Chapter 5

Crawling the Infinite Web

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

This poses a problem to Web crawling, as it must be done in 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 on 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 [PBMW98].

The next section outlines the motivation of this work, namely, the existence of dynamic pages. In Section 5.2, three models of random surfing in dynamic Web sites are presented and analyzed; in Section 5.3, 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.

Portions of this chapter were presented in [BYC04].

5.1 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 can not.

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 5.1.

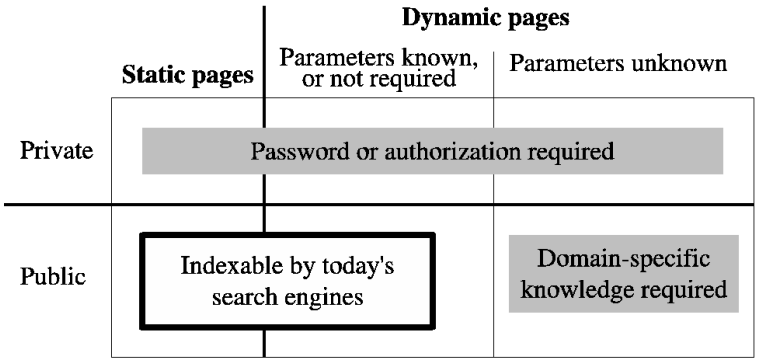

Figure 5.1: The Web can be divided into password-protected and publicly available, and into dynamic and static pages.

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 Web is usually considered as a collection of pages, in the same sense as in traditional Information Retrieval collections, but much larger. Under this assumption, the Web graph has a finite number of nodes in which measures such as diameter are well defined. This is fundamentally wrong. The amount of information in the Web at any given time is certainly finite, but when a dynamic page leads to another dynamic page, *the number of pages can be potentially infinite*. Take for instance a dynamic page that implements a calendar, you can always click on "next month" and from some point on there will be no more data items in the calendar; humans can be reasonably sure that it is very unlikely to find events scheduled 50 years in advance, but a crawler can not. 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 that 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. This is a case of uncertainty, in which the instrument, the Web crawler, affects the object it is attempting to measure.

This poses a problem to Web crawling, as it must be done in such a way that it stops downloading pages from each Web site at some point. Most researchers usually take one of the following approaches to this:

Download only static pages A common heuristic to do so is to avoid downloading URLs containing a question mark, but this heuristic can fail as there are many URLs which are dynamically generated but do not use the CGI standard, encoding the parameters in the rest of the URL. Also, a valuable fraction of the publicly available Web pages is generated dynamically upon request, and it is not clear why those pages should be penalized in favor of static pages.

Download dynamic pages only with one set of parameters When doing this, dynamic pages are either downloaded with the set of parameters of the first time they are found, or with an empty set of parameters. The obvious drawback is that dynamic pages could query a database and a single set of parameters cannot represent the contents of the database.

Download up to a maximum amount of pages This creates a data set that is highly dependent on the crawling strategy. Moreover, this cannot be used to compare, for instance, the amount of information on different domains.

Download up to a certain amount of pages per domain name As a small sum has to be paid for registering a domain name, there is a certain effort involved in creating a Web site under a domain name. However, there are certain domain names such as ".co.uk" which are very large and might require special rules.

Download up to a certain amount of levels per Web site Starting from the home page of each Web site, follow links up to a certain depth. This is the approach we consider in this paper, and the natural question is: how deep must the crawler go?

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 dynamic pages and 15 links in static pages. When we plot the number of pages at a given depth, a profile as the one shown in Figure 5.2 is obtained.

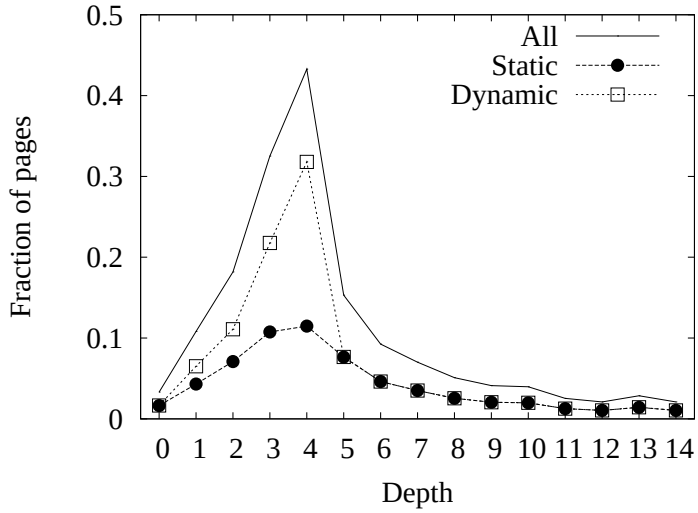

Figure 5.2: Amount of static and dynamic pages at a given depth. Dynamic pages were crawled up to 5 levels, and static pages up to 15 levels. At all depths, static pages represent a smaller fraction of the Web than dynamic pages.

Notice that here we are not using the number of slashes in the URL, but using the real shortest distance in links with the start page(s) of the Web site. The dynamic pages grow with depth, while the static pages follow a different shape, with the maximum number of pages found around 2 or 3 links deep; 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.

We deal with the problem of capturing a relevant portion of the *dynamically generated content 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, should a search engine like to retrieve that page? Clearly, from the Web site or the searcher's point of view, the answer should be yes, but from the search engine's point of view, the answer might be no.

5.2 Random surfer models for an infinite Web site

We will consider a Web site $S = (Pages, Links)$ as a set of pages under the same host name that forms a directed graph. The nodes are $Pages = \{P_1, P_2, \dots\}$ and the arcs are $Links$ such that $(P_i, P_j) \in Links$ iff there exists a hyperlink 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 $\mathbf{u} = (P_1, P_2, \ldots, P_n)$, 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 Web site following a link to an internal page, e.g., if they come from a search engine.

Definition (Page depth) For a page P_i and a session $\mathbf{u}$, we define the depth of the page in the session, $depth(P_i, \mathbf{u})$ as:

$$depth(P_i, \mathbf{u}) = \begin{cases} 0 & \text{if } P_i = u_0 \\ \min depth(P_j, \mathbf{u}) + 1 & P_j \in \mathbf{u}, j < i, (P_j, P_i) \in Links \end{cases}$$

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 a particular 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 that collapses multiple pages at the same level as a single node, as shown in Figure 5.3. That is, the Web site graph is collapsed to a sequential list.

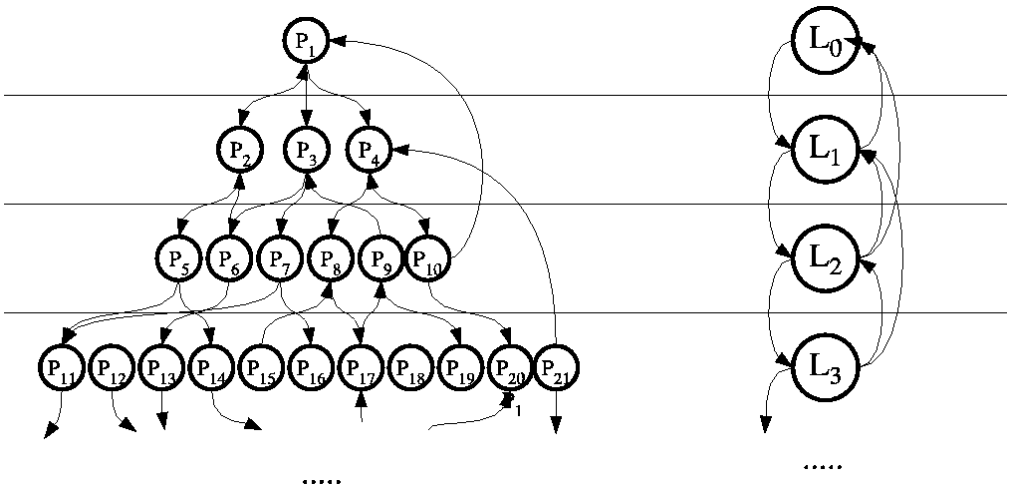

Figure 5.3: A Web site and a sequence of user actions can be modeled as a tree (left). If we are concerned only with the depth at which users explore the Web site, we can collapse the tree to a linked list of levels (right).

The advantage of modeling the Web site graph as a sequential list as that we do not need to model exactly which page a user is visiting, because we do not need this information as our main concern is at what

depth the user is inside a Web site. Also, different Web sites have varying degrees of connectivity, so for considering the entire Web site we would need to model both the number of out-links of each page and the distribution of the overlap of out-links between pages.

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 previous level (action *prev*), go to a different deeper 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 5.4. Regarding this Web site, after leaving, users have only one option: start again in a page with depth 0 (action *start*).

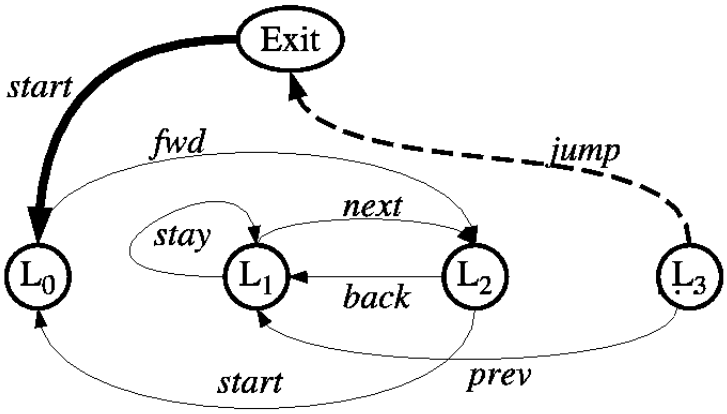

Figure 5.4: Representation of the different actions of the random surfer. The EXIT node represents leaving the Web site, and the transition between that node and the start level has probability one.

As this node `EXIT` has a single out-going link, 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 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 cannot use this idea because we want 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:

- $Pr(next|\ell)$: probability of advancing to the level $\ell+1$.
- $Pr(back|\ell)$: probability of going back to the level $\ell-1$.

- $Pr(stay|\ell)$: probability of staying at the same level ℓ.
- $Pr(start, jump|\ell)$: 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,
- $Pr(prev|\ell)$: probability of going to a previous level that is neither the start level nor the immediate preceding level.
- $Pr(fwd|\ell)$: probability of going to a following level that is not the next level.

As they are probabilities, $\sum_{action \in \mathcal{A}} Pr(action|\ell) = 1$. The probability distribution of all levels 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 paper, we study three models with $Pr(next|\ell) = q$ for all levels, i.e.: the probability of advancing to the next level is constant. Our purpose is to predict how far will a real user go into a dynamically generated Web site. If we know that, e.g.: $x_0 + x_1 + x_2 \geq 0.9$, 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, though 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.

Our models are "birth-and-death" processes, because they have an interpretation in terms of each level being a number representing the population of a certain species, and each transition between two levels represents either a birth of a death of a member. In this context, we note in advance that any given model in which from a certain point over the rate of death (going back to the first levels) exceeds the rate of birth (going deeper), then the population will be bounded (the visits will be found mostly in the first levels).

5.2.1 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 5.5.

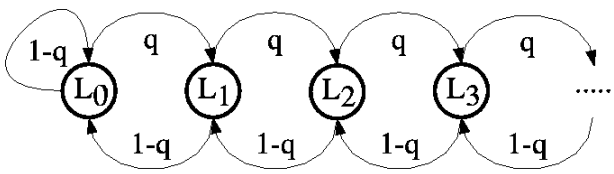

Figure 5.5: Model A, the user can go forward or backward one level at a time.

Transition probabilities are given by:

- $Pr(next|\ell) = q$

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

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

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

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 the solution is $x_i = 0$, and $x_\infty = 1$, so we have an asymptotic absorbing state. In our framework this means that no depth boundary can ensure a certain proportion of pages visited by the users. When $q < 1/2$ and we impose the normalization constraint, $\sum_{i\geq 0} x_i = 1$, 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 $0 \ldots k$ is:

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

This distribution is shown in Figure 5.6. We also calculate the session length, if we consider that a session ends when the user returns to level zero, as actions *start* and *jump* are equivalent. This is equivalent to the average return time to the origin in a Markov chain, which is $1/x_0$ [MT93]. Hence, $E(|\mathbf{u}|) = \frac{1-q}{1-2q}$.

5.2.2 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 5.7.

The transition probabilities are given by:

- $Pr(next|\ell) = q$
- $Pr(back|\ell) = 1-q$ if $\ell = 1$, 0 otherwise
- $Pr(stay|\ell) = 1-q$ for $\ell = 0$

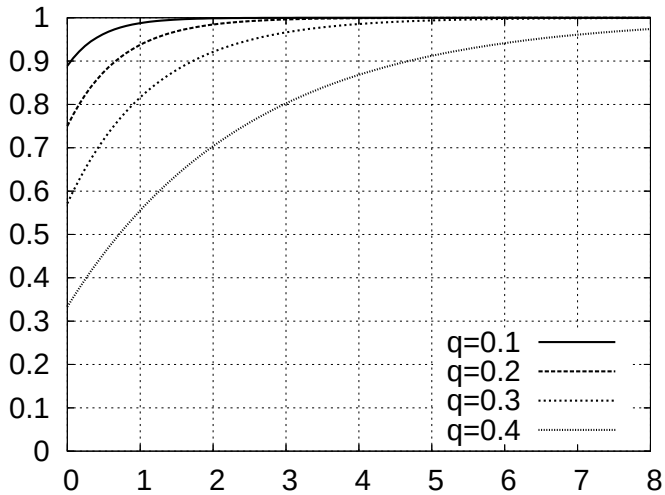

Figure 5.6: Distribution of visits per depth predicted by model A.

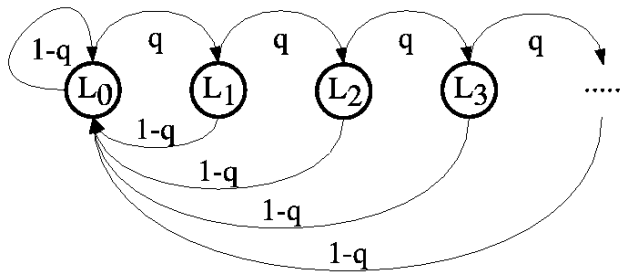

Figure 5.7: Model B, users can go forward one level at a time, or they can go back to the first level either by going to the start page, or by starting a new session.

- $Pr(start, jump|\ell) = 1 - q$ for $\ell \geq 2$
- $Pr(prev|\ell) = Pr(fwd|\ell) = 0$

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

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

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:

$$\sum_{i=0}^{k} x_i = 1 - q^{k+1}$$

This distribution is shown in Figure 5.8. In this case we have $E(|\mathbf{u}|) = \frac{1}{1-q}$.

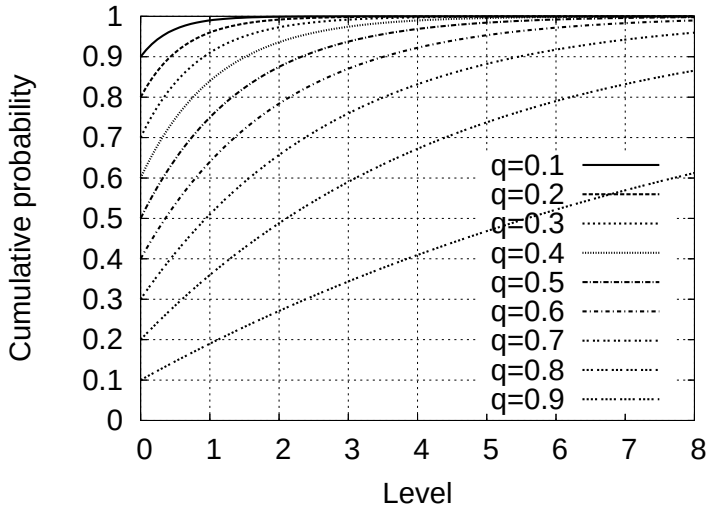

Figure 5.8: Distribution of visits per depth predicted by model B.

5.2.3 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 a user decides to go back to a previously seen level, the level will chosen uniformly from the set of visited levels (including the current one), as shown in the Figure 5.9.

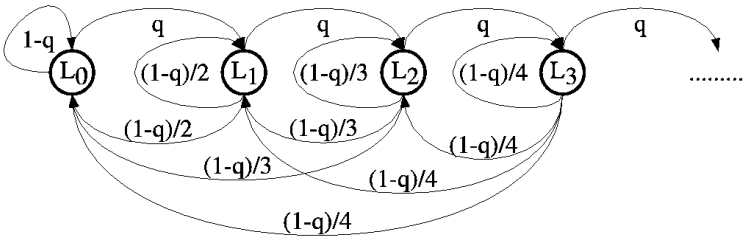

Figure 5.9: Model C: the user can go forward one level at a time, and can go back to previous levels with uniform probability.

The transition probabilities are given by:

- $Pr(next|\ell) = q$
- $Pr(back|\ell) = 1 - q/(\ell+1)$ for $\ell \geq 1$
- $Pr(stay|\ell) = 1 - q/(\ell+1)$
- $Pr(start, jump|\ell) = 1 - q/(\ell+1)$ for $\ell \geq 2$

- $Pr(prev|\ell) = 1 - q/(\ell+1)$ for $\ell \geq 3$
- $Pr(fwd|\ell) = 0$

A stable state $\mathbf{x}$ 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 obtain 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:

$$\sum_{i=0}^{k} x_i = 1 - (2 + k - (k+1)\,q)\,q^{k+1}$$

This distribution is shown in Figure 5.10. In this case we have $E(|\mathbf{u}|) = \frac{1}{(1-q)^2}$.

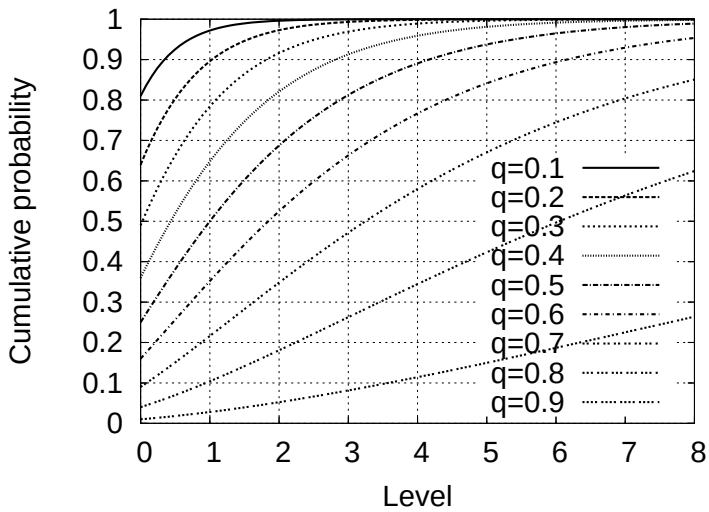

Figure 5.10: Distribution of visits per depth predicted by model C.

5.2.4 Model comparison

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 the same amount of page views.

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:

$$q_A = \frac{q_B}{1+q_B} \ .$$

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 of session depth.

It is worth noticing that a good model should approximate both the distribution of session depth and the distribution of session length. Table 5.1 shows the predicted session lengths.

Table 5.1: Predicted average session length for the models, with different values of q.

q	Model A	Model B	Model C
0.1	1.13	1.11	1.23
0.2	1.33	1.25	1.56
0.3	1.75	1.43	2.04
0.4	3.00	1.67	2.78
0.5	–	2.00	4.00
0.6	–	2.50	6.25
0.7	–	3.34	11.11
0.8	–	5.00	25.00
0.9	–	10.00	100.00

In Table 5.1 we can see that although the distribution of session depth is the same for models A and B, model B predicts shorter sessions. Observed average session lengths in the studied Web sites are mostly between 2 and 3, so reasonable values for q lie between 0.4 and 0.6.

5.3 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 and educational sites, non-governmental organizations, and sites in which collaborative forums play a major role, also known as "Blogs".

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

- Sort the logs by IP address of the client, then by access time stamp.
- 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], as a way of overcoming the issue that multiple clients can be behind the same IP address.
- 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, this required manual inspection of pages with frames.
- Ignore hits to Web applications such as e-mail or content management systems, as they neither respond to the 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. Note that, 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*, so user visits could be actually even less deep.

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 by Tan and Kumar [TK02].
- Identify robots by known `User-Agent` fields.
- Ignore malicious hits searching for security holes, 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].

5.3.1 General characteristics of user sessions

The characteristics of the sample, as well as the results of fitting models B and C to the data are summarized in Table 5.2. The names of the Web sites are not public because some of the log files, specially those of commercial entities, were obtained under the condition of publishing only the statistical results.

Table 5.2: Characteristics of the studied Web sites. The number of user sessions does not reflect the relative traffic of the Web sites, as it was obtained in different time periods. The average number of page views per session is larger in Blogs. "Root entry" is the fraction of sessions starting in the home page.

Code	Type	Country	Recorded sessions	Average session length	Average max. depth	Root entry
E1	Educational	Chile	5,500	2.26	0.98	84%
E2	Educational	Spain	3,600	2.82	1.41	68%
E3	Educational	US	71,300	3.10	1.31	42%
C1	Commercial	Chile	12,500	2.85	0.99	38%
C2	Commercial	Chile	9,600	2.12	1.01	32%
R1	Reference	Chile	36,700	2.08	0.95	11%
R2	Reference	Chile	14,000	2.72	1.21	22%
O1	Organization	Italy	10,700	2.93	1.97	63%
O2	Organization	US	4,500	2.50	1.13	1%
OB1	Organization + Blog	Chile	10,000	3.73	1.89	31%
OB2	Organization + Blog	Chile	2,000	5.58	2.48	84%
B1	Blog	Chile	1,800	9.72	3.56	39%
B2	Blog	Chile	3,800	10.39	2.31	21%

By inspecting Table 5.2, we observe that the average session length involves about 2 to 3 pages, and user sessions in Blogs are larger than in the other Web sites. This is reasonable as Web postings are very short, so a user reads several of them during a session.

5.3.2 Distribution of visits per depth

Figure 5.11 shows the cumulative distribution of visits per page depth to Web sites. We can see that at least 80%-95% of the visits occur at depth ≤ 4 (this is, no more than four "clicks" away from the entry page). It is also noticeable that about 30%-50% of the sessions include only the start page.

The distribution of visits per depth follows a power law, as shown in Figure 5.12. We only selected the log files with more than 10,000 sessions recorded in order to have enough sessions across the entire range of the figure, which is 30 levels.

An interesting observation about the distribution session lengths is that although they are longer in Blogs, they are not much deeper than in the other Web sites, as shown in Table 5.2. This led us to study the relationship between session length and session depth. The result is shown in Figure 5.13, which uses

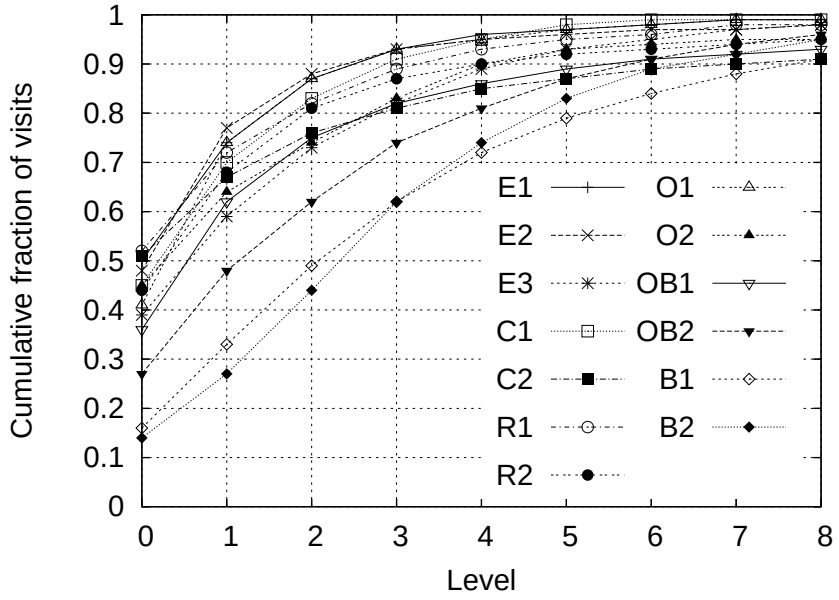

Figure 5.11: Cumulative distribution of visits per level, from access logs of Web sites. E=educational, C=commercial, O=non-governmental organization, OB=Organization with on-line forum, B=Blog (Web log or on-line forum).

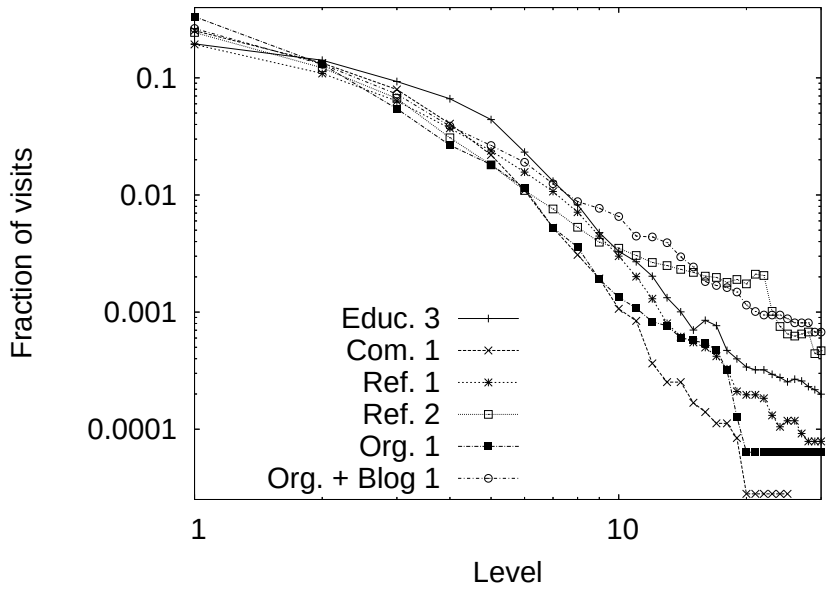

Figure 5.12: Distribution of visits per level. In this figure we only selected the log files with more than 10,000 sessions recorded.

information from all our samples including Blogs. Session depth grows slower than session length, and even long sessions, which are very rare, are not so deep. User browsing is certainly not depth-first.

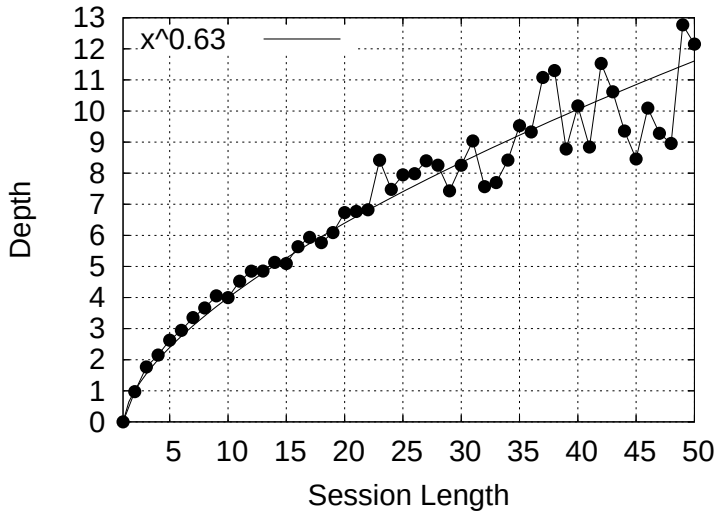

Figure 5.13: Session length vs. average session depth in the studied user sessions. Even very long sessions, which are rare, are not very deep.

The discrepancy between session length and depth is important from the point of view of an alternative model. Suppose the user chooses a session length at random before entering the Web site (this session length could reflect that the user has a certain amount of time or interest in the topic). In this model, the average session depth could be overestimated if we do not account for the fact that the browsing pattern is not depth-first. Figure 5.14 shows the session length distribution which follows a power law with parameter almost -2. This differs from the results of Huberman that had parameter -3/2 [HPPL98].

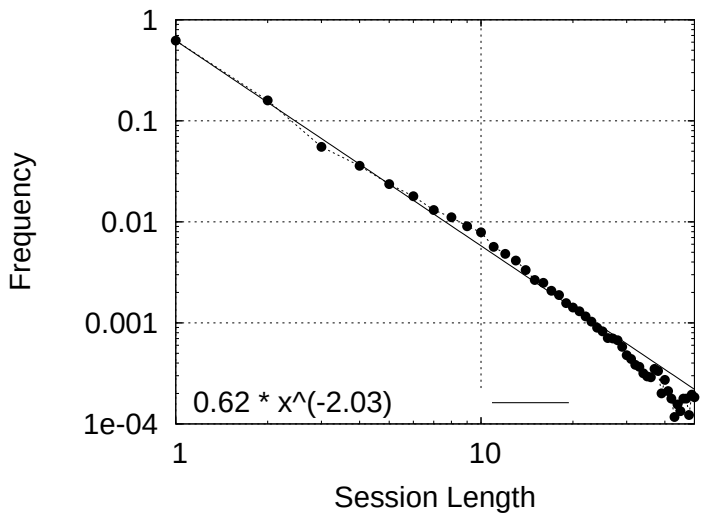

Figure 5.14: Session length distribution.

Table 5.3: Results of fitting models B (equivalent to model A) and C to the distribution of visits per depth in the studied Web sites. The minimum fitting error for each Web site is shown in bold face.

	Model B		Model C	
Code	q	Error	q	Error
Educ. 1	0.51	**0.88%**	0.33	3.69%
Educ. 2	0.51	**2.29%**	0.32	4.11%
Educ. 3	0.64	**0.72%**	0.45	3.65%
Com. 1	0.55	**0.39%**	0.36	2.90%
Com. 2	0.62	**5.17%**	0.41	10.48%
Ref. 1	0.54	**2.96%**	0.34	6.85%
Ref. 2	0.59	**2.75%**	0.39	6.11%
Org. 1	0.54	2.36%	0.35	**2.27%**
Org. 2	0.62	**2.31%**	0.42	5.95%
Org. + Blog 1	0.65	**2.07%**	0.46	5.20%
Org. + Blog 2	0.72	**0.35%**	0.54	2.00%
Blog 1	0.79	0.88%	0.63	**0.70%**
Blog 2	0.78	1.95%	0.63	**1.01%**

5.4 Model fit

We fitted the models of cumulative depth to the data from Web sites. The results are presented in Table 5.3 and Figure 5.18. 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, as seen in Figure 5.19. 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 5.4, in which we consider all the Web sites except for Blogs.

Inspecting Table 5.4, we can see that the actions *next*, *jump* and *back* are the more important ones, which is evidence for the adequacy of models A (back one level) and model B (back to start level).

We can see in Figure 5.15 that $Pr(next|\ell)$ does not vary too much, and lies between 0.45 and 0.6, increasing as ℓ grows. This is reasonable as a user that already have seen several pages is more likely to follow a link. From the point of view of our models, it is certainly not constant, but is almost constant for the first five levels which are the relevant ones. On the other hand, *prev* and *back* are closer to constant.

Table 5.4: 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.

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
11	1649	**0.656**	0.020	0.076	**0.119**	0.018	**0.105**	0.004
12	1273	**0.687**	0.040	0.091	0.091	0.007	0.082	0.001
13	1008	**0.734**	0.015	0.058	**0.112**	0.005	0.054	0.019
14	814	**0.716**	0.005	0.051	**0.113**	0.015	0.080	0.019
15	666	**0.762**	0.025	0.056	0.091	0.008	0.041	0.017

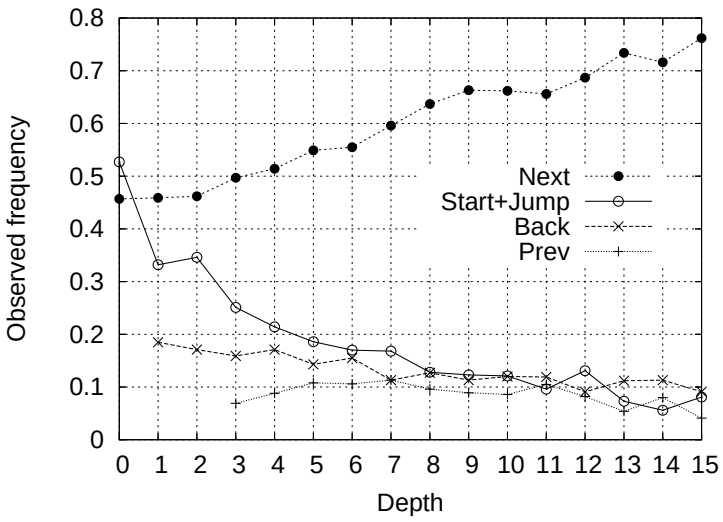

Figure 5.15: Experimental values for our atomic actions.

Actions *start*, *stay* and *fwd* are not very common. These actions include visits to pages that have been already seen, but it seems that pages are only re-visited by going back one level.

5.5 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 Web site 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 Web sites, 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 deciding when to stop while doing Web crawling, the simple model is good enough.
- Model C 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 show evidence of a distribution of visits that 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 3 to 4 clicks away from the entry page in most of the Web sites. In the case of Blogs, we observed deeper user sessions, with 90% of the visits within 6 to 7 clicks away from the entry page. Although our models do not fit well for deep sessions, they are accurate for the first five relevant levels. Also, we would need much more data to get significant results for over six levels.

In theory, as internal pages can be starting points, it could be concluded that Web crawlers must always download entire Web sites. However, entry pages are usually only in the first few levels of a Web site. If we consider the physical page depth in the directory hierarchy of a Web site, we observe that the frequency of surfing entry points per level rapidly decreases, as shown in Figure 5.16. This is consistent with the findings of Eiron *et al.*; they observed that "when links are external to a site, they tend to link to the top level of the site" [EMT04].

Link analysis, specifically Pagerank, provides more evidence for our conclusions. We asked, what fraction of the total Pagerank score 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 in April of

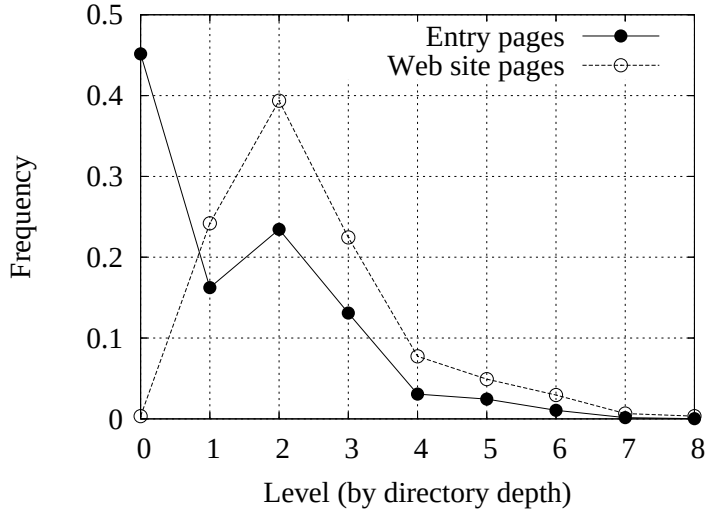

Figure 5.16: Fraction of different Web pages seen at a given depth, and fraction of entry pages at the same depth, considering the directory structure, in the studied Web sites. Frequencies are normalized relative to all pages.

2004, using 150 thousand seed pages that found 53 thousand Web sites. Figure 5.17 shows the cumulative Pagerank score for this sample. Again, the first five levels capture more than 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. These results are consistent with the findings by Najork and Wiener [NW01].

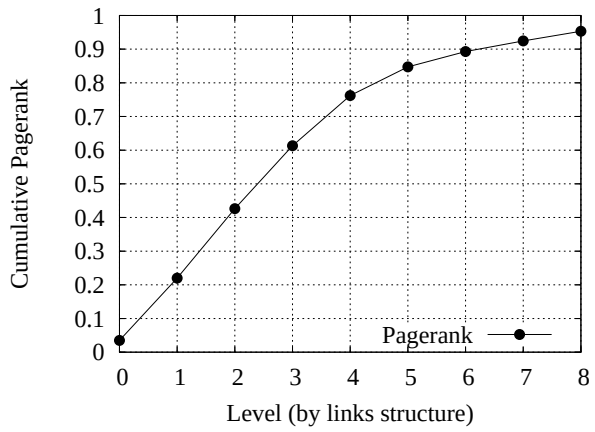

Figure 5.17: Cumulative Pagerank by page levels in a large sample of the Chilean Web.

These models and observations could be used by a search engine, and they can also account for differences in Web sites. For instance, if the search engine's crawler performs 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 content 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 that we visited that page for the first time. A more detailed analysis could also consider the distribution of terms in Web pages and anchor text as the user browses through a Web site.

A different class of models for user browsing, including models based on economical decisions could be used, but those models should be able to fit both, the distribution of session length and the expected session depth.

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.

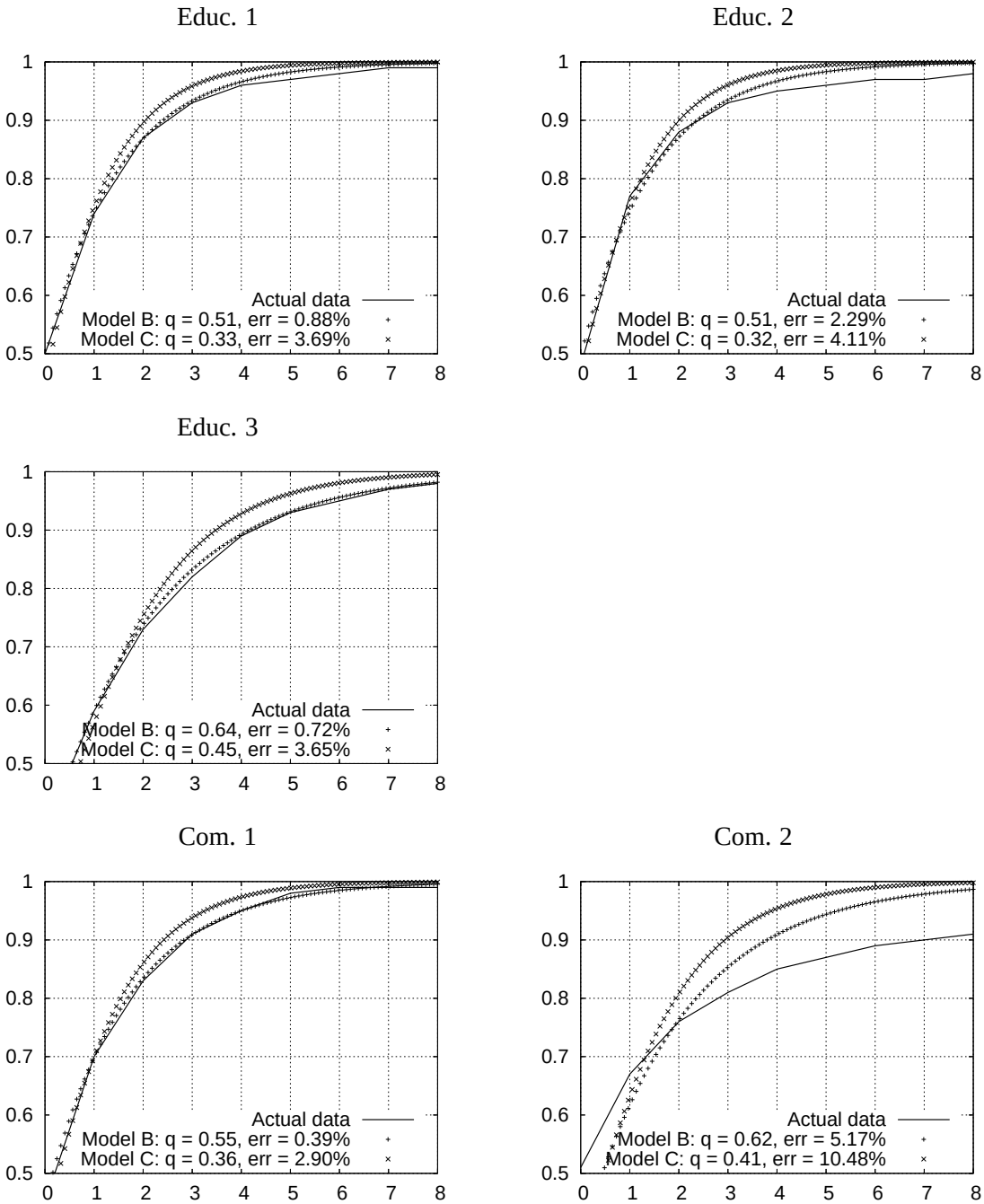

Figure 5.18: Fit of the models to actual data, except for Blogs and non-governmental organizations with Blogs. Model B (back to start level), has smaller errors for most Web sites. The asymptotic standard error for the fit of this model is 5% in the worst case, and consistently less than 3% for all the other cases. Note that we have zoomed in into the top portion of the graph (continues on the next page).

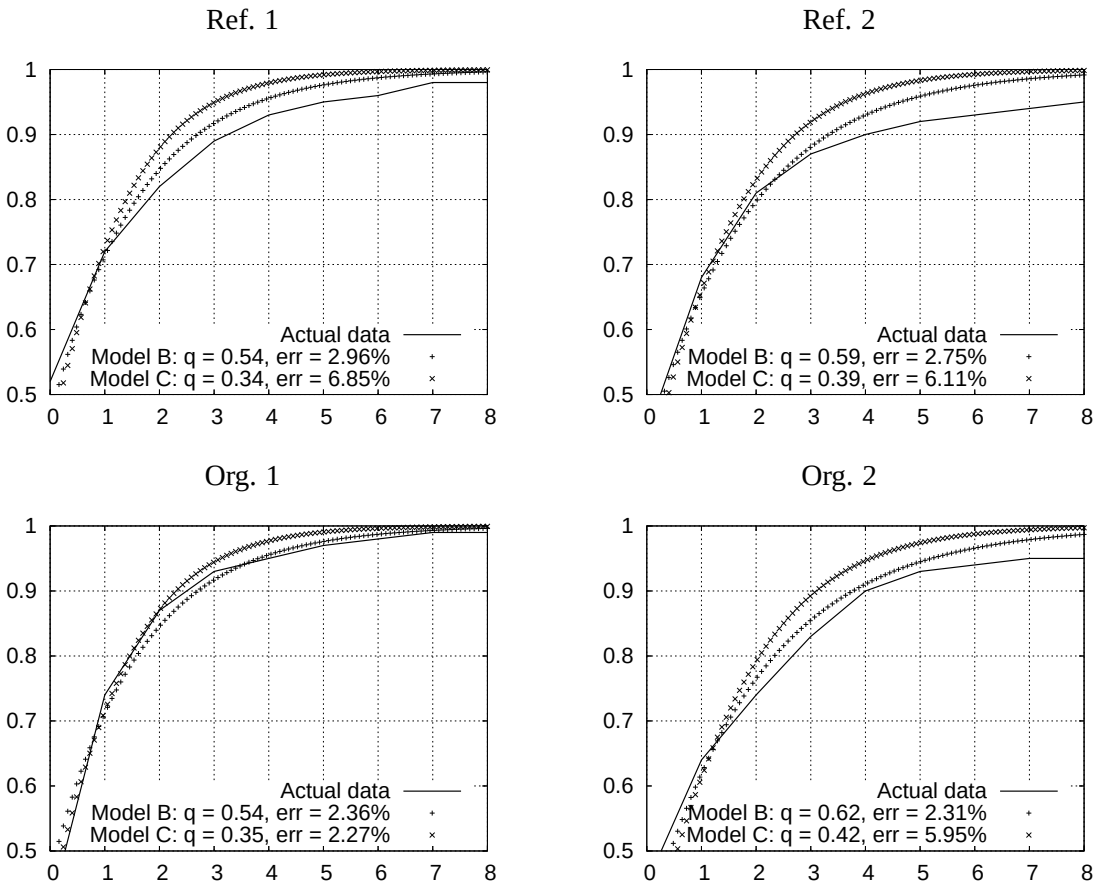

Figure 5.18 (cont.)

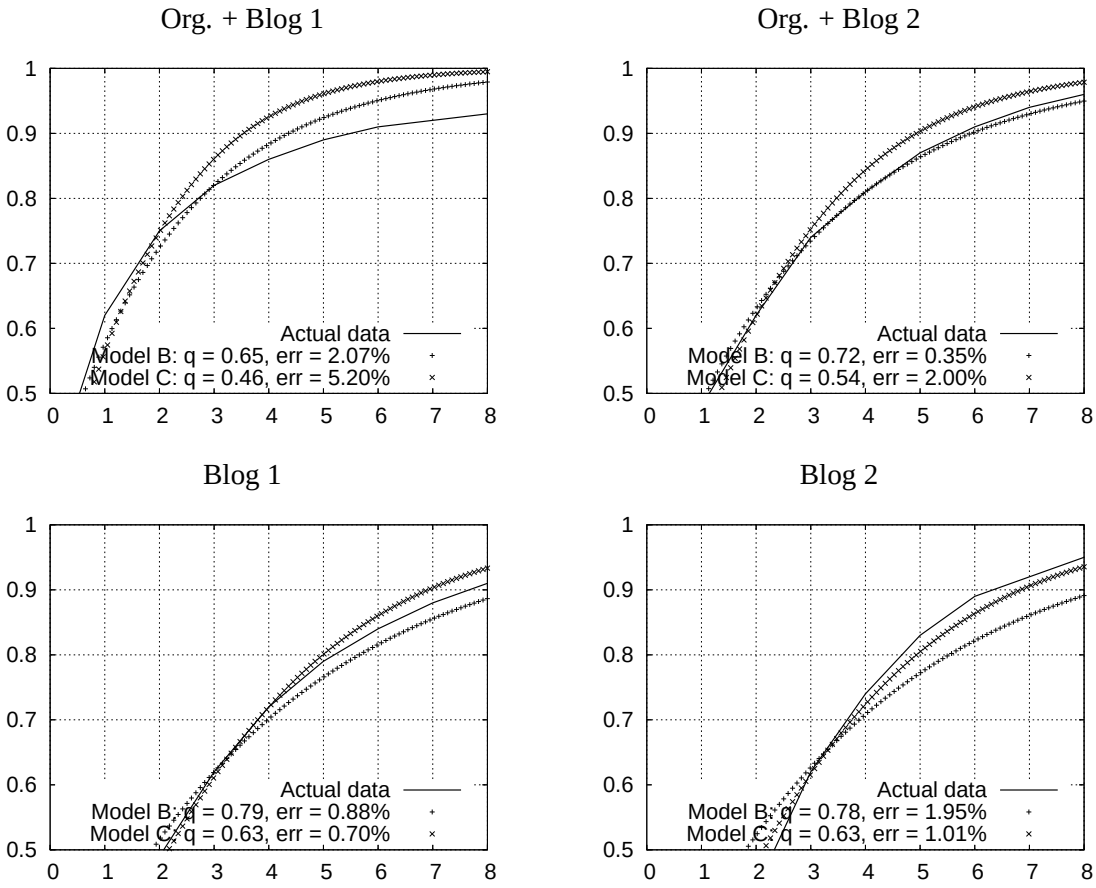

Figure 5.19: Fit of the models to actual data in the case of Blogs. In this case user sessions tend to go deeper inside the Website because more pages are visited per session, probably because Blog postings tend to be short.

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, Santiago, Chile, 2002. IOS Press Amsterdam.

[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)*, volume 3243 of *Lecture Notes in Computer Science*, pages 156–167, Rome, Italy, October 2004. Springer.

[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.

[EMT04] Nadav Eiron, Kevin S. McCurley, and John A. Tomlin. Ranking the web frontier. In *Proceedings of the 13th international conference on World Wide Web*, pages 309–318. ACM Press, 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.

[MT93] S. P. Meyn and R. L. Tweedie. *Markov Chains and Stochastic Stability*. Springer, London, 1993.

[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 Science.

[PBMW98] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The Pagerank citation algorithm: bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

[RGM01] Sriram Raghavan and Hector Garcia-Molina. Crawling the hidden web. In *Proceedings of the Twenty-seventh International Conference on Very Large Databases (VLDB)*, pages 129–138, Rome, Italy, 2001. Morgan Kaufmann.

[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.

[TK02] Pang-Ning Tan and Vipin Kumar. Discovery of web robots session based on their navigational patterns. *Data Mining and Knowledge discovery*, 6(1):9–35, 2002.