Compressed Web Indexes

Flavio Chierichetti*
Dipartimento di Informatica
Sapienza University of Rome
Via Salaria, 113
Roma, 00198, Italy
chierichetti@di.uniroma1.it

Ravi Kumar
Yahoo! Research
701 First Avenue
Sunnyvale, CA 94089, USA
ravikumar@yahoo-inc.com

Prabhakar Raghavan
Yahoo! Research
701 First Avenue
Sunnyvale, CA 94089, USA
pragh@yahoo-inc.com

ABSTRACT

Web search engines use indexes to efficiently retrieve pages containing specified query terms, as well as pages linking to specified pages. The problem of compressed indexes that permit such fast retrieval has a long history. We consider the problem: assuming that the terms in (or links to) a page are generated from a probability distribution, how well compactly can we build such indexes that allow fast retrieval? Of particular interest is the case when the probability distribution is Zipfian (or a similar power law), since these are the distributions that arise on the web.

We obtain sharp bounds on the space requirement of Boolean indexes for text documents that follow Zipf's law. In the process we develop a general technique that applies to any probability distribution, not necessarily a power law; this is the first analysis of compression in indexes under arbitrary distributions. Our bounds lead to quantitative versions of rules of thumb that are folklore in indexing. Our experiments on several document collections show that the distribution of terms appears to follow a double-Pareto law rather than Zipf's law. Despite widely varying sets of documents, the index sizes observed in the experiments conform well to our theoretical predictions.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Algorithms, Experimentation, Measurements

Keywords

Power law, double-Pareto, index size, compression

1. INTRODUCTION

We study the following setting: suppose that we have n web pages in our search engine, each having an integer ID in $\{1, \ldots, n\}$. We have a collection of *adjacency lists* each consisting of a set of IDs: the adjacency list may denote the pages containing a term (thus we have one list per term), or the pages linking to a page (in this case, one list per page). Suppose further that the members of each list are drawn independently from a probability distribution. We wish to build a data structure that compactly stores all the adjacency lists so as to efficiently respond to a query of the form "enumerate the members of a specified list." The trivial solution of a membership vector (a 0/1 n-vector with a position for each integer in $\{1, \ldots, n\}$, with a 1 denoting that the corresponding integer is in the list and 0 otherwise) is prohibitively expensive since most adjacency lists in practice are extremely sparse. The solution is to use a standard inverted index, that writes down the integers in each list, using space $\Theta(k \log n)$ to store a list with k elements. The literature (e.g., [27]) has developed sophisticated coding schemes for encoding the integers in a list to beat the space used by the standard inverted index. How well do these schemes perform, in theory and in practice?

The frequency of occurrence of terms in documents (aka pages) is the subject of Zipf's law [28], which states that the frequency of the ith most frequent term in a page collection is proportional to $1/i$. It is the best known of the class of so-called *power-law distributions*, in which the ith most frequent item in a set occurs with probability proportional to $1/i^{\alpha}$, where α is a positive real number. Such power laws have been observed in a number of natural, physical, sociological, and computational phenomena [24, 4, 18]. For instance, the number of links to a web page has been observed [11] to follow a power law with $\alpha = 2.1$. Text indexing has a long history of folklore rules (for instance, that inverted indexes require about one-third of the storage of the text being indexed). Can a rigorous analysis based on Zipf justify such folklore rules?

We develop a general analysis technique that applies to lists generated by *any* distribution (not necessarily Zipfian); of course, the theoretical bounds for index size depend on the specific distribution. This is the first analysis of compression in indexes under arbitrary distributions. Experiments with text corpora (Section 3.2) suggest that empirical term distributions are in fact somewhat more intricate than Zipfian distributions. The generality of our analytical technique makes it applicable nevertheless, while simultaneously raising the question of how we can better exploit specific properties of Zipfian distributions.

1.1 Scope and outline of this paper

We consider a *collection* of n *pages*, each of which is a sequence of *terms*. The set of all terms in a collection is known as its *vocabulary* and we denote its cardinality by m. An *in-*

[*] Most of this work was done while the author was visiting Yahoo! Research.

WWW 2009, April 20–24, 2009, Madrid, Spain.
ACM 978-1-60558-487-4/09/04.

verted index is a data structure built from a collection that facilitates query-processing in a search engine for the collection. For each term t in the vocabulary, it maintains the set of pages from the collection that contain t; this set is known as the *posting list* (a DB/IR nomenclature for the adjacency list) for t. From such a representation the search engine can efficiently identify, for instance, all pages containing both of two query terms t_1 and t_2. The reader is referred to [3, 14, 27] for introductory material on inverted indexes.

The space used by an inverted index is a major concern in search engine design, especially in web-scale engines where it is commonplace for the index to reside in memory. Accordingly, the postings lists (which account for the bulk of the space used by an inverted index) are typically stored in a compressed form, albeit one that permits quick decompression at query time. In Section 1.3 below we cover the principal forms of such compression. The central question we study: what is the space used by such a compressed inverted index if the terms in a page are stochastically generated? Manning et al. [14] give a heuristic calculation for estimating the space used by an inverted index for pages that follow Zipf's law; however (as they point out) their calculation makes several assumptions that are technically invalid. Our main results (Theorems 2 and 3) give rigorous and tight stochastic bounds on inverted index space for a variety of compression schemes, when the terms in pages follow an arbitrary distribution (with Double Pareto as a special case). We thus significantly generalize, and make rigorous, the calculation in Manning et al. (Our analysis could also be applied to study the compressibility of web graphs, if the links are generated by a process similar to the terms in our model; we will not focus on web graph compressibility in this paper.) We then study the actual term distributions in several benchmark page collections including news articles and wikipedia (Section 3). We observe that these collections follow double-Pareto distributions, rather than simple power laws. We confirm the fit between our analytical bounds and the actual space used by compressed inverted indexes for these collections (Section 6). Despite a widely varying set of documents, the index sizes observed in the experiments conform well to our theoretical predictions.

1.2 Inverted indexes

As common in inverted indexes, we assume each page has a unique page identifier (ID) that is an integer in $[1, n]$. The assignment of IDs to pages can depend on a variety of factors ranging from age to some quality measure. Then, the *postings list* for a term t is a set of integers denoting the pages containing t. For efficient query processing [14], it is customary for the postings list to be sorted in increasing order of page ID. Given such an increasing sequence of integers $d_1, d_2, d_3, \cdots$ a common technique is to store instead a compressed version of the sequence of *gaps*: $d_1, d_2 - d_1, d_3 - d_2, \cdots$; below, we mention some codes commonly used to compress these gaps. Then, at query time, the original postings list can be reconstructed by traversing and decompressing the sequence of gaps while adding them up.

The storage requirement for inverted indexes stems from two sources: (1) storing the terms in the vocabulary and (2) storing the postings. It is known that the latter component is the dominant use of storage; henceforth we focus on the postings alone.

1.3 Codes for compressing the gaps

For concreteness, we focus on the γ and δ codes for compressing the index; see [27] for more background on this. For the rest of the paper, we use lg to denote log to the base 2.

The γ *code* of $x \in Z^+$ is obtained by representing $x - 2^{\lfloor x \rfloor}$ in binary, but prefixing this by a unary representation of the binary length of x. Thus we need $\lfloor \lg x \rfloor$ for the prefix and $\lfloor \lg x \rfloor$ bits for representing $x - 2^{\lfloor \lg x \rfloor}$. Let $S_\gamma(x)$ denote the number of bits used by the γ-code; we have

$$S_\gamma(x) = 1 + 2\lfloor \lg x \rfloor.$$

The δ *code* of $x \in Z^+$ is obtained by representing $x - 2^{\lfloor x \rfloor}$ in binary, but prefixing this by a γ code of its length. If $S_\delta(x)$ denotes the number of bits used by this encoding scheme, we have

$$S_\delta(x) = 1 + \lfloor \lg(x) \rfloor + 2\lfloor \lg(1 + \lfloor \lg(x) \rfloor) \rfloor.$$

1.4 Notation

We will also use the following notation. $\mathrm{H}(x)$ will denote the *binary entropy* of x:

$$\mathrm{H}(x) = -x \lg x - (1-x) \lg(1-x).$$

$\mathrm{H}_{\alpha,n}$ will denote the *nth generalized harmonic number*:

$$\mathrm{H}_{\alpha,n} = \sum_{i=1}^{n} i^{-\alpha}.$$

2. RELATED WORK

Several theoretical explanations have been proposed to explain Zipf's law [28], most notably by Mandelbrot [13] and Simon [22]. Witten and Bell [26] investigate the quality of the fit obtained by the law in natural language text. Li [12] showed that random texts exhibit Zipf's law-like word frequency distribution. Ha et al. [9] study the extension of Zipf's law to word and character n-grams.

An observation known as *Heaps' law* [10] estimates vocabulary size m as a function of collection size n: $m = m(n) = K(Ln)^\theta = \Theta(L(n)n)^\theta$, where $L(n)$ is the average page length, $0 < \theta < 1$ and $K = O(1)$. Typical values for the parameters K and θ are: $30 \le K \le 100$ and $\theta \approx 0.5$.

Other term distribution models, including the K-mixture and two-Poisson model, are discussed by Manning and Schütze [15]. Williams and Zobel present a detailed study of vocabulary growth in large web collections [25]. van Leijenhorst and van der Weide formally derive Heaps' law from a generalized version of Zipf's law [23]; for a more heuristic derivation, see [2]. Gelbukh and Sidorov [8] observe that the coefficients of Zipf's and Heaps' law are language dependent.

Double-Pareto distributions can be viewed as the juxtaposition of two power laws with differing exponents, for different ranges of ordinate values. The use of double-Pareto distributions to model file size distributions is developed by Reed and Jorgensen [19] and Mitzenmacher [16]. For an extensive use of γ and δ codes to represent a web graph, see the work of Boldi and Vigna [6, 7].

While there have been several works on improving the compression on inverted indexes (e.g., [5, 20, 21]), to the best of our knowledge, there has been neither theoretical nor empirical study on the compression of indexes generated by a simple page model. The heuristic calculation in the [14], which inspired our work, is the closest. Our work can be thought of as making this calculation rigorous.

3. TERM DISTRIBUTION IN REAL DATA

First we study the empirical distribution of terms in real data. We will argue that empirical term distributions are in fact somewhat more intricate than Zipfian distributions.

3.1 Data

For our experiments, we use five collections, namely, the TREC data corresponding to news articles from the Associated Press (`trec.ap`), the TREC Wall Street Journal collection (`trec.wsj`), the Reuters collection (`reuters`), Wikipedia (`wiki`), and random web pages (`web`). We chose these collection for repeatability reasons.

The `trec.ap` collection consists of articles from the Associated Press. It has 316,504 pages and 299,702 terms. The average page length is 338.55 terms. The sum of the posting list lengths is 55,678,330.

The `trec.wsj` collection consists of articles from the Wall Street Journal. It has 173,232 pages and 208,620 terms. The average page length is 387.91 terms. The sum of the posting list lengths is 32,495,850.

The `reuters` collection consists of articles from Reuters. It has 806,791 pages and 399,990 terms. The average page length is 218.51 terms. The sum of the posting list lengths is 96,638,730.

Note that the pages in the above three collections are created by professional writers. For comparison purposes, our last two collections are `wiki` and `web`. The first is `wiki`, the collection of all articles in Wikipedia (crawled in October 2007); these articles are edited by online users. It has 2,373,481 pages and 3,962,599 terms. The average page length is 347.48 terms. The sum of the posting list lengths is 369,622,681.

The second is `web`, which is a set of 1,179,206 web pages. These web pages were sampled at random from a repository at Yahoo!. Terms were extracted from these web pages after parsing them to discard HTML elements. The total number of terms is 2,880,011 and the sum of the posting list lengths is 250,093,064. The average page length is 474.64 terms.

3.2 Empirical term distribution

We first study the size of the vocabulary as a function of the size of the collection. Figure 1 (left) shows this plot for `trec.ap`. We then fit a power law to this plot and observe that Heaps' law holds in `trec.ap` with $\theta = 0.508$. We then study the frequency distribution of this collection. We compute the probability of each term and plot the rank of the term vs the probability. Figure 1 (right) shows this plot for `trec.ap`. We see that the term distribution is clearly not a power law since the plot is not a straight line. We observe a similar phenomenon for the other three collections as well (Figure 2).

Based on this empirical observation, we postulate that the term distribution is a double-Pareto distribution. While it is easy to rule out possibilities such as power law with exponential cutoff or log-normal distribution, it is possible that there could be other candidate distributions that fit the observation. We chose double-Pareto mainly for sake of mathematical tractability; moreover, our experimental results seem to endorse this choice.

3.3 Double-Pareto distribution

A *double-Pareto distribution* is made up of two power laws stitched together at a cut off point C. Specifically, it has three parameters (α, β, C), where $0 \leq \alpha < 1 < \beta$ and $C = \omega(1)$. For $1 \leq k \leq C$, the kth term has probability proportional to $\left(\frac{k}{C}\right)^{-\alpha}$. For $k \geq C$, the kth term has probability proportional to $\left(\frac{k}{C}\right)^{-\beta}$. Double-Pareto distributions offer a neat trade-off between the frequent words (exponent α), the rare words (exponent β), and the ratio of their masses (C), expressed below (proved in Appendix B).

THEOREM 1. *The probability that a randomly drawn term t has rank $r(t) = k$ is*

$$\Pr[r(t) = k] = \begin{cases} (1 \pm o(1)) \frac{(1-\alpha)(\beta-1)}{\beta-\alpha} C^{\alpha-1} k^{-\alpha} & \textit{if } k \leq C, \\ (1 \pm o(1)) \frac{(1-\alpha)(\beta-1)}{\beta-\alpha} C^{\beta-1} k^{-\beta} & \textit{if } k > C. \end{cases}$$

Also, $\Pr[r(t) \leq C] = \frac{\beta-1}{\beta-\alpha} \pm o(1)$, *and* $\Pr[r(t) > C] = \frac{1-\alpha}{\beta-\alpha} \pm o(1)$.

Fitting a double-Pareto distribution to the `trec.ap` (also shown in Figure 1 (right)), we observe that $\alpha = 0.911$, $\beta = 2.014$, and $C = 1900$. This fit was obtained by choosing the parameters of the double-Pareto distribution that minimize the sum of squares error. The mass of the distribution until the cut off point C is 0.777. The corresponding fits for the other four collections is given in Table 1. It is clear that web-based collections `wiki` and `web` are closer to each other and differ from the other non-web collections, in terms of many of the parameters. It is also interesting to note that the double Pareto law parameter β is away from 2 and the Heaps' law parameter θ is away from 0.5 for both `wiki` and `web`. This suggests the vocabulary growth in `wiki` and `web` follows a different pattern than usual text collections by having a flatter tail.

Collection	α	β	C	cdf@C	θ
`trec.wsj`	0.896	2.025	700	0.672	0.467
`reuters`	0.869	1.963	700	0.666	0.504
`wiki`	0.900	1.500	2400	0.719	0.725
`web`	0.935	1.325	1100	0.684	0.689

Table 1: Heaps' (θ) and double-Pareto parameters (α, β, C) for other collections. The β and θ of the web collections (`wiki`, `web`) are significantly different from the others .

4. A PAGE GENERATION MODEL

We now state a simple page generation model. We assume each of the n pages to have the same number of terms $L = L(n)$; we will relax this in Section 7.2. Each page in the collection is independently generated by the following process. Each of the L terms in the page is picked independently according to some fixed probability distribution[1] on the vocabulary. For instance, in the power law distribution with exponent $\alpha > 1$, the ith term in the vocabulary is chosen with probability $p_i \propto i^{-\alpha}$. Since the sampling is done with replacement, the number of distinct terms in a page can be less than L.

Let T be the $m \times n$ Boolean *term–document matrix* that has a 1 in entry (i, j) if the term i is present in the page j; all other entries are 0. Let R_i be the random variable

[1] We are mostly interested in power law and double-Pareto distributions, but our results hold for arbitrary ones.

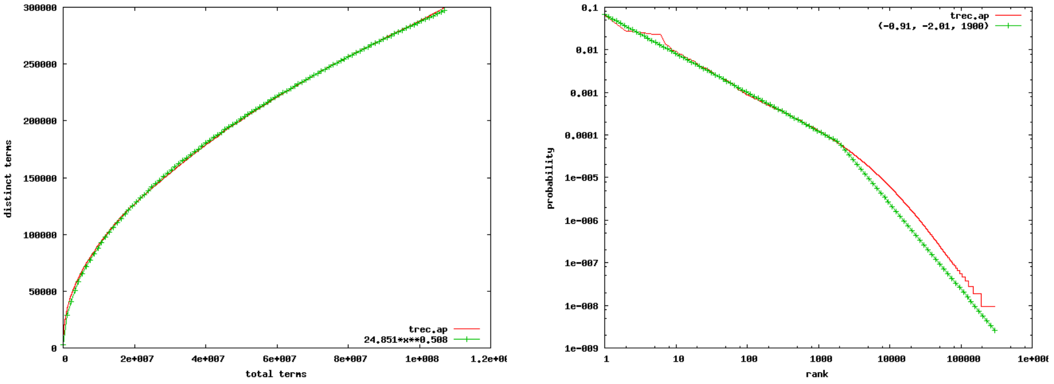

Figure 1: Heaps' law and double-Pareto law for `trec.ap`. The two curves virtually coincide and hence may be hard to distinguish.

representing the ith row of this matrix, i.e., the row corresponding to the ith most frequent term. Thus the rows of T are in order of decreasing term frequency. Note that row i of T is a bit-vector representation of the postings list for term i. When i is a frequent term, this is a space-efficient representation of the postings list; when i is infrequent, the sequence of page IDs (or equivalently, gaps) is more efficient. However, these two representations are equivalent.

5. INDEX SIZE IN THE MODEL

We begin with the main theoretical result of this paper: the δ-compressed index size of pages generated according to the model described in Section 4 is tightly concentrated around its expectation, which is computed in Section 5.1; the concentration is computed in Section 5.2.

5.1 Expected size

We first assume that n pages have been generated by the model described in Section 4. We then measure the compressibility of the term–document matrix of these pages using δ codes. (Our results apply to γ codes as well, but for simplicity of exposition, we do not present these results.)

Let S be the random variable corresponding to size of the δ-compressed term–document matrix of these pages. For a term $1 \leq i \leq m$, let S_i denote the random variable corresponding to the size of row i in the term–document matrix after using δ-codes for compression. Let $P_i = 1 - (1 - p_i)^L$ denote the probability that term i is contained in a given page.

For $1 \leq g \leq n$, let $X_{g,i}$ be the random variable denoting the number of gaps of length g in row i.

Theorem 2. *We have the following:*

$$\begin{aligned} E[X_{g,i}] &= P_i(1-P_i)^{g-1} + P_i^2(1-P_i)^{g-1}(n-g), \\ E[S_i] &= \sum_{g=1}^{n} \left(S_\delta(g) \cdot E[X_{g,i}]\right), \text{ and} \\ E[S] &= \sum_{i=1}^{m} E[S_i]. \end{aligned}$$

Proof. The expressions for $E[S_i]$ and $E[S]$ can be easily proved by the linearity of expectation. Now, we prove the expression for $E[X_{g,i}]$. Consider a generic g and i. Note that none of the first $g-1$ pages can be the terminal endpoint of a length g gap. The gth page is such an endpoint if and only if none of the previous pages was in row i, i.e., if none of them contained the term i. Any other page $j > g$ is an endpoint if and only if it is in the row, along with the $(j-g)$th page, and no other page is in between. Thus the statement follows once again from the linearity of expectation. □

Notice that the above expressions are purely in terms of P_i's, which themselves are determined by the term distribution given by p_i's and the page size L. Also notice that a simple closed-form expression for S in terms of p_i's does not appear plausible; this is true even if p_i's were to follow a power law. Things are made more complicated by the fact that the functions $S_\delta(x)$ and $S_\gamma(x)$ have highly discontinuous behavior when x is small.

5.2 Concentration

While the expectation was computed "row-by-row" of the term–document matrix, for obtaining the concentration bounds, we use a more holistic analysis (using Theorem 7; see Appendix A).

To be able to do this, we first need to make the following very mild assumption. We assume that the number of terms per page is sub-exponential in the number of pages. This is reasonable since in practice, the number of terms per page is much less than the number of pages.

Theorem 3. *If $L = \exp\left(O\left(\frac{n}{\log^4 n}\right)\right)$, the total compressed size S is concentrated, i.e.,*

$$\Pr[|S - E[S]| = o(E[S])] = 1 - O(n^{-c}),$$

for each $c > 0$.

Proof. Recall the page generation process. For each page, we draw L terms (with replacement) from the term distribution. Thus, the random variable S is completely determined by $n \cdot L$ random trials. Note that, no matter how the trials turn out to be, we have $S = \Omega(n)$, as each page contains at least one term (by $L \geq 1$) and the minimum gap cost is 1.

We split the analysis into the following two cases.

(1) Case $L = o(n/\log^3 n)$, i.e., the page is small. We apply Theorem 7, letting its X be our total index size S and its X_i, $i = 1, \ldots, n \cdot L$, be the draw of the ith term; we choose $b_i = O(\log n)$ and $D = O(n \cdot L \cdot \log^2 n)$. The crucial

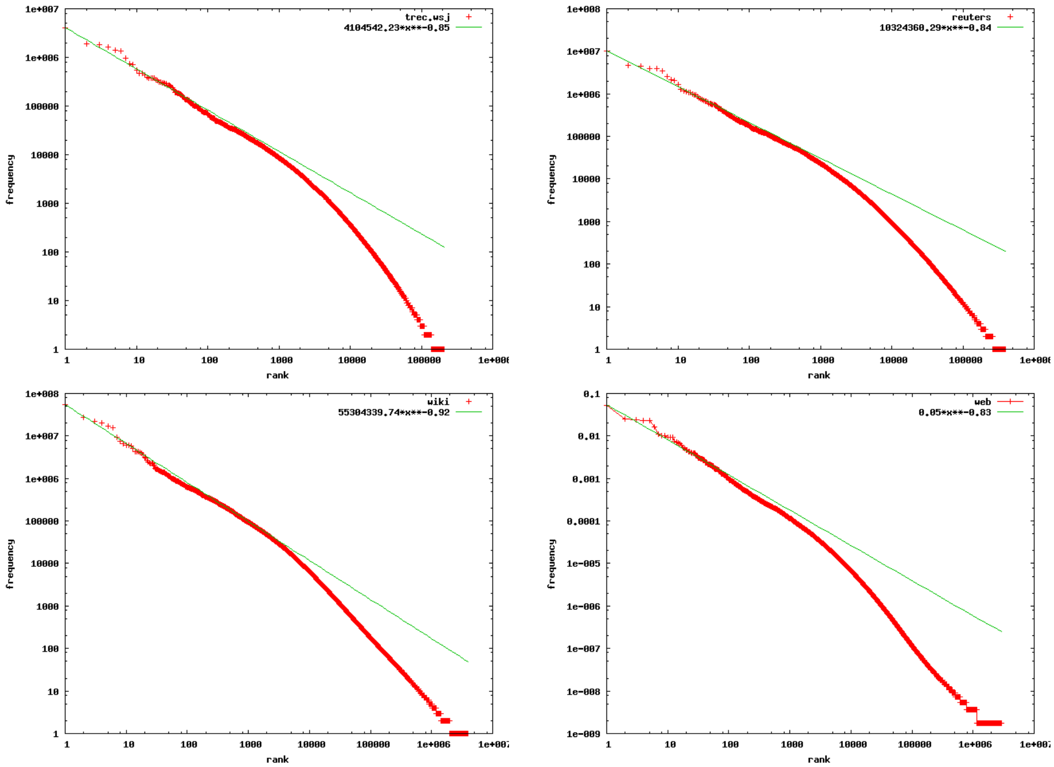

Figure 2: Term distribution for trec.wsj, reuters, wiki, web, suggesting double-Pareto.

observation is that changing a trial could change S by at most a value of the order of the maximum encoding length of a gap, i.e., by at most $O(\log n)$. This follows because a change may only add a new gap and/or remove a gap induced by the previous outcome of the trial. In any case, the maximum difference between the old and the new S is bounded by $O(\log n)$.

Let $t' = n/(L \log^3 n)$. Since $L = o(n/\log^3 n)$, we have $t' = \omega(1)$. Let $t = \sqrt{E[S] \cdot n/t'}$. Using $S = \Omega(n)$, we have $t = o(E[S])$. Then, the statement follows from Theorem 7 by choosing $\mathcal{F}$ to be the whole probability space so that $\Pr[\mathcal{F}] = 1$. (This case could have been proved with the method of bounded differences, but we use Talagrand's inequality to have a uniform presentation.)

(2) Case $L = \omega(n/\log^4 n)$, i.e., the page is large. Call a term i "frequent" if $p_i > 100\frac{\log L}{L}$ and "rare" otherwise. Let P_{rare} be the sum of the p_i's of rare terms. Let $S_{\mathsf{freq}}, S_{\mathsf{rare}}$ be the random variables representing the encoding size of, respectively, the frequent and rare terms' rows. We have $S = S_{\mathsf{freq}} + S_{\mathsf{rare}}$.

Note that by the Chernoff bound, with probability at least $1-O(n^{-2})$, each frequent term will be chosen in every page.[2] The expected number of times each page chooses every such term is at least $100 \log L = (1 - o(1)) \cdot 100 \log n$. So S_{freq} is concentrated.

Let us now consider rare terms and show that S_{rare} is concentrated.

(2a) Case $P_{\mathsf{rare}} \leq 100\frac{\log L}{L}$. Our plan is to bound the deviation of S_{rare} via Theorem 7, as before (this time, though, we choose $X = S_{\mathsf{rare}}$). We do so by first proving the following, where the weights b_i refer to those in Theorem 7.

Lemma 4. *With probability $1 - O(n^{-2})$, only $O(n \log L)$ of the $n \cdot L$ independent random variables making up the term–document matrix will require weights $b_i = O(\log n)$; all the other weights can be set to 0.*

Proof. Let ξ be the random variable denoting a term in the document that is generated according our model. Suppose ξ corresponds to a frequent term. Then ξ makes zero contribution to S_{rare} and so modifying ξ cannot decrease the value of S_{rare}. On the other hand, if ξ was for a rare term, then we just apply the trivial bound $O(\log n)$, i.e., modifying ξ cannot cost more than $O(\log n)$ towards the compressed index size.

By the Chernoff bound, and the union bound, each single page will choose at most $O(\log L)$ rare terms, with probability at least $1 - O(n^{-2})$. Thus the number of rare terms is at most $O(n \log L)$ and the proof is complete. □

Given Lemma 4, we apply Theorem 7 with

$$D = O(n \log L \log^2 n) = O\left(\frac{n^2}{\log^2 n}\right)$$

and

$$t = \Theta(\sqrt{D \log n}) = O\left(\frac{n}{\sqrt{\log n}}\right),$$

to show that with probability $1 - n^{-2}$, the deviation of S_{rare} from $E[S_{\mathsf{rare}}]$ is at most by $o(n)$. (It might very well be that

[2] The -2 exponent can be decreased and depends on the lower bound on the p_i's of the frequent terms. We are not interested in optimizing this constant, however.

$E[S_{\mathsf{rare}}]$ is asymptotically much smaller than this deviation, but since $S = \Omega(n)$ and we interested in showing the concentration of S, the above error term suffices.)

(2b) Case $P_{\mathsf{rare}} > 100\frac{\log L}{L}$.

Note that, by the Chernoff bound, each single page will make $\Theta(P_{\mathsf{rare}} \cdot L)$ choices among the rare terms with probability at least $1 - O(n^{-2})$.

Thus, in Theorem 7 we will have at most $O(n \cdot P_{\mathsf{rare}} \cdot L)$ different b_i's upper bounded by $O(\log n)$ and the rest are set to 0. Now, choosing $D = O(n \cdot P_{\mathsf{rare}} \cdot L \cdot \log n)$ and $t = \Theta(\sqrt{D \log n})$ will then give us a high probability statement with error term $\epsilon = O(\sqrt{n \cdot P_{\mathsf{rare}} \cdot L} \cdot \log n)$.

It only remains to be shown $\epsilon = o(E[S_{\mathsf{rare}}])$. Again by domination and the Chernoff bound, with probability at least $1-O(n^{-2})$, there is no page that chooses any single rare term more than $O(\log L)$ times. (Note that for the union bound on the pages to work out, we need $\log L = \Omega(\log n)$, which is ensured by the assumption $L = \omega(n/\log^4 n)$.) Thus, with probability at least $1 - O(n^{-2})$, no page will have less than $\Omega(\frac{P_{\mathsf{rare}} \cdot L}{\log L})$ rare terms. This implies that $E[S_{\mathsf{rare}}] \geq n \cdot P_{\mathsf{rare}} \cdot \frac{L}{\log L}$, which is asymptotically larger than the error term ϵ. □

The reader may wonder why we had to use this particular version of Talagrand's inequality to prove concentration. The difficulty is caused by Case (2a) of the previous proof, i.e., we needed to bound the error term of a possibly small quantity that depended on *many* random variables. With Lemma 4 we showed that, while the number of random variables was large, with high probability only few of them had a role in determining our quantity — the version of Talagrand's inequality we used allowed us to prove concentration from that observation.

6. EXPERIMENTAL RESULTS

In this section we present the results of our experiments on the five data-sets. We present three sets of the numbers corresponding to various compression schemes. For each collection, we first randomly permute the page ids in order to remove any biases associated with the page numbering.

Predicted and actual estimates. The specific compression schemes we consider are the γ and δ codes, with and without byte alignment.[3] The byte-aligned versions are denoted with a suffix -8. The first set of numbers (called *Pareto-fit*) are the theoretical estimates, using our estimation of the parameters of the double-Pareto law, namely, α, β, and C. We compute the estimates by applying Theorem 2; recall that Section 5.2 shows that the actual bounds are tightly concentrated around the expectation. The second set of numbers (called *frequency-fit*) are the theoretical estimates using the observed term frequencies. Recall that our analysis is general enough to handle any distribution, not just power laws. We set up a distribution using the term frequencies and once again apply Theorem 2 to estimate the compressed index size. For both these predicted values, for computational ease, we assume that all pages have the same length. The third set of numbers are the *actual* values, obtained by indexing the collection.

Measures. We consider the following measures that we use to compare the theoretical and empirical estimates. In the following tables, S denotes compressed index size and U denotes uncompressed index size; thus S/U is the *compression ratio*. The next two rows give the compression ratios for the index before and after the cutoff C. The final row gives the fraction of the index size used by the top 100 terms (most probably, the stop-words).

Results. The results are presented in Table 2. From the entries in the tables, it can be seen that overall there is a reasonable agreement between both the analytical predictions and actual index metrics. Even though there is arguably a little discrepancy between the predicted values and the actual values, the closeness of the values themselves should be viewed as quite striking. This is because the actual values are computed for the pages in `trec.ap`, `trec.wsj`, `reuters`, `wiki`, and `web` collections, *not* for the models derived from them. The discrepancy is mainly due to two simplifications in our analysis, namely, that all the pages are of the same length (for computational ease) and the pages and the terms in a page are generated independently (model assumption for analytical tractability). Clearly these assumptions do not hold in practice.

Nevertheless, in most cases, the frequency-fit estimates are closer to the actual values than the Pareto-fit estimates; this is not surprising given that the double Pareto fit is only meant to be an approximation to the term distribution. So, even though the `wiki` and `web` pages seemed to have different vocabulary growth mechanisms, they all are comparable in terms of compression ratios. Notice that the compression ratios are worse than what is usually observed in practice. This is because the page ids were randomly permuted; to achieve the best compression often requires a careful page ordering [5, 20, 21]. In fact, our experiments suggest that without a good page ordering, it is hard to obtain good compression ratios; this is similar in spirit to the work on web graph compression [6, 7], where locality in url ordering is exploited for improving the compression.

The values in the tables also show that compression is significant below the cutoff C, and further that there is fairly good agreement between prediction and reality. The compression is dramatic below the top 100 terms (perhaps the stop-words). It is interesting to note that `wiki` and `web` achieve far more compression in this regime than the other three non-web collections.

Figure 3 shows the compression obtained for each posting list using Pareto-fit, frequency-fit, and the actual values. Once again the actual values are more or less aligned with the predicted values.

7. DISCUSSIONS

7.1 When is the entropy bound achieved?

In this section we analyze the conditions under which the compressed size of a row of the term–document matrix approaches the entropy bound. We prove that δ-codes achieve the entropy bound for row i for $\omega\left(\frac{\log n}{n}\right) = P_i = o(1)$. In practice, this means that we can approximate the expected size of those[4] rows by

$$(1 - o(1))n \cdot \mathrm{H}(P_i) \leq E[S_i] \leq (1 + o(1))n \cdot \mathrm{H}(P_i).$$

[3] Byte aligning a bit string of size k means padding $(8 - k \bmod 8)$ zero bits to the end; see [27] for the practical benefits of byte alignment.

[4] For other rows, we can use the formulas of Theorem 2.

	Pareto-fit				Frequency-fit				Actual			
	γ	δ	γ-8	δ-8	γ	δ	γ-8	δ-8	γ	δ	γ-8	δ-8
`trec.ap`												
S (MB)	61.851	59.906	103.79	99.912	75.368	71.120	117.75	111.45	58.520	54.626	87.723	82.467
S/U	.360	.349	.478	.460	.424	.399	.524	.496	.489	.457	.550	.517
$S_{\leq C}/U_{\leq C}$	.251	.263	.399	.399	.261	.272	.406	.406	.313	.321	.427	.427
$S_{>C}/U_{>C}$	.762	.663	.770	.686	.774	.673	.778	.689	.857	.739	.807	.706
$S_{\leq 100}/S$	.070	.083	.177	.184	.056	.068	.147	.155	.065	.079	.140	.149
`trec.wsj`												
S (MB)	26.777	26.777	50.793	49.366	43.243	41.160	69.574	66.188	33.181	31.217	50.077	47.256
S/U	.299	.299	.426	.414	.417	.397	.503	.479	.503	.474	.538	.508
$S_{\leq C}/U_{\leq C}$	.181	.201	.349	.349	.197	.216	.356	.356	.259	.280	.377	.377
$S_{>C}/U_{>C}$	.620	.567	.636	.591	.664	.599	.668	.616	.751	.670	.701	.640
$S_{\leq 100}/S$	.091	.103	.229	.235	.055	.066	.151	.159	.063	.077	.132	.140
`reuters`												
S (MB)	95.073	93.311	165.95	160.17	139.79	130.43	214.06	201.15	107.227	98.967	157.464	146.815
S/U	.313	.307	.455	.440	.422	.394	.539	.506	.465	.429	.569	.531
$S_{\leq C}/U_{\leq C}$	.198	.215	.368	.368	.214	.230	.380	.380	.249	.262	.402	.402
$S_{>C}/U_{>C}$	.637	.567	.701	.640	.687	.603	.741	.667	.730	.634	.775	.689
$S_{\leq 100}/S$	.113	.131	.242	.251	.073	.088	.164	.175	.081	.100	.155	.167
`wiki`												
S (MB)	731.264	662.282	1075.9	992.58	754.87	680.68	1085.8	996.73	505.863	448.814	692.737	627.040
S/U	.431	.391	.582	.537	.455	.410	.600	.551	.546	.485	.655	.592
$S_{\leq C}/U_{\leq C}$	.239	.247	.417	.416	.249	.256	.422	.422	.320	.319	.463	.462
$S_{>C}/U_{>C}$	.803	.669	.901	.769	.789	.661	.888	.759	.889	.735	.944	.791
$S_{\leq 100}/S$	.044	.055	.118	.128	.041	.051	.105	.114	.049	.063	.096	.106
`web`												
S (MB)	466.60	416.47	683.54	622.94	404.05	373.82	621.855	581.89	290.626	265.198	420.073	389.012
S/U	.464	.414	.595	.542	.403	.373	.543	.508	.487	.444	.587	.543
$S_{\leq C}/U_{\leq C}$	.186	.201	.369	.369	.184	.199	.369	.369	.243	.257	.398	.398
$S_{>C}/U_{>C}$	.781	.657	.852	.739	.661	.577	.747	.671	.747	.644	.788	.699
$S_{\leq 100}/S$	.035	.045	.110	.121	.041	.050	.126	.135	.049	.062	.112	.121

Table 2: Predicted and actual index sizes for `trec.ap`, `trec.wsj`, `reuters`, `wiki`, and `web`.

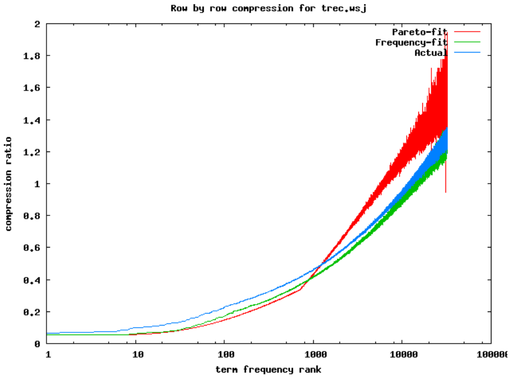

Figure 3: Term-by-term compression for `trec.wsj` using γ codes.

The lower bound is trivial, since each row is compressed by itself. The entropy of the random variable representing the row is exactly $n \cdot \mathrm{H}(P_i)$ (as each row i is completely determined by a sequence of n independent P_i-biased coin flips), and the entropy is a lower bound for the expected compression size of any code.

Lemma 5. *Let i be a row such that $\omega\left(\frac{\log n}{n}\right) = P_i = o(1)$. Then, for all $c > 0$, $\Pr[S_i \leq (1+o(1))n\mathrm{H}(P_i)] = 1 - O(n^{-c})$.*

Proof. If the length g of a gap is $\omega(1)$, then its δ-compressed size is $\lg g + o(\log g)$. We will start by proving that at least a constant fraction of the gaps will have a size $\omega(1)$ w.h.p. This will let us disregard constant length gaps and will allow us to use the previous asymptotic formula for the remaining gaps' lengths.

Let $p = P_i$ and let $\ell = \lfloor \frac{1}{p} \rfloor + \lceil \frac{1}{p} \rceil$, and let k be the maximum integer such that $\ell k \leq n$, that is $\frac{n}{\ell} - 1 \leq k \leq \frac{n}{\ell}$. Subdivide the sequence $(0, \ldots, n-1)$ in $k+1$ subsequences, each of length ℓ, with the possible exception of the last (having length $n - \ell k \leq \ell$). Then, for $0 \leq i < k$, the ith subsequence will be $\sigma_i = (i \cdot \ell, \ldots, (i+1) \cdot \ell - 1)$. Subdivide it again in $\sigma_i' = (i \cdot \ell, \ldots, i \cdot \ell + \lfloor \frac{\ell}{2} \rfloor - 1)$ and $\sigma_i'' = (i \cdot \ell + \lfloor \frac{\ell}{2} \rfloor, (i+1) \cdot \ell - 1)$, of length $\lfloor \frac{\ell}{2} \rfloor = \lfloor \frac{1}{p} \rfloor$ and $\lceil \frac{\ell}{2} \rceil = \lceil \frac{1}{p} \rceil$ respectively.

The probability that no element is chosen in σ_i' is

$$(1-p)^{\lfloor \frac{1}{p} \rfloor} \geq (1-p)^{\frac{1}{p}} = q',$$

and for large n, $q' \approx e^{-1}$. The probability that at least an element is chosen in σ_i'' is

$$1 - (1-p)^{\lceil \frac{1}{p} \rceil} \geq 1 - (1-p)^{\frac{1}{p}} = q'',$$

and for large n, $q'' \approx 1 - e^{-1}$.

Since these two events are independent, the probability of them happening together is $q = q' \cdot q'' = \Theta(1)$; let X_i be the indicator variable of their intersection. If $X_i = 1$, then at least a gap of length $\Omega(\frac{1}{p})$ will end in some element of σ_i. Thus, $X = \sum_{i=1}^{k} X_i$ is a lower bound for the number of gaps of length $\Omega(\frac{1}{p})$.

There exists at least $\frac{n}{\ell} - 1$ different σ_i's, with $\frac{n}{\ell} - 1 = \Theta(np)$. Thus $E[X] = \Theta(np) = \omega(\log n)$. As the X_i are binary i.i.d. random variables, we can apply the Chernoff bound to their sum X,

$$\Pr[X \leq \frac{E[X]}{2}] = \exp\left(-\Omega(E[x])\right) = \exp(-\omega(\log n)) = O(n^{-c}),$$

for $c > 0$. The number of gaps of length $\Omega(1/p)$ will be, w.h.p., at least $\Omega(np)$. This will allow us to disregard, in the total compression size, the contribution of constant length gaps.

Let Y denote the random variable counting the number of gaps (or pages) in the row. Then, $E[Y] = np$. By the Chernoff bound,

$$\Pr[|Y - np| > \sqrt{3c\, n\, p \log n}] \le 2\exp(-c\log n) = 2n^{-c}.$$

As $p = \omega(\log n/n)$, we have that, w.h.p., $Y = np \pm o(np)$.

In the rest of the proof we assume that both $Y = np \pm o(np)$ and $X = \Omega(np)$. The intersection of these two events has probability at least $1 - O(n^{-c})$, for all $c > 0$.

Consider the gap lengths $g_1, \ldots, g_Y$ (recall that they are random variables) induced by these Y pages. Obviously $\sum_{i=1}^{Y} g_i \le n$.

In particular let L be the set of indices of the "little" gaps and B the set of indices of the "big" gaps — say, for each $\ell \in L$, $g_\ell < \log \frac{1}{p}$, while for each $b \in B$, $g_b > \sqrt{\frac{1}{p}}$.

Note that $X = \Omega(np)$ implies $|B| = \Omega(np)$. Also, by $Y = np \pm o(np)$, we obtain $|L| = O(np)$. The total compression size of the little gaps is asymptotically smaller than the total compression size of the big gaps, as $|B| \,/\, |L| = \Omega(1)$, and each of the big gaps has a $\omega(\log \frac{1}{p})$ compression size, while each of the little gaps has a $O(\log\log \frac{1}{p})$ compression size. That is,

$$\sum_{\ell\in L} S_\delta(g_\ell) < o\left(\sum_{b\in B} S_\delta(g_b)\right) \le o\left(\sum_{i=1}^{Y} S_\delta(g_i)\right).$$

The total space needed by the coding will thus be

$$\begin{aligned} S_i = \sum_{i=1}^{Y} S_\delta(g_i) &= \left(\sum_{t\notin L} S_\delta(g_t)\right) + \left(\sum_{\ell\in L} S_\delta(g_\ell)\right) \\ &= \left(\sum_{t\notin L} \lg g_t + o(\log g_t)\right) + o(S_i) \\ &= \sum_{t\notin L} \lg g_t + o(S_i) \\ &\le \sum_{i=1}^{Y} \lg g_i + o(S_i) \\ &= \lg \prod_{i=1}^{Y} g_i + o(S_i). \end{aligned}$$

The arithmetic mean-geometric mean inequality states, given a sequence $x_1, \ldots, x_t$ of non-negative reals,

$$\sqrt[t]{x_1 \cdot x_2 \cdots x_t} \le \frac{x_1 + x_2 + \cdots + x_t}{t}.$$

By applying the inequality to the g_i's we get

$$\prod_{i=1}^{Y} g_i \le \left(\frac{1}{Y}\sum_{i=1}^{Y} g_i\right)^{Y} \le \left(\frac{n}{Y}\right)^{Y},$$

thus, we can obtain

$$\begin{aligned} S_i &\le Y \lg \frac{n}{Y} + o(S_i) = (np \pm o(np)) \lg \frac{1}{p \pm o(p)} + o(S_i) \\ &\le (1 + o(1)) n \mathrm{H}(p). \end{aligned}$$

which proves our claim. □

We now address the question: how tight is Lemma 5, i.e., do δ-codes achieve the entropy bound outside the range of P_i's given by Lemma 5? Notice that the upper bound is strict: if $P_i = 1$, then the entropy of the row is 0 while we use $\Omega(n)$ bits to represent it. (If $P_i = c$, for some constant $c < 1$, we would use a number of bits that is larger than the entropy bound by some function of c). The lower bound can possibly be improved a little, but it cannot be entirely removed. If $P_i \le n^{-c}$, for $c > 1$, then $n\mathrm{H}(P_i) \ge c \cdot n^{1-c} \log n$; on the other hand, $E[S_i] = (1 + o(1)) \cdot n^{1-c} \log n$, since we pay at most $(1+o(1))\cdot \log n$ bits per page, and the expected number of pages is $\le n^{1-c}$). Thus the ratio between the expected length of our encoding of the row and its entropy is $\le 1/c$.

The above might appear to contradict the entropy lower bound. However, this is not the case. Indeed, if a term is not chosen by any page, we completely ignore that term while building the index, we don't encode its row at all (instead of encoding an *empty* row). So if $P_i \le n^{-c}$, for $c > 1$, with probability $1 - o(1)$, row i will be empty and we won't need to encode it. On the other hand, if $P_i = \omega(\log n/n)$, with high probability the row will be non-empty, i.e., we will have to encode it and the entropy bound can be used to lower bound the encoding size.

7.2 Varying length pages

So far, we have assumed that the length of each page is fixed to be some integer L. We remark here that the results hold even if we allow pages to have varying sizes, under mild assumptions. Suppose pages are generated as follows. For each $i = 1, \ldots, n$, let L_i be a fixed integer. Page i is generated by sampling L_i terms from the term distribution. Suppose each $L_i = O(n^{1-\epsilon})$ (or even $\sum_{i=1}^{n} L_i = o(n^2/\log^3 n)$), then we can obtain the same concentration result of Theorem 3: the index size is concentrated around its expectation. We omit the details in this version.

The expression for $E[X_{g,i}]$ would become significantly more complicated, though. Indeed, two different pages may have different probabilities of containing the same term. So $E[X_{g,i}]$ would become

$$\begin{aligned} E[X_{g,i}] \;=\; & \sum_{k=g+1}^{n} \left[P_{i,k-g} \cdot P_{i,k} \prod_{j=k-g+1}^{k-1} (1 - P_{i,j}) \right] + \\ & + P_{i,g} \prod_{j=1}^{g-1} (1 - P_{i,j}), \end{aligned}$$

where $P_{i,j}$ is the probability that page j contains term i: $P_{i,j} = 1 - (1 - p_i)^{L_j}$.

7.3 Computational costs

We compare the computational costs of answering a query (i.e., a conjunction or disjunction of terms) using an uncompressed vs a compressed index. There are (at least) two ways of assessing the computational efficiency: the number of bits read and the number of basic operations performed by the CPU (say, arithmetic, logical, shift operations). It is easy to see that for a k-term query, where the terms are generated independently from the term distribution, we have:

LEMMA 6. *The expected number of bits read is* $\sum_{i=1}^{n}(1 - (1 - p_i)^k) E[S_i]$.

Here, S_i is the size of the posting list for term i. This follows from the independence assumption and concrete bounds for compressed indexes can be obtained using Theorem 2. (Unfortunately, one can show that this random variable is not concentrated.) This brings us to an interesting question: how to achieve the best computation-compression trade-off?

Suppose the term–document matrix is stored in a sparse, but uncompressed form (each gap is encoded using $\lceil \lg n \rceil$ bits). Then, reading a word would require $\lceil \lg n \rceil$ operations. Suppose we compress the gaps using δ-codes. Decoding a gap of length ℓ uses $\Theta(\log \ell)$ operations. Asymptotically, this could be smaller than the cost paid for uncompressed gaps — except when the gaps are very large, $\Omega(n)$ say: in this case leaving them uncompressed is a better choice.

To summarize, to achieve a good computational-compression trade-off, a reasonable strategy could be: (i) either remove very frequent terms (stop words) or store their rows uncompressed in term–document matrix, (ii) store the rows corresponding to rare terms without compression, and (iii) δ-compress the gaps of the posting lists of the other terms (that, overall, make up the largest part of the corpus).

8. CONCLUSIONS

In this paper we analyze the compressibility of a term–document matrix, where the pages are generated by a simple model in which the terms are chosen independently from a given distribution. We show that the size of the index is tightly concentrated around its expectation. We then analyze five data-sets and use a double-Pareto law to model the distribution of terms in the pages. The parameters of the double-Pareto distribution are used to predicate various aspects of the size of the compressed index. We show that the predicted index sizes closely match the actual index sizes. It will be interesting to consider more nuanced page generation models and carry out similar analysis.

Although our work was primarily motivated by compressibility of web indexes under the Zipfian term distribution, our analysis did not use any specific properties of the Zipf distribution. The general problem of analyzing algorithms and data structures under power law inputs is, however, an important area that demands further study. First, it appears that in a wide variety of practical settings, power laws are quite common. Second, analyzing algorithms and data structures under power law inputs demands the development of new tools, beyond what is available in the classical probabilistic analysis of algorithms. This is because classical probabilistic analysis depends substantially on inputs drawn identically and independently from a common generative process. On the other hand, such i.i.d. inputs do not yield power laws. Accordingly, the analysis of algorithms and data structures with power law inputs demands new tools in probabilistic analysis.

9. REFERENCES

[1] T. M. Apostol. *Introduction to Analytic Number Theory.* Springer-Verlag, 1976.

[2] R. Baeza-Yates and G. Navarro. Block-addressing indices for approximate text retrieval. *Journal of the American Society for Information Science*, 51(1):69–82, 2000.

[3] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval.* Addison Wesley, 1999.

[4] A.-L. Barabasi. *Linked: How Everything is Connected to Everything Else and What It Means.* Penguin Group, 2003.

[5] D. Bladford and G. Blelloch. Index compression through document reordering. In *Proceedings of the Data Compression Conference*, pages 342–351, 2002.

[6] P. Boldi and S. Vigna. The Webgraph framework i: Compression techniques. In *Proceedings of the 13th International Conference on World Wide Web*, pages 595–602, 2004.

[7] P. Boldi and S. Vigna. The Webgraph framework ii: Codes for the world-wide web. In *Data Compression Conference*, 2004.

[8] A. Gelbukh and G. Sidorov. Zipf and Heaps laws' coefficients depend on language. In *Proceedings of the 2nd International Conference on Computational Linguistics and Intelligent Text Processing*, pages 332–335, 2001.

[9] L. Q. Ha, E. I. Sicilia-Garcia, J. Ming, and F. J. Smith. Extension of Zipf's law to word and character n-grams for English and Chinese. *Computational Linguistics and Chinese Language Processing*, 8(1):77–102, 2003.

[10] H. S. Heaps. *Information Retrieval: Computational and Theoretical Aspects.* Academic Press, New York, 1978.

[11] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Trawling the Web for emerging cyber-communities. *Computer Networks*, 31(11–16):1481–1493, 1999.

[12] W. Li. Random texts exhibit Zipf's-law-like word frequency distribution. *IEEE Transactions on Information Theory*, 38(6):1842–1845, 1992.

[13] B. Mandelbrot. An information theory of the statistical structure of language. In W. Jackson, editor, *Communication Theory*, pages 486–502. Academic Press, 1953.

[14] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval.* Cambridge University Press, 2008.

[15] C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing.* MIT Press, Cambridge, MA, 1999.

[16] M. Mitzenmacher. Dynamic models for file sizes and double Pareto distributions. *Internet Mathematics*, 1(3):305–333, 2003.

[17] M. Molloy and B. Reed. *Graph Coloring and the Probabilistic Method.* Springer-Verlag, 2002.

[18] M. Newman, A.-L. Barabasi, and D. J. Watts. *The Structure and Dynamics of Networks.* Princeton University Press, 2006.

[19] W. J. Reed and M. Jorgensen. The double Pareto-lognormal distribution - A new parametric model for size distributions. *Communications in Statistics: Theory and Methods*, 33(8):1733–1753, 2004.

[20] W.-Y. Shieh, T.-F. Chen, J. J.-J. Shann, and C.-P. Chung. Inverted file compression through document identifier reassignment. *Information Processing and Management*, 39(1):117–131, 2003.

[21] F. Silvestri, R. Perego, and S. Orlando. Assigning document identifiers to enhance compressibility of web search indexes. In *Proceedings of the Symposium on Applied Computing*, pages 600–605, 2004.

[22] H. A. Simon. On a class of skew distribution functions. *Biometrika*, 42:425–440, 1955.

[23] D. C. van Leijenhorst and T. P. van der Weide. A formal derivation of Heap's law. *Information Sciences*, 170:263–272, 2005.

[24] D. Watts. *Six Degrees: The Science of a Connected Age.* W. W. Norton, 2003.

[25] H. E. Williams and J. Zobel. Searchable words on the web. *International Journal on Digital Libraries*, 5(2):99–105, 2005.

[26] I. H. Witten and T. C. Bell. Source models for natural language text. *International Journal Man-Machine Studies*, 32(5):545–579, 1990.

[27] I. H. Witten, A. Moffat, and T. C. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images.* Morgan Kaufmann, 1999.

[28] G. K. Zipf. *Human Behavior and the Principle of Least Effort.* Addison-Wesley, Cambridge MA, 1949.

APPENDIX

A. TAIL BOUNDS

To prove concentration bounds (i.e., to prove that random variables are strongly concentrated around their expectation), we use the following version of Talagrand's inequality [17].

THEOREM 7 (TALAGRAND'S INEQUALITY). *Let X be a non-negative random variable, not identically 0, determined by n independent random variables $X_1, \ldots, X_n$ such that $X = f(X_1, \ldots, X_n)$. Fix some $D > 0$, and let $\mathcal{F}$ be the event that for the outcome $x = (x_1, \ldots, x_n)$ of the trials, there exists a list of non-negative weights $b_1, \ldots, b_n$ such that*

1. *$\sum_{i=1}^{n} b_i^2 \le D$, and*
2. *for any outcome y, it holds $X(y) \ge X(x) - \sum_{x_i \neq y_i} b_i$.*

Then, for any $0 \le t \le E[X]$,

$$\Pr[|X - E[X]| > t + 60\sqrt{D}] \le 4 \exp\left(-\frac{t^2}{8D}\right) + 2(1 - \Pr[\mathcal{F}]).$$

We will also use the well-known Chernoff bound to prove the concentration bounds.

THEOREM 8 (CHERNOFF BOUND). *Let $X_1, \ldots, X_n$ be i.i.d. Bernoulli random variables, 1 with probability p and 0 otherwise. Let $X = \sum_{i=1}^{n} X_i$. Then*

$$\Pr[|X - np| > t] < 2 \exp\left(-\frac{t^2}{3np}\right).$$

B. PROOF OF THEOREM 1

PROOF. Let us use H to denote the "head" of the distribution, $H = \sum_{k=1}^{C} \left(\frac{k}{C}\right)^{-\alpha}$.

If $h = \sum_{k=1}^{C} k^{-\alpha}$, then $H = C^{\alpha} h$. By applying simple calculus,

$$\frac{C^{1-\alpha} - 1}{1 - \alpha} \le \int_0^C (x+1)^{-\alpha} dx \le h \le 1 + \int_1^C x^{-\alpha} dx \le \frac{C^{1-\alpha}}{1 - \alpha}$$

we obtain

$$h = \frac{C^{1-\alpha}}{1 - \alpha} \pm O(1),$$

thus

$$H = \frac{C}{1 - \alpha} \pm O(C^{\alpha}) = \frac{C}{1 - \alpha} \pm o(C).$$

Now let us consider the "tail" $T = \sum_{k=C+1}^{m} \left(\frac{k}{C}\right)^{-\beta}$. Let $t = \sum_{k=C+1}^{m} k^{-\beta}$.

It is known [1] that, for $\beta > 1$,

$$\sum_{k=1}^{s} k^{-\beta} = \zeta(\beta) - \frac{s^{1-\beta}}{\beta - 1} \pm O(s^{-\beta}).$$

This implies

$$\begin{aligned} t &= \sum_{k=1}^{m} k^{-\beta} - \sum_{k=1}^{C} k^{-\beta} \\ &= \zeta(\beta) - \frac{m^{1-\beta}}{\beta - 1} \pm O(m^{-\beta}) - \zeta(\beta) + \frac{C^{1-\beta}}{\beta - 1} \pm O(C^{-\beta}) \\ &= \frac{C^{1-\beta}}{\beta - 1} \pm O(m^{1-\beta} + C^{-\beta}), \end{aligned}$$

and

$$T = C^{\beta} t = \frac{C}{\beta - 1} \pm O(m^{1-\beta} C^{\beta} + 1) = \frac{C}{\beta - 1} \pm o(C).$$

where the third equality is justified by $m^{1-\beta} = o(C^{1-\beta})$, which holds as $C = o(m)$.

The normalizing factor of the probabilities is $(T + H)^{-1}$. That is,

$$\begin{aligned} (T + H)^{-1} &= \left(\frac{C}{1 - \alpha} + \frac{C}{\beta - 1} \pm o(C)\right)^{-1} \\ &= \left(\frac{C(\beta - \alpha) \pm o(C)}{(1 - \alpha)(\beta - 1)}\right)^{-1} \\ &= \frac{(1 - \alpha)(\beta - 1)}{C(\beta - \alpha) \pm o(C)} \\ &= \frac{1}{C} \cdot \frac{(1 - \alpha)(\beta - 1)}{\beta - \alpha} \pm o(C^{-1}). \end{aligned}$$

This proves the first part of the theorem.

The probability that a newly drawn term t is chosen within the first c terms (that is, the probability that it happens to be in the "head") is

$$\begin{aligned} \Pr[r(t) \le C] &= \frac{H}{T + H} \\ &= \frac{\frac{C}{1-\alpha} \pm o(C)}{\frac{C}{1-\alpha} + \frac{C}{\beta-1} \pm o(C)} \\ &= \frac{\frac{1}{1-\alpha} \pm o(1)}{\frac{1}{1-\alpha} + \frac{1}{\beta-1} \pm o(1)} \\ &= \frac{1 \pm o(1)}{1 - \alpha} \cdot \frac{(1 - \alpha)(\beta - 1)}{(\beta - 1) + (1 - \alpha) \pm o(1)} \\ &= \frac{\beta - 1 \pm o(1)}{\beta - \alpha \pm o(1)} \\ &= \frac{\beta - 1}{\beta - \alpha} \pm o(1). \end{aligned}$$

The claim follows. □