APPROXIMATE COUNTING: A DETAILED ANALYSIS

PHILIPPE FLAJOLET

INRIA, Domaine de Voluceau-Rocquencourt, B.P. 105, 78153 Le Chesnay CEDEX, France

Abstract.

Approximate counting is an algorithm proposed by R. Morris which makes it possible to keep approximate counts of large numbers in small counters. The algorithm is useful for gathering statistics of a large number of events as well as for applications related to data compression (Todd et al.). We provide here a complete analysis of approximate counting which establishes good convergence properties of the algorithm and allows to quantify precisely complexity-accuracy tradeoffs.

Introduction.

As shown by an easy information-theoretic argument, maintaining a counter whose values may range in the interval 1 to M essentially necessitates $\log_2 M$ bits. This lower bound is of course achieved by a standard binary counter. R. Morris [8] has proposed a *probabilistic algorithm* that maintains an *approximate count* using only about $\log_2 \log_2 M$ bits. This paper is devoted to a detailed analysis of characteristic parameters of that algorithm. We provide precise estimates on the probabilities of errors, from which the soundness of the method can be assessed and complexity-accuracy trade-offs can be quantified.

The algorithm itself is useful for gathering statistics on a large number of events in a storage efficient way [8]. It was proposed for applications to data compression [9] when building an adaptive encoding scheme to represent "non-random" data (see e.g. [4] for adaptive Huffman codes and [7] for arithmetic coding); there, typically a large number of counters need to be maintained to gather statistics on the data to be compressed, but high accuracy of each counter is not a critical factor in the design of almost-optimal codes. Actually Todd et al. report the overall performance of a system using approximate counting which is only a few percent off a reference system using exact counts.

There are other cases like data base systems where probabilistic counting methods prove useful. We mention a related algorithm, called "Probabilistic Counting" that has been proposed in [3]. This algorithm makes it possible to determine the approximate value of the number of *distinct* elements in a file in a single pass using a few operations per element and only $O(1)$ additional storage.

Received October 1982. Revised August 1984.

The plan of the paper is as follows. We start with a simple version of the algorithm: approximate counting with base 2, which is very easy to implement on a binary computer. It appears (Theorems 1, 2) that this algorithm can maintain an approximate count up to M using about $\log_2 \log_2 M$ bits, with an error that is typically of one binary order of magnitude.

The analytic techniques that we use in Sections 2, 3, 4, involve manipulation of generating functions related to a discrete time birth-process to which the algorithm is equivalent, certain properties of the Mellin integral transform, and finally some simple identities that properly belong to the theory of integer partitions. In Sections 5, 6, we discuss the more general version of the algorithm with an arbitrary base. The analysis shows that, using suitable corrections, one can count up to M keeping only $\log_2 \log_2 M + \delta$ bits with an accuracy of order $O(2^{-\delta/2})$.

A preliminary report on this work has been presented at the International Seminar on Modelling and Performance Evaluation Methodology ("On Approximate Counting": Proceedings, Volume 2, pp. 205–236, Paris, January 1983).

1. Approximate counting with a binary base.

If the requirement of accuracy is dropped, a counter of value n can be replaced by another counter C containing $\lfloor \log_2 n \rfloor$ which only requires storing about $\log_2 \log_2 n$ bits. However since the fractional part of $\log_2 n$ is no longer preserved, there now arises the problem of deciding when to update the logarithmic counter in the course of successive incrementations. The idea of [8], [9] is to base this decision on probabilistic choices.

Approximate counting starts with counter C initialized to 1. After n increments, we expect C to contain a good approximation to $\lfloor \log_2 n \rfloor$; we should thus increase C by 1 after another n increments approximately. Since the exact value of n has not been kept and only C is known, the algorithm has to base its decision on the content of C alone. Approximate counting then treats the incrementation by the following procedure.

```
procedure increment ($C$: integer);

    Let DELTA ($C$) be a random variable which takes value 1 with probability
    $2^{-C}$ and value 0 with probability $1-2^{-C}$;
    $C := C + DELTA(C)$
```

The interesting fact about this procedure is the following: if C_n is the random variable representing the content of counter C after n applications of the increment procedure, then the expectation of 2^{C_n} bears a simple relation to n (as we shall prove at the end of Section 2).

PROPOSITION 0 [8]: *The expectation and variance of 2^{C_n} satisfy*

$$E(2^{C_n}) = n+2; \quad \sigma^2(2^{C_n}) = n(n+1)/2.$$

Thus 2^C-2 represents an unbiased estimator of n. In the sequel, we give a detailed analysis of the probability distribution of C_n and characterize its mean and variance.

THEOREM 1. *After n successive increments, the counter of approximate counting has average value*

$$\bar{C}_n = \log_2 n+\gamma/ln2-\lambda+\tfrac{1}{2}+\omega(\log_2 n)+O(n^{-0.98}),$$

where $\lambda = \sum_{n\geq 1} 1/(2^n-1) = 1.6067,\ldots, \gamma = 0.577721,\ldots,$ *is the Euler constant, and ω is a periodic function of mean value* 0 *and amplitude less than* 10^{-5}.

The constant after $\log_2 n$ gives the *asymptotic drift* of C_n compared to $\log_2 n$, and its numerical value is $-0.27395,\ldots$; furthermore, calculations developed hereafter show that the actual drift for finite n varies very little with n: for $n = 10$, 100, 20000, the values of $\bar{C}_n - \log_2 n$ are respectively $+0.0453$, -0.2383, -0.2737.

Another interesting feature of the algorithm is the relatively low dispersion of the results it produces. We can prove:

THEOREM 2. *After n successive increments, the standard deviation of the contents of the counter satisfies*

$$\sigma_n^2 = \sigma_\infty^2+\pi(\log_2 n) +o(1),$$

where $\sigma_\infty = 0.8736,\ldots$ *is a constant and π is a periodic function of mean value* 0 *and amplitude less than* 10^{-4}. *The constant σ_∞ has the explicit expression*

$$\sigma_\infty^2 = \frac{\pi^2}{6\log_2 2} - \sum_{n\geq 1} \frac{2^n}{(2^n-1)^2} + \frac{1}{12} - \frac{1}{\log 2} \sum_{k\geq 1} \frac{1}{k \sin h(\theta k)},$$

$$\theta = 2\pi\cdot(\log 2)^{-1}.$$

In particular, corresponding to $n = 10$, 100, 20000, we have $\sigma_n = 0.7776$, 0.8618, 0.8734. Thus typically C_n estimates $\log_2 n$ with an error less than 1.

Finally, the methods developed here also permit evaluation of the probabilities of error. The distribution of values of approximate counting after n increments appears to be fairly narrowly centered around the average (better than merely predicted from the variance analysis using the Chebyshev inequalities), and for instance in the case of $n = 1024$ (so that $\log_2 n = 10$) the following probabilities for the counter values, determined by Proposition 1 below, are:

counter value	7	8	9	10	11	12	13
probability	0.0011	0.0602	0.3424	0.4218	0.1538	0.0195	0.0001

Thus in that case C_n will differ from $\log_2 n$ by more than 1 unit in only 8% of the cases. More generally, Proposition 3 provides a sort of limiting distribution result for the probabilities of counter values.

2. Basic probabilities.

The possible evolutions of the algorithm can be seen as an evergrowing tree: we start from the counter set to 1; from this two situations can result: either the counter keeps its value 1 (this has probability $\frac{1}{2}$) or it is increased to 2 (with probability $\frac{1}{2}$); each of these possible stages has itself two possible outcomes. The corresponding tree of possibilities is given in Figure 1, with edges labelled with the probabilities of corresponding transitions. From it, we see for instance that when $n = 3$, the probabilities of observing counter values 1, 2, 3, 4 are respectively $\frac{8}{64}, \frac{38}{64}, \frac{17}{64}, \frac{1}{64}$.

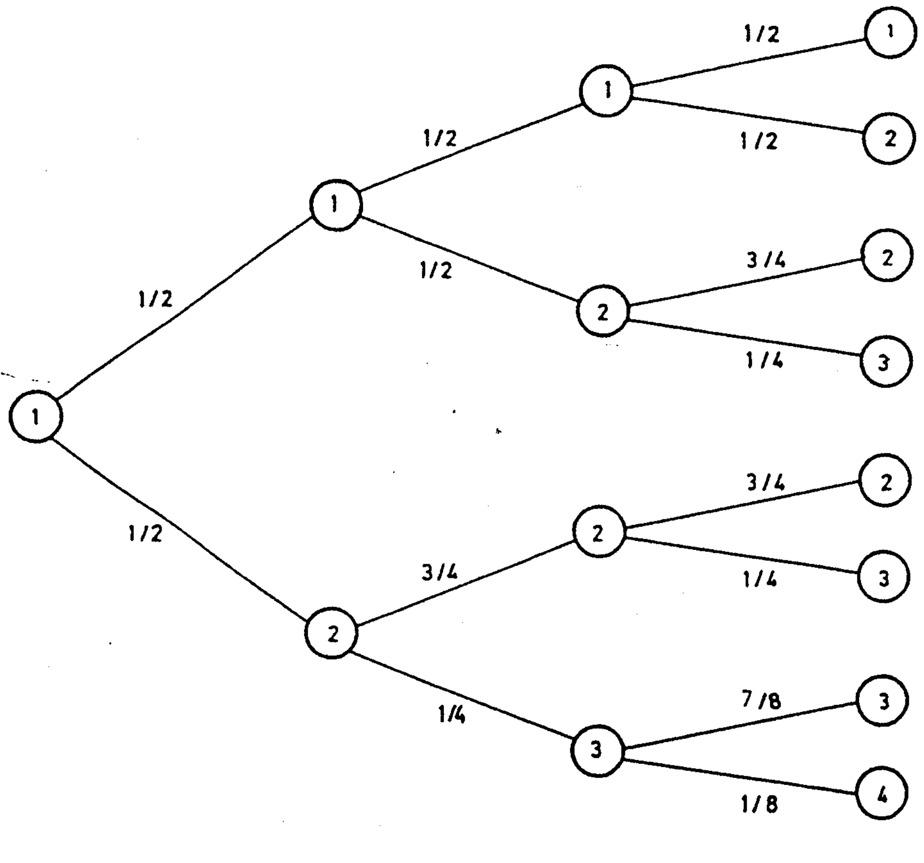
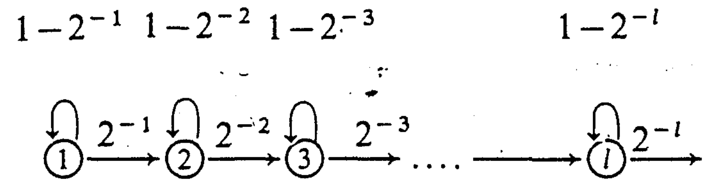


Fig. 1. The possible evolutions of approximate counting for $n = 1, 2, 3$.

Another way of viewing the evolutions is by drawing a state diagram:



This is to be interpreted as follows: at state $l = 1, 2, 3, \ldots$, (i.e. when the counter contains value l), one increment causes the transition to state $l+1$ with probability 2^{-l}, and the transition to state l with probability $1-2^{-l}$. This is formally a discrete time pure birth process [5].

Let $p_{n,l}$ be the probability that the counter contains the value l after n applications of the stochastic increment procedure. To compute $p_{n,l}$, observe that the probability of reaching state l through n_1 transitions from state 1 to state 1, n_2 transitions from state 2 to state $2, \ldots, n_l$ transitions from state l to state l is:

$$(1-2^{-1})^{n_1}2^{-1}(1-2^{-2})^{n_2}2^{-2}\ldots(1-2^{-(l-1)})^{n_{l-1}}2^{-(l-1)}(1-2^{-l})^{n_l},$$

with the condition that:

$$n_1+n_2+\ldots+n_l+l-1 = n.$$

Summing over all possible intermediary transitions, we thus find

$$p_{n,l} = 2^{-l(l-1)/2} \sum (1-2^{-1})^{n_1}(1-2^{-2})^{n_2}\ldots(1-2^{-l})^{n_l} \tag{1}$$

with summation over $n_1+n_2+\ldots n_l+l-1 = n$.

If we introduce the corresponding generating functions (for each l):

$$H_l(x) = \sum_{n \geqq 0} p_{n,l}x^n, \tag{2}$$

we observe that (1) expresses the $p_{n,l}$ as the coefficients of a Cauchy product of simpler functions, so that:

$$H_l(x) = \frac{2^{-l(l-1)/2}x^{l-1}}{(1-\alpha_1 x)(1-\alpha_2 x)\ldots(1-\alpha_l x)} \quad \text{with } \alpha_j = 1-2^{-j}. \tag{3}$$

We obtain an expression different from (1), and indeed simpler to estimate numerically, decomposing H_l into partial fractions. Since we expect the asymptotically dominant contributions in the $p_{n,l}$ to come from the dominant poles, we set:

$$H_l(x) = \sum_{j=0}^{l-1} \frac{C_j}{1-x\alpha_{l-j}}, \tag{4}$$

and start evaluating $C_0, C_1, \ldots$.

We have

$$C_j = \lim\nolimits_{x \to \alpha_{l-j}^{-1}} H_l(x)(1-\alpha_{l-j}x), \quad \text{so that}$$

$$C_0 = 2^{-l(l-1)/2}(1-2^{-l})^{-(l-1)}(1-\alpha_1/\alpha_l)^{-1}(1-\alpha_2/\alpha_l)^{-1}\dots(1-\alpha_{l-1}/\alpha_l)^{-1}$$

which after simplification using $(\alpha_l-\alpha_j) = 2^{-j}(1-2^{-l+j})$ gives:

$$C_0 = (1-\tfrac{1}{2})^{-1}(1-1/4)^{-1}\dots(1-2^{-(l-1)})^{-1}.$$

Similarly, we find in general

$$C_j = (-1)^j 2^{-j(j-1)/2} Q_j^{-1} Q_{l-1-j}^{-1}, \tag{5}$$

where for all k:

$$Q_k = \prod_{i=1}^{k} (1-2^{-i}) \quad \text{and} \quad Q_0 = 1. \tag{6}$$

Now from (4), there immediately follows an expression for the coefficients $p_{n,l}$ of $H_l(x)$:

$$p_{n,l} = \sum_{j=0}^{l-1} C_j \alpha_{l-j}^n,$$

whence with (5), (6):

PROPOSITION 1. *The probability $p_{n,l}$ of having counter value l after n increments is*

$$p_{n,l} = \sum_{j=0}^{l-1} (-1)^j 2^{-j(j-1)/2}(1-2^{-(l-j)})^n \prod_{i=1}^{j} (1-2^{-i})^{-1} \prod_{i=1}^{l-1-j} (1-2^{-i})^{-1}. \tag{7}$$

This expression permits an easy numerical calculation of the probabilities involved in approximate counting. We notice also that from their definition the quantities $p_{n,l}$ satisfy the recurrence:

$$p_{n+1,l} = (1-2^{-l})p_{n,l} + 2^{-(l-1)}p_{n,l-1}$$

from which by induction follows the already mentioned equality:

$$E(2^{C_n}) = n+2.$$

3. Continuing with approximations.

The expression of Proposition 1 is not as bad as it looks. First the product

$$Q = \prod_{i=1}^{\infty} (1-2^{-i}) \tag{8}$$

is convergent and simple comparisons with the geometric series show that

$$|Q-Q_k| = O(2^{-k})$$

with Q_k defined in (6). In particular the Q_k are always in the interval defined by $Q = 0.288788,\dots$ and 1, and the denominators in (7) are bounded.

Second, the very fast decrease of the coefficients $2^{-j(j-1)/2}$ shows that numerically the significant contribution comes from small values of the index j, and asymptotically only values of j less than $O(\sqrt{}\log_2 n)$ need to be considered.

Last, the exponential approximation $(1-a)^n \simeq e^{-an}$ is usually justified in this class of problems (see *e.g.* [6, p. 131]).

We first prove that for l small enough compared to n, the probabilities $p_{n,l}$ are small.

Proposition 2. *For $l < \log_2 n - 2\log_2 \ln n$, the probabilities $p_{n,1}$ satisfy*
$$p_{n,l} = O(\ln n \exp(-(\ln n)^2))$$
uniformly in n and l.

Proof. Since we have $(1-2^{-l})^n > (1-2/2^l)^n > (1-4/2^l)^n > \dots$ and $Q_k > Q$ for all k, the $p_{n,l}$ can be bounded by

$$p_{n,l} < Q^{-2}l(1-2^{-l})^n = Q^{-2}l\exp(n\ln(1-2^{-l})). \tag{9}$$

Now observing that for
$$u \in]0;1[\ : \exp(n\ln(1-u)) = \exp(-nu-nu^2/2-\dots) < e^{-nu},$$
we obtain from (9):

$$p_{n,l} < Q^{-2}l\exp(-n2^{-l}) = O(\ln n \exp((-\ln n)^2))$$

which is thus exponentially small. ■

Now when l is large enough, we can prove that the $p_{n,l}$ approach a *limiting distribution* in the following sense:

Proposition 3. *Let ϕ be the function defined by*

$$\phi(x) = Q^{-1}\sum_{j=0}^{\infty}(-1)^j 2^{-j(j-1)/2}\exp(-x2^j)\prod_{i=1}^{j}(1-2^{-i})^{-1}.$$

Then for $l > \log_2 n - 2\log_2 \ln n$, we have $p_{n,l} = \phi(n2^{-l}) + O(n^{-0.99})$ where the $O(.)$ term is uniform in n and l.

PROOF. The proof proceeds by stages using the previously mentioned approximations.

(i) Truncation of the sum: let $r = r(n) = 2(\log_2 n)^{1/2}$. We set

$$p'_{n,l} = \sum_{j=0}^{r} (-1)^j 2^{-j(j-1)/2} Q_j^{-1} Q_{l-1-j}^{-1}(1-2^{-(l-j)})^n. \tag{10}$$

Obviously

$$|p_{n,l}-p'_{n,l}| \leqq Q^{-2} \sum_{j>r} 2^{-j(j-1)/2} = Q(n^{-2}) \tag{11}$$

(ii) Simplification of the denominators: define

$$p''_{n,l} = Q^{-1} \sum_{j=0}^{r} (-1)^j 2^{-j(j-1)/2} Q_j^{-1}(1-2^{-(l-j)})^n \tag{12}$$

using the fact that $|Q-Q_{l-1-j}| = O(2^{-l+1+j})$ since the sum of $p''_{n,l}$ comprises $(r+1)$ terms:

$$|p'_{n,l}-p''_{n,l}| = O(r(n)2^{-l+1+r(n)}) = O(n^{-1}r(n)2^{r(n)}(\ln n)^2) = O(n^{-0.99}). \tag{13}$$

(iii) Using the exponential approximation: given the conditions on l and j, $u = 2^{-(l-j)}$ is always small, so that:

$$(1-u)^n = e^{n\ln(1-u)} = e^{-nu}e^{O(nu^2)} = e^{-nu}(1+O(nu^2))$$

since $nu^2 < 1$ for n large enough. Thus setting:

$$p'''_{n,l} = Q^{-1} \sum_{j=0}^{r} (-1)^j 2^{-j(j-1)/2} Q_j^{-1} \exp(-n2^{j-l}) \tag{14}$$

we have

$$|p''_{n,l}-p'''_{n,l}| = O(r(n)n2^{2r(n)}2^{-2l}) = O(n^{-0.99}) \tag{15}$$

(iv) Completing the sum: $p'''_{n,l}$ is a partial sum of $\phi(n2^{-l})$; using again the majorization of (11), we find:

$$|p'''_{n,l}-\phi(n2^{-l})| = O(n^{-2}). \tag{16}$$

Thus putting together equations (10) to (16) proves Proposition 3. ■

Finally we need information on the tail of the distribution, corresponding to values of l larger than $\log_2 n$.

Proposition 4. *For $l = 2\log_2 n + \delta$ with $\delta \geqq 0$, we have $p_{n,l} = O(2^{-\delta} n^{-0.99})$ uniformly in n and δ.*

Proof (sketch). The proof mimics the previous one; let us choose this time

$$r = \log_2 n + \delta$$

as the splitting value for the index in the sum giving $p_{n,l}$. In part (i), we now have:

$$\left| \sum_{j=r+1}^{l-1} (-1)^j 2^{-j(j-1)/2} Q_j^{-1} Q_{l-1-j}^{-1} (1-2^{-(l-j)})^n \right| < Q^{-2} \sum_{j>r} 2^{-j(j-1)/2} = \tag{17}$$

$$= O(2^{-\delta} 2^{-(\ln n)2}).$$

Parts (ii) and (iii) now lead to error bounds of the form $O(2^{-\delta} n^{-0.99})$ since $2^{-(l-j)} = O(n^{-1})$.

Finally we can again complete the sum as in (iv) introducing error terms of the form (17).

We have thus proved:

$$p_{n,l} = \phi(n2^{-l}) + O(2^{-\delta} n^{-0.99}). \tag{18}$$

Since $\phi(x)$ is clearly differentiable at $x = 0$, we have

$$\phi(n2^{-l}) = O(n2^{-l}) = O(2^{-\delta} n^{-1}). \tag{19}$$

Thus combining (18) and (19) completes the proof of the proposition. ■

In the sequel we shall need properties of the function ϕ. Some of them appear to be related to classical identities in the theory of partitions. Our starting point is the following identity [1]:

$$\prod_{m=0}^{\infty} (1+ut^m) = \sum_{k=0}^{\infty} \frac{u^k t^{k(k-1)/2}}{(1-t)(1-t^2)\dots(1-t^k)} \tag{20}$$

(with the usual convention for $k = 0$ that an empty product is equal to 1).

Equation (20) is also valid analytically for all u, provided that $|t| < 1$.

The coefficient of $u^k t^n$ in the left hand side member counts the number of partitions of the integer n into distinct parts and, with a simple transformation on partitions, the right-hand side can be similarly interpreted (see also [1] for an algebraic proof). Instantiating (20) with $u = -1$ and $t = 1/2$, shows that

$$\sum_{k=0}^{\infty} (-1)^k 2^{-k(k-1)/2} Q_k^{-1} = 0 \tag{21}$$

and thus $\phi(0) = 0$ as could be expected. We shall also need the following identities:

$$\sum_{k=1}^{\infty} (-1)^k 2^{-k(k-1)/2} k (1-2^{-1})^{-1} \dots (1-2^{-k})^{-1} = \tag{22}$$

$$= -(1-\tfrac{1}{2})(1-\tfrac{1}{4})(1-\tfrac{1}{8})\dots,$$

$$\sum_{k=2}^{\infty} (-1)^k k(k-1) 2^{-k(k-1)/2} (1-2^{-1})^{-1} \dots (1-2^{-k})^{-1} = \tag{23}$$

$$= 2[(1-\tfrac{1}{2})(1-\tfrac{1}{4})(1-\tfrac{1}{8})\dots] \sum_{n=1}^{\infty} \frac{1}{2^n-1},$$

which are easily obtained by successive differentiation of (20) with respect to u, setting then $u = -1$ and $t = \frac{1}{2}$.

4. Determination of asymptotic expansions.

The developments above suggest approximating $\bar{C}_n$ with the value $F(n)$ where function F is defined for all $x \geqq 0$ by:

$$F(x) = \sum_{l \geqq 1} l\phi(x 2^{-l}). \tag{24}$$

For large x, F can be estimated using Mellin transform techniques.

We first prove

LEMMA 1. *The expected value $\bar{C}_n$ satisfies*:

$$\bar{C}_n = F(n) + O(n^{-0.98}).$$

PROOF. Let us define the 3 intervals:

$$\begin{aligned} I_1 &= [1, \log_2 n - 2\log_2 \ln n[\\ I_2 &= [\log_2 n - 2\log_2 \ln n, 2\log_2 n[\\ I_3 &= [2\log_2 n, \infty[, \end{aligned}$$

and for $j = 1, 2, 3$:

$$C^{(j)} = \sum_{l \in I_j} l p_{n,l}: \qquad F^{(j)} = \sum_{l \in I_j} l\phi(2^{-l} n).$$

By Proposition 2, we have:

$$|C^{(1)} - F^{(1)}| = O((\ln n)^2 e^{-(\ln n)^2});$$

similarly, by Proposition 3

$$|C^{(2)} - F^{(2)}| = O(n^{-0.99} \ln n),$$

and by Proposition 4:

$$|C^{(3)} - F^{(3)}| = O\left(\sum_{\delta > 0} 2^{-\delta} n^{-0.99} \right) = O(n^{-0.99}).$$

The three last equalities imply Lemma 1. ■

We are thus left with estimating the behaviour of $F(x)$ as given by (24). To that purpose, we use the *Mellin integral transform* which for a real function f is defined by (see [2]):

$$f^*(s) = \mathcal{M}[f(x); s] = \int_0^\infty f(x) x^{s-1} dx. \tag{25}$$

This transform is useful for studying harmonic sums like (24): from the obvious functional property

$$\mathcal{M}[f(ax); s] = a^{-s} f^*(s), \qquad a > 0, \tag{26}$$

it follows formally that the Mellin transform of F is

$$F^*(s) = (\sum_{l \geqq 1} l\, 2^{ls}) \phi^*(s). \tag{27}$$

The Mellin transform of ϕ is itself computed using (26) repeatedly: from the definition of ϕ (again formally) we expect

$$\phi^*(s) = Q^{-1} \sum_{j \geqq 0} (-1)^j 2^{-j(j-1)/2 - js} Q_j^{-1} \Gamma(s) \tag{28}$$

since, as is classically known [10]:

$$\mathcal{M}[e^{-x}; s] = \int_0^\infty e^{-x} x^{s-1} dx = \Gamma(s). \tag{29}$$

Thus formally, we have:

$$F^*(s) = 2^s \Gamma(s) (2^s - 1)^{-2} \xi(s), \tag{30}$$

where

$$\xi(s) = Q^{-1} \sum_{j=0}^{\infty} (-1)^j 2^{-j(j-1)/2 - js} (1 - 2^{-1})^{-1} \dots (1 - 2^{-j})^{-1}. \tag{31}$$

Analytically the integral in (29) is defined for $\mathrm{Re}(s) > 0$. For s: $-1 < \mathrm{Re}(s) < 0$, we have

$\int_0^\infty (e^{-x}-1)x^{s-1}dx = \Gamma(s)$. Using (21), we also have

$$\phi(x) = Q^{-1} \sum_{j \geqq 0} (-1)^j 2^{-j(j-1)/2} Q_j^{-1}(\exp(-2^j x - 1),$$

whence the integral defining the Mellin transform of ϕ is defined for $-1 < \mathrm{Re}(s) < 0$. Actually (28) holds for any $s, \mathrm{Re}(s) > -1$, and ϕ^* has a removeable singularity at $s = 0$. It is finally easy to see that (27) holds provided the sum there is convergent, which requires $\mathrm{Re}(s) < 0$. Thus equations (30), (31) are justified for s in the strip $-1 < \mathrm{Re}(s) < 0$; there the integral of the form (25) expressing $F^*(s)$ is absolutely convergent.

The singularities of $F^*(s)$ are related to the terms in the asymptotic expansion of $F(x)$ when $x \to \infty$ [2]. To see that, we use the inversion theorem for Mellin transforms which gives

$$(32) \qquad F(x) = \frac{1}{2i\pi} \int_{d-i\infty}^{d+i\infty} F^*(s) x^{-s} ds,$$

where d can be taken arbitrarily inside the domain of absolute convergence of the integral giving $F^*(s)$. Here, we may take any d in the interval $]-1, 0[$.

By Cauchy's residue theorem, assuming the contour of integration can be moved to the right with $F^*(s)$ meromorphic:

$$(33) \qquad F(x) = \frac{1}{2i\pi} \int_{E+i\infty}^{E-i\infty} F^*(s) x^{-s} ds - \sum_s \mathrm{Res}(F^*(s)x^{-s})$$

where the summation is extended to all poles of $F^*(s)$ in the strip $d < \mathrm{Re}(s) < E$.

The first integral should be $O(x^{-E})$ representing smaller and smaller terms (for large x) as E increases. A simple computation shows that if $F^*(s)$ has a pole of order k at $s_0 = \sigma_0 + it_0$, then

$$\operatorname*{Res}_{s=s_0} (F^*(s)x^{-s}) = x^{-s_0} P_{k-1}(\ln x),$$

where P_{k-1} is a polynomial of degree $k-1$. Since $x^{-s_0} = x^{-\sigma_0} e^{-it_0 \ln x}$ we thus see that successive poles of F^* starting from the left yield successive terms in the asymptotic expansion of $F(x)$ for $x \to \infty$.

We shall therefore first identify singularities of $F^*(s)$ for $\mathrm{Re}(s) \geqq 0$ and then return to a formal justification of (33).

(i) $F^*(s)$ has a double pole at $s = 0$ as the following expansions show:

$$\Gamma(s) = s^{-1}\Gamma(s+1) = s^{-1}(1-\gamma s+O(s^2)) \qquad \text{(see e.g. [10])} \tag{34}$$

$$2^s(2^s-1)^{-2} = s^{-2}(\ln 2)^{-2}(1+O(s^2)) \tag{35}$$

$$\xi(s) = \xi(0)+s\xi'(0)+(s^2/2)\xi''(0)+O(s^3). \tag{36}$$

We already know from (21) that $\xi(0) = 0$. Using (22), we can transform $\xi'(0)$:

$$\xi'(0) = -Q^{-1}\sum_{j\geqq 0}(-1)^j 2^{-j(j-1)/2} jQ_j^{-1}\ln 2 = \ln 2.$$

Similarly with (23):

$$\xi''(0) = Q^{-1}\sum_{j\geqq 0}(-1)^j 2^{-j(j-1)/2} j^2 Q_j^{-1}(\ln 2)^2$$
$$= Q^{-1}(\ln 2)^2\left\{2Q\sum_{n\geqq 1}(2^n-1)^{-1}-Q\right\} = (\ln 2)^2\left\{2\sum_{n\geqq 1}(2^n-1)^{-1}-1\right\}.$$

Thus for ξ around 0:

$$\xi(s) = s\ln 2(1+\ln 2(\lambda-\tfrac{1}{2})+O(s^2)) \tag{37}$$

with $\lambda = \sum_{n\geqq 1}(2^n-1)^{-1}$. We also have $x^{-s} = e^{-s\ln x} = 1-s\ln x+O(s^2(\ln x)^2)$

so that the residue of $F^*(s)x^{-s}$ at 0 can be evaluated exactly, and we find from (34) and (37):

$$\operatorname*{Res}_{s=0}(x^{-s}F^*(s)) = -\log_2 x-\gamma/\ln 2+\lambda-\tfrac{1}{2}. \tag{38}$$

(ii) $F^*(s)$ also has a simple pole at $\chi_k = 2ik\pi/\ln 2$ for all $k\in\mathbb{Z}\backslash 0$. Due to the periodicity of $\xi(s)$ and 2^s, we can use some of the previous expansions; in particular around χ_k:

$$\xi(s) = (s-\chi_k)\xi'(0)+O((s-\chi_k)^2) = \ln 2(s-\chi_k)+O((s-\chi_k)^2.$$

$$2^s(2^s-1)^{-2} = (s-\chi_k)^{-2}(\ln 2)^{-2}(1+O(s-\chi_k));$$

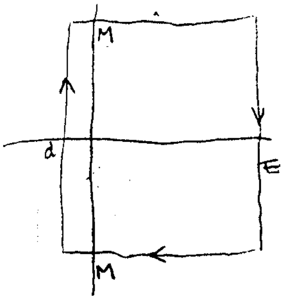
$$\Gamma(s) = \Gamma(\chi_k)+O(s-\chi_k).$$

Thus:

$$(39) \qquad \operatorname*{Res}_{s=\chi_k} (x^{-s}F^*(s)) = \Gamma(\chi_k)e^{-\chi_k \log x}/\ln 2.$$

To conclude with the proof of the theorem, we only need to establish (33). To that purpose we use the rectangular contours

$$R(M, E) = R_1 + R_2 + R_3 + R_4, \quad \text{where}$$
$$R_1 = \{d+it \mid t \in [-M; M]\}$$
$$R_2 = \{u+iM \mid u \in [d; E]\}$$
$$R_3 = \{E+it \mid t \in [-M; M]\}$$
$$R_4 = \{u-iM \mid u \in [d; E]\}$$



with R oriented clockwise. For any positive d and iM not equal to one of the χ_k, we have by Cauchy's theorem applied to the contour R and the integrand $F^*(s)x^{-s}$:

$$(2\pi i)^{-1}\left[\int_{d-iM}^{d+iM} + \int_{d+iM}^{E+iM} + \int_{E+iM}^{E-iM} + \int_{E-iM}^{d-iM}\right] = -\sum \operatorname{Res}(F^*(s)x^{-s})$$

where the sum is extended to all poles s with

$$-M < \operatorname{Im}(s) < +M, \qquad d < \operatorname{Re}(s) < E.$$

If we let M tend to infinity – keeping E fixed – in such a way that $M = (2k+1)\pi/\ln 2$ for some integer k, we observe that, along the contour, $\xi(s)$ and $2^s/(2^s-1)^2$ stay uniformly bounded. The very fast decrease of $\Gamma(s)$ when $\operatorname{Im}(s)$ tends to infinity [10] verifies that the second and fourth integrals then tend to 0.

The first term converges to $F(x)$ by the inversion formula (32). As to the third one, it is bounded in modulus by:

$$(2\pi)^{-1}x^{-E}\int_{-\infty}^{+\infty} |F(E+it)|\,dt < A(E)x^{-E}$$

for all $x > 0$. On the right hand side, the sum is a partial sum of a Fourier series of $\log_2 x$, which is also convergent.

We have therefore established that

$$(40) \qquad F(x) = \log_2 x + \gamma/\ln 2 - \lambda + \tfrac{1}{2} - (\ln 2)^{-1} \sum_{k \in \mathbb{Z}\backslash 0} \Gamma(\chi_k)e^{-2ik\pi \log_2 x} + O(x^{-E})$$

for any positive E. Combining (40) with Lemma 1, and taking $E = 1$ establishes Theorem 1. In passing, we have proved:

COROLLARY. *The periodic function that expresses the fluctuations of $\bar{C}_n$ is*

$$(41) \qquad \omega(u) = -(\ln 2)^{-1} \sum_{k \in \mathbb{Z} \setminus 0} {}^{\Gamma(\chi_k) e^{-2ik\pi u}}$$

Such periodicities are not of infrequent occurrence in the analysis of algorithms: a function similar to ω turns up in the analysis of radix exchange sort, as shown in [6, p. 131] where an integration contour similar to ours is used.

Let us last briefly mention how to prove Theorem 2 relative to the variance. After n increments, the variance of the counter content is:

$$(42) \qquad V_n = \sum l^2 p_{n,l} - \bar{C}_n^2.$$

To handle the sum, we first approximate it by $G(n)$ where

$$(43) \qquad G(x) = \sum_l l^2 \phi(2^{-l} x)$$

introducing only vanishing error terms. The Mellin transform of (43) is

$$(44) \qquad G^*(s) = 2^s(2^s+1)/(2^s-1)^{-3}\Gamma(s)\xi(s)$$

which now has a triple pole at $s = 0$, and double poles at $s = 2k\pi i/\ln 2$.

Thus $G(x) = O(\log^2 x)$ as $x \to \infty$.

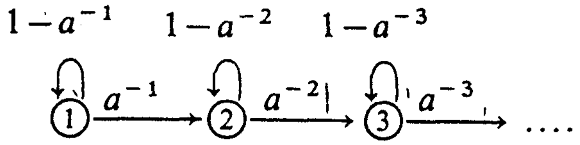
One can actually determine the terms in the asymptotic expansion of G up to $O(1)$ error terms. The main terms in $G(n)$ cancel with those of $\bar{C}_n^2$ and we are left with the result of Theorem 2.

5. Extensions to an arbitrary base.

The previous analysis has shown precisely that the performances of approximate counting (with base 2) remain remarkably stable with the number of increments. However, for certain applications, the expected error of about one binary order of magnitude might be prohibitively large.

The performance of the algorithm might for instance be improved by keeping several counters and averaging their contents which can be done in a storage efficient manner (keeping only one counter and a set of differences). It turns out, however, that an effect similar to averaging is achieved more elegantly – and in a way simpler to implement – by using a different base: in the increment procedure of Section 1, only change the definition of DELTA(C), letting DELTA(C) be a random variable that takes the value 1 with probability

a^{-C} and the value 0 with probability $1-a^{-C}$. The number a is called the *base*. The corresponding state transition diagram is then:



Apart from notational details, this is exactly Morris' original algorithm [8]. If we take $a < 2$, the value of the counter content after n increments will be larger than with a binary base, and we should expect a smoother behaviour of the counter contents as a function of n, thus giving a better accuracy (see [8]). From a practical standpoint, the transition probabilities need not be recomputed each time, and can be stored once and for all in a table.

As in the binary case (see the end of Section 2), we can easily prove:

PROPOSITION 5 [8]: *If C_n is the value of the counter of approximate counting after n increments, then $E(a^{C_n}) = n(a-1)+a$ so that*

$$D_n = (a^{C_n}-a)/(a-1) \tag{45}$$

is an unbiased estimator of n. The variance of D_n is $\sigma^2(D_n) = (a-1)n(n+1)/2$.

In the sequel, we give the generalization of our previous results to the case of an arbitrary base and concentrate on the corrections necessary to obtain an unbiased estimator of $\log_2 n$.

We let $\bar{C}_n(a)$ denote the expectation of C_n and use similar obvious generalizations of our previous notation for other quantities of interest. The calculations develop in a way similar to before (replacing essentially 2 by a in most formulae) and we find:

(i) For the probability distribution of counter values:

$$P_{n,l}(a) = \sum_{j=0}^{l-1} (-1)^j a^{-j(j-1)/2} Q_j(a)^{-1} Q_{l-1-j}(a)^{-1} (1-a^{-(l-j)})^n \tag{46}$$

with $Q_n(a) = \prod_{i=1}^{m} (1-a^{-i})$.

(ii) For the expected counter value after n increments:

$$\bar{C}_n(a) = \log_a n + \gamma/\ln a - \lambda(a) + \tfrac{1}{2} + \omega(a;\log_a n) + O(1). \tag{47a}$$

$$\lambda(a) = \sum_{n \geq 1} (a^n-1)^{-1}.$$

(iii) For the standard deviation of the counter values:

(47b) $\sigma_n^2(a) = \sigma_\infty^2(a) + \pi(a; \log_a n) + o(1)$, where

$$\sigma_\infty^2(a) = \frac{\pi^2}{6\ln^2 a} - \sum_{n \geqq 1} \frac{a^n}{(a^n-1)^2} + \frac{1}{12} - \frac{1}{\ln a} \sum_{k \geqq 1} \frac{1}{k \sinh(\theta k)}, \quad \theta = 2\pi^2/\ln a.$$

There follows from these equations that for the content of the counter (with base a), the normalized value

$$(48) \qquad X = (C - K(a)) \cdot \log_2 a \quad \text{where} \quad K(a) = \gamma/\ln a - \sum_{n \geqq 1} (a^n - 1)^{-1} + \tfrac{1}{2}$$

is apart from negligible periodic fluctuations, an *asymptotically unbiased estimator of* $\log_2 n$. The values $K(a)$ for $a = 2, 2^{1/2} \ldots 2^{1/16}$ are:

$K(2) = -0.2729$; $K(2^{1/2}) = -2.8030$; $K(2^{1/4}) = -9.8598$
$K(2^{1/8}) = -27.9714$; $K(2^{1/16}) = -72.1936$.

Figure 2 displays the expectations of X_n (the value of the normalized variable X of (48) after n increments) for $n = 128$, 1024 and a few values of a, together with the corresponding standard deviations of X_n defined by:

$$(49) \qquad \tau_n^2(a) = E((X_n - E(X_n))^2).$$

a	$E(X_n)$	τ_n
2	7.01581	0.864
$2^{1/2}$	7.03139	0.596
$2^{1/4}$	7.06266	0.409
$2^{1/8}$	7.12340	0.276
$2^{1/16}$	7.23907	0.174

$n = 128$

a	$E(X_n)$	τ_n
2	10.00113	0.872
$2^{1/2}$	10.00307	0.607
$2^{1/4}$	10.00712	0.425
$2^{1/8}$	10.01516	0.298
$2^{1/16}$	10.03027	0.217

$n = 1024$

Fig. 2. Bias and accuracy of the normalized X value for sample values of a and n.

Table 2 shows that X is a very good estimate of $\log_2 n$ even for small values n ($n \sim 10^3$). For smaller n ($n \sim 10^2$) there is a slight bias which increases when a gets closer to 1. If necessary, corrections for smaller values of n could be easily tabulated using (47a) and introduced in the algorithm.

The accuracy of that version of the algorithm is thus essentially determined by the dispersion of results it produces. The values of τ_n for finite n are remarkably close to the asymptotic limit $\tau_\infty(a) = \sigma_\infty(a)/\ln a$ as shown by a comparison of results in Figure 2 with the values:

$$\tau_\infty(2) = 0.873; \quad \tau_\infty(2^{1/2}) = 0.609; \quad \tau_\infty(2^{1/4}) = 0.427;$$
$$\tau_\infty(2^{1/8}) = 0.302; \quad \tau_\infty(2^{1/16}) = 0.212.$$

Thus to determine the effect of smaller bases on the accuracy of the algorithm, we only need to determine the dependence of $\tau_\infty(a)$ on a. To do so, it proves convenient, as we shall see, to study the behaviour of $\tau_\infty(a)$ as $a \to 1$; this will lead to very good numerical estimates on $\tau_\infty(a)$ for general a. From (47), we have:

$$\tau_\infty^2(a) = (\log_2 a)^2 \cdot \left[\frac{\pi^2}{6\ln^2 a} - \sum_{n \geqq 1} \frac{a^n}{(a^n-1)^2}\right] + o(\ln^2 a), \quad a \to 1.$$

The asymptotics of the function appearing in the expression above

$$c(x) = \sum_{n \geqq 1} e^{-nx}/(e^{-nx}-1)^2$$

for small x are easily determined, again by Mellin transform techniques since the transform of $c(x)$ is:

$$c^*(s) = \Gamma(s)\zeta(s)\zeta(s-1),$$

so that, when $x \to 0$:

$$c(x) \sim \sum_\sigma \operatorname{Res}(c^*(s)x^{-s}; \sigma), \quad \sigma = 2, 1, 0, -1, \ldots.$$

$$c(x) = (\pi^2/6)x^{-2} - 2^{-1}x^{-1} + O(1).$$

Thus, using this result in the expression of $\tau_\infty(a)$:

$$\tau_\infty(a) \sim (\log_2 a/\ln 4)^{1/2}, \quad a \to 1. \tag{50}$$

6. Final conclusions.

We have examined in Section 5, two possible ways of using the idea of approximate counting with a general base.

(i) The first one, which corresponds to Morris' original algorithm, estimates n by means of the random variable D defined by formula (45) and produces an unbiased estimate of n.

(ii) The second one estimates $\log_2 n$ by means of the random variable X

defined by formula (48) and (apart from negligible fluctuations) leads to an asymptotically unbiased estimate of $\log_2 n$.

As measures of the accuracy of these algorithms, it is reasonable to consider:

(i) The quotient between the standard deviation of the estimate D and n, which provides a measure of the *relative* accuracy of the algorithm. By Proposition 5, this ratio is asymptotic to

$$\mu_1(a) = ((a-1)/2)^{1/2}.$$

(ii) The standard deviation of the estimate X of $\log_2 n$ which from equation (50) is closely approximated by the function

$$\mu_2(a) = (\log_2 a/\ln 4)^{1/2}.$$

The meaning of these formulas is probably best understood if we set $a = 2^{1/m}$ so that one gets better accuracy when m gets large. Using approximations for large m, we find

$$\mu_1(2^{1/m}) \sim (\ln 2/2m)^{1/2}; \qquad \mu_2(2^{1/m}) \sim (m \ln 4)^{-1/2}$$

(both approximations are fairly tight and for instance the approximation of μ_1 is at most 3 % off the exact value of $\tau_\infty(a)$ for all $m \geqq 1$).

As for the storage requirement of the algorithm, it is $E(1+\lfloor \log_2 C_n \rfloor)$; a quantity upper bounded by (and actually close to) $1+\log_2 E(C_n)$, which by our previous results is itself close to $\log_2 \log_2 n + \log_2 m$. Thus setting now $\delta = \log_2 m$ we can roughly summarize the situation as follows:

Fact. *Using approximate counting with base $a = 2^{2^{-\delta}}$ one can count up to n using storage of about $\log_2 \log_2 n + \delta$ bits; the accuracy of the results is close to $0.59\, 2^{-\delta/2}$ and $0.85\, 2^{-\delta/2}$ respectively for the linear estimate algorithm (version (i) based on the variate D) and for the logarithmic estimate algorithm (version (ii) based on the variate X).*

As an example, consider taking as base $a = 2^{1/16} = 1.044\ldots$; such a configuration leads to an expected error on the estimate of $\log_2 n$ close to 0.2125. If an unbiased estimate of n is sought using (45) the relative error given by (45) is typically less than 15 %. This value of a makes it for instance possible to count up to about 65000 ($2^{16} = 65536$) using 8 bits since $\log_2(\log_2 2^{16}/\log_2 2^{1/16}) = 8$, and thus results in this particular case in halving the storage requirement of standard binary counters.

Figure 3 displays the result of a sample run of Approximate Counting with D_n plotted against n for $n = 0\ldots 10^5$, using base $a = 2^{1/16}$ and confirms the good behaviour of the algorithm.

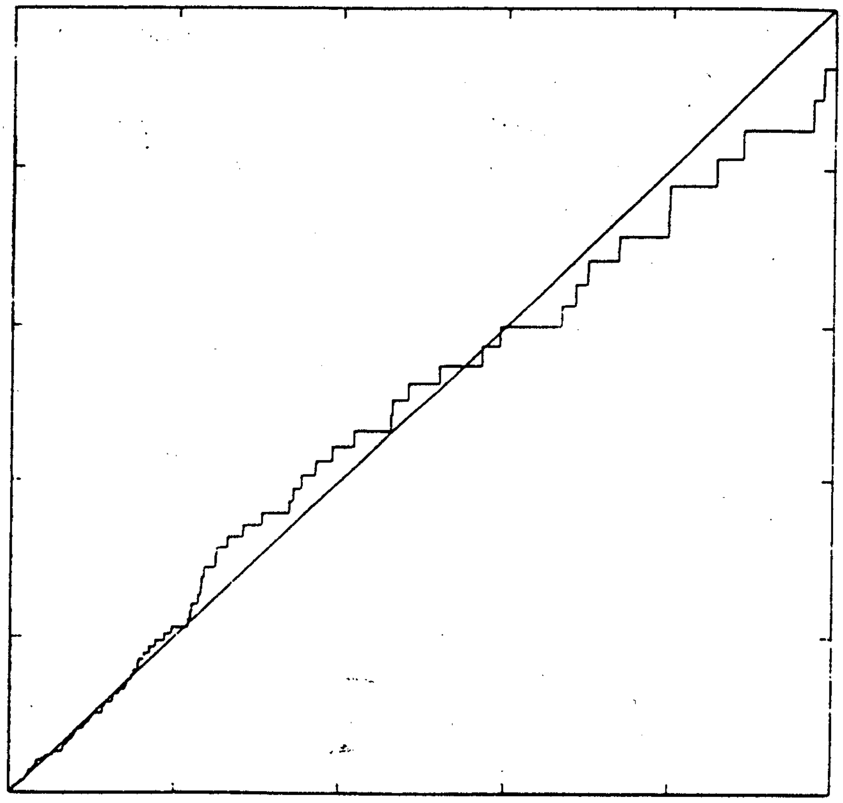
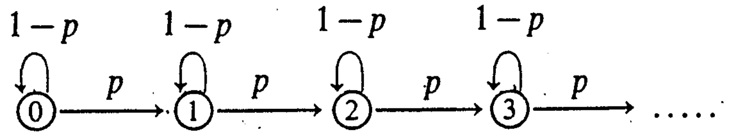


Fig. 3. A simulation of the linear estimate D_n of approximate counting plotted against n, for $0 \leqq n \leqq 10^5$.

Notice finally that a simpler solution to the problem of approximate counting would be a *direct sampling* method in which the approximate counter C is increased with a fixed probability p (instead of using a probability that decreases geometrically with the counter value C). For instance, if $p = 1/256$, on can still count up to $M = 65536 = 2^{16}$ using an average of $\log_2(Mp) = 8$ bits. The corresponding algorithm then simply provides $p^{-1}C_n$ as an estimate of n and the corresponding transition diagram is



That direct sampling algorithm is trivial to analyze since the distribution of counter values is the Bernoulli probability:

$$Pr(C_n = k) = \binom{n}{k} p^k (1-p)^{n-k}.$$

However, it turns out that direct sampling has the major disadvantage of providing greatly inaccurate estimates for n small while approximate counting leads to an expected *constant relative accuracy* of the estimate.

Figure 4 exemplifies (to some extent) this situation. Here we have used $p = 1/256$ for direct sampling and base $a = 2^{1/16}$ for approximate counting, so

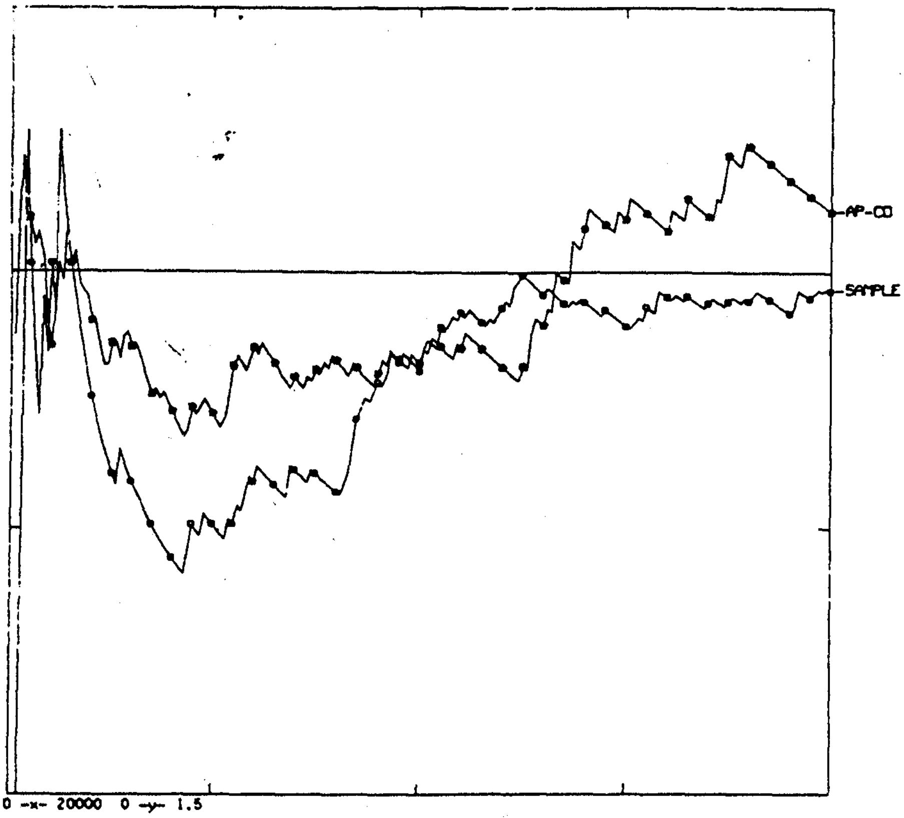


Fig. 4. The ratio between estimates of approximate counting with base $2^{1/16} = 1.044$ (AP – CO) or of direct sampling (SAMPLE) with $p = 1/256$ and exact counter values for $0 \leqq n \leqq 20000$ (simulations).

that both algorithms allow to count up to $2^{16} = 65536$ using only 8 bits.

Considering values of $n = 100, 200, \ldots, 2000$, we notice that the relative accuracy of SAMPLE becomes better as n increases (where the results become more accurate than AP – CO). However, on that particular simulation, while the accuracy (ratio of estimate to exact value) of AP – CO was always between 0.70 and 1.25, that of SAMPLE varied from 0.00 to 1.26; for $n = 100, 200, 300$ the estimate of AP – CO were respectively 88, 218, 366 while those of SAMPLE were 0, 0, 253; for $n = 4300$, SAMPLE still underestimated n by more than a factor of 2 (the accuracy was 0.41).

As a last conclusion approximate counting appears as the method of choice when a fairly constant relative accuracy is needed over a large range of values while saving storage for keeping incremental counters.

Acknowledgements.

The author would like to express his gratitude to Nigel Martin for introducing him to the subject and for several enriching discussions on approximate counting.

Gaston Gonnet and R. Rivest kindly provided reference [8]. Some of the computations have been made using the MACSYMA system for symbolic calculations.

Recently K. Melhorn and K. Simon have shown the author some interesting connections of this work with the analysis of topological sorting under a random graph model; in particular they had obtained independently the first term in our expansion (47a).

BIBLIOGRAPHY

1. L. Comtet, *L'Analyse Combinatoire*, 2 vol., P.U.F., Paris (1970).
2. G. Doetsch, *Handbuch der Laplace Transformation*, Birkhauser Verlag, Basel (1955).
3. P. Flajolet and N. Martin, *Probabilistic counting*, in Proc. 24th Annual Symp. on Foundations of Comp. Sc., Tucson, Arizona (1984), pp. 76–82.
4. R. G. Gallager, *Variations on a theme by Huffmann*, IEEE Trans. IT, 24 (1978) pp. 669–674.
5. L. Kleinrock, *Queuing Systems*, Wiley Interscience, New York (1976).
6. D. E. Knuth, *The Art of Computer Programming: Sorting and Searching*, Addison-Wesley, Reading (1973).
7. G. Langdon and J. Rissanen, *Compression of black white images with binary arithmetic coding*, IEEE Trans. on Communications (1981).
8. R. Morris, *Counting large numbers of events in small registers*, Comm. ACM, 21 (1978), pp. 840–842.
9. S. Todd, N. Martin, G. Langdon and D. Helman, *Dynamic statistics collection for compression coding*, Unpublished manuscript, 12 p. (1981).
10. E. T. Whittaker and G. N. Watson, *A Course in Modern Analysis*, (1907); 4th edition, Cambridge Univ. Press, 1927.