Modeling skewed distributions using multifractals and the '80-20 law'

Christos Faloutsos[*]
Dept. of Computer Science and
Institute of Systems Research
Univ. of Maryland
College Park, MD 20742
christos@cs.umd.edu

Yossi Matias *Avi Silberschatz*
Bell Laboratories
Murray Hill, NJ 07974
{matias,avi}@research.att.com

Abstract

The focus of this paper is on the characterization of the skewness of an attribute-value distribution and on the extrapolations for interesting parameters. More specifically, given a vector with the highest h multiplicities $\vec{m} = (m_1, m_2, ..., m_h)$, and some frequency moments $F_q = \sum m_i^q$, (e.g., $q = 0, 2$), we provide effective schemes for obtaining estimates about either its statistics or subsets/supersets of the relation.

We assume an 80/20 law, and specifically, a $p/(1-p)$ law. This law gives a distribution which is commonly known in the fractals literature as '*multifractal*'. We show how to estimate p from the given information (first few multiplicities, and a few moments), and present the results of our experimentations on real data. Our results demonstrate that schemes based on our multifractal assumption consistently outperform those schemes based on the uniformity assumption, which are commonly used in current DBMSs. Moreover, our schemes can be used to provide estimates for supersets of a relation, which the uniformity assumption based schemes can not not provide at all.

[*]This work was partially supported by the National Science Foundation under Grants No. CDR-8803012, EEC-94-02384, IRI-8958546 and IRI-9205273), with matching funds from Empress Software Inc. and Thinking Machines Inc. Part of the work performed while visiting AT&T Bell Laboratories.

Proceedings of the 22nd VLDB Conference
Mumbai(Bombay), India, 1996

1 Introduction

The goal of this paper is to estimate several measures for a distribution of attribute values, given the 'standard' information that commercial RDBMSs keep about the distributions. Typically [16] the RDBMSs maintain several statistics regarding the attribute values. These include the total number of records N for a relation and the total number of distinct values F_0 for a given attribute. Other statistics considered recently [10] are the high-biased histogram (that is, the first few most common values, along with their multiplicity = occurrence frequency), and the size of the self-join, also denoted as the *second frequency moment* F_2. Very recent works [6, 2, 1] have suggested efficient, on-line probabilistic methods to keep track of the high-end histograms, as well as the self-join size and other frequency moments F_q of the distribution.

This is typically the information that we keep track of, in order to estimate selectivities for query optimization. For the attribute values that we have no information about, the typical assumption is the uniformity assumption [9]. In this work, we propose an alternative, more realistic assumption, and we show that it can help us model multiplicity distributions in a more accurate way, and therefore to provide better estimates, as well as to allow extrapolations for subsets or supersets of the relation.

Sample scenarios and applications are listed next. For concreteness, consider a relation of *sales(product-*

name, customer-id, amount-spent). Also assume that we keep the high-end histograms for *product-name*, and, of course the total number of distinct products F_0 and the total number of sales records N. Then, we have the following classes of queries of interest:

- **Estimates for subsets:** Given the above information, focus on sales of $100 and above, and estimate the number of distinct products involved in such sales

- **Median and percentiles:** How many (distinct) products account for 50% of the sales? or 90% of the sales?

- **Extrapolations for supersets:** Suppose that the above relation concerns the domestic sales only; what is our best estimate for the number of distinct products for the international sales, when we only know the total number of sales $N_{international}$? What is our best estimate for the total amount of the international sales?

- **Self-joins selectivity estimation:** What is our best estimate for the moments F_q of the distribution? Recall that the q-th moment corresponds to the cardinality of q successive joins of the relation with itself.

- **Spatial databases:** Consider a geographic database, with the schema: *cities(lattitude, longtitude, name)*; consider a multi-dimensional histogram, which stores the count of cities in each grid-cell; the goal is to estimate the selectivity of spatial queries, given the above histogram. For example, a spatial-join query would be '*estimate the number of pairs of cities that are closer than 10 miles to each other*' [3].

For all the above scenarios, we propose to assume that the unknown multiplicities were derived from a multifractal distribution, which is a more general case than the familiar '80-20 law'. Based on this assumption, we can estimate the parameters of the multifractal distribution, and subsequently extrapolate, to try to answer the above classes of questions.

We illustrate the reasons why a multifractal distribution should appear often in real datasets, how it includes the uniform distribution as a special case, and how its predictions compare with the predictions of the uniformity assumption.

Section 2 gives the survey and background information. Section 3 defines the problem and the proposed solution. Section 4 shows experimental results on real data. Section 5 lists the conclusions and future research directions.

2 Survey - Background

Here we present the state of the art in histogram methods, a discussion on previous models for skewed distributions ('Zipf' and 'generalized Zipf' [17] etc.) and some related methods for estimation using sampling; we also give an introduction to multifractals.

2.1 Histograms

DeWitt and Muralikrishna [12] studied multi-dimensional histograms. Ioannidis and Poosala [10] suggest keeping the frequencies of a few frequent attributes, and making the uniformity assumption for the rest. These are called 'high-biased' histograms, and seem to be the state of the art in current commercial systems. Ioannidis and Christodoulakis [9] showed that they have the smallest error among several classes of histograms for self-joins.

Recent works [6, 2, 1] have proposed efficient on-line algorithms to maintain probabilistically the first few largest multiplicities, as well as a few frequency moments $F_q = \sum m_i^q$, where the summation is over all the attribute values i, and m_i is the multiplicity of i. These algorithms make no assumptions about the distribution of the data.

There are two main ideas that distinguish the present work from the current state-of-the-art: The first is the proposal to use the *multifractal* assumption, as opposed to the uniformity assumption. The second idea is to also use information about the frequency moments, to help us better estimate the parameters of the multifractal distribution.

To make the discussion more concrete, we need the following definitions:

Definition 2.1 The q-th frequency moment F_q of a frequency distribution $\vec{m}$ is defined as

$$F_q \equiv \sum_{i=1} m_i^q \tag{1}$$

Example 2.1 For the frequency ($\equiv$ multiplicity) vector

$$\vec{m} = (5, 3, 2, 2, 1, 1, 1, 1) \tag{2}$$

we have

$$\begin{aligned} F_0 &= 5^0 + 3^0 + 2^0 + 2^0 + 1^0 + 1^0 + 1^0 + 1^0 = 8 \\ F_1 &= 5^1 + 3^1 + 2^1 + 2^1 + 1^1 + 1^1 + 1^1 + 1^1 = 16 \\ F_2 &= 5^2 + 3^2 + 2^2 + 2^2 + 1^2 + 1^2 + 1^2 + 1^2 = 46 \end{aligned}$$

□

Obviously, F_0 gives the number of distinct values (or 'vocabulary', borrowing terminology from text databases), $F_1 \equiv N$ (the total number of records), and F_2 is the size of the self-join of the relation on this attribute. It is computationally more efficient to group identical multiplicities together:

Definition 2.2 Let c_m denote the count of distinct attribute values that have multiplicity m.

Then, the frequency moments can also be computed as follows:

$$F_q = \sum_{m=1} c_m m^q \tag{3}$$

Example 2.2 For the multiplicity vector of Example 2.1, we have $c_5 = 1$, $c_3 = 1$, $c_2 = 2$, $c_1 = 4$ and we can compute the moments as follows, using Eq. 3:

$$\begin{aligned} F_0 &= 1 \cdot 5^0 + 1 \cdot 3^0 + 2 \cdot 2^0 + 4 \cdot 1^0 = 8 \\ F_1 &= 1 \cdot 5^1 + 1 \cdot 3^1 + 2 \cdot 2^1 + 4 \cdot 1^1 = 16 \\ F_2 &= 1 \cdot 5^2 + 1 \cdot 3^2 + 2 \cdot 2^2 + 4 \cdot 1^2 = 46 \end{aligned}$$

□

The above definitions of the frequency moments can be extended for non-integer values of q, and keeping track of such frequency moments can also be handled by the probabilistic algorithms of [2, 1]. The frequency moments are useful to characterize the skewness of the distribution. Note that the q-th frequency moment gives the size of joining the table q times with itself on the attribute under discussion.

A typical tool for the study of skewed distributions is the so-called rank-frequency plot:

Definition 2.3 The rank-frequency plot of a set of multiplicities sorted in descending order is the plot of m_r versus the rank r, with both axes *logarithmic*.

As an example, Figure 1 shows the rank-frequency plot for the first names from a telephone book ('VFN' dataset, as described in section 4). It is interesting to report some specific numbers, to highlight the skewness of this distribution: there are 11,657 records in total, while the number of distinct first names is surprisingly small: F_0=3,269. The most common name appears m_1=288 times, while the vast majority of names (2,345 out of the 3,629 distinct ones) appear only once! As we show in the experiments section, such skewed distributions are the rule, as opposed to the exception!

2.2 Models for non-uniformity

Probably the earliest model for non-uniform distributions is the Zipf distribution [17]. According to this model, the r-th highest multiplicity m_r is given by the formula:

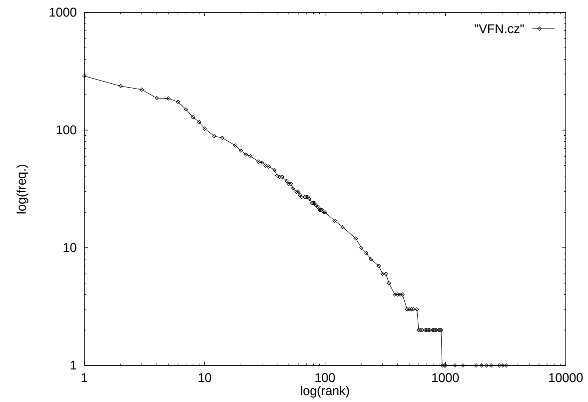


Figure 1: *rank-frequency plot of first names from a telephone directory*

$$m_r \approx C / r^{\theta} \tag{4}$$

where r stands for the rank.

For $\theta = 1$ we have the Zipf distribution; for $\theta \neq 1$ we have a 'generalized Zipf' distribution with parameter θ. Clearly, the rank-frequency plot of a generalized Zipf distribution is a straight line with slope equal to $-\theta$.

As Zipf showed experimentally [17]), the above distribution gives a good approximation for the occurrence frequencies of words in natural text, including English as well as several other languages. More specifically, for text, Schroeder [15] gives the following formula (adapted to our notation):

$$m_r \approx \frac{N}{r \ln(1.78 F_0)} \tag{5}$$

However, there are two weaknesses of the Zipf (and generalized Zipf) distributions:

- As even Zipf himself noted, real datasets typically show the 'top-concavity', that neither the Zipf distribution nor any generalized Zipf distribution can match. Figures 4-6 show several rank-frequency plots of real distributions; notice that the top part of the curve typically tilts horizontally, giving a concave shape to the whole distribution.

- There is no explanation for the Zipf distributions: there is no physical process that would generate a (plain or generalized) Zipf distribution. Moreover, these distributions can not help us predict the chances that a new record will introduce a brand-new attribute value (as opposed to match one of the already existing attribute values). Thus, the

Zipf distributions can not do extrapolations for supersets, when given a sample of a relation.

For these reasons, we do not examine the Zipf distribution any further.

2.3 Sampling

One of the uses of a good model for a skewed distribution is the ability to do extrapolations from a subset. As we show later, we can estimate the number of distinct values F_0 for a subset or a superset of a given relation. The state of the art in this area seems to be the work of Haas *et al.* [7] which uses two different estimators, and, depending on the perceived skewness, it chooses the appropriate one each time. Previous work includes [8], whose estimators are superseded by [7].

As we show later, our proposed multifractal assumption leads to very good estimates, with estimation error about the same as the best available estimator.

2.4 Introduction to Multi-fractals

An excellent introduction to multifractals is in [14]. Their relationship with the '80-20 law' is very close, and seem to appear often: Schroeder [15] claim that several real distributions follow a rule reminiscent of the 80-20 rule in databases. For example photon distributions in physics, or commodities (water, gold, etc) distributions on earth etc., follow a rule like '*the first half of the region contains a fraction p of the gold, and so on, recursively, for each sub-region.*' Similarly, financial data and salary distributions follow similar patterns (Pareto's law of income distribution [11]).

With the above rule, we assume that the address space (*e.g.*, the unit interval) is recursively decomposed at k levels; each decomposition halves the input interval in two. Thus, eventually we have 2^k subintervals (also called buckets, or slots) of length 2^{-k}.

We consider the following distribution of probabilities, as illustrated in Figure 2: At the first level, the left half is chosen with probability $(1-p)$, while the right one with p; the process continues recursively, for k levels. Thus, the left half of buckets will host $1-p$ of the probability mass, the left-most quarter will host $(1-p)^2$ etc. We shall refer to the p and k parameters as the <u>bias</u> and the <u>order</u> of the multifractal distribution, respectively.

Definition 2.4 A distribution of N records is defined as a *binomial multifractal* distribution (or simply *multifractal* distribution) with parameters (N, p, k), if it has 2^k possible attribute values (buckets), each attracting records with the bias parameter p, as described above. In particular, the assignment of a record to a bucket can be viewed as a probabilistic (binary) decision tree of depth k; starting at the root, we choose the right sub-tree with probability p and (of course) the left sub-tree with probability $(1-p)$, until we reach a leaf (= bucket = an attribute value).

Notice that the uniform distribution is a *special case*, by setting $p = 0.5$.

Next we derive some formulas which are useful for the up-coming estimations. Let C_a^k denote the k-choose-a combinations. For a binomial multifractal distribution (N, p, k), there are C_a^k attribute values for which the expected relative frequency is $p^{(k-a)}(1-p)^a$. This is easy to observe by considering the probabilistic decision tree. In our previous terminology (Def. 2.2), we expect to have

$$c_m \approx C_a^k \quad (6)$$

distinct attribute values, each of which occurring

$$m = N \cdot p^{(k-a)}(1-p)^a \quad (7)$$

times.

3 Problem Definition and Proposed Solution

The general problem is as follows: Given some partial information about the distribution (*e.g.*, first few multiplicities, a few frequency moments, a small sample, etc.), find a way to characterize its skewness and to enable predictions about measures of interest (*e.g.*, median value number of distinct values in a superset or subset etc). We propose to use multifractals, or equivalently, a generalization of the 80-20 law.

Given a data set with unknown distribution of attribute values, we maintain the hypothesis that the distribution can be well approximated by some multifractal distribution, the parameters of which are initially unknown. The problem is to identify the 'bias' p and the order k, that will lead to a good match of the given set of multiplicities and other available information about the distribution.

As we mentioned, this problem is very realistic: many commercial systems keep some 'high-end biased' histograms [10] for query optimization; probabilistic on-line algorithms for maintaining such histograms efficiently have just recently been proposed [6].

There are two sets of results: The first set tries to express the p and k parameters as functions of the given data. More concretely, we have the following goal given the hypothesis:

- **Given**
  - the first few of the multiplicities m_i, $i = 1, 2, \ldots, h$ and
  - the number of distinct attribute values F_0,

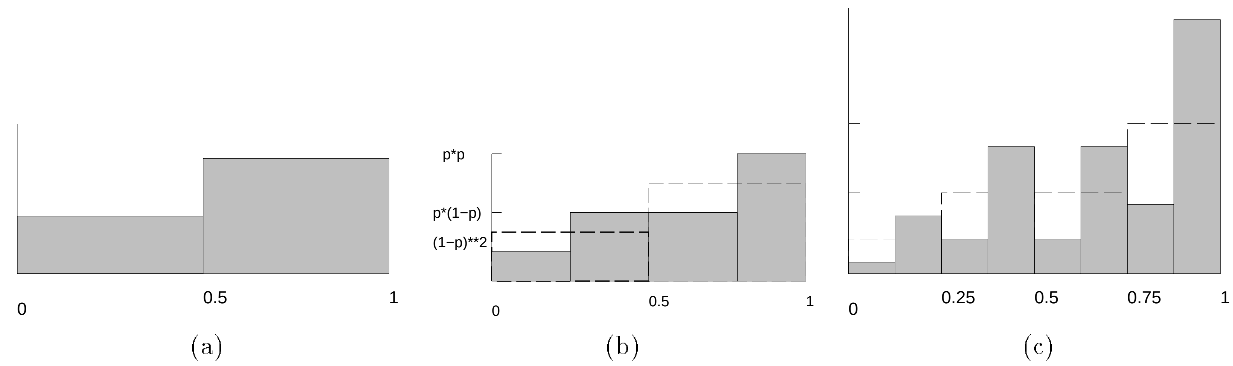


Figure 2: *Generation of a 'multifractal' - first three steps*

- **Estimate** the p and k parameters to yield a multifractal distribution that will match the given data.

The second set of results tries to estimate other quantities of interest (*e.g.*, median value etc), for a given multifractal distribution with parameters p and k. Table 1 contains the symbols and their definitions.

3.1 Estimating the p and k parameters

We use the following observations:

Observation 3.1 The bias parameter p can be estimated as

$$p = (m_{max}/N)^{1/k} \tag{8}$$

Indeed, the highest multiplicity $m_{max} = m_1$ will be on the average $N \times p^k$.

Theorem 3.1 For a binomial multifractal distribution with N records, bias p and order k, the expected number of distinct values $\hat{F}_0$ is given by the following equation

$$\hat{F}_0 \;=\; \mathcal{F}_0(N, p, k) \;=\; \sum_{a=0}^{k} C_a^k (1 - (1 - p_a)^N) \tag{9}$$

where

$$p_a = p^{k-a}(1-p)^a \tag{10}$$

Proof: The idea is to focus on one of the 2^k buckets. We can estimate the probability that this specific bucket will be hit at least once by one of the N records, and then, average over all these buckets. **QED**

Thus our estimation algorithm needs only m_{max}, F_0 and N. See Figure 3 for the pseudo-code. The Appendix A gives the code for Step 3 of the algorithm.

3.2 Extrapolations

If our distribution follows a multifractal distribution with (known) parameters p and k, we can use this fact to estimate several useful measures.

Estimation of number of distinct values for subsets/supersets:

We can use our 'multifractal assumption' to do extrapolation from a sample of N' ($< N$) records, out of the total N records. Given the sample, we compute its p and k parameters; if the full collection comes from a multifractal distribution, it will have the same parameters p and k. Thus, we just substitute the values N, p and k in the formula for $\hat{F}_0$ (Eq. 9), to obtain an estimate for the number of distinct values of the collection.

Thus, if the original distribution is approximated by a multifractal distribution with N records, bias p and order k, for a subset of N' records we estimate its 'vocabulary' $\hat{F}'_0$ as follows:

$$\hat{F}'_0 \;=\; \mathcal{F}_0(N', p, k) \;=\; \sum_{a=0}^{k} C_a^k C_m^{N'} p_a^m (1 - p_a)^{(N'-m)} \tag{11}$$

Median and percentiles:

Salaries and incomes follow very skewed distributions [15, p. 35] [13], [11]. Our upcoming experiments (see section 4) show that sales patterns seem to do the same. Thus, given a relation with salaries, the question is to find the median salary, given little information (*e.g.*, the first few top salaries). Assuming a multifractal distribution, we can compute p and k, and estimate several statistics (median, percentiles etc). For concreteness, we repeat the standard definitions of median and percentiles:

Definition 3.1 The median rank $r_{50\%}$ of a multiplicity vector $\vec{m}$ (sorted in descending order) is the smallest rank, so that the elements up to and including that rank $r_{50\%}$ account for at least 50% of the occurrences:

$$\sum_{r=1}^{r_{50\%}-1} m_r \;<\; 0.5\, N \;\leq \sum_{r=1}^{r_{50\%}} m_r \tag{12}$$

Symbol	Definition
N	total number of records
p	'bias': fraction of 'mass' that goes to the right half, in each subdivision of the multifractal
k	order of multifractal distr. (number of subdivisions)
m_{max}	$=m_1$: the highest multiplicity
c_m	count for multiplicity m (number of distinct attr. values with multiplicity m)
F_q	frequency moment of order q
$F_0 = V$	number of distinct values = vocabulary
h	number of values kept in a high-biased histogram
C_n^m	combinations m-choose-n

Table 1: *Symbols and definitions*

Input: N, m_{max} *and* F_0
Output: the p *and* k *parameters*

1 let $k = \lceil \log F_0 \rceil$ as a first estimate

2 estimate p using Eq. 8.

3 estimate $\hat{F}_0$ using Eq. 9. *It will be an under-estimate of the real* F_0.

4 $k++$, and repeat the steps 2-4, until $\hat{F}_0$ matches F_0 within a desired tolerance ϵ.

Figure 3: *Algorithm to estimate the bias* p *and order* k *of a multifractal distribution*

Definition 3.2 Median frequency $m_{r_{50\%}}$ is the frequency of the element with the median rank.

Example 3.1 For the multiplicity vector of Example 2.1, the median rank $r_{50\%}$=2 and the median frequency $m_{r_{50\%}}$=3. □

In a real setting, where we are given a high-end histogram with the highest h multiplicities $m_1, \ldots, m_h$, we estimate the median rank $r_{50\%}$ as follows: we use the given first h multiplicities as well as the estimates for the rest of the multiplicities from Eqs. (6-7); we keep including more elements, until we reach or exceed 50% of the number of records N.

Estimating the frequency moments:

If the given multiplicity vector was the result of a binomial multifractal process, with a parameter p and k, then we would have

$$\begin{aligned} F_q &= \sum_m (c_m m^q) \\ &= \sum_m \left(C_a^k \left(N p^{k-a} (1-p)^a\right)^q \right) \\ F_q &= N^q \left(p^q + (1-p)^q\right)^k \qquad (13) \end{aligned}$$

which allows a fast estimate of the moments, given the parameters N, p and k of the multifractal distribution.

Recall that k is the order of the multifractal distribution, that is, the number of recursive subdivisions of the address space, resulting in 2^k possible distinct values.

This concludes the mathematical derivations that pertain to a multifractal distribution. The question now is to find out how accurate our predictions are, when we try to approximate a real distribution of frequencies with a multifractal distribution. This is exactly the topic of the next section.

4 Experiments

In this section we use real datasets, and we test the accuracy of the predictions using the multifractal assumption. We used several real datasets. Table 2 shows the characteristics of each dataset, that is, the total number of records N, the highest multiplicity $m_1 \equiv m_{max}$, and the total number of distinct attribute values ('vocabulary') $F_0 \equiv V$. The description of each dataset follows:

- 'VFN' consists of the first names from an on-line telephone catalog [5]. Actually, we used the 'very first names', keeping only the first one in the case of multiple first names: For example 'Maria Teresa' would be registered as 'Maria'.

- 'SALES', which contains the dollar amounts of sales for customers, rounded to the nearest 1-, 10- and 100-dollar amount, for 'SALES1', 'SALES10' and 'SALES100' respectively.

- 'BIBLE': the words in the Bible (Old and New Testament), along with their occurrence frequency. We also used sub-sets of the BIBLE, namely 'GENESIS' (the book of Genesis), 'ROMANS' (the letter to the Romans), 'PSALMS' (the Psalms), 'JEREMIAH' (the prophesies of Jeremiah), 'PJ' (the PSALMS and JEREMIAH datasets combined, to provide a ≈10% sample of the BIBLE).

- 'WUTHERING': the book 'Wuthering Heights'.

Dataset	N	F_0	m_{max}
VFN	11657	3269	288
SALES1	213603	246	71565
SALES10	21507	246	7157
SALES100	2309	246	716
BIBLE	791448	12561	63924
PSALMS	42732	2884	2884
JEREMIAH	42729	2592	3838
PJ	85461	3944	6722
GENESIS	38520	2448	3678
ROMANS	9439	1317	597
WUTHERING	120951	10042	4747

Table 2: *Datasets and their characteristics*

Figures 4-6 show the rank-frequency plots for our datasets: 'diamonds' with solid lines correspond to the actual values,, 'crosses' with dashed lines correspond to our predictions using multifractals (Eq. 6-7). In some of the plots we show some straight dotted lines, which correspond to Zipf and generalized Zipf distributions. Notice that the actual curves can not be approximated with a straight line of *any* slope, while the curves suggested by the multifractal distribution are closer to the real curves, exhibiting the 'top-concavity' that we mentioned earlier.

This concludes the first set of experiments, where we visually illustrate that several real distributions are matched well by a carefully selected multifractal distribution. In the next two subsections we study the accuracy of the predictions of a multifractal distribution (a) for the number of distinct values in a subset or superset of a relation and (b) for the median rank and percentiles.

4.1 Vocabulary estimation of a sample

The problem is: given a high-biased histogram $m_i, i = 1, ..., h$ of length h, the number records N and the number of distinct values F_0, estimate the number of distinct values for a subset of N' records.

As mentioned before, assuming a multifractal distribution, we compute the N, p, k parameters, and then use Eq. 9 to estimate the vocabulary of the subset/super-set.

Under the uniformity assumption, the best we can do is to consider a generalization of Cardenas' formula [4]: we know that we have F_0 buckets and N' records; we also know the frequency that the first h buckets are chosen; thus each bucket is chosen with probability p_i, which is computed as follows:

$$\begin{aligned} p_i &= m_i/N \quad i \leq h && (14) \\ p_i &= p_u = \\ & (N - N_h)/N/(F_0 - h) \quad h < i \leq F_0 && (15) \end{aligned}$$

where N_h is the sum of the frequencies of the histogram.

Then, the expected number $\hat{F}'_{unif}$ of non-empty buckets (after N' choices) is estimated by

$$\hat{F}'_{unif} = \sum_{i=1}^{F_0} (1 - (1 - p_i)^{N'})$$

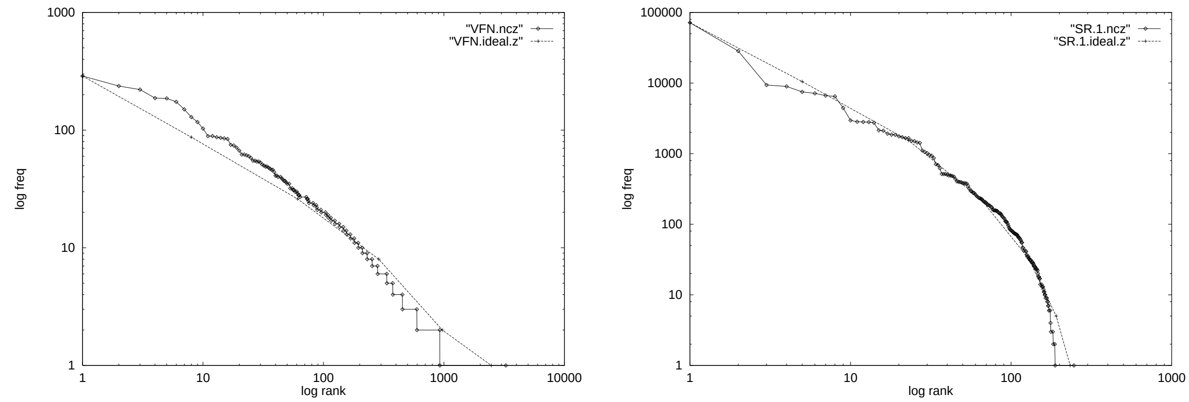
or

$$\hat{F}'_{unif} = \sum_{i=1}^{h} (1 - (1 - p_i)^{N'}) + (F_0 - h)(1 - (1 - p_u)^{N'})$$

where the probabilities p_i are given by Eq. 14-15.

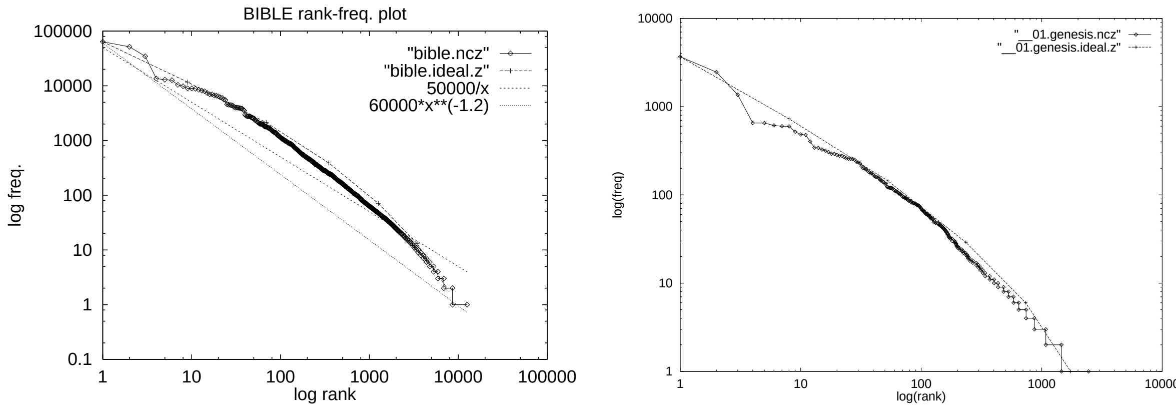
Table 3 gives the results of these estimators on the real datasets. Based on the BIBLE dataset, we estimated the samples of it (ROMANS, PSALMS and JEREMIAH). Notice that the work of Haas et al. [7] is not directly applicable, because it assumes that we know *all* the multiplicities of the given dataset, as opposed to only the h highest, that is our setting. Notice that our estimates give low errors (45-62%), which are comparable to the errors of much more sophisticated estimation algorithms: Haas et al [7], using all the statistics about the dataset, report that, for a 10% sample of 'highly skewed' distributions, the relative error ($\equiv |\hat{F}_0 - F_0|/F_0$) was on the average 23% (maximum: 95%) for the so-called *Shlosser* estimator, which was the best performer for 'high-skew distributions'. Interestingly, among the methods they tried, the worst competitor had 158% average and 1235% maximum relative error.

Table 4 shows the reverse: given a sub-set (*e.g.*, the PJ set), we can estimate the vocabulary of the superset (BIBLE). In this case, the uniformity assumption gives poor results, exactly because it does not have the ability to predict the appearance of new words in the larger set. Again, the 54% relative error compares well



(a) VFN (b) SALES1

Figure 4: *The rank-frequency plots of the 'VFN' and 'SALES1' datasets. Real ('diamonds') and estimated ('crosses') values.*



(a) BIBLE (b) GENESIS

Figure 5: *The rank-frequency plots for the 'BIBLE' and 'GENESIS' datasets; real ('diamonds') and estimated ('crosses') values. Dotted lines indicate Zipf and generalized Zipf distributions.*

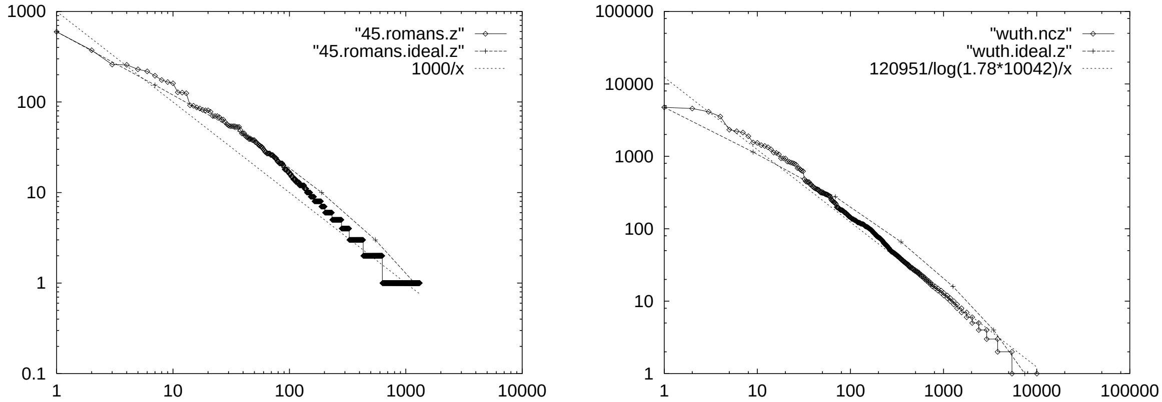
with the errors of the more sophisticated algorithms by Haas et al (23% average, 95% maximum, as mentioned before).

4.2 Estimation of median & percentiles

Table 5 shows the estimates for the median rank for several datasets, given a high-biased histogram with h entries. We used the multifractal and the uniformity assumption; in either method, we exploited the fact that the first h multiplicities are known, and we estimated the unknown multiplicities $m_{h+1}, \ldots$, and summed them, until we reached 50% of the count. Notice that the estimates of the uniformity assumption are often 1 or 2 orders of magnitude away.

5 Conclusions

We have shown that the multifractal theory formalizes and generalizes the 80-20 'law'; that it includes the uniform case as a special case (p=0.5) and that it matches reality better than the Zipf distribution. Using the multifractal assumption, we provided a simple, but accurate way to estimate the multiplicity vector, given only easy-to-maintain values: the highest multiplicity m_{max}, the number of records N and the number of distinct values V. A good estimate of the multiplicity vector helps in doing extrapolations for several useful statistical quantities, both of the original relation, as well as of super-sets and sub-sets of it. For example, it can help compute percentiles and median ranks ('*how many of our customers account for 90% of our sales*', or '*how many distinct products would the female portion of our customer base be interested in?*'). Such



(a) ROMANS (b) WUTHERING

Figure 6: *The rank-frequency plots for the 'ROMANS' and 'WUTHERING' datasets; real ('diamonds') and estimated ('crosses') values. Dotted lines indicate Zipf distributions.*

Dataset	Size N (in words)	Vocabulary size			
		uniformity	multifractal		actual
			estimate	rel. error	F_0
ROMANS	9,439	4686	1963	49%	1,317
PSALMS	42,732	11036	4,208	45%	2,884
JEREMIAH	42,729	11035	4,208	62%	2,592

Table 3: *Estimates for the vocabulary of a sample from the BIBLE (N=791,448 p=0.84557 k=15). For the 'uniform', the h=20 highest multiplicities are kept.*

estimates are useful in numerous applications, such as (a) traditional query optimization, supplementing the high-biased histogram methods that are currently the state of the art [10], (b) decision support systems, where extrapolations for subsets and supersets are important.

Experiments on several real datasets showed that the multifractal assumption gives significantly better estimates than the 'uniformity' assumption, for several useful statistical quantities.

Future work could examine the application of multifractals to several other settings, such as join size estimation and spatial-join selectivity estimation in geographic information systems.

References

[1] N. Alon, P. Gibbons, Y. Matias, and M. Szegedy. Dynamic probabilistic maintenance of self-join sizes in limited storage. Manuscript, Bell Labs, February 1996.

[2] N. Alon, Y. Matias, and M. Szegedy. The space complexity of approximating the frequency moments. *Proc. 28th ACM Symp. on Theory of Computing*, pages 20–29, May 1996.

[3] Alberto Belussi and Christos Faloutsos. Estimating the selectivity of spatial queries using the 'correlation' fractal dimension. *Proc. of VLDB*, pages 299–310, September 1995.

[4] A.F. Cardenas. Analysis and performance of inverted data base structures. *CACM*, 18(5):253–263, May 1975.

[5] Christos Faloutsos and H.V. Jagadish. On b-tree indices for skewed distributions. In *18th VLDB Conference*, pages 363–374, Vancouver, British Columbia, August 1992.

[6] P. Gibbons, Yossi Matias, and A. Witkowski. Practical maintenance algorithms for high-biased histograms using probabilistic filtering. Technical report, ATT Bell Laboratories, December 1995.

[7] Peter J. Haas, Jeffrey F. Naughton, S. Seshadri, and Lynne Stokes. Sampling-based estimation of the number of distinct values of an attribute. *Proc. of VLDB*, pages 311–322, September 1995.

[8] Wen-Chi Hou and Gultekin Ozsoyoglu. Statistical estimators for aggregate relational algebra

Dataset	Size N (in words)	Vocabulary size			
		uniformity	multifractal		actual
			estimate	rel. error	
BIBLE	791,448	3944	5749	**54.5%**	12561

Table 4: *Estimates for the vocabulary of the BIBLE from a sample (PJ set: N=85461, p=0.822349 k= 13)*

Dataset	uniformity value	multifractal value	actual median	actual F_0
VFN (h=20)	1178	227	130	3269
SALES1 (h=0)	123	5	3	246
SALES10 (h=0)	123	5	3	246
SALES100 (h=0)	123	8	4	246
SALES100 (h=2)	31	5	4	246
BIBLE (h=0)	6281	90	43	12561
BIBLE (h=20)	2419	64	43	12561
ROMANS (h=20)	267	48	39	1317
PSALMS (h=20)	547	52	35	2884
GENESIS (h=20)	460	50	39	2448
JEREMIAH (h=20)	437	46	37	2592
WUTHERING (h=20)	2539	94	68	10042

Table 5: *Estimates for the median rank of datasets, given a high-biased histogram of h entries.*

queries. *ACM TODS*, 16(4):600–654, December 1991.

[9] Yannis E. Ioannidis and Stavros Christodoulakis. Optimal histograms for limiting worst-case error propagation in the size of join results. *ACM TODS*, 18(4):709–748, December 1993.

[10] Yannis E. Ioannidis and Viswanath Poosala. Balancing histogram optimality and practicality for query result size estimation. *ACM SIGMOD*, pages 233–244, June 1995.

[11] B.B. Mandelbrot. The stable paretian income distribution when the apparent exponent is near zero. *Int. Econ. Rev.*, 4:111–115, 1963.

[12] M. Muralikrishna and David J. DeWitt. Equi-depth histograms for estimating selectivity factors for multi-dimensional queries. *Proc. ACM SIGMOD*, pages 28–36, June 1988.

[13] V. Pareto. *Oeuvres Completes*. Droz, Geneva, 1896.

[14] Heinz-Otto Peitgen, Hartmut Juergens, and Dietmar Saupe. *Chaos and Fractals: New Frontiers of Science*. Springer-Verlag New York Inc., 1992.

[15] Manfred Schroeder. *Fractals, Chaos, Power Laws: Minutes From an Infinite Paradise*. W.H. Freeman and Company, New York, 1991.

[16] P.G. Selinger, D.D. Astrahan, R.A. Chamberlain, R.A. Lorie, and T.G. Price. Access path selection in a relational database management system. *Proc. ACM-SIGMOD*, pages 23–34, 1979.

[17] G.K. Zipf. *Human Behavior and Principle of Least Effort: an Introduction to Human Ecology*. Addison Wesley, Cambridge, Massachusetts, 1949.

A Appendix: AWK code for the estimation of F_0

Here we give the code to estimate the number of distinct values F_0, for a multifractal distribution with N samples, bias p and order k. The file is ready to execute under UNIX(TM).

```
#!/bin/sh -f
# echo "$0 working on $1" >&

echo $1 $2 $3 | nawk '
# reads N, p, k of a binomial multifractal
# and estimates the number
# of distinct values F0
#
    function power( x, y ) {
        res = exp( y * log(x) );
```

```
        return( res );
    } # end function power
    function comb( NN, MM){
        cres = 1;
        for( ii=1; ii<=MM; ii++ ){
            cres = cres * (NN - ii + 1) / ii;
        }
        return ( cres );
    } # end function comb

    # estimates F0, the expected number
    # of distinct values
    function estF0( NN, pp, kk){
        rres = 0;
        for(aa=0; aa<=kk; aa++){
            pa = power(pp, kk-aa) * power( 1-pp, aa)
            if( pa*NN > 50 ) { tmp = 0.0 }
                # guard against underflow of power()
            else { tmp = power( 1-pa, NN); }
            rres = rres +  comb(kk,aa) * ( 1 - tmp );
        }
        return (rres)
    } # end function estF0
{
    N = $1        # number of records
    p = $2        # bias factor
                  # (= split probability)
    k = $3        # number of divisions
}
END{
  print "number of records N=", N
  print "bias p=", p
  print "number of splits k=", k
  F0hat = estF0(N,p,k)
  print "est. number of distinct values F[0]=", F0hat
}
'
```