Reorganizing Large Web Sites

David J. Aldous

1. INTRODUCTION. Sometimes you're confident that the information you're seeking is somewhere in a large web site, but it takes a long sequence of clicks on links until you find the page you want, or you give up in frustration. "Couldn't this site be organized better?", you think. This essay is my attempt to make an interesting mathematical question out of such a thought, and to relate it to some classical (pre-Internet) work.

A document you see using your browser is a *page*. Oversimplifying, let's say a *content page* is one containing information that the site owner is making publicly accessible; an *index page* contains only links to other pages, intended to help the user navigate through the site to find a desired content page. Section 2 elaborates on the unrealistic simplifications of the model. A small web site could just have one index page (conventionally called `index.html`) with links to each content page, but a site with thousands of pages needs more elaborate organization, typically a hierarchical scheme of index pages (an *index tree*) illustrated in Figure 1.

As a concrete illustration of searching a large site, suppose you're gotten to my department's site `www.stat.berkeley.edu` and are looking for a draft of this paper. The *root* index page offers about 45 links. Some are to content pages (available faculty positions, list of tutors, etc.) and some are to other index pages (faculty, staff, course offerings, etc.). The "faculty" link takes you to a list of 35 faculty, then the "David Aldous" link takes you to my personal home page. This page has 10 links, one of which is "preprints and recent publications", and that link leads to a list of about 25 papers. Thus, having reached the department site, to see a copy of this paper on your browser requires 4 more clicks: the page is at *depth* 4.

But there's something strange about these numbers. If a "typical" page on the site required 4 clicks with the same sequence of numbers of choices, there would be about

$$45 \times 35 \times 10 \times 25 \approx 400{,}000$$

pages on the site, which is a huge overestimate (in fact there are about 4,000 pages). The site design makes finding this particular page rather harder than it need be. How can we quantify the effort of finding a page? Let's say that the *cost* of looking at an index page i with $d(i)$ links and choosing a link is

$$1 + \frac{d(i)}{\gamma}$$

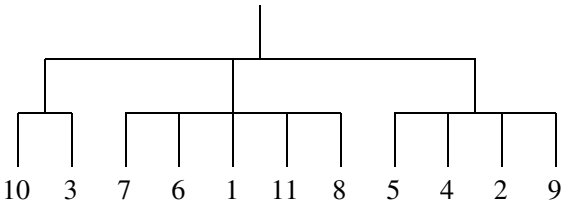

Figure 1. An index tree with 11 content pages (numbered) and 4 index pages (horizontal lines).

where γ is a parameter reflecting your preferences for numbers of links per page (larger γ means you prefer more links). So the cost c_j of finding a content page j at depth h is

$$c_j = (1 + \tfrac{d(i_0)}{\gamma}) + (1 + \tfrac{d(i_1)}{\gamma}) + \cdots + (1 + \tfrac{d(i_{h-1})}{\gamma}) \tag{1}$$

where i_0 (= root), $i_1, \ldots, i_{h-1}$ is the sequence of index pages you must traverse to reach the content page j. See Section 2.3 for more discussion of this cost function, and an interpretation of γ. Now from a practical viewpoint, seeking to minimize such costs is only a *secondary* goal in site design; the primary goal is a user-friendly logical structure with well-chosen names on links, so that users can choose the correct links in their search for a content page. In other words, the *categories*—the phrases such as `faculty, staff, course offerings` used to link to index pages, which mathematically represent subsets of content pages—have to be human-interpretable categories, not arbitrary subsets. But it's not so clear how to make mathematical sense of "human-interpretable category", so let's continue to discuss cost for a while. Suppose we know the relative frequencies $\mathbf{p} = (p_j)$ of access of the different content pages j (on many sites, frequencies are recorded automatically). Then for any index tree there is an average cost

$$\bar{c} = \sum_j p_j c_j$$

and so there is some "optimal" index tree that minimizes $\bar{c}$. Now the optimal index tree typically does not have human-interpretable categories, (so it would be useless in practice), but it can serve as a basis for comparison. We show in Section 3 that the optimal tree, and any other index tree with a certain "approximate balance" property, has an average cost whose order of magnitude is

$$\tfrac{1}{\log \gamma} \times \mathcal{E}(\mathbf{p}) \tag{2}$$

where $\mathcal{E}(\mathbf{p}) = -\sum_j p_j \log p_j$ is the *entropy* of the distribution $\mathbf{p}$. That entropy arises is no surprise to the reader acquainted with classical information theory—see Section 3.

We can now state a conceptual question about web site organization.

> Suppose we start with an organizational scheme that reflects the logical structure of the site, as perceived by users, without paying attention to the frequencies p_j. Then we look at the frequencies, and wish to make an index tree that achieves (up to a constant factor) the optimal cost while preserving as much of the logical structure as possible. What kind of "logical structure" can we guarantee to preserve?

One way to formalize this question is to assume that the logical structure is represented by a binary tree, as illustrated in Figure 2 for the case of 11 content pages (keep in mind we are really interested in sites with thousands of content pages). The interpretation of Figure 2 is that pages 4 and 2 are more closely related to each other than to any other page; that page 9 is less closely related to 2 and 4 than they are inter-related, but more closely than to any other page; and so on.

Because the phrase "binary tree" has several different meanings, we borrow a term from biology [**4**, Chapter 2] and call this type of tree a *cladogram*. Any branchpoint (internal node) specifies a subset of pages "all pages beneath the branchpoint", such as {4, 2, 9} or {7, 6, 1, 11, 8}—call such a subset a *clade*. Mathematically, a clade may

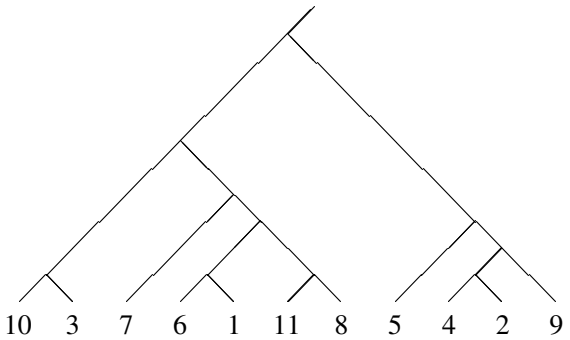

Figure 2. A cladogram on 11 pages.

consist of a single content page $\{j\}$. Section 2.5 says more about how such a cladogram could be obtained in the first place. Conceptually, we are supposing that each clade is a human-interpretable category.

Now a cladogram on n content pages could itself be regarded as an index tree with $n-1$ index pages, each having exactly two links. But that is inefficient—we want more links per index page. As a first way to get more links, consider a simple notion of *contraction* of index trees. If index page i' has children $i_1, \ldots, i_d$ and parent i, then one can obtain a new tree by "contracting edge (i, i') to a point", that is by deleting page i' and making pages $i_1, \ldots, i_d$ be new children of page i. Say a tree $\mathbf{t}$ is a contraction of $\hat{\mathbf{t}}$ if it can be obtained from $\hat{\mathbf{t}}$ by a sequence of such steps. For example, the tree in Figure 1 is a contraction of the tree in Figure 2. It is an easy exercise to show in general

Lemma 1. $\mathbf{t}$ *is a contraction of* $\hat{\mathbf{t}}$ *if and only if each clade of* $\mathbf{t}$ *is a clade of* $\hat{\mathbf{t}}$.

One might hope that, given a cladogram and page frequencies, one could always use some contraction to obtain a near-optimal index tree, but the following extreme example shows this is not always possible. Consider the completely unbalanced cladogram on pages $1, 2, \ldots, n$, illustrated for $n = 10$ in Figure 3. Suppose the frequency of the rightmost page n is almost 1. Precisely, define $\mathbf{p}_n$ by

$$p_n(n) = 1 - \tfrac{1}{\log n}; \quad p_n(i) = \tfrac{1}{(n-1)\log n},\ 1 \le i \le n-1.$$

This distribution is chosen so that $\mathcal{E}(\mathbf{p}_n)$ is bounded as $n \to \infty$. Any contraction tree must look like the tree in Figure 4, with page n at some depth.

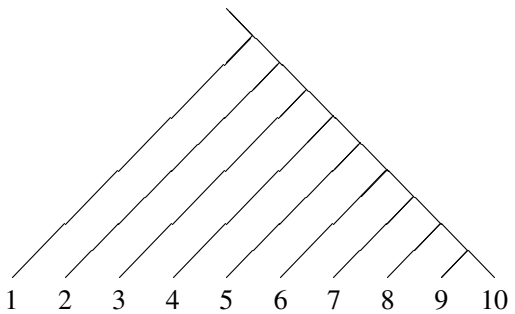

Figure 3. An unbalanced cladogram.

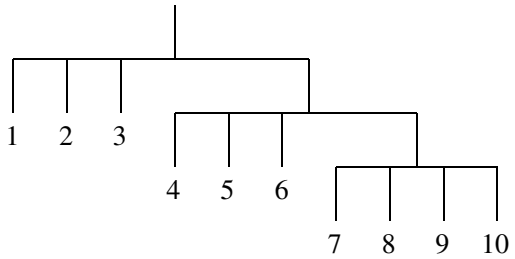

Figure 4. A contraction of the Figure 3 cladogram.

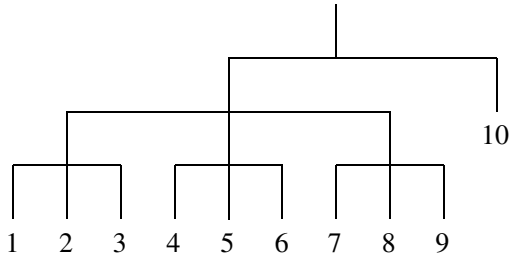

Figure 5. An alternative index tree.

Regardless of the number of levels we choose, the cost of such a contraction tree with frequencies $\mathbf{p}_n$ is not bounded in n, though the optimal cost is bounded (as we show next). The natural index tree to use in this example is as in Figure 5. It is easy to check that the cost of using this index tree with frequencies $\mathbf{p}_n$ is bounded. This tree isn't a contraction of the original cladogram because the clade $\{1, 2, \ldots, 9\}$ of the left branch at the root is not a clade of the original tree. Instead, it is of the form "clade minus subclade", that is $\{1, 2, \ldots, 10\} \setminus \{10\}$. The following theorem, proved in Section 4, is the mathematical content of this essay.

Theorem 2. *Given any cladogram* $\hat{\mathbf{t}}$ *and any page-frequencies* $\mathbf{p}$*, there exists an index tree* $\mathbf{t}$ *such that*

(i) each clade of $\mathbf{t}$ *is either a clade of* $\hat{\mathbf{t}}$ *or is of the form* $A \setminus B$*, where* $B \subset A$ *are clades of* $\hat{\mathbf{t}}$*;*

(ii) the average cost $\bar{c} = \bar{c}(\mathbf{t}, \mathbf{p})$ *is within a constant factor of the minimum cost over all index trees with frequencies* $\mathbf{p}$*.*

The proof of Theorem 2 uses a simple algorithm to construct an index tree with the required properties. The conceptual point of Theorem 2 is that the original cladogram specifies $2n - 1$ clades (the n content pages and $n - 1$ subsets), which are supposed to be human-interpretable categories. So there are $O(n^2)$ possible subsets of form (i), which are only slightly more complicated to interpret. The index tree given in Theorem 2 uses only these subsets as categories, rather than arbitrary subsets taken from all 2^n subsets.

The appearance of a constant factor in Theorem 2 makes such a result rather distant from practical concerns; it is the mathematical price to pay for complete generality of $\hat{\mathbf{t}}$ and $\mathbf{p}$.

1.1. Summary

1. Modelling web sites as trees, and taking the cost to be affine (1) in number of links per index page, there is a theoretical expression (2) for the minimum-cost index tree, which involves the entropy of the probability distribution over content pages. This is analogous to classical coding theory.
2. Given a logical structure for a site, as represented by a cladogram, one can construct an index tree whose cost is within a constant factor of optimal; it uses index categories no more complicated than "clade minus subclade".

2. FURTHER DISCUSSION OF THE MODEL

2.1. Simplifications in the Model

(i) Many large sites also offer searching via key words, as an alternative to clicking on links.

(ii) There is not really a rigid distinction between content pages and index pages.

(iii) We are simplifying the real-world problem by assuming each user is seeking only a single page.

(iv) Our affine model of cost is perhaps oversimplistic.

2.2. Removing the Restriction to Trees. It's often said that the point of the web and hypertext is to allow you to pursue links sideways, to investigate a topic in any direction you like rather than in the linear order of a book. Such browsing is indeed valuable, but in the context of looking for specific information on a particular site it seems reasonable to consider indexing structure with only downward links. But there is no practical reason to restrict index pages to form a *tree*; in many contexts it is desirable to have different paths to the same page. For example, in designing a department's site one might put a link to "technical reports" that provides an alternate route to a technical paper. So it would be more realistic to model the index structure as a directed acyclic graph (DAG).

In a sense, in formulating Theorem 2 we are cheating by assuming that the logical structure is given by a cladogram to begin with. It would be more realistic to start with some collection of subsets that represent human-interpretable categories, and then try to construct a DAG index structure using only sets that can be obtained easily from the given subsets. Formulating some analog of Theorem 2 in this context is an interesting challenge.

2.3. The Cost Function. In (1), the cost of making a choice from an index page with d downward links is assumed to be $1 + d/\gamma$ for some parameter $\gamma > 0$. The constant term (which, by scaling, we may take to be 1) reflects the time for an index page to be downloaded. In some contexts, e.g., choosing a name from an alphabetical list, the effort to choose a link is sublinear in d. But more typically, a user doesn't mentally formulate a category and then search for it; rather, the user scans through the given categories to find a match, and in this context linear dependence seems appropriate.

To interpret γ, suppose you have the choice of a single list of m categories or a two-level tree with $\sqrt{m}$ categories at each level. Which do you prefer? For small m the former is easier, while for large m the latter is easier. So there is some crossover or "indifference" point m_0. Since the costs are $1 + m/\gamma$ and $2(1 + \sqrt{m}/\gamma)$, you can derive your γ from your m_0 by solving

$$1 + \tfrac{m_0}{\gamma} = 2(1 + \tfrac{\sqrt{m_0}}{\gamma})$$

to get

$$\gamma = m_0 - 2\sqrt{m_0}.$$

Personally, I choose $m_0 \approx 30$ so that $\gamma \approx 20$.

2.4. Expert Advice. Web site design is a substantial commercial enterprise, as can be seen by typing `web site design` into a search engine. More intellectual analysis can be found in texts such as [**8**], in the IBM web design guidelines [**6**], and in [**10**] and other papers from the same conference. For broader academic analysis of the world wide web see [**11**]. This literature is not overtly mathematical. The consensus advice from experts in web design is:

> Have few vertical levels.

Thus, the experts tell us to take γ larger rather than smaller; enable the user to get to content pages via few links. It turns out that our algorithm for constructing an index tree in the proof of Theorem 2 does tend to create many links (see Figure 8) and, in particular, forbids the lazy habit of creating a single "miscellaneous" category comprised of logically unrelated topics.

2.5. Building the Cladogram. Classification trees arise in several overlapping subdisciplines of statistics, e.g., cluster analysis, statistical learning theory, machine learning theory [**3**]. In a typical setting, a set of data in R^d is recursively partitioned into clusters, with the aim of identifying informative subpopulations within a population. One could seek to apply such existing methods to web site organization. Martin [**9**] suggests asking potential users to sort content pages into logically related groups, and then input these data into a cluster analysis algorithm to produce an index tree.

2.6. Trees as Data Structures. Trees have been studied classically as data structures for sorting and searching; see [**7**, Chapter 6]. Issues of "balance" and rebalancing arise there also. However, in the context of data structures or classification trees one envisages the tree as sitting inside the computer and invisible to the user; the key distinction is that in our context the tree has to be meaningful to a human navigator.

Web directories such as Yahoo seek to create user-friendly hierarchical taxonomies of the entire web. Intriguing software from Inxight [**1**] makes visible (as a "hyperbolic tree") the index structure of web sites.

2.7. Tree Balance and Phylogenetics. In biological systematics (taxonomy), the word *cladogram* is one formalization of *phylogenetic tree*, such trees being used to describe relationships betwen species. The notion of *balance* of a tree, which plays a role in this paper, has also been discussed in biological systematics: see [**2**] for a recent survey.

3. CRUDELY BALANCED INDEX TREES. In this section we formalize (Corollary 5) the idea that the optimal tree, and any other index tree with a certain "approxi-

mate balance" property, has an average cost whose order of magnitude is

$$\tfrac{1}{\log \gamma} \times \mathcal{E}(\mathbf{p}).$$

Fix an index tree $\mathbf{t}$ and frequencies $\mathbf{p}$. We can regard p_j as the probability that a user seeks content page j. So for each index page i there is an induced probability

$$q(i) = \sum_{j \in C(i)} p_j; \quad C(i) = \text{ clade defined by } i$$

that index page i is used in the search for a random content page. If i' is a child of i then $q(i'|i) = q(i')/q(i)$ is the conditional probability of going to page i' given that you are on page i. For notational compatibility write $q(j) = p_j$ for a content page j.

Intuitively, for an index tree to be cost-efficient it should be approximately *balanced*, in the following sense. Fix an integer $b \geq 2$. Each index page i should have about b children i', and the conditional probabilities $q(i'|i)$ should be about $1/b$.

Lemma 3. *Suppose an index tree* $\mathbf{t}$ *and frequencies* $\mathbf{p}$ *satisfy*

(a) $q(i'|i) \leq 2/b$ *whenever* i *and* i' *are both index pages.*

(b) $d(i) \leq 3b$ *for each index page* i.

Then

$$\bar{c} \leq (1 + \tfrac{3b}{\gamma})(1 + \tfrac{\mathcal{E}(\mathbf{p})}{\log b/2}).$$

Proof. Consider a content page j at depth h. By (b) and the definition of c_j

$$c_j \leq (1 + \tfrac{3b}{\gamma})h$$

whereas by (a)

$$p_j \leq (2/b)^{h-1}. \tag{3}$$

Rearranging (3) gives

$$h - 1 \leq -\tfrac{\log p_j}{\log b/2}$$

and so

$$c_j \leq (1 + \tfrac{3b}{\gamma})(1 - \tfrac{\log p_j}{\log b/2}).$$

Since $\bar{c} = \sum_j p_j c_j$ the desired inequality follows. ∎

Given $\mathbf{p}$ and b, it is easy to construct an index tree that satisfies the hypotheses of Lemma 3. To get the categories at the first level, first use each content page j with $q(j) \geq 1/b$ as a category; then gather the remaining content pages into subsets (index pages) of total probability between $1/b$ and $2/b$, except that the final subset may have probability $< 1/b$. Repeat recursively for each index page i, using conditional probabilities $q(\cdot|i)$ instead of unconditional probabilities.

Since we are free to choose b, we can choose $b \approx \gamma$ and then the cost of the index tree is of order $\mathcal{E}(\mathbf{p})/\log\gamma$. In order-of-magnitude assertions we are taking $n \to \infty$ and $\mathcal{E}(\mathbf{p}_n) \to \infty$.

This construction is very similar to the construction of the *Huffman code* [**5**, Chapter 4], in classical information theory. Given probabilities (p_j) and b, the Huffman code represents each item j as a base-b string of length $h(j)$; equivalently, as a leaf of a b-ary tree.

We now outline how a fundamental result of information theory implies a converse to Lemma 3.

Lemma 4. *For any index tree* $\mathbf{t}$ *with frequencies* $\mathbf{p}$,

$$\bar{c} \geq \mathcal{E}(\mathbf{p})/\log\gamma.$$

Proof. The fundamental result [**5**, Section 6.5] is that in any coding of a distribution $\mathbf{p}$, in particular in any variation of the Huffman code, the mean number of bits used is at least the entropy $\mathcal{E}(\mathbf{p})$. An index tree implicitly codes each content page j as a string $u_1 u_2 \ldots u_h$ where $1 \leq u_k \leq d(i_{k-1})$ for each $k \leq h$; that is, from the $k-1$'st index page i_{k-1} we choose one link (the u_k'th, say) from the $d(i_{k-1})$ possible links. Coding an integer from $\{1, \ldots, d\}$ requires $\log_2 d$ bits, so coding such a string requires $c_j^* := \sum_k \log_2 d(i_{k-1})$ bits. So the fundamental result implies that for any index tree

$$\mathcal{E}_2(\mathbf{p}) \leq \sum_j p_j c_j^*, \tag{4}$$

where $\mathcal{E}_2$ denotes entropy using $\log_2$. But a calculus exercise shows that for fixed $\gamma \geq 2$

$$\log_2 x \leq (1 + \tfrac{x}{\gamma})\ \log_2 \gamma, \quad 1 \leq x < \infty,$$

which implies, using the definition (1) of c_j, that $c_j^* \leq c_j \log_2 \gamma$. Then by (4), in any index tree the average cost $\bar{c}$ satisfies

$$\mathcal{E}_2(\mathbf{p}) \leq \bar{c}\ \log_2 \gamma. \qquad \blacksquare$$

Summarizing, Lemmas 3 and 4 imply

Corollary 5. *Suppose an index tree* $\mathbf{t}$ *and frequencies* $\mathbf{p}$ *satisfy the hypothesis of Lemma 3. Then the average cost* $\bar{c} = \bar{c}(\mathbf{t}, \mathbf{p})$ *is within a constant factor of the minimum cost over all index trees with frequencies* $\mathbf{p}$.

4. PROOF OF THEOREM 2. Fix a cladogram $\hat{\mathbf{t}}$ and frequencies $\mathbf{p}$. Fix an integer $b \geq 2$. We need to construct an index tree that satisfies the "clade minus subclade" condition and the hypotheses of Lemma 3; then the theorem follows from Corollary 5. We show that it is essentially enough to describe the construction of the top level of the index tree, because we can then proceed recursively.

First define a *pruned cladogram* as follows. Retain all vertices i (either content page or branchpoint) such that $q(i) \geq 1/b$. Say a vertex i with $q(i) \geq 1/b$ is a *major leaf* if i is a content page or if i is an index page and all of its children i' have $q(i') < 1/b$. Also retain a vertex i' with $q(i') < 1/b$ if its parent vertex i has $q(i) \geq 1/b$ and i is not a major leaf; call such i' a *minor leaf*. Delete all other vertices.

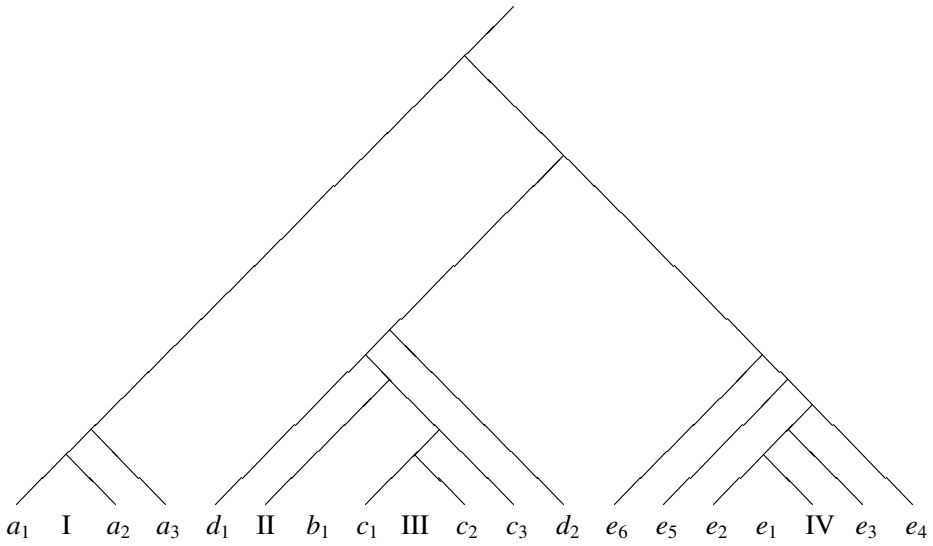

Figure 6. A pruned cladogram.

Figure 6 illustrates a pruned cladogram with 4 major leaves {I, II, III, IV} and 15 minor leaves $\{a_i, b_i, c_i, d_i\}$. The top level of the index tree (illustrated in Figure 8) has a category for each major leaf, and also has categories for subsets of minor leaves. The point of the following construction is to define these subsets so that they preserve the "clade minus subclade" property.

The pruned cladogram has some number k of major leaves; one can check that $k \geq 1$. Consider (Figure 7) the spanning tree on the root and the k major leaves. This tree has $k-1$ branchpoints, and has $2k-1$ line segments that link a branchpoint to a major leaf or to another branchpoint or to the root. To each of these line segments there may be minor leaves attached. In our example, there are no minor leaves attached to the two segments (root,α) and (α,β), but there are minor leaves attached to the other five segments. Thus the minor leaves are partitioned into subsets $(a_i), (b_i), \ldots, (e_i)$ and each subset is ordered (as a convention) so that the position of attachment to the line-segment increases from bottom to top. For a line segment with minor leaves (z_i), if $\sum_i q(z_i) < 2/b$ then in the index tree we make a single category at level 1 for the set (z_i). Otherwise (as happens with (e_i) in our example) we make categories of the form

$$\{z_1, z_2, \ldots, z_{L_1}\}, \quad \{z_{L_1+1}, \ldots, z_{L_2}\}, \quad \{z_{L_2+1}, \ldots\}, \tag{5}$$

where

$$L_1 = \min\left\{ j : \sum_{i=1}^{j} q(z_i) > 1/b \right\},$$

$$L_2 = \min\left\{ j : \sum_{i=L_1+1}^{j} q(z_i) > 1/b \right\},$$

and so on.

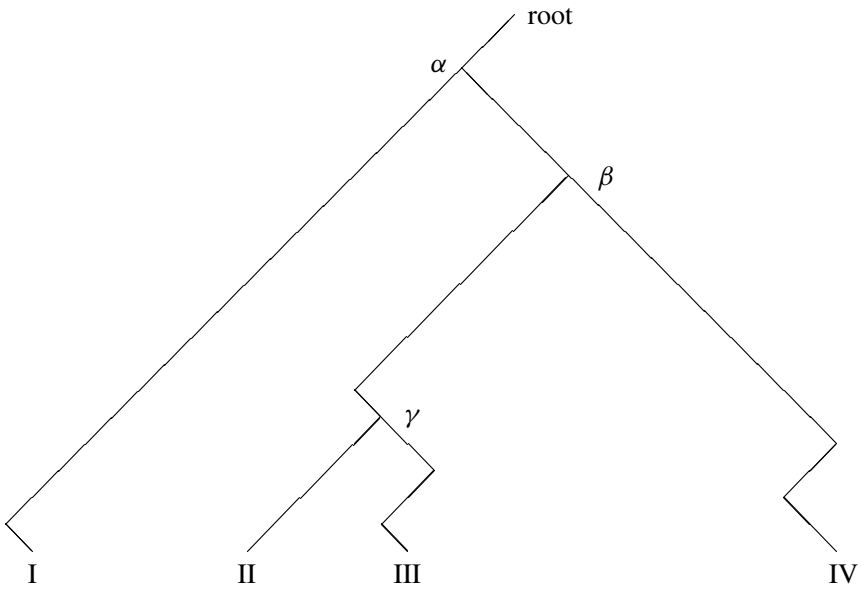

Figure 7. The spanning tree on the major leaves in Figure 6.

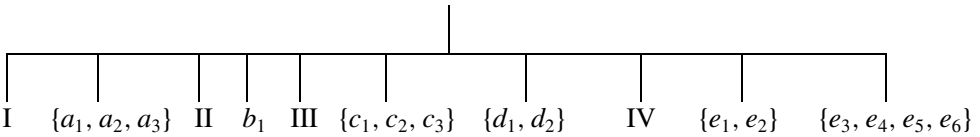

Figure 8. The top level of the derived index tree.

This completes the construction of the first level of the index tree; we next check it has the desired properties. A major leaf X that is not a content page must have $q(X) < 2/b$ (otherwise one of its two children i' has $q(i') \geq 1/b$, which is forbidden by the definition of *major leaf*). And each subset of form (5) has probability $< 2/b$ by construction, since each $q(z_i) < 1/b$ for a minor leaf z_i. This checks the property of Lemma 3(a) for the first level. Next, the number of categories at level one can be written as

$$d = k + \sum_{\ell} n(\ell),$$

where k is the number of major leaves, ℓ runs over the $2k - 2$ line segments and $n(\ell)$ is the number of subsets of form (5) attached to segment ℓ. So

$$d \leq (2k - 2) + \left(k + \sum_{\ell} (n(\ell) - 1)^+ \right).$$

But the total probability of the minor leaves attached to ℓ is at least $(n(\ell) - 1)^+/b$ and the probability of each major leaf is at least $1/b$, so summing gives

$$1 \geq \left(k + \sum_{\ell} (n(\ell) - 1)^+ \right) /b.$$

Therefore

$$\left(k + \sum_{\ell} (n(\ell) - 1)^+\right) \leq b,$$

which implies

$$d \leq 3b - 2. \tag{6}$$

This checks the property of Lemma 3(b) for the first level. We now need to check the "clade minus subclade" property. The categories given by major leaves ({I, II, III, IV} in Figure 8) are clearly clades in the original cladogram $\hat{\mathbf{t}}$. For an edge-segment ℓ such as (β, γ) in Figure 7, the content pages below the minor leaves attached to ℓ are a subset of the form

$$\text{clade rooted at left child of } \beta \ \setminus \ \text{clade rooted at } \gamma.$$

If this subset is made into a single category in the first level of the index tree, it is clearly of the required form. Alternatively, it may have been made into several categories, as is $\{e_1, \ldots, e_6\}$ in the example. Let us argue in the notation of the example. Because the subset $\{e_1, \ldots, e_6\}$ is ordered so that the position of attachment to the line-segment increases from bottom to top, there is some branchpoint δ of the original cladogram such that the content pages below $\{e_1, e_2\}$ are a subset of the form

$$\text{clade rooted at } \delta \ \setminus \ \text{clade rooted at IV},$$

while the content pages below $\{e_3, e_4, e_5, e_6\}$ are a subset of the form

$$\text{clade rooted at right child of } \beta \ \setminus \ \text{clade rooted at } \delta.$$

So in each case the content pages are of the required form.

We have now constructed the first level of an index tree with the required properties. Proceed recursively. If X is a major leaf of the pruned cladogram, then X can be regarded as the root of a sub-cladogram with content-page probabilities $q(\cdot|X)$. Similarly, for a category that is a subset (z_i) attached to an edge-segment $\ell = (\xi, \zeta)$, one can identify this category with the cladogram

$$\text{cladogram rooted at } \xi \ \setminus \ \text{cladogram rooted at } \zeta$$

and can assign to its content pages the appropriate conditional probabilities. In either case we can apply the first-level construction to the sub-cladogram to get the second level of the index tree; and so on recursively.

One detail of the recursion requires a modification, however. Consider a category, such as $\{c_1, c_2, c_3\}$ in Figure 8, that is of the form "clade minus subclade". When we expand to find the subcladogram for the start of the construction of the second level, we may get a picture like the one in Figure 9: suppose c_2 and c_3 are content pages and c_1 is an index page.

Now at the second level the algorithm might combine $\{s, c_2\}$ as a category. But this is not of the required form, because at the first level the subclade III was removed from

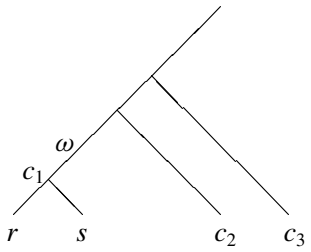

Figure 9.

attachment point ω. However, this difficulty is easily avoided by splitting this category into two categories; since the total number of categories increases by at most 1, the desired bound $d \leq 3b$ remains true because of the slack in (6).

ACKNOWLEDGMENTS I thank my colleague Mike Jordan for helpful comments.

REFERENCES

1. Hyperbolic tree demos, http://www.inxight.com/demos/ht/index.html.
2. D. J. Aldous, Stochastic models and descriptive statistics for phylogenetic trees, from Yule to today, preprint, U.C. Berkeley Statistics Dept., 2000. `www.stat.berkeley.edu/users/aldous/`.
3. L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*, Wadsworth, Pacific Grove, CA, 1984.
4. N. Eldredge and J. Cracraft, *Phylogenic Patterns and the Evolutionary Process*, Columbia University Press, New York, 1980.
5. R. W. Hamming, *Coding and Information Theory*, 2nd ed., Prentice-Hall, Englewood Cliffs, NJ, 1986.
6. IBM Austin Ease of Use group, Web design guidelines, 1999, `www-3.ibm.com/ibm/easy/eou_ext.nsf/Publish/572`.
7. D. E. Knuth, *The Art of Computer Programming*, vol. 3, Addison-Wesley, Reading, MA, 1973.
8. P. J. Lynch and S. Horton, *Web Style Guide*, Yale University Press, New Haven, CT, 1999.
9. S. Martin, Cluster analysis for web site organization, 1999, `www-3.ibm.com/ibm/easy/eou_ext.nsf/LicenceView3/410`.
10. E. Tiller and P. Green, Web navigation: How to make your site fast and usable, in: 5th Conference on Human Factors & The Web, 1999, `zing.ncsl.nist.gov/hfweb/proceedings/proceedings.en.html`.
11. Various, Proceedings of 8th international world wide web conference, *Computer Networks* 31 (1999) 1079–1751.

DAVID J. ALDOUS holds a B.A. and Ph.D. from Cambridge University, and since 1979 has been on the U.C. Berkeley faculty. He is the first recipient of the Loeve Prize in Probability. His research covers a broad range of theoretical and applied probability—rare events, weak convergence, Markov chains, randomized algorithms, random trees and other combinatorial objects—and is further described at www.stat.berkeley.edu/users/aldous.
Department of Statistics, University of California, Berkeley CA 94720
aldous@stat.berkeley.edu