep" users go inside a Web site.
- We validate these models using actual data from Web sites, as well as using a link analysis measure such as Pagerank.

The next section outlines the motivation of this work, namely, the existence of dynamic pages. In Section 7.2 three models of random surfing in dynamic websites are presented and analyzed; in Section 7.3, these models are compared with actual data from the access log of several Web sites. The last section concludes with some final remarks and recommendations for practical web crawler implementations.

The results presented here were obtained in a joint work with Ricardo Baeza-Yates [BYC04].

7.1 Static and dynamic pages

Most studies about the web refer only to the "publicly indexable portion" [LG98]; excluding a portion of the web that has been called "the hidden web" [RGM01] or the "deep web" [Ber01, GA04]. The non-indexable portion is characterized as all the pages that normal users could eventually access, but automated agents such as the crawlers used by search engines cannot.

Certain pages are not indexable because they require previous registration or some special authorization such as a password, or are only available when visited from within a certain network, such as a corporate intranet. Others are **dynamic pages**, generated after the request has been made. Some times they are not indexable

because they require certain parameters as input, e.g. query terms, and those query terms are unknown at crawling time. The different portions of the Web are depicted in Figure 7.1.

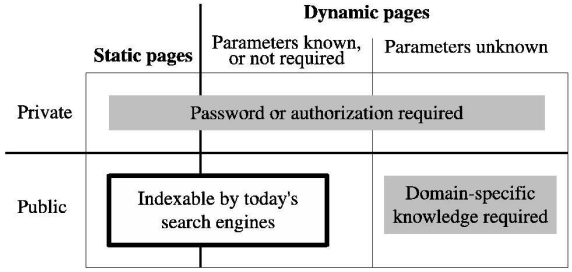

Figure 7.1: The web can be divided into password-protected and publicly available, and into dynamic and static pages.

However, many dynamic pages are indexable, as the parameters for creating them can be found by following links. This is the case of, e.g. typical product catalogs in Web stores, in which there are links to navigate the catalog without the user having to pose a query.

The amount of information in the Web is certainly finite, but when a dynamic page leads to another dynamic page, *the number of pages can be potentially infinite*. Take, for example, a dynamic page which implements a calendar; you can always click on "'next month" and from some point over there will be no more data items in the calendar; we can be reasonably sure that it is very unlikely to find events scheduled 50 years in advance, but a crawler cannot. A second example would be a calculator, such as a dynamic page that calculates approximations of π using an iterative method. A crawler cannot tell when two pages reflect the same information. There are many more examples of "crawler traps" that involve loops and/or near-duplicates (which can be detected afterwards, but we want to avoid downloading them).

Also, personalization is a source of a large number of pages; if you go to `www.amazon.com` and start browsing your favorite books, soon you will be presented with more items about the same topics and automatically generated lists of recommendations, as the Web site assembles a vector of preferences of the visitor. The visitor is, in fact, creating web pages as it clicks on links, and an automated agent such as a web crawler generates the same effect[1].

The web of dynamically generated content is crawled superficially by many web crawlers, in some cases because the crawler cannot tell a dynamic URL from a static one, and in other cases purposefully. However, few crawlers will go deeper, unless they know when to stop and how to handle dynamic pages with links to more dynamic pages. In our previous experiences with the WIRE crawler [BYC02], we usually limit the depth at which pages are explored, typically to 5 links in static pages and 15 links in dynamic pages. When we plot the number of pages at a given depth, a profile as the one shown in Figure 7.2 is obtained.

Notice that here we are not using the number of slashes in the URL, but using the real distance in links with the start page(s) of the website. The dynamic pages grow with depth, while the static pages follow a different shape, with the maximum number of pages found around 2 – 3 links depth; this is why some search engines use the heuristic of following links to URLs that seems to hold dynamically generated content only from pages with static content. This heuristic is valid while the amount of information in static pages continues to be large, but that will not be the case in the near future, as large Web sites with only static pages are very hard to maintain.

In this work, we deal with the problem of capturing a relevant portion of the *dynamically generated content*

[1]This is a case of uncertainty, in which the instrument, the Web crawler, affects the object it is attempting to measure.

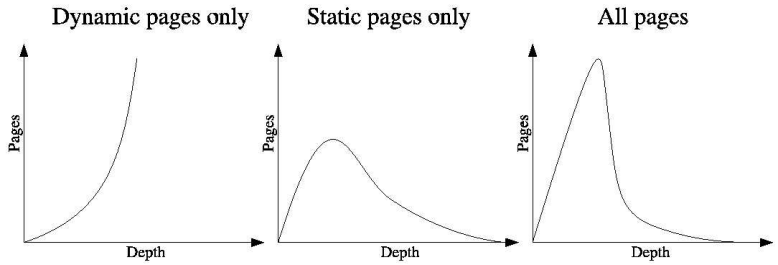

Figure 7.2: Amount of static and dynamic pages at a given depth.

with known parameters, while avoiding the download of too many pages. We are interested in knowing if a user will ever see a dynamically generated page. If the probability is too low, would a search engine like to retrieve that page? Clearly, from the Web site point of view the answer is yes, but perhaps from the search engine's point of view, the answer is no. In that case, our results are even more relevant.

7.2 Random Surfer Models for a Website with Infinite Pages

We will consider a Web site $S = (Pages, Links)$ as a set of pages under the same host name, with $Pages = \{P_1, P_2, ...\}$ and $Links$ such as $(P_i, P_j) \in Links$ iff there exists a hyper link from page P_i to page P_j in the Web site.

Definition (User session) We define a user session $\mathbf{U}$ as a finite sequence of page views $\mathbf{U} = (P_1, ...P_n)$, with $P_i \in Pages$, and $(P_i, P_{i+1}) \in Links$. The first request U_0 does not need to be the start page located at the root directory of the server, as some users may enter to website following a link to an internal page, e.g.: if they come from a search engine.

Definition (Page depth) We define for a page P_i and a session $\mathbf{U}$ the depth $depth(P_i, \mathbf{U})$ as:

$$depth(P_i, \mathbf{U}) = \begin{cases} 0 & \text{if } P_i = U_0 \\ \min depth(P_j, \mathbf{U}) + 1 & P_j \in \mathbf{U}, j < i, (P_j, P_i) \in Links \end{cases} \tag{7.1}$$

The depth is basically the length of the shortest path from the start page through the pages actually seen during a session. Note that the depth of a page is not only a function of the Web site structure, it is the *perceived* depth during each session $\mathbf{U}$.

Definition (Session depth) We define the depth of session $\mathbf{U}$ as $\max depth(P_i, U)$ with $P_i \in U$. We are interested in this variable as its distribution is relevant from the point of view of search engines.

For random surfing, we can model each page in *Pages* as a state in a system, and each hyperlink in *Links* as a possible transition. This kind of model has been studied by Huberman *et al.* [HPPL98, AH00]. We propose to use a related model in which we collapse multiple pages at the same level as a single node, as shown in Figure 7.3. That is, the Web site graph is collapsed to a sequential list.

At each step of the walk, the surfer can perform one of the following atomic actions: go to the next level (action *next*), go back to the previous level (action *back*), stay in the same level (action *stay*), go to a different

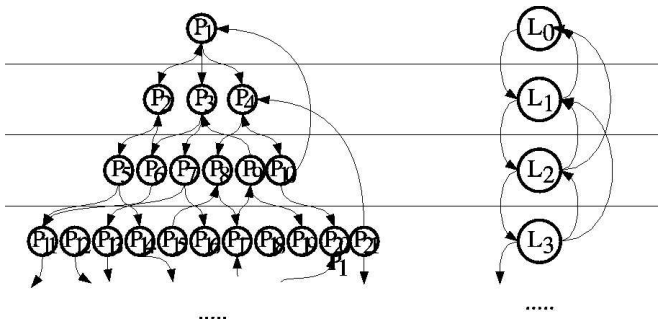

Figure 7.3: A website modeled as a tree (left) and as a sequence of levels (right).

previous level (action *prev*), go to a different higher level (action fwd), go to the start page (action *start*) or jump outside the Web site (action *jump*).

For action *jump* we add an extra node `EXIT` to signal the end of a user session (closing the browser, or going to a different Web site) as shown in Figure 7.4. Regarding this Web site, after leaving users have only one option: start again in a page with depth 0 (action *start*).

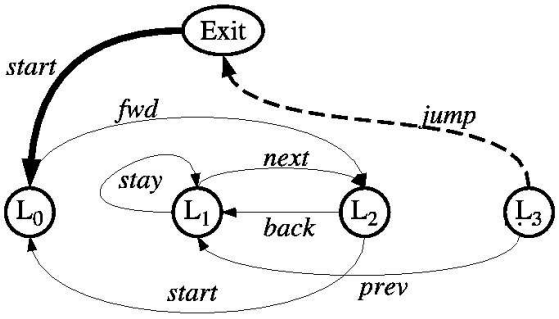

Figure 7.4: Representation of the different actions of the random surfer. The EXIT node represents leaving the Web site, and the transition between that node and the start level has probability 1.

As this node `EXIT` has a single out-going link with probability 1.0, it does not affect the results for the other nodes if we remove the node `EXIT` and change this by transitions going to the start level L_0. Another way to understand it is that as this Markovian process has no memory, *going back to the start page or starting a new session are equivalent*, so actions *jump* and *start* are indistinguishable in terms of the resulting probability distribution for the other nodes. As a response to the same issue, Levene *et al.* [LBL01] proposed to use an absorbing state representing leaving the Web site; but we want to avoid absorbing states to be able to calculate and compare stationary probability distributions.

The set of atomic actions is $\mathcal{A} = \{next, start/jump, back, stay, prev, fwd\}$ and the probabilities if the user is currently at level ℓ, are:

- $x(next|\ell)$: probability of advancing to the level $\ell+1$.
- $x(back|\ell)$: probability of going back to the level $\ell-1$.
- $x(stay|\ell)$: probability of staying at the same level ℓ.
- $x(start/jump|\ell)$: probability of going to the start page of this session, when it is not the previous two cases; this is equivalent in our model to begin a new session,

- $x(prev|\ell)$: probability of going to another previous level, which is neither start nor the immediate preceding level.
- $x(fwd|\ell)$: probability of going to another following level, which is not the next level.

As they are probabilities, $\sum_{action \in \mathcal{A}} x(action|\ell) = 1$. The probability distribution at a given time is the vector $\mathbf{x}(t)$. When there exists a limit, we will call this $\lim_{t\to\infty} \mathbf{x}(t) = \mathbf{x}$.

In this article, we study three models with $x(next|\ell) = q\ \forall \ell$, i.e.: the probability of advancing to the next level is constant for all levels. Our purpose is to predict how far will a real user go into a dynamically generated website. If we know that, e.g.: $x_0 + x_1 + x_2 \geq 90\%$, then the crawler could decide to crawl just those three levels.

The models we analyze were chosen to be as simple and intuitive as possible, without sacrificing correctness. We seek more than just fitting the distribution of user clicks, we want to *understand and explain user behavior in terms of simple operations*.

7.2.1 Model A: back one level at a time

In this model, with probability q the user will advance deeper, and with probability $1-q$ the user will go back one level, as shown in Figure 7.5.

$x(next|\ell) = q$
$x(back|\ell) = 1-q$ for $\ell \geq 1$
$x(stay|\ell) = 1-q$ for $\ell = 0$
$x(start, jump|\ell) = 0$
$x(prev|\ell) = x(fwd|\ell) = 0$

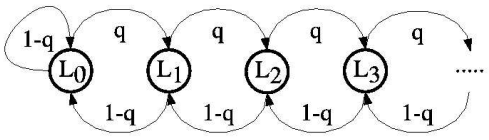

Figure 7.5: Model A, the user can go forward or backward one level at a time.

A stable state $\mathbf{x}$ is characterized by:

$$\begin{aligned} x_i &= q x_{i-1} + (1-q) x_{i+1} \quad (\forall i \geq 1) \\ x_0 &= (1-q) x_0 + (1-q) x_1 \end{aligned}$$

and $\sum_{i\geq 0} x_i = 1$.

The solution to this recurrence is: $x_i = x_0 \left(\frac{q}{1-q}\right)^i \quad (\forall i \geq 1)$.

If $q \geq 1/2$ then we we have that the solution is $x_i = 0\ \forall i$, and $x_{\text{inf}} = 1$ (that is, we have an absorbing state); which in our framework means that no depth can ensure a certain proportion of page views visited by the user. When $q < 1/2$ and we impose the normalization constraint, we have a geometric distribution:

$$x_i = \left(\frac{1-2q}{1-q}\right)\left(\frac{q}{1-q}\right)^i$$

The cumulative probability of levels $0..k$ is:

$$\sum_{i=0}^{k} x_i = 1 - \left(\frac{q}{1-q}\right)^{k+1}$$

7.2.2 Model B: back to the first level

In this model, the user will go back to the start page of the session with probability $1-q$. This is shown in Figure 7.6.

- $x(next|\ell) = q$
- $x(back|\ell) = 1-q$ if $\ell = 1$, 0 otherwise.
- $x(stay|\ell) = 1-q$ for $\ell = 0$
- $x(start, jump|\ell) = 1-q$ for $\ell \geq 2$
- $x(prev|\ell) = x(fwd|\ell) = 0$.

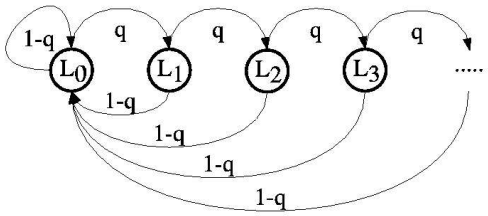

Figure 7.6: Model B, the user can go forward one level at a time, or he can go back to the first level either by going to the start page, or by starting a new session.

A stable state **x** is characterized by:

$$\begin{aligned} x_0 &= (1-q)\sum_{i\geq 0} x_i = (1-q) \\ x_i &= qx_{i-1} \quad (\forall i \geq 1) \end{aligned}$$

and $\sum_{i\geq 0} x_i = 1$.

As we have $q < 1$ we have another geometric distribution: $x_i = (1-q)q^i$.

The cumulative probability of levels $0..k$ is: $\sum_{i=0}^{k} x_i = 1-q^{k+1}$.

Note that the cumulative distribution obtained with model A ("back one level") using parameter q_A, and model B ("back to home") using parameter q_B are equivalent if: $q_A = \left(\frac{q_B}{1+q_B}\right)$.

So, as the distribution of session depths is equal, except for a transformation in the parameter q, we will consider only model B for charting and fitting the distributions.

7.2.3 Model C: back to any previous level

In this model, the user can either discover a new level with probability q, or go back to a previous visited level with probability $1-q$. If it decides to go back to a previously seen level, will choose uniformly from the set of visited levels (including the current one), as shown in the Figure 7.7.

- $x(next|\ell) = q$
- $x(back|\ell) = 1-q/(\ell+1)$ for $\ell \geq 1$
- $x(stay|\ell) = 1-q/(\ell+1)$
- $x(start, jump|\ell) = 1-q/(\ell+1)$ for $\ell \geq 2$
- $x(prev|\ell) = 1-q/(\ell+1)$ for $\ell \geq 3$
- $x(fwd|\ell) = 0$.

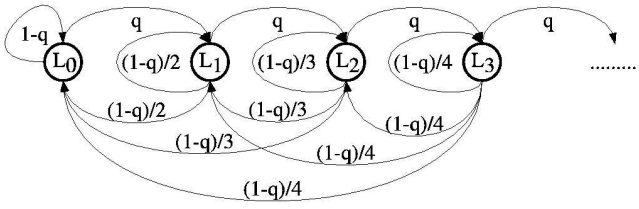

Figure 7.7: Model C: the user can go forward one level at a time, and can go back to previous levels with uniform probability.

A stable state **x** is characterized by:

$$\begin{aligned} x_0 &= (1-q)\sum_{k\geq 0}\frac{x_k}{k+1} \\ x_i &= qx_{i-1}+(1-q)\sum_{k\geq i}\frac{x_k}{k+1} \qquad (\forall i>1) \end{aligned}$$

and $\sum_{i\geq 0} x_i = 1$.

We can take a solution of the form: $x_i = x_0\,(i+1)\,q^i$.

Imposing the normalization constraint, this yields: $x_i = (1-q)^2\,(i+1)\,q^i$.

The cumulative probability of levels $0..k$ is: $\sum_{i=0}^{k} x_i = 1-(2+k-(k+1)\,q)\,q^{k+1}$.

7.2.4 Comparison of the Models

In terms of the cumulative probability of visiting the different levels, models A and B produce equivalent results except for a transformation of the parameters. Plotting the cumulative distributions for models B and C yields Figure 7.8.

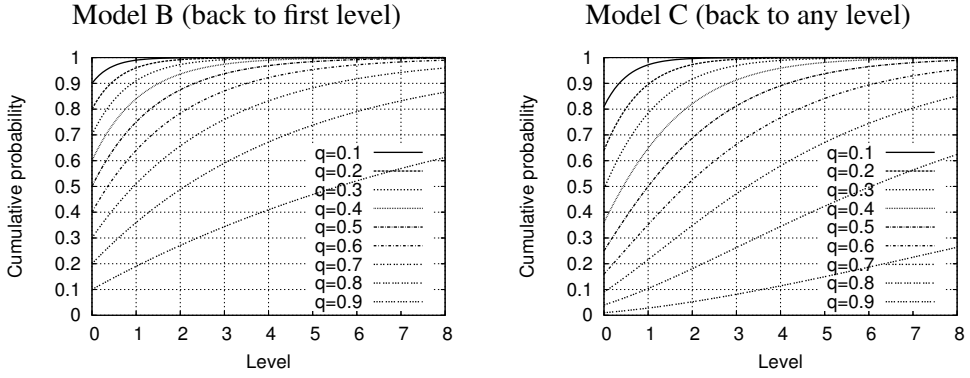

Figure 7.8: Cumulative probabilities for models B and C, for varying values of q.

We can see that if $q \leq 0.4$, then in these models there is no need for the crawler to go past depth 3 or 4 to capture more than 90% of the pages a random surfer will actually visit, and if q is larger, say, 0.6, then the crawler must go to depth 6 or 7 to capture this amount of page views.

7.3 Data from user sessions in Web sites

We studied real user sessions on 13 different Web sites in the US, Spain, Italy and Chile, including commercial, educational, non-governmental organizations and Web logs (collaborative Web sites in which the forums are the largest part, also known as "Blogs"); characteristics of this sample, as well as the results of fitting models B and C to the data, are summarized in 7.1.

We obtained access logs with anonymous IP addresses from these Web sites, and processed them to obtain user sessions using the following procedure:

- Sort the logs by IP address of the client, then by access time stamp.

Collection					Fit		
Code	Type	Country	Recorded sessions	Average page views	Best model	q	Error
E1	Educational	Chile	5,500	2.26	B	0.51	0.88%
E2	Educational	Spain	3,600	2.82	B	0.51	2.29%
E3	Educational	US	71,300	3.10	B	0.64	0.72%
C1	Commercial	Chile	12,500	2.85	B	0.55	0.39%
C2	Commercial	Chile	9,600	2.09	B	0.62	5.17%
R1	Reference	Chile	36,700	2.08	B	0.54	2.96%
R2	Reference	Chile	14,000	2.72	B	0.59	2.75%
O1	Organization	Italy	10,700	2.93	C	0.35	2.27%
O2	Organization	US	4,500	2.50	B	0.62	2.31%
OB1	Organization + Blog	Chile	10,000	3.73	B	0.65	2.07%
OB2	Organization + Blog	Chile	2,000	5.58	B	0.72	0.35%
B1	Blog	Chile	1,800	9.72	C	0.79	0.88%
B3	Blog	Chile	3,800	10.39	C	0.63	1.01%

Table 7.1: Characteristics of the studied Web sites and results of fitting models B and C. The number of user sessions does not reflect the relative traffic of the Web sites, as the data was obtained in different time periods. The average number of page views per session is larger in Blogs.

- Consider only `GET` requests for static and dynamic HTML pages or documents such as Word, PDF or Postscript.
- Consider that a session expires after 30 minutes of inactivity, as this is common in log file analysis software, and is based on empirical data [CP95].
- Consider that a session expires if the `User-Agent` changes [CMS99].
- Consider multiple consecutive hits to the same page (page reload) as a single page view.
- In pages with frames, consider all the frames as a single page.
- Ignore hits to Web applications such as e-mail or content management systems, as they neither respond to a logic of page browsing, nor are usually accessible by Web crawlers.
- Expand a session with missing pages (e.g.: if the user clicks "back" in his browser, and then follow a link). This information is obtained from the `Referrer` field, and is a way of partially overcoming the issue of caching.

Additionally, manual inspection of the data led to the following heuristics to discard automated agents:

- Identify robots by their accesses to the `/robots.txt` file, as suggested in [TK01].
- Identify robots by known `User-Agent` fields.
- Ignore malicious hits searching for exploits, which are usually a sequence of requests searching for buffer overflows or other software bugs. These requests are usually done by automated agents like Nessus [Der04].

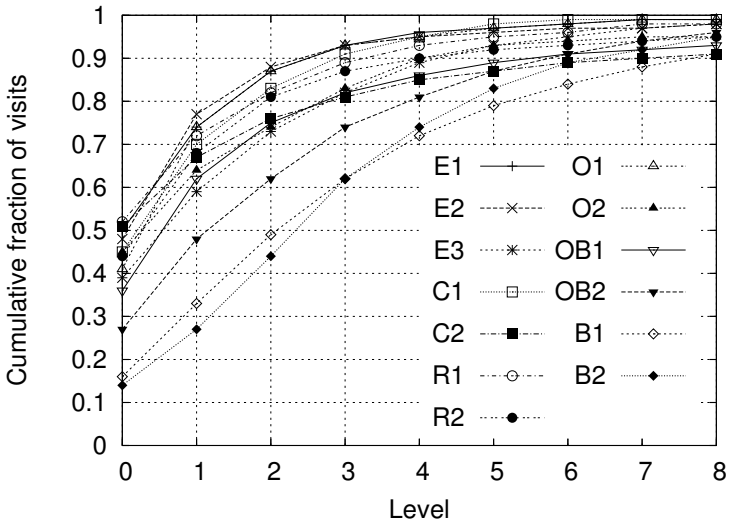

Figure 7.9: Distribution of visits per level, from access logs of Web sites. E=educational, C=commercial, O=non-governmental organization, OB=Organization with on-line forum, B=Blog (Web log or on-line forum).

As re-visits are not always recorded because of caching [TG97], data from log files *overestimates the depth at which users spent most of the time*. Figure 7.9 shows the cumulative distribution of visits per page depth to Web sites. Three observations arise:

- At least 80%-95% of the visits occur at depth ≤ 4.
- About 50% of the sessions include only the start page.
- The average session length is 2 to 3 pages.
- In the case of Web logs sessions tend to be longer. This is reasonable as Web postings are very short, so a user reads several of them during a session.

We fitted the models to the Web sites, as shown in Table 7.1 and Figure 7.11. In general, the curves produced by model B (and model A) are a better approximation to the user sessions than the distribution produced by model C, except for Blogs. The approximation is good for characterizing session depth, with error in general lower than 5%.

We also studied the empirical values for the distribution of the different actions at different levels in the Web site. We averaged this distribution across all the studied Web sites at different depths. The results are shown in Table 7.2, in which we consider all the Websites except for Blogs.

Our observations regarding Table 7.2 are:

- The actions *next*, *jump* and *back* are the more important ones, which is evidence in favor of models A (back one level) and model B (back to start level).
- $x(next|\ell)$ doesn't vary too much, and lies between 0.45 and 0.6. It increases as ℓ grows which is reasonable as a user that already have seen a several pages is more likely to follow a link.

Level	Observations	Next	Start	Jump	Back	Stay	Prev	Fwd
0	247985	**0.457**	–	**0.527**	–	0.008	–	0.000
1	120482	**0.459**	–	**0.332**	**0.185**	0.017	–	0.000
2	70911	**0.462**	**0.111**	**0.235**	**0.171**	0.014	–	0.001
3	42311	**0.497**	0.065	**0.186**	**0.159**	0.017	0.069	0.001
4	27129	**0.514**	0.057	**0.157**	**0.171**	0.009	0.088	0.002
5	17544	**0.549**	0.048	**0.138**	**0.143**	0.009	**0.108**	0.002
6	10296	**0.555**	0.037	**0.133**	**0.155**	0.009	**0.106**	0.002
7	6326	**0.596**	0.033	**0.135**	**0.113**	0.006	**0.113**	0.002
8	4200	**0.637**	0.024	**0.104**	**0.127**	0.006	0.096	0.002
9	2782	**0.663**	0.015	**0.108**	**0.113**	0.006	0.089	0.002
10	2089	**0.662**	0.037	0.084	**0.120**	0.005	0.086	0.003

Table 7.2: Average distribution of the different actions in user sessions of the studied Web sites, except for Blogs. Transitions with values greater than 0.1 are shown in boldface.

- $x(jump|\ell)$ is higher than $x(back|\ell)$ for the first levels.
- $x(jump|\ell)$ is much higher than $x(start|\ell)$, and about half of the user sessions only involve one page from the Web site.
- $x(start|\ell)$, $x(stay|\ell)$ and $x(fwd|\ell)$ are not very common actions.

7.4 Conclusions

The models and the empirical data presented lead us to the following characterization of user sessions: they can be modeled as a random surfer that either advances one level with probability q, or leaves the Website with probability $1-q$. In general $q \approx 0.45-0.55$ for the first few levels, and then $q \approx 0.65-0.70$. This simplified model is good enough for representing the data for Websites, but:

- We could also consider Model A (back one level at a time), which is equivalent in terms of cumulative probability per level, except for a change in the parameters. Based on the empirical data, we observe that users at first just leave the Web site while browsing (Model B), but after several clicks, they are more likely to go back one level (Model A).
- A more complex model could be derived from empirical data, particularly one that considers that q depends on ℓ. We considered that for our purposes, which are related to Web crawling, the simple model is good enough.
- There is a better model (Model C) presented in this article which appears to be better for Blogs. A similar study to this one, focused only in the access logs of Blogs seems a reasonable thing to do since Blogs represent a growing portion of on-line pages.

In all cases, the models and the data shown evidence a distribution of visits which is strongly biased to the first few levels of the Web site. According to this distribution, more than 90% of the visits are closer than 4 to 5 clicks away from the start page in most of the Web sites. The presence of Web logs or Blogs produces user sessions that go deeper into the Web site, with 90% of the visits 7 to 8 clicks away from the start page.

Link analysis, specifically Pagerank, provides more evidence for our conclusions. That is, what fraction of the Pagerank is captured by the pages on the first ℓ levels of the Web sites? To answer this we crawled a large portion of the Chilean Web (.cl) obtaining around 3 million pages on April of 2004, using 150 thousand seed pages that found 53 thousand Web sites. Figure 7.10 shows the cumulative Pagerank for this sample. Again, the first five levels capture 80% of the best pages. Note that the levels here are obtained in terms of the global Web structure, considering internal and external links, not user sessions. The results are consistent with the findings by Najork and Wiener [NW01].

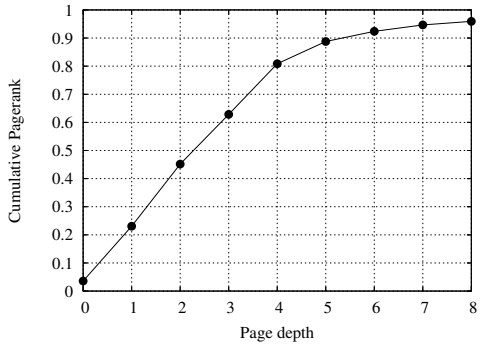

Figure 7.10: Cumulative Pagerank by page levels.

These models and observations could be used by a search engine, and we expect to do future work in this area. For instance, if the search engine's crawler is using a breadth-first crawling and can measure the ratio of new URLs from a Web site it is adding to its queue vs. seen URLs, then it should be able to infer how deep to crawl that specific Web site. The work we presented in this article provides a framework for that kind of adaptivity.

An interesting enhancement of the models shown here is to consider the contents of the pages to detect duplicates and near-duplicates. In our model, downloading a duplicate page should be equivalent to going back to the level at which we visited that page for the first time. A more detailed analysis could also consider the distribution of terms in Web pages and link text as the user browses through a Web site.

As the amount of on-line content that people, organizations and business are willing to publish grows, more Web sites will be built using Web pages that are dynamically generated, so those pages cannot be ignored by search engines. Our aim is to generate guidelines to crawl these new, practically infinite, Web sites.

Bibliography

[AH00] Eytan Adar and Bernardo A. Huberman. The economics of web surfing. In *Poster Proceedings of the Ninth Conference on World Wide Web*, Amsterdam, Netherlands, May 2000.

[Ber01] Michael K. Bergman. The deep web: Surfacing hidden value. *Journal of Electronic Publishing*, 7(1), 2001.

[BYC02] Ricardo Baeza-Yates and Carlos Castillo. Balancing volume, quality and freshness in web crawling. In *Soft Computing Systems - Design, Management and Applications*, pages 565–572. IOS Press, 2002.

[BYC04] Ricardo Baeza-Yates and Carlos Castillo. Crawling the infinite Web: five levels are enough. In *Proceedings of the third Workshop on Web Graphs (WAW)*, Rome, Italy, October 2004. Springer LNCS. (To appear).

[CMS99] Robert Cooley, Bamshad Mobasher, and Jaideep Srivastava. Data preparation for mining world wide web browsing patterns. *Knowledge and Information Systems*, 1(1):5–32, 1999.

[CP95] L. Catledge and J. Pitkow. Characterizing browsing behaviors on the world wide web. *Computer Networks and ISDN Systems*, 6(27), 1995.

[Der04] Renaud Deraison. Nessus: remote security scanner. http://www.nessus.org/, 2004.

[GA04] Thanaa M. Ghanem and Walid G. Aref. Databases deepen the web. *Computer*, 37(1):116 – 117, 2004.

[HPPL98] Bernardo A. Huberman, Peter L. T. Pirolli, James E. Pitkow, and Rajan M. Lukose. Strong regularities in world wide web surfing. *Science*, 280(5360):95–97, April 1998.

[LBL01] Mark Levene, Jose Borges, and George Loizou. Zipf's law for web surfers. *Knowledge and Information Systems*, 3(1):120–129, 2001.

[LG98] Steve Lawrence and C. Lee Giles. Searching the World Wide Web. *Science*, 280(5360):98–100, 1998.

[NW01] Marc Najork and Janet L. Wiener. Breadth-first crawling yields high-quality pages. In *Proceedings of the Tenth Conference on World Wide Web*, pages 114–118, Hong Kong, May 2001. Elsevier.

[RGM01] Sriram Raghavan and Hector Garcia-Molina. Crawling the hidden web. In *Proceedings of the Twenty-seventh International Conference on Very Large Databases*, 2001.

[TG97] Linda Tauscher and Saul Greenberg. Revisitation patterns in world wide web navigation. In *Proceedings of the Conference on Human Factors in Computing Systems CHI'97*, 1997.

[TK01] P.N. Tan and V. Kumar. Discovery of web robots session based on their navigational patterns. *Data Mining and Knowledge discovery*, 2001.

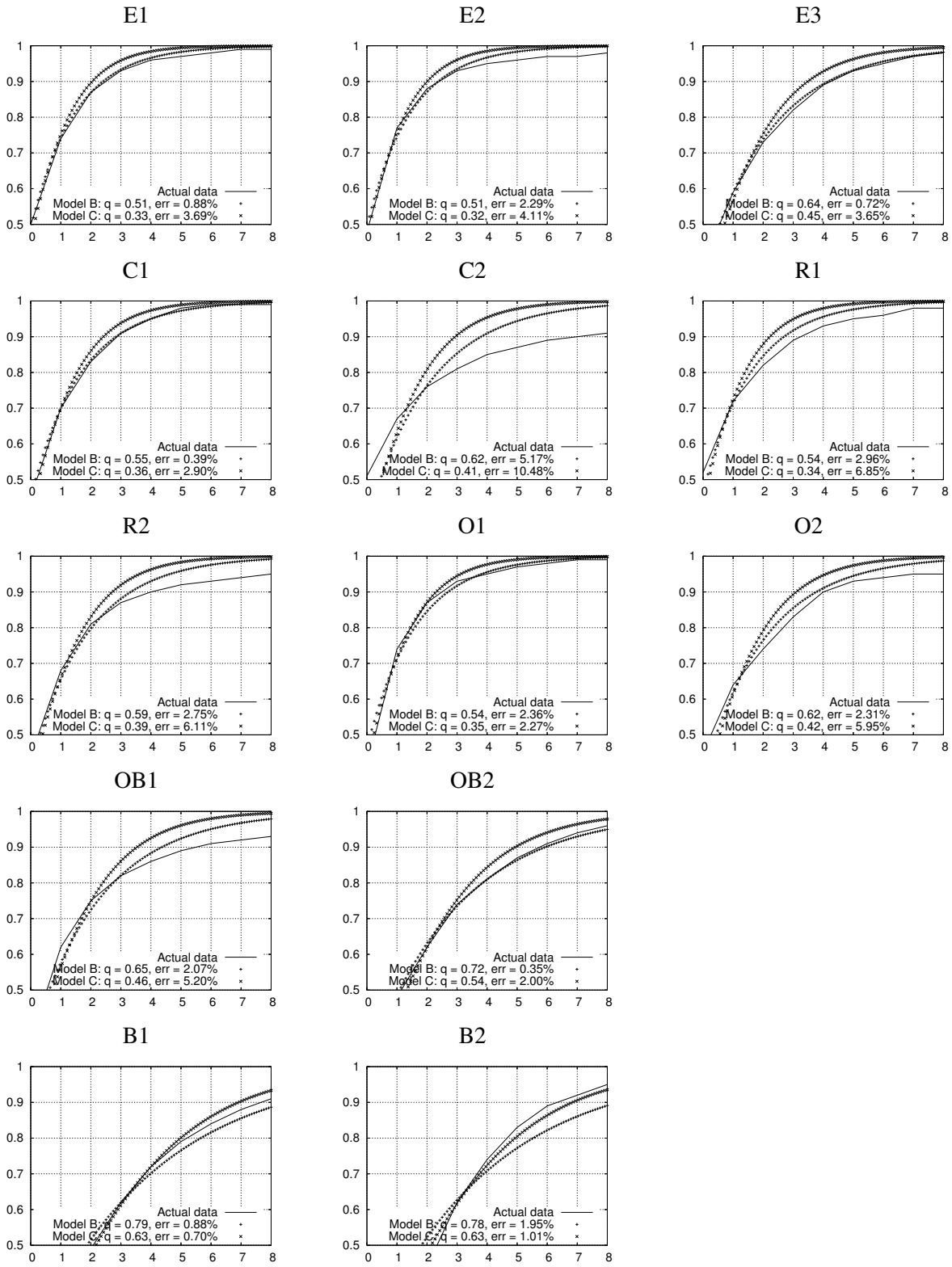

Figure 7.11: Fit of the models to actual data. Model B (back to start level), has smaller errors for most Websites, except for Blogs and Web site O1. The asymptotic standard error for the fit of this model is 5% in the worst case, and consistently less than 3% for all the other cases. Note that we have zoomed in into the upper portion of the graph, starting in 50% of cumulative page views.